ALTAIR

ONLY FORWARD

Altair Radioss 2022

User Guide

Updated: 03/21/2022

Contents

Intellectual Property Rights Notice iv
Technical Support viii

User Guide 10

Run Radioss 11
Solution Process 27
Recommendations for Using Multiple Cores 29
Advanced Techniques 32
Explicit Structural Finite Element Analysis 37
Time Step 37
Finite Elements 82
Modeling Tools 159
Materials 172
Failure 236
Composites 265
Connections 309
Kinematic Constraints 326
Interfaces 344
Loads 374
Airbag Modeling 379
Debugging Guidelines 433
Appendix 453
Implicit Structural Finite Element Analysis 457
Implicit Features and Compatibility 458
Implicit Analysis Activation 461
Implicit Solvers 463
Fluid-Structure Interaction Analysis with Radioss and AcuSolve 465
Quasi-static Analysis Notes 475
Nonlinear Implicit Analysis Notes 476
Fluid-Structure Interaction Analysis with Radioss and AcuSolve 479
Output Messages 481
Troubleshooting 483
Appendix 485
Fluid and Fluid-Structure Simulation 487
Arbitrary Lagrangian-Eulerian (ALE) and Computational Fluid Dynamics (CFD) 487
Smooth Particle Hydrodynamics (SPH) 518
SPH Cells Distribution 518
Maximum Stored Number of Neighbors 522
SPH Symmetry Conditions 523
Solid to SPH Option (Sol2SPH) 528
Multi-Domain Technique 533
Theoretical Speedup 534

Model Setup 535
Single Input File Format 537
Multiple Input File Format 546
Design Optimization 555
Implementation 555
Error Message Database 560
Engine Error Messages 571
Warning Message Database 577

Index 580

Intellectual Property Rights Notice

Copyright © 1986-2022 Altair Engineering Inc. All Rights Reserved.

This Intellectual Property Rights Notice is exemplary, and therefore not exhaustive, of intellectual property rights held by Altair Engineering Inc. or its affiliates. Software, other products, and materials of Altair Engineering Inc. or its affiliates are protected under laws of the United States and laws of other jurisdictions. In addition to intellectual property rights indicated herein, such software, other products, and materials of Altair Engineering Inc. or its affiliates may be further protected by patents, additional copyrights, additional trademarks, trade secrets, and additional other intellectual property rights. For avoidance of doubt, copyright notice does not imply publication. Copyrights in the below are held by Altair Engineering Inc. except where otherwise explicitly stated. Additionally, all non-Altair marks are the property of their respective owners.

This Intellectual Property Rights Notice does not give you any right to any product, such as software, or underlying intellectual property rights of Altair Engineering Inc. or its affiliates. Usage, for example, of software of Altair Engineering Inc. or its affiliates is governed by and dependent on a valid license agreement.

Altair Simulation Products

Altair® AcuSolve® ©1997-2022

Altair Activate® ©1989-2022

Altair® BatteryDesigner™ ©2019-2022

Altair Compose® ©2007-2022

Altair® ConnectMe™ ©2014-2022

Altair® EDEM™ ©2005-2022 Altair Engineering Limited, ©2019-2022 Altair Engineering Inc.

Altair® ElectroFlo™ ©1992-2022

Altair Embed® ©1989-2022

Altair Embed® SE ©1989-2022

Altair Embed®/Digital Power Designer ©2012-2022

Altair Embed® Viewer ©1996-2022

Altair® ESAComp® ©1992-2022

Altair® Feko® ©1999-2022 Altair Development S.A. (Pty) Ltd., ©1999-2022 Altair Engineering Inc.

Altair® Flow Simulator™ ©2016-2022

Altair® Flux® ©1983-2022

Altair® FluxMotor® ©2017-2022

Altair® HyperCrash® ©2001-2022

Altair® HyperGraph® ©1995-2022

Altair® HyperLife® ©1990-2022

Altair® HyperMesh® ©1990-2022

Altair® HyperStudy® ©1999-2022

Altair® HyperView® ©1999-2022

Altair® HyperWorks® ©1990-2022

Altair® HyperXtrude® ©1999-2022

Altair® Inspire™ ©2009-2022

Altair® Inspire™ Cast ©2011-2022

Altair® Inspire™ Extrude Metal ©1996-2022

Altair® Inspire™ Extrude Polymer ©1996-2022

Altair® Inspire™ Form ©1998-2022

Altair® Inspire™ Friction Stir Welding ©1996-2022

Altair® Inspire™ Mold ©2009-2022

Altair® Inspire™ PolyFoam ©2009-2022

Altair® Inspire™ Play ©2009-2022

Altair® Inspire™ Print3D ©2022

Altair® Inspire™ Render ©1993-2016 Solid Iris Technologies Software Development One PLLC, ©2016-2022 Altair Engineering Inc

Altair® Inspire™ Resin Transfer Molding ©1990-2022

Altair® Inspire™ Studio ©1993-2022

Altair® Material Data Center™ ©2019-2022

Altair® MotionSolve® ©2002-2022

Altair® MotionView® ©1993-2022

Altair® Multiscale Designer® ©2011-2022

Altair® nanoFluidX® ©2013-2022 FluiDyna GmbH, ©2018-2022 Altair Engineering Inc.

Altair® OptiStruct® ©1996-2022

Altair® PollEx™ ©2003-2022

Altair® Pulse™ ©2020-2022

Altair® Radioss® ©1986-2022

Altair® SEAM® ©1985-2019 Cambridge Collaborative, Inc., ©2019-2022 Altair Engineering Inc.

Altair® SimLab® ©2004-2022

Altair® SimSolid® ©2015-2022

Altair® ultraFluidX® ©2010-2022 FluiDyna GmbH, ©2018-2022 Altair Engineering Inc.

Altair® Virtual Wind Tunnel™ ©2012-2022

Altair® WinProp™ ©2000-2022

Altair® WRAP™ ©1998-2022 Altair Engineering AB

Altair® S-FRAME® ©1995-2022 Altair Engineering Canada, Ltd., ©2021-2022 Altair Engineering Inc.

Altair® S-STEEL™ ©1995-2022 Altair Engineering Canada, Ltd., ©2021-2022 Altair Engineering Inc.

Altair® S-PAD™ ©1995-2022 Altair Engineering Canada, Ltd., ©2021-2022 Altair Engineering Inc.

Altair® S-CONCRETE™ ©1995-2022 Altair Engineering Canada, Ltd., ©2021-2022 Altair Engineering Inc.

Altair® S-LINE™ ©1995-2022 Altair Engineering Canada, Ltd., ©2021-2022 Altair Engineering Inc.

Altair® S-TIMBER™ ©1995-2022 Altair Engineering Canada, Ltd., ©2021-2022 Altair Engineering Inc.

Altair® S-FOUNDATION™ ©1995-2022 Altair Engineering Canada, Ltd., ©2021-2022 Altair Engineering Inc.

Altair® S-CALC™ ©1995-2022 Altair Engineering Canada, Ltd., ©2021-2022 Altair Engineering Inc.

Altair Packaged Solution Offerings (PSOs)

Altair® Automated Reporting Director™ ©2008-2022

Altair® e-Motor Director™ ©2019-2022

Altair® Geomechanics Director™ ©2011-2022

Altair® Impact Simulation Director™ ©2010-2022

Altair® Model Mesher Director™ ©2010-2022

Altair® NVH Director™ ©2010-2022

Altair® Squeak and Rattle Director™ ©2012-2022

Altair® Virtual Gauge Director™ ©2012-2022

Altair® Weld Certification Director™ ©2014-2022

Altair® Multi-Disciplinary Optimization Director™ ©2012-2022

Altair HPC & Cloud Products

Altair® PBS Professional® ©1994-2022

Altair® Control™ ©2008-2022

Altair ®Access™ ©2008-2022

Altair® Accelerator™ ©1995-2022

Altair® Accelerator™ Plus©1995-2022

Altair® FlowTracer™ ©1995-2022

Altair® Allocator™ ©1995-2022

Altair® Monitor™ ©1995-2022

Altair® Hero™ ©1995-2022

Altair® Software Asset Optimization (SAO) ©2007-2022

Altair Mistral™ ©2022

Altair Drive ©2021-2022

Altair® Grid Engine® ©2001, 2011-2022

Altair® DesignAI™ ©2022

Altair Breeze™ ©2022

Altair Data Analytics Products

Altair® Knowledge Studio® ©1994-2022 Altair Engineering Canada, Ltd., ©2018-2022 Altair Engineering Inc.

Altair® Knowledge Studio® for Apache Spark ©1994-2022 Altair Engineering Canada, Ltd., ©2018-2022 Altair Engineering Inc.

Altair® Knowledge Seeker™ ©1994-2022 Altair Engineering Canada, Ltd., ©2018-2022 Altair Engineering Inc.

Altair® Knowledge Hub™ ©2017-2022 Datawatch Corporation, ©2018-2022 Altair Engineering Inc.

Altair® Monarch® ©1996-2022 Datawatch Corporation, ©2018-2022 Altair Engineering Inc.

Altair® Panopticon™ ©2004-2022 Datawatch Corporation, ©2018-2022 Altair Engineering Inc.

Altair® SmartWorks™ ©2021-2022

Altair SmartCore™ ©2011-2022

Altair SmartEdge™ ©2011-2022

Altair SmartSight™ ©2011-2022

Altair One™ ©1994-2022

2022

January 10, 2022

Technical Support

Altair provides comprehensive software support via web FAQs, tutorials, training classes, telephone, and e-mail.

Altair One Customer Portal

Altair One (https://altairone.com/) is Altair's customer portal giving you access to product downloads, a Knowledge Base, and customer support. We recommend that all users create an Altair One account and use it as their primary portal for everything Altair.

When your Altair One account is set up, you can access the Altair support page via this link: www.altair.com/customer-support/

Altair Community

Participate in an online community where you can share insights, collaborate with colleagues and peers, and find more ways to take full advantage of Altair's products.

Visit the Altair Community (https://community.altair.com/community) where you can access online discussions, a knowledge base of product information, and an online form to contact Support. These valuable resources help you discover, learn and grow, all while having the opportunity to network with fellow explorers like yourself.

Altair Training Classes

Altair's in-person, online, and self-paced trainings provide hands-on introduction to our products, focusing on overall functionality. Trainings are conducted at our corporate and regional offices or at your facility.

For more information visit: https://learn.altair.com/

If you are interested in training at your facility, contact your account manager for more details. If you do not know who your account manager is, contact your local support office and they will connect you with your account manager.

Telephone and E-mail

If you are unable to contact Altair support via the customer portal, you may reach out to technical support via phone or e-mail. Use the following table as a reference to locate the support office for your region.

When contacting Altair support, specify the product and version number you are using along with a detailed description of the problem. It is beneficial for the support engineer to know what type of workstation, operating system, RAM, and graphics board you have, so include that in your communication.

Location	Telephone	E-mail
Australia	+61 3 9866 5557	anzsupport@altair.com
Brazil	+55 113 884 0414	br_support@altair.com

Location	Telephone	E-mail
Canada	+1 416 447 6463	support@altairengineering.ca
China	+86 400 619 6186	support@altair.com.cn
France	+33 141 33 0992	francesupport@altair.com
Germany	+49 703 162 0822	hwsupport@altair.de
Greece	+30 231 047 3311	eesupport@altair.com
India	+91 806 629 4500 +1 800 425 0234 (toll free)	support@india.altair.com
Israel		israelsupport@altair.com
Italy	+39 800 905 595	support@altairengineering.it
Japan	+81 3 6225 5830	support@altairjp.co.jp
Malaysia	+60 32 742 7890	aseansupport@altair.com
Mexico	+52 55 5658 6808	mx-support@altair.com
New Zealand	+64 9 413 7981	anzsupport@altair.com
South Africa	+27 21 831 1500	support@altair.co.za
South Korea	+82 704 050 9200	support@altair.co.kr
Spain	+34 910 810 080	support-spain@altair.com
Sweden	+46 46 460 2828	support@altair.se
United Kingdom	+44 192 646 8600	support@uk.altair.com
United States	+1 248 614 2425	hwsupport@altair.com

If your company is being serviced by an Altair partner, you can find that information on our web site at https://www.altair.com/PartnerSearch/.

See www.altair.com for complete information on Altair, our team, and our products.

User Guide

1

This chapter covers the following:

- Run Radioss (p. 11)
- Explicit Structural Finite Element Analysis (p. 37)
- Implicit Structural Finite Element Analysis (p. 457)
- Fluid and Fluid-Structure Simulation (p. 487)
- Smooth Particle Hydrodynamics (SPH) (p. 518)
- Multi-Domain Technique (p. 533)
- Design Optimization (p. 555)
- Error Message Database (p. 560)
- Engine Error Messages (p. 571)
- Warning Message Database (p. 577)

Run Radioss

The Radioss solver can be executed using different methods described here.

From Altair Compute Console (ACC)

The Altair Compute Console (ACC) is the easiest way to launch a Radioss computation.

It includes an interactive GUI for selecting input files, defining run options, submit multiple solver runs using a queue, schedule a delay, stop a job, and has all the environment variables needed to run Radioss predefined.

On Windows, the Compute Console (ACC) can be launched using **Start** > **2022** > **Compute Console**.

The Compute Console (ACC) can be launched from within the software by selecting the **Help** > **Compute Console Manual**.

On all platforms, this utility can be started from the command line

Linux	`<install_dir>/altair/scripts/radioss -gui`
Windows	`<install_dir>/hwsolvers/scripts/radioss.bat -gui`

Input Filename

Based on the input filename, the Compute Console (ACC) automatically detects if it should run the Starter or Engine.

- By default, all the Starter and Engine files in the directory of the input file will be ran. Multiple Engine files can easily be ran by placing them in the same directory as the Starter file.
- If an Engine file is selected, then the Engine file will be ran followed by any other Engine files in the directory.
- If the `-onestep` option is used, only the selected input file will be ran. Only the Starter is ran, if the input file is a Starter file and only the Engine file is ran, if the input file is an Engine file.

Commonly Used Run Options

Option	Argument	Description
`-onestep`	N/A	Disable auto continuation to next Engine, run only Starter for single file format.
`-nt`	Number of threads	Number of SMP threads per SPMD MPI domain. (Example: `radioss ROOTname_0000.rad -nt 2`)
`-np`	Number of SPMD MPI domains	Number of Radioss SPMD MPI domains.

Option	Argument	Description
		(Example: `radioss ROOTname_0000.rad -np 12`)

All Available Run Options

Any arguments containing spaces or special characters must be quoted in {}, for example:

```
-mpipath {C:/Program Files/MPI}
```

File paths on Windows may use backward "\" or forward slash "/", but must be within quotes when using a backslash "\".

Option	Argument	Description	Available Platforms
`-altver`		Suffix for alternate solver version.	All Platforms
`-appfile`	N/A	Spec file with some MPI for SPMD run.	All Platforms
`-args`		Extra arguments that are passed to the solver. Example: `\"\{-x\} \{-f=C:/file name/path\}\"`	All Platforms
`-check`		Check the model for warnings and errors. Disable domain decomposition and restart file writing.	All Platforms
`-checkpoint`	N/A	Run Radioss Engine starting from checkpoint savefile.	All Platforms
`-debug`	`1`, `2`	Output scripting debugging information.	All Platforms
`-delay`	Number of seconds	Delays the start of a Radioss run for the specified number of seconds. This functionality does not use licenses, computer memory or CPU before the start of the run (the delay expires). This option may be used to wait for any computing resources which are currently usable but are not yet freed by the previous run. If the previous run crashed, then the license may still be locked by the license server depending on the timeout value, or the memory may still be reserved by the Operating System.	All Platforms

Option	Argument	Description	Available Platforms
		Note: 1. The `-delay` option applies to the current job. If the job is submitted to the Compute Console (ACC) queue, then the delay will start after the Compute Console (ACC) releases this job to run. 2. If the run is started using the Compute Console (ACC), the Schedule delay option should be used to delay starting the queue.	
`-dylib`	`FILE`	Set name to the dynamic library for Radioss User Subroutines	All Platforms
`-hostfile`	`TEXT FILE`	Text file that list the hosts and number of cores for a multi-computer (node) run. Refer to *MPI document* for exact format.	All Platforms
`-HSTP_READ`	NA	Read an `hst_input.hstp` file and replace the input deck's `/PARAMETER` with the ones defined in the read file.	All Platforms
`-HSTP_WRITE`	NA	Write an `<root_name>_0000.rad2hst` file containing the model `/PARAMETER` information.	All Platforms
`-mpi`	`i , pl, ms`	Specify which MPI software is used for multiprocessor (SPMD) run. *i* Intel MPI *pl* IBM Platform-MPI (formerly HP-MPI) *ms* Microsoft MPI	Not all platforms support all MPI software.

Option	Argument	Description	Available Platforms
		Note: An argument for `-mpi` is optional. If an argument is not specified, Intel MPI is used by default.	
`-mpipath`	`DIRECTORY`	Specify the path where `mpirun` is located. (Altair Simulation includes the MPI software so this typically is not needed).	All Platforms
`-noh3d`	N/A	Do not generate `h3d` file from Radioss animation files when the simulation has ended.	All Platforms
`-norst`	N/A	Remove all restart (`*.rst`) files after the simulation has ended.	All Platforms
`-np` or `-nspmd`	Number of SPMD MPI domains	Number of Radioss SPMD MPI domains. Example: `radioss -np 4 ROOTname_0000.rad`	All Platforms
`-nt` or `-nthread`	Number of thread	Number of SMP threads per SPMD MPI domain. Example: `radioss -i ROOTname_0000.rad -np 4 -nt 2`	All Platforms
`-nt_r2r`		Number of cpu threads for multi-domain process.	All Platforms
`-onestep`	N/A	Disable auto continuation to next Engine or mod file, run only Starter for single file format.	All Platforms
`-outfile`	`DIRECTORY`	Defines the output file directory for all files created by the Starter and Engine.	All Platforms
`-radfiles`	N/A	List all files in the current folder with detected types (Starter/Engine/mod).	All Platforms
`-radopt`	`.radopt` file	Option to run Radioss optimization with OptiStruct. It specifies to request an optimization run for an Radioss input deck.	All Platforms

Option	Argument	Description	Available Platforms
		Note: The Radioss Starter and input files supporting the optimization input should be available in the same directory as the `<name>.radopt` file. Refer to Design Optimization for more information.	
`-rseed`	`seed`	Optional value: Set the `seed` value for nodal random noise (0 ≤ `seed` < 1).	All Platforms
`-rxalea`	`xalea`	Activation of nodal random noise with `xalea` value (`xalea` ≥ 0).	All Platforms
`-sp`	N/A	Specify to run a single precision version of Radioss.	All Platforms
`-v`		Required version of solver if multiple are installed. If not specified, newest version installed is used. Example: `radioss -v 14.0.220 ROOTname_0000.rad`	All Platforms

Solver Control

When the **Use solver control** option is checked in the initial Compute Console (ACC) input screen, a solver view window will appear when the analysis starts running. This will allow the user to select one of the options (INFO, STOP, KILL, H3D, ANIM, RFILE, and CHKP) and click **Send Command**. For more details on all of these options, refer to Control File (C-File).

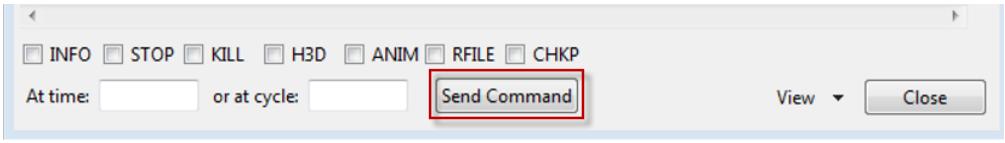


Figure 1:

Note: The ANIM or H3D option can be used to create an extra animation file at the current or specified time or cycle. This is very useful when wanting to see the status of a simulation. The STOP option can stop the analysis and create a restart file that can later be used to continue the solution.

From an Altair Simulation Script

Launch Radioss from a command line using the script included in HyperWorks.

This script should be used when running Radioss on high-performance computing cluster via a job scheduler because many environment variables are already defined so it simplifies the setup. All the options available in the Compute Console (ACC) can be used from the command line.

To run on Linux from the command line, type the following:

```
<install_dir>/altair/scripts/radioss  "filename" -option argument
```

To run from a Windows DOS prompt, type the following:

```
<install_dir>\hwsolvers\scripts\radioss.bat "filename" -option argument
```

The script options are described in All Available Run Options.

From HyperMesh

When creating a Radioss finite element model in HyperMesh, the simulation can be ran directly in HyperMesh by going to the Radioss panel.

The panels can be accessed through the Analysis page, through the Utility menu, or through the Applications menu. The panel will ask for the input filename. After clicking the **Solver** button, the model is exported using the given export options. Then the Solver runs the script that is provided locally on the machine. After Solver execution, the results can be viewed in HyperView by clicking HyperView.

Note: On Linux, when running Radioss from HyperMesh, a shell is spawned with the DISPLAY setting `<hostname>:0.0`. If this is different from the DISPLAY setting for HyperMesh, 25/38 Altair units (in addition to the 21 Altair units being used for HyperMesh) will be checked out. To avoid the checking out of additional units, be sure that the DISPLAY is set to `<hostname>:0.0` before starting HyperMesh.

Run Executables Directly

The Radioss executables can be executed directly, but various environment variables need to be defined.

It is, therefore, recommended and easier to utilize the Radioss script included in Altair Simulation when launching jobs on a computer server. If it is still necessary to run the executables directly, the following environment variables must be set.

`RADFLEX_PATH = <install_dir>/hwsolvers/common/bin/<platform>/`

`RAD_H3D_PATH = <install_dir>/hwsolvers/radioss/lib/<platform>/`

`HWSOLVERS_TEMPLEX = <install_dir>/hwsolvers/common/bin/<platform>/`

`ABF_PATH = <install_dir>/hwsolvers/radioss/bin/<platform>/`

`RAD_CFG_PATH = <install_dir>/hwsolvers/radioss/cfg/`

The per thread stacksize environment variable should also be defined when using the SMP version.

Linux	`KMP_STACKSIZE`=400m
Windows	`KMP_STACKSIZE`=400m

Windows Only

Update environment variable, `PATH=%PATH%;\<install_dir>\hwsolvers\common\bin\win64;\<install_dir>\hwsolvers\radioss\lib\win64`

Linux Only

`LD_LIBRARY_PATH=<install_dir>/hwsolvers/common/bin/linux64/:<install_dir>/hwsolvers/radioss/lib/linux64`

The stacksize memory should be set to unlimited using these commands depending on login shell.

Using csh shell

limit stacksize unlimited

Using sh shell

unlim `-s` unlimited

The Radioss executables can be found in `<install_dir>\hwsolvers\radioss\bin\<platform>\`.

Now to run the Radioss Starter and Engine:

```
./s_<version>_<platform> -i <arguments> <prefix>_0000.rad
./e_<version>_<platform> -i <arguments> <prefix>_0001.rad
```

Refer to the *Altair Simulation Advanced Installation Guide* for information on using Radioss with MPI software.

Starter and Engine Exit Codes

The Starter and Engine provide exit codes that can be used by scripts.

The values are:

0

Normal termination.

2

Model or computation error.

3 or 128 + Signal number

Process failure (core dump, segmentation fault).

In Linux the error code is set by the operating system: 136 (floating point exception) or 139 (segmentation fault). The exit code can be obtain using the '`$?`' command in a Linux shell script.

```
#!/bin/bash
s_2021.1_linux64 -input [Starter_input]_0000.rad -np 96
Exit_code=$?
echo $Exit_code
```

In Windows, use the "`errorlevel`" variable in a script.

```
s_2021.1_win64.exe -input [Starter_input]_0000.rad -np 96
echo %errorlevel%
```

Starter Command Line Arguments

The Radioss Starter supports the following command line arguments.

Argument	Short Form	Description
`-help`	`-h`	Print help message
`-version`	`-v`	Print Radioss release information
`-input [FILE]`	`-i`	Set Radioss Starter input file
`-nspmd [INTEGER]`	`-np`	Set number of SPMD domains
`-nthread [INTEGER]`	`-nt`	Set number of threads per SPMD domains
`-notrap`		Disable error trapping
`-check`		Option to check the model. Disable domain decomposition and restart file writing.
`-outfile=[output path]`		Defines the output file directory for all files created.
`-HSTP_WRITE`		Write an `<root_name>_0000.rad2hst` file containing model the `/PARAMETER` information.
`-HSTP_READ`		Read in a `hst_input.hstp` file and replace the input deck's `/PARAMETER` with the ones defined in the `.hstp` file.
`-rxalea [REAL]`		Activation of nodal random noise with a [value] `xalea`.
`-rseed [REAL]`		Optional value: Set the `seed` value for nodal random noise.
`-dylib [FILE]`		Set name of the dynamic library for Radioss User Subroutines.
`-mds_libpath [PATH]`		Set path to the dynamic library for Multiscale Designer.

-help Argument

Prints help information on the command line arguments.

Radioss Starter exits after the printout.

-version Argument

Prints Radioss release information.

Information printed includes:

- Radioss title
- Radflex name
- Version to use
- Build information (date, time of build and build tag)

Radioss Starter exits after the printout.

-input [FILE] Argument

Sets the Radioss Starter input file.

The `FILE` argument must be a Radioss Starter input file with the following format: `[ROOTNAME]_[RUN NUMBER].rad`

where:

`ROOTNAME` is the dataset rootname

`RUN NUMBER` is the run number expressed in four numbers.

```
e.g. CRA2V51_0000.rad
ROOTNAME: CRA2V51
RUN NUMBER: 0000
```

Note:

- `ROOTNAME` and `RUN NUMBER` are extracted from the input file.
- The dataset `ROOTNAME` and `RUN NUMBER` settings in /BEGIN are ignored when the `-input` command option is used.
- If the file name doesn't have the correct format, the file is rejected.
- The file is open. Standard input is no longer used.
- If `-input` is not set, Radioss Starter opens standard input to read input.

Example

```
[user@machine]$  ./s_2021.1_linux64 -input CRA2V51_0000.rad
```

-nspmd [SPMD DOMAIN NUMBER] Argument

Sets the number of SPMD domains.

Where:

SPMD DOMAIN NUMBER: INTEGER is the number of SPMD domains

Example

```
[user@machine]$ ./s_2021.1_linux64 -nspmd 4
```

Note:

- If the SPMD DOMAIN NUMBER value is not properly set, Radioss Starter stops with error message.
- If `-nspmd` is set, the number of domains set in `/SPMD` input card is ignored. `/SPMD` card can be omitted, if other parameters are kept as default settings.
- The total number of processes used for the Radioss Engine computation will be equal to SPMD DOMAIN NUMBER * SMP THREAD NUMBER .

-nthread [SMP THREAD NUMBER] Argument

Sets the number of SMP threads per SPMD domain. Each SPMD domain will be computed using a multi-threaded process composed of SMP THREAD NUMBER.

Where, SMP THREAD NUMBER: INTEGER is the number of SMP threads per SPMD domain.

Example:

```
[user@machine]$ ./s_2021.1_linux64 -nthread 2
```

Note:

- If the SMP THREAD NUMBER value is not properly set, Radioss Starter stops with error message.
- If `-nthread` is set, the number of threads set in /SPMD input card is ignored. `/SPMD` card can be omitted if other parameters are kept as default settings.
- The total number of processes used for the Radioss Engine computation will be equal to SPMD DOMAIN NUMBER * SMP THREAD NUMBER.

-notrap Argument

Disables the trapping error of Radioss Starter.

By default, in case of error, Radioss Starter will display an error message that corresponds to the error occurred and stop properly. If `-notrap` is used, Radioss Starter will continue and potentially core dumps where the error occurs. This behavior is sometimes helpful to trace the error with external tools.

General Behavior, Error Handling

1. `-version`
2. `-help`
3. `-nspmd`, `-input`, `-nthread`, `-notrap`

> **Note:** Arguments 1, 2 and 3 cannot be used together.
>
> `-nspmd`, `-nthread`, `-notrap` and `-input` can be used together; and the order does not matter.

If an error is encountered like:

- Missing argument to `-input` or `-nspmd` or `-nthread`
- `-input` argument is not a Radioss file format
- `-nspmd` or `-nthread` argument are not an integer value
- unknown command line argument used

An error message is printed out with the `-help` printouts.

-check Argument

Option to stop the Starter after model check.

The domain decomposition calculation and writing of the restart files (`.rst`) is not done. This reduces the time to run for the Starter model to run when doing a model check.

-outfile=[output path] Argument

The Starter writes all output and restart files in the defined output path.

The output path can be absolute or relative. If there is a space in the output path then, quotes must be used in the path. On Windows, either "/" or "\" can be used between directories.

Example

Linux:

```
s_2022_linux64 -i TENSI36_0000.rad -outfile=/storage/files/run1
s_2022_linux64 -i TENSI36_0000.rad -outfile="../my directory"
s_2022_linux64 -i TENSI36_0000.rad -outfile="/storage/files/my directory"
```

Windows:

```
s_2022_win64.exe -i TENSI36_0000.rad -outfile=c:\storage\files\run1
s_2022_win64.exe -i TENSI36_0000.rad -outfile="../my directory"
s_2022_win64.exe -i TENSI36_0000.rad -outfile="c:/storage/files/my directory"
```

-HSTP_WRITE Argument

Write a file called `<root_name>_0000.rad2hst` containing all model parameters (`/PARAMETER`) information defined in the model (`_0000.rad`).

The `<root_name>_0000.rad2hst` file for HyperStudy with *schema*="`altair_pdd`" and *schemaVersion*="`hstp_v_5`".

-HSTP_READ Argument

Replace parameter (`/PARAMETER`) values defined in the model (`_0000.rad`) by the ones read in the `hst_input.hstp` file.

The file `hst_input.hstp` is an xml file for HyperStudy with *schema*="`altair_pdd`" and *schemaVersion*="`hstp_v_5`".

-rseed [seed] Argument

Option to define the seed for random noise on the node coordinate with the value [`seed`], a real value between 0 and 1.

-rxalea [xalea] Argument

Option to initialize random noise on the node coordinates with the value [`xalea`], a positive real value.

-dylib [FILE] Argument

Option to add the file name with full or relative path of Radioss User Subroutines for the Radioss execution.

-mds libpath [PATH] Argument

Option to specify the path of the dynamic library for Multiscale Designer material laws. By default, Radioss uses the path defined in the HyperWorks environment.

Engine Command Line Argument

Radioss Engine HMPP executables are available (as of version 11.0) with MPI and are fully Hybrid capable mixing SPMD and SMP parallelization.

Simplified executables also exist, which are only SMP capable but do not require MPI.

To distinguish easily between these two types of executables, MPI executables have a `_mpp` suffix.

- For binaries compiled with MPI, they need to be run through the `mpirun` command. The general form is as follows:

```
mpirun [list of mpirun option] engine_executable_mpp [list of engine option]
```

> **Note:** The list of `mpirun` options depends on the MPI vendor. Refer to the MPI documentation.

- For binaries compiled without MPI:

```
engine_executable [list of engine option]
```

> **Note:** The list of Radioss Engine options is the same for both types of executables.

Radioss Engine supports the following command line arguments. Each argument has a long and a short form. These are executable arguments.

Argument	Short Form	Description
`-help`	`-h`	Print help message
`-version`	`-v`	Print Radioss release information
`-input [FILE]`	`-i`	Set Radioss Engine input file
`-nthread [INTEGER]`	`-nt`	Set Number of SMP threads per SPMD domain
`-notrap`		Disable error trapping
`-norst`		Disable restart `*.rst` file writing during and at the end of the computation
`-outfile=[output path]`		Defines the output file directory for all files created.
`-dylib [FILE]`		Set name of the dynamic library for Radioss User Subroutines.
`-mds_libpath [PATH]`		Set path to the dynamic library for Multiscale Designer.

-help Argument

Prints help information on the command line arguments.

Radioss Engine exits after the printout.

-version Argument

Prints Radioss release information.

Information includes:

- Radioss title
- Radflex name
- Version to use
- Build information (date, time of build and build tag)

Radioss Engine exits after the printout.

-input [FILE] Argument

Sets the Radioss Engine input file.

The `FILE` argument must be a Radioss Engine input file with the following format: `[ROOTNAME]_[RUN NUMBER].rad`

where:

- `ROOTNAME` is the dataset rootname
- `RUN NUMBER` is the run number expressed in four numbers.

> **Note:**
>
> - `ROOTNAME` and `RUN NUMBER` are extracted from the input file.
> - The dataset `ROOTNAME` and `RUN NUMBER` settings in /RUN are ignored, when the `-input` command option is used.
> - If the file name doesn't have the correct format, the file is rejected.
> - The file is open. Standard input is no longer used.
> - If `-input` is not set, Radioss Engine opens standard input to read input.

Example

```
[user@machine]$  mpirun -np 4 ./e_2019_linux64 impi -nt 2 -input CRA2V51_0001.rad
**************************************************************************
**                                                                      **
**                                                                      **
**                         RADIOSS ENGINE 2019                          **
**                                                                      **
**            Non-linear Finite Element Analysis Software               **
**                   from Altair Engineering, Inc.                      **
**                                                                      **
**                                                                      **
**             Linux 64 bits, Intel compiler, Intel MPI                 **
**                                                                      **
**                                                                      **
**                                                                      **
** Build tag: 0479348_2413000                                           **
**************************************************************************
**  COPYRIGHT (C) 1986-2019                Altair Engineering, Inc.   **
** All Rights Reserved.  Copyright notice does not imply publication.   **
** Contains trade secrets of Altair Engineering Inc.                    **
** Decompilation or disassembly of this software strictly prohibited.   **
**************************************************************************
```

```
ROOT: CRA2V51  RESTART: 0001
NUMBER OF HMPP PROCESSES    8
17/10/2008
NC=        0 T= 0.0000E+00 DT= 1.7916E-03 ERR=  0.0% DM/M= 0.0000E+00
    ANIMATION FILE: CRA2V51A001 WRITTEN
NC=      100 T= 1.7826E-01 DT= 1.7700E-03 ERR=  0.0% DM/M= 0.0000E+00
...
NC=   13700 T= 1.9871E+01 DT= 1.4032E-03 ERR= -1.2% DM/M= 0.0000E+00

                          ** CPU USER TIME **

#PROC CONT.SORT CONT.F ELEMENT MAT KIN.COND INTEGR. I/O TASK0 ASSEMB RESOL
1 .0000E+00 .0000E+00 .1144E+02 .2910E+01 .7000E-01 .5600E+00 .3100E+00 .5000E+00 .3700E+00 .1364E+02
2 .0000E+00 .0000E+00 .1102E+02 .2870E+01 .7000E-01 .5600E+00 .0000E+00 .0000E+00 .3300E+00 .1364E+02


ELAPSED TIME :      7.79 s

    RESTART FILE: CRA2V51_0001_0000.rst WRITTEN
    NORMAL TERMINATION
    TOTAL NUMBER OF CYCLES:   13793
[user@machine]$
```

-nthread [SMP THREAD NUMBER] Argument

Sets the number of SMP threads per SPMD domain. Each SPMD domain will be computed using a multi-threaded process composed of SMP THREAD NUMBER.

Where, SMP THREAD NUMBER: INTEGER is the number of SMP threads per SPMD domain.

Example

```
[user@machine]$ mpirun -np 4 ./e_2019_linux64 impi -nthread 2 -input CRA2V51_0001.rad
```

Note:

- If the INTEGER value is not properly set, the Radioss Starter stops with error message.
- If `-nthread` is set, the number of threads defined in the Starter run (`/SPMD` input card, `-nthread` command line) or previous Engine run is ignored.
- The environment variable OMP_NUM_THREADS can also be used to change the number of threads. `-nthread` remains prior to OMP_NUM_THREADS.
- For performance reason, it is recommended to use OMP_NUM_THREADS instead of `-nt`. This informs the system of the true number of threads used.
- The total number of processes used for the Radioss Engine computation will be equal to SPMD DOMAIN NUMBER * SMP THREAD NUMBER.

General Behavior, Error Handling

1. `-version`
2. `-help`
3. `-input`, `-nthread`, `-notrap`

The argument order in the command line does not matter.

If an error is encountered like:

- Missing argument to `-input` or `-nthread`

- `-input` argument is not a Radioss file format
- `-nthread` argument is not an integer

An error message is printed with the `-help` printouts.

MPI and Radioss Options Coherency

The `-nspmd` value (or *Nspmd* field of `/SPMD` Starter input card) must match the `mpirun -np` value. If this is not the case, Radioss Engine will stop with an appropriate error message.

Example of incorrect commands:

```
[user@machine]$ ./s_2019_linux64 -nspmd 4 -input CRA2V51_0000.rad
```

```
[user@machine]$ mpirun -np 2 ./e_2019_linux64 impi -input CRA2V51_0001.rad
```

In this case, Radioss will display an error message in standard output and listing file:

```
THE REQUIRED NUMBER OF MPI PROCESSES DOES NOT MATCH MPIRUN
  PLEASE, RUN WITH THE PROPER NUMBER OF MPI PROCESSES
  REQUIRED  (NSPMD)       =           4
  AVAILABLE (mpirun -np)  =           2
  E R R O R     T E R M I N A T I O N
  TOTAL NUMBER OF CYCLES  :           0
```

Example of correct commands:

```
[user@machine]$ ./s_2019_linux64 -nspmd 4 -input CRA2V51_0000.rad
[user@machine]$ mpirun -np 4 ./e_2019_linux64 impi -input CRA2V51_0001.rad
```

-notrap Argument

Add this option to force core dump.

This is a useful option to get call stack traceback.

-norst Argument

Option to disable restart `*.rst` file writing during and at the end of the computation.

This makes the Engine run faster because of the reduction in files written.

-outfile=[output path] Argument

The Engine writes all output and restart files in the defined output path.

The output path can be absolute or relative. If there is a space in the output path then, quotes must be used in the path. On Windows, either "/" or "\" can be used between directories.

Example

Linux:

```
e_2022_linux64 -i TENSI36_0000.rad -outfile=/storage/files/run1
```

```
e_2022_linux64 -i TENSI36_0000.rad -outfile="../my directory"
e_2022_linux64 -i TENSI36_0000.rad -outfile="/storage/files/my directory"
```

Windows:

```
e_2022_win64.exe -i TENSI36_0000.rad -outfile=c:\storage\files\run1
e_2022_win64.exe -i TENSI36_0000.rad -outfile="../my directory"
e_2022_win64.exe -i TENSI36_0000.rad -outfile="c:/storage/files/my directory
```

-dylib [FILE] Argument

Option to add the file name with full or relative path of Radioss User Subroutines for the Radioss execution.

-mds libpath [PATH] Argument

Option to specify the path of the dynamic library for Multiscale Designer material laws. By default, Radioss uses the path defined in the HyperWorks environment.

Solution Process

There are two parts to the Radioss simulation, the Starter and Engine. The Starter is an input data check and must successfully complete without errors before the simulation can be completed in the Engine.

Starter and Engine

The Radioss Starter takes as input the model or commonly called Starter input file `runname_0000.rad` and creates the Starter output file `runname_0000.out`. The Radioss Starter is responsible for checking the model consistency and reporting any errors or warnings in the output file. If there are no errors in the model, the Radioss Starter creates initial restart file(s), `runname_0000_CPU#.rst`. There is one restart file created for each SPMD MPI domain requested for the solution.

The second part of the Radioss computation is called the Radioss Engine. The Radioss Engine takes as input the Radioss Engine file, `runname_0001.rad` and the initial restart file(s) created by the Radioss Starter. The Radioss Engine files describes the solution control and output for the simulation. While the Radioss Engine is running, an Engine output file, `runname_0001.out`, is created which contains statistics about the simulation including time, time step, current system energeies, energy error, and mass error.

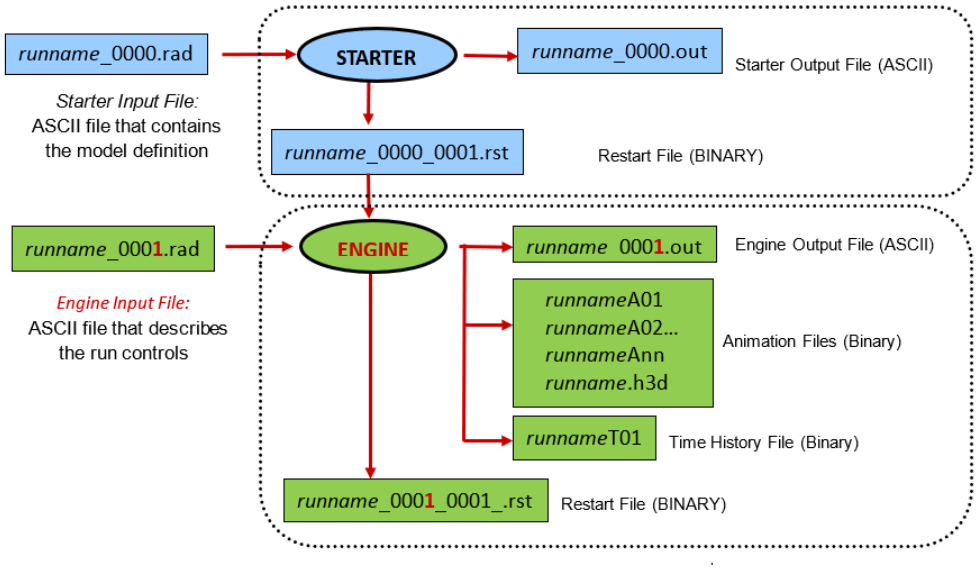


Figure 2: Radioss Calculation Process when a Single MPI Domain is Used

Memory and Disk Requirements

Memory is allocated automatically by the Radioss Starter and the amount used is listed in the Starter output file.

```
STARTER MEMORY USAGE       1486 MB
```

Next the Starter gives a memory estimation and restart file size for every MPI process.

```
     LOCAL ENGINE STORAGE EVALUATION FOR SPMD PROCESSOR     1
     ------------------------------------------------------
        MEMORY USED FOR REALS           80 MB
        MEMORY USED FOR INTEGERS        35 MB
        TOTAL MEMORY EVALUATION        115 MB
        RESTART FILE SIZE     60091KB
...
     LOCAL ENGINE STORAGE EVALUATION FOR SPMD PROCESSOR    18
     ------------------------------------------------------
        MEMORY USED FOR REALS           83 MB
        MEMORY USED FOR INTEGERS        38 MB
        TOTAL MEMORY EVALUATION        122 MB

        RESTART FILE SIZE     63349KB
```

After the Radioss Engine is done running, the Engine output file contains details about actual memory and disk space used by a simulation.

```
                  ** MEMORY USAGE STATISTICS **

 TOTAL MEMORY USED .........................:    10284 MB
 MAXIMUM MEMORY PER PROCESSOR...............:      605 MB
 MINIMUM MEMORY PER PROCESSOR...............:      562 MB
 AVERAGE MEMORY PER PROCESSOR...............:      571 MB


                  ** DISK USAGE STATISTICS **

 TOTAL DISK SPACE USED .....................:    1421485 KB
 ANIMATION/TH/OUTP SIZE ....................:     200950 KB
 RESTART FILE SIZE .........................:    1220535 KB
```

Recommendations for Using Multiple Cores

Modern CPUs typically have multiple cores which can be utilized to reduce solution times.

Three different parallel computing methods are available, Shared-Memory Parallelism (SMP), single program multiple data (SPMD) and Hybrid (a combination of SMP and SPMD) to solve a problem.

The SMP method uses the specified cores (often called *threads*) to solve the whole model.

The SPMD method splits a model into separate domains which are solved separately and communication between the domains is handled by a message passing interface (MPI) software.

In Hybrid mode, the model is split into separate domains and then multiple cores (*threads*) are used to solve each domain. Hybrid mode reduces the amount of communication because there are less domains as compared to using only the SPMD method. This is useful when using a very large number of cores on a computer cluster or when using two workstations connect via a slow network.

The number of cores to use for each method is specified using the Compute Console (ACC) input option:

- `-nt` NumThreads for SMP
- `-np` NumDomains for SPMD
- `-nt` NumThreads `-np` NumDomains for Hybrid

Recommendations depend on the computer setup available.

Single Workstation

- Use only SPMD by specifying the run option, `-np` NumDomains
- NumDomains should be the number of cores available on the workstations' CPU
- If Hyperthreading is enable for the CPU, the computer will appear to have twice as many cores as listed in the CPU specification. These extra cores are virtual and thus provide only a small amount of speedup. A 5% speedup can be obtained by using these extra cores but extra licenses will be used because the number of license depends on the number of cores requested. If utilizing hyperthreading then the hybrid mode with 2 SMP threads, `-nt 2`, will give the best speedup. For example if an 8 core CPU is being used then, `radioss -nt 2 -np 8 model_0000.rad`.

Computer Cluster

- When the number of cores used < 256, the fastest solution times can be obtained by using only SPMD via, `-np` NumDomains. Note the default value of `-nt`=1 so it does not need to be included.
- When the number of cores used > 256, hybrid mode should be used, `-nt` NumThreads `-np` NumDomains. NumThreads= 2 and NumDomains= (# cores to be used) / 2.

  For example, if a cluster node contains 2 CPU, with each CPU having 8 cores. To use 512 cores, NumThreads=2 and NumDomains = 512/2=256.
- All the cores available on a compute node should be used and dedicated to the Radioss solution, that is, if a compute node has 16 cores, then the number of cores available to use should be multiples of 16.

- Hyperthreading cores should not be used in a solution and if possible, hyperthreading can be disabled in the system bios.

It is recommended to use a job scheduler, like Altair PBS to manage the cluster.

Two or Three Workstations

It is possible to use two or three workstations connected with gigabit ethernet to solve one Radioss simulation.

- For best results, all machines should be identical hardware or at least have the same CPU and be connected with gigabit ethernet or faster network.
- Hybrid mode can be used to minimize network communication, `-np` NumDomains `-nt` NumThreads. Start by setting NumThreads=2 and NumDomains=(total # cores available on all machines)/ NumThreads. Run a benchmark model and compare the time to running the model on just one workstation.
- Next, increase the NumThreads and rerun the benchmark to see if there is any additional speedup. NumThreads should be ≤ number of cores on 1 CPU. NumDomains must be a multiple of the of number of computers used.

  For example, if using 2 Workstations each with two CPU and each CPU has 8 cores. 2*2*8=32 cores available.

  - `-nt 2 -np 16` = OK
  - `-nt 4 -np 8` = OK
  - `-nt 8 -np 4` = OK
- If hyperthreading is enable for the CPUs, do not use the extra hyperthreading cores.

Additional setup is required as detailed in the *Altair Simulation Advanced Installation Guide*.

Model Size

When using multiple cores to solve a simulation, there must be a reasonable number of elements in the simulation.

A good balance of speedup and throughout is obtained by making sure there are at least 10000 elements in the model for every core used in the solution. So, for a model with 320,000 elements, 320,000/10000=32 cores. Usually, additional speedup can be obtained down to 1000 elements per core.

Use a Different Number of Cores

The Starter can use multiple cores via SMP parallelization.

For extremely large models that will run using a large number of SPMD domains, the Starter will run faster if multiple SMP cores are utilized for domain decomposition and to create the restart files. When using the Compute Console (ACC) or included script, the Starter and Engine must be ran separately

using the `-onestep` option. For example, assuming a very large 3 million element model that will run on 120 cores on a compute server with 12 cores per CPU.

The Starter will use 12 cores to calculate the 120 SPMD domains and create the restart files.

```
radioss -nt 12 -np 120 -onestep model_0000.rad
```

The Engine will use 120 cores using SPMD parallelization.

```
radioss -nt 1 -np 120 -onestep model_0001.rad
```

Advanced Techniques

There are various advanced techniques which can be useful to check on the status of a simulation, stop a simulation, modify some simulation options, or use single precision.

Control File (C-File)

The optional control file is used to get information about a currently running analysis.

While a simulation is running, it can be very useful to find out more information than is printed in the Engine output file. This can be done using a Radioss control file which has the same prefix name as the current Engine input file but ends in `*.ctl`. If the current running Engine file is named `TensileTest_0001.rad`, then the control file should be named `TensileTest_0001.ctl`.

Some reasons to use a control file include:

- Retrieve the current state of the simulation including current simulation time, time step, cycle, energy information, and estimated remaining time
- Stop the computation immediately, or at a specified time or cycle number
- Create an animation immediately, or at a specified time or cycle number

If a simulation is launched in the Altair Compute Console with solver control options checked, the following dialog will appear and a control file will be created automatically using the selected options when the **Send Command** button is selected.

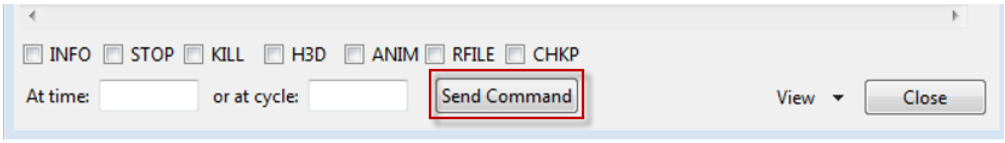


Figure 3:

If the Compute Console (ACC) is not being used, the control file can be created using a text editor and saved in the directory where Radioss is writing the Engine output file.

The following options are available to be used in the control file with each option placed on its own line. Unless /TIME or /CYCLE is included, the action is taken immediately after Radioss reads the control file. Otherwise, the action is taken at the /TIME or /CYCLE specified.

Action	Result
/ANIM	Create an extra animation file (`A###`). The output frequency defined in the Engine file will not change when this option is used.
/CHKPT	Create a file named `CHECK_DATA` which contains /RERUN commands to continue a simulation if it is stopped. Usually used in combination with /STOP to stop the simulation. Not available with the implicit solution.
/CYCLE/Nbcycle	The control file commands will be executed at the specified cycle number.

Action	Result
/H3D	Write animation data to the `*.h3d` file. The output frequency defined in the Engine file will not change when this option is used.
/INFO	Returns information on current cycle, current global energies, current time step. This information is always written for all options.
/KILL	Kill the simulation and do not create a restart file.
/RFILE	Create a restart file.
/STOP	Stop the simulation and create a restart file, so it can be restarted.
/TIME/timeValue	The other control file commands will be executed at the specified simulation time.

When Radioss detects the control file, the current status of the simulation is written in the same control file using the options were entered.

For example, a simulation is running on remote system and thus not started using the Altair Compute Console. Since the Engine input file is named, `TensileTest_0001.rad`, a control file named `TensileTest_0001.ctl` is created with the following information and uploaded to the working directory of the simulation.

```
/TIME/.1
/ANIM
/CHKPT
/STOP
```

When Radioss detects the control file, the following information is written into the same `TensileTest_0001.ctl` file with information about the current state of the simulation and execution of the control file commands. At t=0.1 in the simulation, Radioss will create an animation file, a `CHECK_DATA` file, a restart file, and stop the simulation.

```
RUN CONTROL FILE
 ----------------

 WRITE NEXT ANIMATION FILES AT TIME  0.1000000

 WRITE NEXT RESTART FILE AT TIME  0.1000000

 RUN STOP AT TIME  0.1000000


 CURRENT STATE:
 --------------
 CYCLE      =     13040
 TIME       = 0.8153611E-02
 TIME STEP  = 0.6252749E-06SOLID      70
 ENERGY ERROR      =  0.0%
 INTERNAL ENERGY   = 0.1188329E-02
 KINETIC ENERGY    = 0.1050914E-07
 ROT. KIN. ENERGY  =  0.000000
 EXTERNAL WORK     = 0.1188431E-02
 MASS.ERR (M-M0)/M0=  0.000000

 CURRENT ELAPSED TIME    =          4.88 s
```

```
 REMAINING TIME ESTIMATE =        594.04 s
```

Multiple Engine Files

It is possible to define multiple Engine files to change the solution control and output during a simulation.

Some common use cases of this powerful feature include:

1. Initializing gravity or pre-tension before a simulation.
2. Changing write frequency of output files, such as animation files, (`/ANIM/DT`, `/H3D/DT`), time history (`/TFILE`), and restart (`/RFILE`).
3. Changing the time step control options used, (`/DT/Keyword`).
4. Modifying system damping (`/DAMP`, `/DYREL`, or `/KEREL`).
5. Adding or removing boundary conditions (`/BCS/Keyword`) or defining initial conditions (`/INIV/Keyword`).
6. Removing elements (`/DEL/Eltyp`) or turning on or off rigid bodies (`/RBODY/Keyword2`).

Define and Run Multiple Engine Input Files

As mentioned above, the first Engine file is named `runname_0001.rad`. When creating a second Engine file, the number is increased by 1 for the next Engine file to a maximum of 9999 Engine files. So, the second Engine file would be named `runname_0002.rad`. To avoid a warning message, the Engine file number should also match the Run Number listed in the keyword, `/RUN`.

Example: First Engine file, `TensileTest_0001.rad`

```
/RUN/TensileTest/1
5.0
/TFILE
0.005
/ANIM/DT
0.  0.5
/ANIM/TENS/STRESS/MEMB
/PRINT/-1000
/DT/NODA/CST
0.9 1e-6
```

Example: Second Engine file `TensileTest_0002.rad` with changed options.

```
/RUN/TensileTest/2
10.0
/TFILE
0.0005
/ANIM/DT
0.  0.05
/ANIM/TENS/STRESS/MEMB
/PRINT/-10
/DYREL
1.00    0.002
/DT/NODA/CST
0.9 1e-7
```

See Also

RD-E: 1200 Jumping Bicycle

Checkpoint File

A checkpoint file is used to continue a stopped simulation.

If no changes need to be made to the Engine file, a stopped Radioss simulation can be continued by using a checkpoint file named `CHECK_FILE` containing the `/RERUN/Run Name/Run Number` option. The `CHECK_FILE` can be made using a text editor or by stopping the simulation using the Compute Console (ACC), solver control options STOP and CHKPT.

The `CHECK_FILE` can be created using a text editor by copying the original Engine file to the `CHECK_FILE` name, changing the `/RUN` line changed to `/RERUN`, and deleting all other lines from the file. When using the Altair Compute Console or Radioss script to restart the solution, the Starter or Engine file can be selected as the input file and use the same input options and number of processors that were defined before and add the `-checkpoint` option. For example,

```
radioss -np 4 -checkpoint tensileTest_0000.rad
```

If running Radioss directly with the executables, the `CHECK_FILE` should be given as input to the Engine executable.

Single Precision Computation

The single precision version of Radioss is referred to as extended single precision, since most calculations are single precision but some remain double precision to obtain a more accurate result.

By default, Radioss uses a double precision floating-point format when completing a calculation as opposed to other explicit solvers which utilize single precision by default. The double precision format uses from 15 to 17 significant decimal digits precision; whereas, the single precision format uses 6 to 9 significant decimal digits precision.

Although model dependent, the single precision version runs 30% - 40% faster than the double precision version.

When single precision is used in a calculation, there is more round off error in the floating point calculations due to the reduced number of significant digits in single precision. Since explicit calculations calculate one cycle at a time and the next cycle is based on the previous cycles calculations, simulations with a large number of cycles can accumulate rounding error as the simulation progresses resulting in a less precise result.

To minimize round off error, it is recommended to use the default version of Radioss which is double precision. This is most important in simulations that have these characteristics:

- Larger number of solution cycles due to a small time step or long simulation time.
- Small element size
- Airbag simulations
- Implicit simulations
- Fluid and Fluid - Structure interaction simulations, ALE, CEL
- High-speed impact such as ballistic simulations

The easiest way to determine if a simulation will give accurate results using single precision calculations is to compare the single precision results to the double precision results to identify any large differences. The single and double precision results will not be identical, but they should be very close.

Start a Single Precision Simulation

When using the Compute Console (ACC), the single precision version can be launched using the `-sp` option.

```
radioss -np 24 -sp tensile_model_0000.rad
```

If using the Radioss executables directly, the single precision version is ran using the executables that include the "`_sp`" name for both the Radioss Starter and Engine.

```
./s_<version>_<platform>_sp -i <prefix>_0000.rad
```

```
./e_<version>_<platform>_sp -i <prefix>_0001.rad
```

See Also

Run Radioss (FAQ)

Explicit Structural Finite Element Analysis

In this section, available explicit features for different explicit analyses are presented.

Time Step

An explicit is solved by calculating results in small time increments or time steps. The size of the time step depends on many factors but is automatically calculated by Radioss.

Results are calculated for each time step or cycle in a simulation. Therefore, the smaller the time step, the longer a simulation will take to solve because more cycles and calculations are done. As discussed in Dynamic Analysis of the *Radioss Theory Manual*, a direction integration method is used to solve the equations of motion. The direct integration method used in Radioss is derived from Newmark time integration scheme. This method solves the equations of motion using a step-by-step procedure using a numerically stable time step, Δt. Numerical Stability of Undamped Systems of the *Radioss Theory Manual* shows that a system without damping will remain stable if $\Delta t \leq \frac{2}{\omega_{\max}}$. Where, $\omega_{\max}$ is the highest angular frequency in the system. For a discrete finite element simulation, the solution remains stable if the shock wave traveling through the mesh does not travel through more than one element during one time step. In this way, the shock wave does not miss any nodes when traveling through the mesh and thus excites all the frequencies in the finite element mesh. Using the speed of sound in a material c and the characteristic element length l_c of a finite element, the time for the wave to travel across one element length is:

$$\Delta t = \frac{l_c}{c} \tag{1}$$

For the discrete solution to remain stable, the time step should be less than or equal to the time needed for the wave to travel across one element:

$$\Delta t \leq \frac{l_c}{c} \tag{2}$$

This stability criterion is often called the *Courant condition* after the research first done by Courant et al. in 1928. [2]

Element Time Step

The default time step calculation used by Radioss is the element time step.

For one finite element, the element time step can be written as:

$$\Delta t_{Element} = \frac{l_c}{c} \tag{3}$$

Where,

l_c	The element's characteristic length, whose calculation depends on the element type being used and is described in Finite Elements of the *User Guide*.
c	Speed of sound in the material, which can be expressed as $c = \sqrt{\frac{E}{\rho}}$
ρ	Material density
E	Young's modulus

For every simulation cycle, each element's characteristic length and sound speed are used to calculate the critical time step of every element in the model. The time step calculation occurs at the same time as the internal force calculation. To ensure stability, the simulation's cycle time step is found by multiplying the minimum time step of all the elements by a scale factor (default is 0.9). The scale factor is needed to ensure that the simulation remains stable if there is a sudden change in the model stiffness.

The element time step method is applied for the entire model using:

/DT

$\Delta T_{sca}\ \Delta T_{\min}$

Where,

ΔT_{sca}	Scale factor for the element time step
$\Delta T_{\min}$	Minimum time step that activates specific actions listed in /DT

Nodal Time Step

The nodal time step calculates the time step based on the nodal mass and nodal stiffness in the model.

The nodal time step is calculated after the computation of all the internal forces at each node using:

$$\Delta t_{nodal} = \sqrt{\frac{2m}{k}} \tag{4}$$

Where,

m	Nodal mass
k	Equivalent nodal stiffness

The equivalent nodal stiffness is calculated using one half of the eigenvalue from each attached element stiffness matrix. If the node is also involved in contact, the contact stiffness is also included in the equivalent nodal stiffness calculation. The stiffness is calculated during internal force computation. To ensure stability, the simulation's cycle time step is found by multiplying the minimum time step of all the nodes by a scale factor (default is 0.9).

For a high quality finite element mesh, the element time step and nodal time step conditions are nearly identical. Most typical finite element meshes include some elements with poor element quality and for these situations the nodal time step calculation is usually higher than the element time step calculation.

The nodal time step method is applied for the entire model using:

`/DT/NODA`

$\Delta T_{sca} \Delta T_{\min}$

Where,

ΔT_{sca}	Scale factor for the element time step
$\Delta T_{\min}$	Not used and is entered as zero

See Also
/DT/NODA/Keyword3/Iflag (Engine)

Global Time Step

The global time step (GTS) method can be used to calculate the time step of a model based on the natural frequency of the model.

For some models the global time step method results in a higher time step when compared to calculating the time step based on each element or node. An iterative algorithm is used to calculate the global time step which is more expensive than calculating the element or nodal time step. Therefore, the global time step calculation is done every 20 cycles by default.

`/DT/GLOB/Iflag`

ΔT_{sca}

If *Iflag*=1, insert the next line

Tol_GTS Ncycles

Where,

ΔT_{sca}	Scale factor for the element time step
Tol_GTS	Convergence tolerance for the iterative algorithm
Ncycles	Number cycles between the calculation of the global time step

The global time method is useful in models without a large change in stiffness such as drop test simulations.

The Global Time Step option is available for the following features:

- Solid elements: `/TETRA4`, `/TETRA10`, hexa with the flags below
- `/PROP/SOLID`: I_{solid} = 1 (Belytschko), 14 (HA8), 17 (H8C), 24 (HEPH) and 16 (20-noded)
- Shell elements: I_{shell} = 1 (Belytschko-Tsay), 12 (Batoz), and 24 (HEPH)
- Truss element
- Beam element

- Contact interfaces: TYPE7, TYPE24 (without edge-to-edge option), and TYPE2 with $Spot_{flag}$=25 and 27

Contact Interface Time Step

The contact interface time step is calculated in two different ways. First, based on stiffness and second, based on the velocity of the secondary nodes.

The contact interface time step based on stiffness is only calculated with TYPE7, TYPE10, TYPE11, TYPE19, TYPE21, TYPE24 and TYPE25. No matter which time step method is used, nodes that are involved in contacts have an increased stiffness which make it necessary to calculate a nodal contact interface time step using the nodal mass, interface stiffness and element stiffness.

$$\Delta t_{inter\ nodal} = \sqrt{\frac{2M_{nodal}}{\sum\left(K_{interface} + K_{element}\right)}} \tag{5}$$

Where,

M_{nodal}	Nodal mass
$K_{interface}$	Interface stiffness
$K_{element}$	Element stiffness

The kinematic contact time step of interfaces TYPE7, TYPE11 and TYPE19 is used to make sure that a secondary node does not penetrate past a main segment during one cycle:

$$\Delta t_{kin} = \frac{1}{2}\left[\frac{Gap - p}{\frac{dp}{dt}}\right] \tag{6}$$

Where,

Gap	Contact gap
p	Penetration amount
$\frac{dp}{dt}$	Relative velocity of the penetrating secondary node (or line for TYPE11 contact) with respect to the main segment (or line for TYPE11 contact).

Time Step Output from a Model

The time step of the initial model is output to the Starter output file. Whereas the time step of a running model can be output to the animation files.

Initial Model Time Step

Various methods are available to find the time step of a model. Some preprocessors can calculate and display the time step of a mesh. When the Radioss Starter is executed, the time step, based on element

types and nodes is written in the Starter output file `runame_0000.out`. These time step values are calculated using the formulas above and do not include the time step scale factor.

SHELL ELEMENTS TIME STEP

TIME STEP	ELEMENT NUMBER
5.1217130219225E-08	1041
5.1298454158612E-08	1042
5.1402124677458E-08	1038
7.6281740390682E-07	6518
7.7034282889530E-07	7751
7.7069143391558E-07	2839
7.8642021051951E-07	1606
7.9110842135121E-07	4065
7.9440193221468E-07	5299
8.0278552984776E-07	4069
8.2549417437159E-07	3808
8.3369750227134E-07	523
8.3475351235039E-07	10657
8.4319561232916E-07	12290
8.4326249981130E-07	11974
8.4827514817688E-07	11540

TRIANGULAR SHELL ELEMENTS TIME STEP

TIME STEP	ELEMENT NUMBER
4.9044927553079E-08	9279
4.9751012932874E-08	9295
4.1091106922772E-07	9652
4.1464052589915E-07	9401
4.1488518301530E-07	9737
4.3906776879877E-07	9316
4.4561567504973E-07	9567
4.4736076154325E-07	9484
5.9166324535877E-07	25255
6.0141903476523E-07	9385
6.0173722939190E-07	9636
6.2817012535627E-07	26811
6.6506699638492E-07	26816
6.8685110404286E-07	26801
6.8726908423285E-07	26831
6.9001216040665E-07	26793

NODAL TIME STEP (estimation)

TIME STEP	NODE NUMBER
3.2906865424675E-07	341
3.4713722736254E-07	19763
3.4895519582563E-07	392
3.5255651364406E-07	393
3.5725539574746E-07	19764
3.7575795222277E-07	342
3.8549505990958E-07	373
4.0072422982473E-07	372
5.9091304769716E-07	324
6.1747488219924E-07	573
7.9360621572418E-07	20404
8.0056657120236E-07	6905
8.2646405517907E-07	3340
8.7228859655617E-07	9415
8.7461845764403E-07	9374
8.7487756693950E-07	9473

Figure 4: Time Step Calculation in the Starter Output File

A histogram showing the distribution of the nodal time step in the model is also printed in the Starter output file.

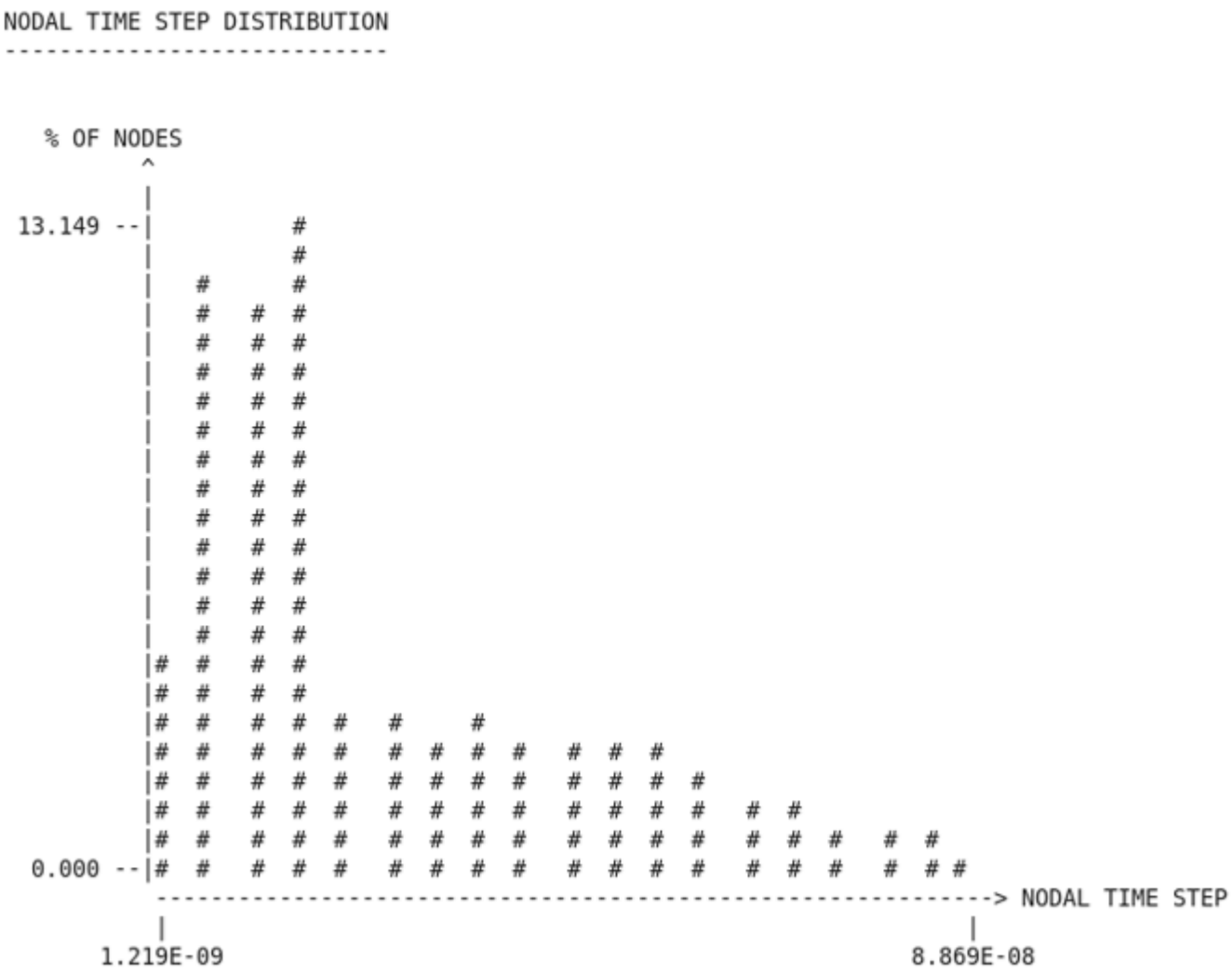


Figure 5: Nodal Time Step Distribution Histogram

Running Model Time Step

When the Radioss Engine is running, the nodal time step can also be written to the animation output files using the output option `/ANIM/NODA/DT` or `/H3D/NODA/DT`. The element time step is available using `/ANIM/ELEM/DT` or `/H3D/ELEM/DT`. The animation file can then be contour plotted in HyperView to observe the time step of the elements in a simulation.

The time step can also be viewed in the Engine output file. The frequency can be set using the `/PRINT/#`, where the time step is written every # cycles, or the time step information of the current cycle can be output using the `/INFO` command in a Control File (C-File).

If element time step is being used, the element type and ID of the element with the minimum time step will be printed in the Engine output file. If there is a contact interface in the model, a node or interface can be listed as having the minimum time step and described in Time Step Control Methods.

Time Step Controlled by an Element

```
  CYCLE    TIME      TIME-STEP  ELEMENT          ERROR
     0   0.000      0.2043E-03 SHELL        202   0.0%
  1000   0.2043     0.2043E-03 SHELL        201   0.0%
  2000   0.4086     0.2043E-03 SHELL        201   0.0%
  3000   0.6129     0.2043E-03 SHELL        201   0.0%
  4000   0.8171     0.2043E-03 SHELL        201   0.0%
```

If nodal time step control is being used, the Engine output file will typically say that a NODE is controlling the time step and list the node ID. Type of time step taken in computation. describes a situation where the time step, based on the contact stiffness is the minimum time step in a simulation based on:

$$\Delta t_{inter\ nodal} = \sqrt{\frac{2M_{nodal}}{\sum\left(K_{interface} + K_{element}\right)}} \tag{7}$$

Time Step Controlled by a Node

```
 CYCLE     TIME      TIME-STEP  ELEMENT          ERROR
     0   0.000      0.5317E-06 NODE       20404   0.0%
  1000   0.5317E-03 0.5317E-06 NODE       20404   0.0%
  2000   0.1063E-02 0.5317E-06 NODE       20404  -0.4%
  3000   0.1595E-02 0.5317E-06 NODE       20404  -1.0%
  4000   0.2127E-02 0.5317E-06 NODE       20404  -0.9%
```

If the kinematic time step of a contact interface has the minimum time step in the model, the Engine output file will say that INTER is controlling the time step and list the contact interface ID. When this happens, secondary node(s) of a contact interface are moving so fast that they will pass through the main contact surface in one time step cycle and thus to prevent this from happening the time step of the simulation is reduced. Time step taken in /DT/INTER/DEL. provides more details.

Time Step Controlled by a Contact Interface

```
  CYCLE    TIME      TIME-STEP  ELEMENT          ERROR
     0   0.000      0.1093E-04 INTER          35   0.0%
  1000   0.1093E-4  0.1093E-04 INTER          35   0.0%
  2000   0.2186E-4  0.1093E-04 INTER          35   0.0%
  3000   0.3279E-4  0.1093E-04 INTER          35   0.0%
  4000   0.4372E-4  0.1093E-04 INTER          35   0.0% %
```

Time Step Control Methods

The time step can often be increased using some of these time step control methods.

One of the most common issues with an explicit finite element simulation is the amount of time the simulation takes to solve. The larger the model's time step, the less cycles that are required to solve a simulation which results in a lower solution time.

Ignoring contact stiffness, the minimum time step of a simulation is a function of the mesh size, material stiffness, and density. Therefore, the first way to increase the time step of a model is to improve the mesh quality by modifying elements with small edge lengths relative to the average element length of the mesh.

Once the mesh is improved, another common problem is a reduction in time step, due to deformation of the mesh. As mentioned earlier, the minimum time step is calculated during each cycle of a simulation

and as deformation occurs the elements size may reduce which cause a reduction in time step. If there is a large reduction in time step, there will be a large increase in simulation time. There are various methods to deal with the reduction in time step.

With some minor variation, all of these methods use this input format.

```
/DT/option/Keyword3/Iflag
```

$\Delta T_{sca} \Delta T_{\min}$

Where,

ΔT_{sca}	Scale factor for the critical nodal time step
$\Delta T_{\min}$	Minimum model time step that activates the time step control

For all options, the time step control is activated when $\Delta T_{\min} \geq \Delta T_{sca} * \Delta t_{option}$; where, Δt_{option} is the time step calculation, based on the option being used such as nodal, element, or interface.

Lagrangian elements whose volume becomes negative during a simulation will automatically switch to the strain formulations to allow the simulation to continue. When this occurs, a Warning message will be printed in the Engine output file.

Table 1: Supported Options

Element Type and Formulation	Strain Formulation	Negative Volume Handling Method
/BRICK, I_{solid} =1, 2, 14, 17, 24 /TETRA4, I_{tetra} = 0 /TETRA10	Full geometric nonlinearities I_{smstr} = 2, 4	Switch to small strain using element shape from cycle before negative volume
	Lagrange type total strain I_{smstr} = 10, 12	Lagrange type total strain with element shape at time=0.0

The automatic switch to small strain can be disabled by setting *Keyword2* to *STOP* in /NEGVOL.

Nodal Time Step Control

The nodal time step control is activated for a simulation by using the option:

```
/DT/NODA/Keyword3/Iflag
```

$\Delta T_{sca} \Delta T_{\min}$

With the time step of a node in the mesh defined as:

$$\Delta t_{nodal} = \sqrt{\frac{2m}{k}} \quad (8)$$

Where,

m	Nodal mass
k	Equivalent nodal stiffness

Note: If the mass of a node increases or the stiffness decreases, then the time step of the simulation increases.

By default, the nodal timestep is applied to the entire model. If *Iflag*=1, then an additional line with a group node set ID is input after the scale factor and time step. This can be used to apply the nodal time step control to a group of nodes `/GRNOD` that have been defined in the Starter.

Note: No matter which *Keyword3* option is used, only one `/DT/NODA/Keyword3/Iflag` option can be used at a time in an Engine file. If multiple ones are included, only the last one is used.

/DT/NODA/CST

`/DT/NODA/CST` is by far the most popular option for maintaining or increasing the time step in a simulation. Radioss will automatically add mass to nodes to maintain the entered value. If needed, the mass will be added at the beginning of the simulation to meet the entered value.

Advantages and Disadvantages

Increase the nodal mass via `/DT/NODA/CST` is the easiest way to increase the time step of a model or prevent a time step from dropping below a certain value during the simulation. Good engineering judgement must be used to determine how much mass is an acceptable amount to be added to a model. Adding too much mass can affect the physics by increasing the kinetic energy of a drop or impact simulation. This is because the object being simulated weighs more than the real part. Increased mass can also change the high frequency behavior of a model which can be very important in very high speed impacts such as ballistics, or explosions.

In general, it is recommended to keep the amount of mass added to less than 5%. However, larger mass increases may be acceptable in some types of simulation. For example, in quasi-static simulations the velocities are usually small, so adding mass does not greatly increase the kinetic energy. For those reasons, it is recommended to check the mass increase in the model by running a simulation without or with reduced mass scaling and comparing the results. If added mass results in added kinetic energy, the energy error calculated by Radioss will be positive.

Choose a Time Step for Mass Scaling

To determine how much mass will be added to a model, Radioss calculates a target time step for different amounts of percent mass increase. The target time step is calculated for percent mass increases from 0.5% to 10% for time step scale factors (ΔT_{sca}) of 0.9 and 0.67. This information is printed in a table and a graph in the Starter output file.

Note: This estimation is valid at first cycle only. The added mass can increase during the run because of element deformation or contacts which means the added mass can become higher than the value expected from the Starter estimation.

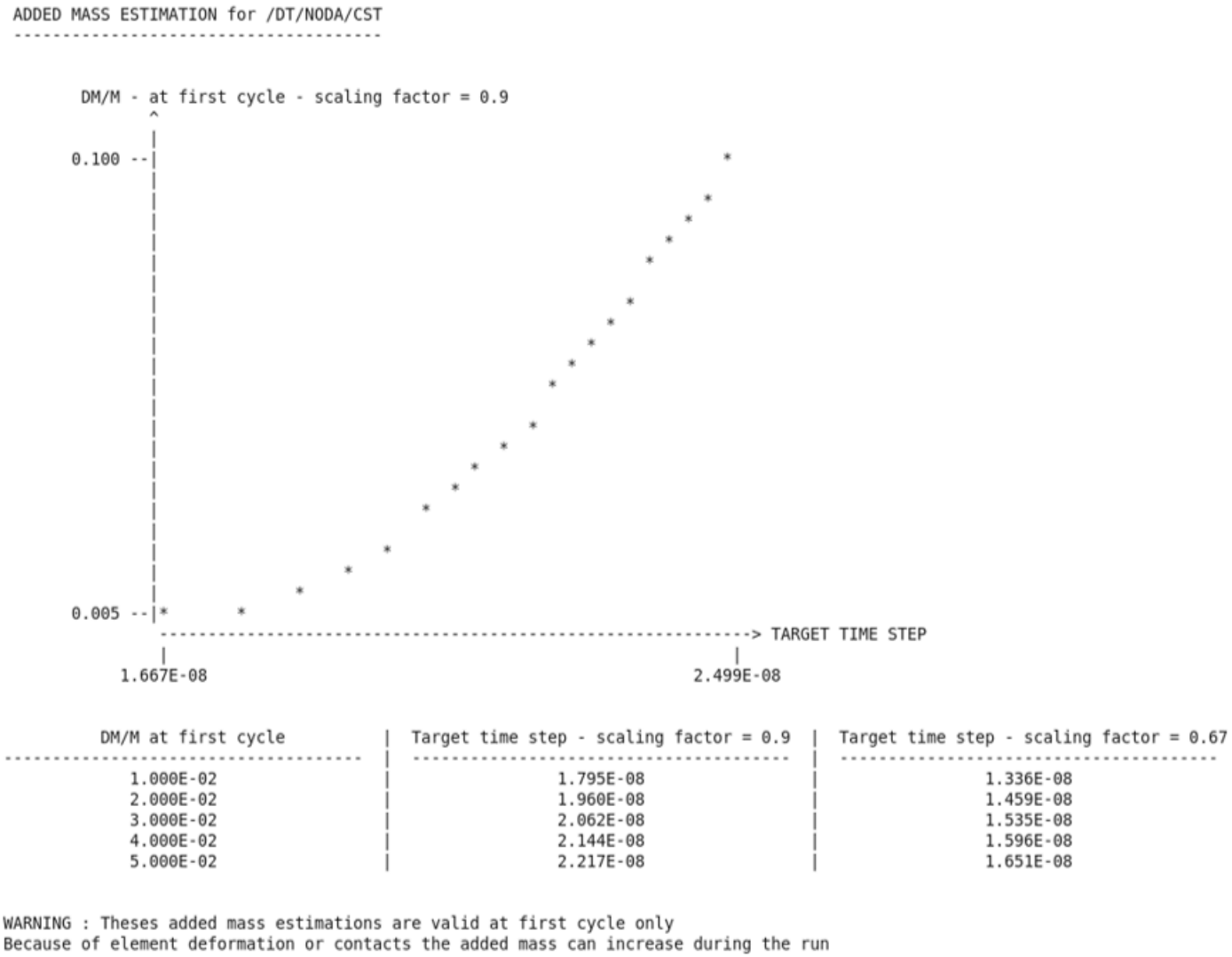


Figure 6: Mass Added for Different Target Time Steps

The target time step and scale factor can then used in `/DT/NODA/CST`.

Check for Mass Increase

The total mass increase is listed in the Engine output file in the last column titled, `MAS.ERR`. Using the animation output option `/ANIM/NODA/DMASS` or `/H3D/NODA/DMASS`, the relative mass increase per node can be visualized in a post-processor as a contour plot. Both the total mass error and nodal mass error represent the change in mass divided by the original mass at the beginning of the simulation.

$$MASS.ERR \text{ or } DMAS = \left(\frac{DM}{M_0}\right) \tag{9}$$

With, $DM = M - M_0$

Where,

M_0 Initial mass at the beginning of the simulation for each Engine file

M	Current mass

> **Note:** The initial mass is reset at the beginning of every Engine file simulation and the total mass increase should be added from all the Engine output files.

The global time history mass and energy curves can be plotted to understand how the increased mass effects the simulation.

> **Note:** That part mass output using `/TH/PART` does not include the mass added, due to mass scaling.

/DT/NODA/SET

Reduces the equivalent nodal stiffness (k) to maintain the entered $\Delta T_{\min}$ value. This reduction in stiffness also changes the physics of a simulation and is typically only used when modeling fluids.

/DT/NODA/STOP

Stops the simulation when the simulation's time step drops below the entered $\Delta T_{\min}$ value. Many times, a reduction in time step is caused by a model's instability, so stopping the simulation can be useful to diagnose the issue.

Element Time Step Control

The element time step can be controlled using the Engine options:

`/DT/Eltyp/Keyword3/Iflag`

$\Delta T_{sca} \Delta T_{\min}$

Where,

Eltype	Element type
Keyword3	Time step control type used

If *Iflag*=1, then a second line with additional input is entered for some element types and times step control methods.

For all the *Keyword3* options, the action taken will occur when with the calculation of the element time step defined in Finite Elements. For element time step control, only one *Keyword3* option is available per *Eltyp*, which means it is not possible to use `/DT/BRICK/CST` and `/DT/BRICK/DEL` in the same model, but `/DT/SHELL/STOP` and `/DT/BRICK/CST` can be used together.

/DT/Eltyp/CST

Eltyp=BRICK or SHELL.

This option controls the times step by switching elements to the small strain formulation if they cause the simulations time step to be less than $\Delta T_{\min}$. This only occurs when using certain I_{smstr} values defined in the property definition `/PROP` for the element. Although, switching to small strain formulation

is not as accurate as large strain, the reduction in accuracy is often necessary to obtain a solution in a reasonable time.

If using `/DT/BRICK/CST/1`, then a second extra line is added.

$\Delta T_{sca}\Delta T_{\min}$
ASP_min VDEF_min

Where,

ASP_min	Minimum aspect
VDEF_min	Minimum volume ratio (V/V_0)

Elements will be switched to small strain formulation when one of the two minimum values for *ASP_min* or *VDEF_min* has been reached. *ASP_min* uses the same tet collapse calculation as HyperMesh element quality check. Elements failing the *ASP_min* criteria will be switched to the small strain formulation at the beginning or during a simulation. *VDEF_min* can be used to switch highly deformed or compressed elements to the small strain formulation during the simulation thus avoiding negative volume issues that sometimes arise during a simulation.

When the switch to small strain formulation occurs, a constant Jacobian matrix is assumed and all spatial variables (characteristic lengths, volumes, …) are considered in their initial configuration. At this point, the element deformation is not taken in account anymore when calculating the time step. If the sound speed is constant, the time step thus becomes and stays constant. For some solid material laws, the sound velocity is pressure dependent. It is, therefore, possible that the time step becomes a little smaller than $\Delta T_{\min}$. If an elements time step is less than $\Delta T_{\min}$ at the beginning of a simulation, then the element will switch to the small strain formulation and assuming that element has the smallest time step the simulation will run at a time step less than $\Delta T_{\min}$. When an element switches to the small strain formulation, a message is printed to the Engine output file.

Advantages and Disadvantages

The switch to small strain formulation stops a decrease in time step which if allowed to continue would cause a simulation to run much longer. Most times the slight loss in accuracy caused by a few elements using the small strain formulation is worth it to make sure results are obtained in a reasonable time. Switching to small strain is better than adding mass and thus kinetic energy in dynamic simulation or removing element and thus removing kinetic energy. Switching to small strain to maintain a time step can be a better solution in dynamic simulations rather than adding or losing kinetic energy associated with adding mass or deleting elements.

There are less limitations for shell elements than for solid elements. In crash applications, the main shell deformation is bending and the small strain formulation has no effect on the bending description if membrane deformation is small.

A large number of elements switching to the small strain formulation typically indicates a model issue that needs investigated.

/DT/Eltyp/DEL

Deletes the element that causes the simulation time step to become less than $\Delta T_{\min}$. When elements are deleted, the physics of the model is changed because removing an element changes the mass,

inertia and possibly the stiffness of the model. When an element is deleted, a message is printed to the Engine output file.

/DT/Eltyp/STOP

Stops the simulation when the simulation's time step drops below the entered $\Delta T_{\min}$ value. Many times, a reduction in time step is caused by a model's instability, so stopping the simulation can be useful to diagnose the issue.

Contact Interface Time Step Control

The contact interface time step can be controlled using the Engine options:

`/DT/INTER/Keyword3/Iflag`

$\Delta T_{sca} \Delta T_{\min}$

For all the *Keyword3* options, the action taken will occur when $\Delta T_{\min} \geq \Delta T_{sca} * \Delta t_{interface}$ where, $\Delta t_{interface} = \min(\Delta t_{inter\ nodal}, \Delta t_{kin})$ is the minimum of contact stiffness time step and the kinetic time step as defined in Contact Interface Time Step. If *Iflag*=1, then a second line with additional input is entered for `/DT/INTER/AMS`.

> **Note:** This control is only used with contact interfaces with nonlinear stiffness, TYPE7, TYPE10, TYPE11, TYPE19, and TYPE21.

/DT/INTER/CST

Automatically increases the mass of nodes involved in contact to maintain the entered value. Normally this option is not useful since contact secondary nodes with low time steps typically have high penetration into the contact gap and when mass is added their kinetic energy increases which causes increased penetration and thus model instability. The amount of added mass should be monitored during and checked after the simulation in the same manner as `/DT/NODA/CST`.

/DT/INTER/DEL

Removes nodes involved in a contact interface that are causing the time step of the simulation to drop below $\Delta T_{\min}$. This option can be very useful to maintain a specified time step, but if the number of removed nodes is too large, then the contact will not be correct, and the results may not be accurate.

/DT/INTER/STOP

Stops the simulation when the contact interface time step drops below the entered $\Delta T_{\min}$ value. Many times, a reduction in time step is caused by a model's instability, so stopping the simulation can be useful to diagnose the issue.

/DT/INTER/AMS

Applies Advanced Mass Scaling to interface secondary nodes to maintain the constant time step, $\Delta T_{\min}$. Refer to Advanced Mass Scaling (AMS) for more information.

Advanced Mass Scaling (AMS)

See Also
/AMS (Starter)
/DT/AMS (Engine)
RD-E: 4400 Blow Molding with AMS

Increase Computation Speed and Maintain Accuracy

AMS (Advanced Mass Scaling) saves significant computation time by increasing the time step of the model for an explicit computation. This is similar to traditional mass scaling, except that the added mass does not increase the translational kinetic energy of the system.

A non-diagonal mass matrix is used to increase the time step on each line of the mass matrix. The lumped mass, M0, is increased with some M value compensated with non-diagonal terms such that the total mass to remain constant. [1] Unlike traditional mass scaling, AMS only modifies high frequencies and does not significantly affect low frequencies of the model.

The advantage of AMS versus traditional user controlled mass scaling is that translational kinetic energy is not increased. This allows the time step to be increased to significantly higher values as compared to traditional mass scaling without significantly affecting the results quality.

Since AMS does not modify the global mass, even at large time steps, the global momentum of the nodes affected by AMS is conserved. At large time steps, traditional `/DT/NODA/CST` can add a significant amount mass to a computation which increases translational kinetic energy.

AMS has a computational cost associated with calculating the mass matrix. The computational cost is model dependent, but for a highly nonlinear model it could be 50% of the total computational cost. So, although the cost per cycle has increased, the number of calculation cycles is reduced, due to the increased time step. For example, using a time step of 10 times the traditional `/DT/NODA/CST`, the total time for the calculation was reduced by a factor of 3. Therefore, to see a reasonable reduction in elapsed time, 10 times the `/DT/NODA/CST` time step is the recommended starting point.

Computational convergence and accurate results can be obtained by setting a target time step to 10 to 20 times higher than traditional mass scaling. In manufacturing simulations, 50 times the traditional mass scaling time step can be used. Since the Courant condition remains to be respected, the stability of the model must be achieved with the targeted time step to apply AMS.

Several modifications in the model may help increase its stability with high time step. Below are some recommendations and suggestions in order to insure the stability of the model.

Note: Advanced Mass Scaling is specific to Radioss. It is advanced because it can be applied to the entire model without degrading computing performances and result quality.

14.0 New AMS Features

- Compatibility with RBE2 and RBE3

13.0 New AMS Features

- Compatibility with moving rigid walls (`/RWALL` with *node_ID* > 0)

- Fixed rigid walls were corrected (`/RWALL` with *node_ID* = 0 or blank)
- Tolerance default value was changed from 1E-4 to 1E-3 (*Tol_AMS* = 0 # 0.001)
- Conjugate Gradient (C.G.) convergence criteria was improved
- Non-diagonal added mass matrix was optimized

> **Note:** The AMS tolerance was changed in order to compensate a slight loss of computation time performance, due to above listed improvements but it should not affect the results accuracy.

Starter

Only Keyword

Only `/AMS` must be present in order to apply the AMS to either a designated group of parts, or the entire model, if followed by a blank line (Radioss Starter Input).

> **CAUTION:** `/DT/AMS` is mandatory in Radioss Engine Input followed by the scale factor and the targeted increased time step to activate AMS (Radioss Engine Input). Without `/DT/AMS` in the Engine file, `/AMS` in the Starter file is ignored.

HMPP / SPMD

If *DOMDEC* is set to 0, the Starter points to 3 (Multi-level Kway decomposition) by default. Up until version 11.0.230, if `/AMS` is applied, `DOMDEC` must be manually switched to 5 (DOF-based Multi-level Kway decomposition). As of version 11.0.240, the Starter automatically sets `DOMDEC` to 5, if it is set to 0 and if `/AMS` is present, otherwise, `DOMDEC` will be set to 3, if `/AMS` is absent.

Interfaces

- It is recommended to use Interface stiffness flag I_{stf} = 4 $K = \max\left[St_{\min}, \min\left(St_{\max}, K_n\right)\right]$ with $K_n = \min\left(K_m, K_s\right)$ and Stiffness scale factor *Stfac* = 1 (default value) for contact interfaces TYPE7, TYPE11, and TYPE19.
- If friction is involved, it is advised to set the friction penalty formulation type, I_{form} to 2 (incremental stiffness for Coulomb friction), whenever this option is available (`/INTER/TYPE7`, and `/TYPE19`.
- For AMS, like in standard mass scaling, it is recommended to not have friction in a `TYPE11` contact, if a `TYPE7` (already handling friction) contact is already defined for the same parts. This avoids drops of time step and helps model convergence. This recommendation becomes obsolete, if `TYPE11` friction is using I_{form}=2 for edge-to-edge contacts, as of version 13.0.
- With contact interfaces `TYPE7`, `TYPE11`, and `TYPE19` using nonlinear penalty stiffness for contact, it might be necessary to use the `/DT/INTER/DEL` option in the Radioss Engine input deck. Otherwise, AMS may slowly converge, or may diverge.

1. Morancay, Lionel, and Gérard Winkelmuller. "Dynamic condensation and selective mass scaling in Radioss Explicit." 19ème Congrès Français de Mécanique (2009).

- Using `/DT/INTER/AMS` is meaningless when using `/DT/AMS`, but `/DT/INTER/AMS` can be used, instead of `/DT/NODA/CST` when contacts highly affect the time step.
- Initial intersections and penetrations should be removed, as much as possible (as well as in traditional mass scaling, notably if it is used for comparison with the AMS results).

> **CAUTION:** Barriers, other impactors, and more particularly dummies contain a lot of interfaces that must NOT be modified since their validation is based on their original `/INTER` options (if modified, their validation is not guaranteed).

Rigid Bodies

- Small rigid bodies inertia should rather be spherical with I_{spher} set to 1.

> **CAUTION:** Dummies contain a lot of small rigid bodies that must NOT be modified since their validation is based on their original `/RBODY` options (if modified, their validation is not guaranteed).

- Flying nodes: are not a limitation but they should be removed, as well as unconnected small rigid bodies that are not needed.
- Rayleigh damping: in few cases where the elastic domain remains large along the simulation, poor AMS performances with noisy time history curves (example, force-time) and Harlequin patterns in contour plots (example, von Mises) may be observed. A significant improvement should occur by inserting a Rayleigh damping card (`/DAMP` in the Starter file) affecting the parts subject to AMS using these parameters:

  $\alpha = 0$ and $\beta = 0.05\Delta t_{AMS}$

  (with Δt_{AMS} is AMS target time step, so β is 5% times the AMS target time step). Such a modification may allow to increase the AMS target time step for much better AMS performances.

> **Note:** It is not advised to use numerical damping (dm or Navier-Stokes), instead of Rayleigh damping, here.

Engine

Only Keyword: only `/DT/AMS` must be present in order to invoke AMS with the targeted increased elementary time step (Radioss Engine Input).

> **CAUTION:** `/AMS` is mandatory in the Radioss Starter Input in order to designate the selected parts or the entire model on which AMS is applied.

In `/DT/AMS/Iflag` if I_{flag}= 1, the tolerance for AMS convergence, *Tol_AMS*, must be provided. If I_{flag} is ignored, the default value (1E-3) for *Tol_AMS* is applied. It is not recommended to modify AMS tolerance, if it is not needed. Note that *Tol_AMS* was 1E-4 in 12.0.210.

In `/DT/AMS/Iflag`, if I_{flag} = 2, the display of the number of AMS iterations may help in debugging or monitoring convergence at no extra CPU cost. Maximum allowed iterations before sending a divergence

message is 1000 by default. It is advised to NOT change this value. Anything above this maximum value, Radioss stops with the following error message:

```
** ERROR ** AMS IS LIKELY DIVERGING
```

- Monitoring the number of iterations per cycle can help to understand the resulting AMS performance:
- 75 to 100 iterations is a sign of a really bad convergence 50 iterations may not provide any speedup 30 iterations or less is considered a good convergence.

It is advised to use 0.67 as the scale factor, ΔT_{sca}, rather than 0.9 (as well as in traditional nodal mass scaling `/DT/NODA/CST`, especially if it is used for comparison with the AMS results).

Target a minimum time step, $\Delta T_{\min}$, from 10 to 20 times than the one used in `/DT/NODA/CST`, that is, start with 10 times, then check how far to go by verifying the numerical convergence, the gain of elapsed time, and the results quality.

`/DT/AMS` will not modify the global mass, therefore:

- The global momentum of the related nodes is conserved.
- A comparison with traditional mass scaling requires a minor added mass along the `/DT/NODA/CST` computation, if used as a reference.

`/DT/Eltyp/Keyword3` is compatible with `/DT/AMS`, except `/DT/INTER/CST` (refer to Capabilities and Limitations for further details).

Use `/DT/INTER/DEL` as a workaround for slow convergence, or possible divergence, when invoking a nonlinear penalty stiffness with contact interfaces `TYPE7`, `TYPE11`, and `TYPE19`.

`/ANIM/NODA/NDMAS` may be added in order to output non-diagonal mass variation.

`/ANIM/NODA/DINER` may be added in order to output added inertia per nodes.

Capabilities and Limitations

AMS is compatible with the following options:

Kinematic Conditions

- Boundary conditions: `/BCS`
- Imposed velocities: `/IMPVEL`
- Imposed displacements: `/IMPDISP`
- Imposed accelerations: `/IMPACC`
- Tied contacts: `/INTER/TYPE2` *Spot*$_{flag}$ 0, 1, 2 (kinematic formulation), 25, 27 or 28 (penalty formulation)
  - Rigid bodies: `/RBODY` (provided they are not sensor activated or deactivated)
  - Fixed rigid walls: `/RWALL` with *node_ID*= 0 or blank (in case of friction, it is still advised to use a rigid meshed impactor, due to AMS cost)
  - Moving rigid walls: `/RWALL` with *node_ID* > 0 (in case of friction, it is still advised to use a rigid meshed impactor, due to AMS cost)
  - Rigid links: `/RLINK`

  - Kinematic joints: `/KJOINT` (`/PROP/TYPE33`) and `/KJOINT2` (`/PROP/TYPE45`)
  - Cylindrical joints: `/CYL_JOINT`
  - Active sections: `/SECT`
  - Velocities: `/VEL/TRA` or `/VEL/ROT`
  - Initial velocities: `/INIV/TRA` or `/INIV/ROT`
- Rigid bodies can be activated and deactivated with `/RBODY/ON` and `OFF` as Engine keywords
- Rigid elements: `/RBE2`
- Interpolation constraint elements: `/RBE3`

Elements

- All classical: trusses, beams, springs, shells, and solids
- 4-node tetra: `/TETRA4`, only if I_{tetra4}=0
- 10-node tetra: `/TETRA10`, only if $I_{tetra10}$=0
- Element formulation:
  - `QEPH` I_{shell}=24 and all other I_{shell} options
  - `HEPH` I_{solid}=24

Interfaces

`/DT/AMS` is compatible with interfaces.

- `TYPE2` (see Tied contacts above)
- `TYPE7`
- `TYPE10`
- `TYPE11`
- `TYPE19`
- `TYPE21`
- `TYPE24`
- `TYPE25`
- Dynamic relaxation: `/DYREL`, `/ADYREL`
- Rayleigh damping: `/DAMP`
- `/DT/Eltyp/Keyword3` is compatible with `/DT/AMS`

  This includes `/DT/Eltyp/CST` for which the small strain formulation is activated for an element when

$$\Delta T_{sca} \cdot \frac{l}{c} < \Delta T_{\min} \tag{10}$$

  Where,

l	Element characteristic length,
c	Speed of sound into the material

- /DT/AMS applied to selected parts (in /AMS from Starter Input file) does not exclude concurrent usage of /DT/NODA/CST applied to non-AMS designated parts of the rest of the model. This is even advised in order to minimize computing time in models where AMS only applies to selected parts.

  Note: It is pointless to use a higher time step in AMS than the one used in classic mass scaling since the smaller time step limits the performance.

- Airbags without reference metrics should work

See Also
/AMS (Starter)
/DT/AMS (Engine)

Limitations (may coexist, but out of the selected AMS part group)

CAUTION: /PARITH/ON is not yet insured with AMS, if the number of domains changes. But, for one same domain number, results are same whatever the number of threads.

- SPH: AMS does not increase time step of SPH (example: tank sloshing in crash)
- ALE - Euler - FVM: formulations are not supported (but can be put out of /AMS part, if time step is decent)
- 2D analysis: /AMS is not available
- Parts involving:
  - Lagrange multipliers: /MPC and /GJOINT
  - /FXBODY

In such cases, a warning is output by Radioss Engine and the computation may even diverge.

- Elements:
  - 4-node tetra: /TETRA4 with I_{tetra4}=1 or 3
  - 10-nodes tetra: /TETRA10 with $I_{tetra10}$=2
  - 16-node shell: /SHEL16
  - 20-node bricks: /BRIC20
  - Multi-strand: /XELEM

In such cases, the computation will most likely diverge.

- Rigid bodies cannot be activated and deactivated with sensors
- Interfaces: /DT/INTER/CST is not compatible with /DT/AMS
  - Nodes that are secondary of both: a tied interface TYPE2 and a contact interface (TYPE7 or TYPE11) will have their contact stiffness removed a Starter message in the 0.out file is then issued:

```
** WARNING SECONDARY NODE OF AN INTERFACE TYPE2 & AMS
  INTERFACE TYPE[7 or 11], ID=xxxxxx:
  SECONDARY NODE ID=yyyyyyy IS ALSO SECONDARY NODE OF AN INTERFACE TYPE2
 THE NODE CONTACT STIFFNESS WILL BE DE-ACTIVATED CASE OF /DT/AMS
```

This contact deactivation can be avoided by using $Spot_{flag}$=25, 27 or 28 (`TYPE2` penalty formulation).

- Kinematic formulation (`TYPE2`) for spotwelds may alter AMS performance when there are many spotwelds, particularly hexa spotwelds.

- Airbags with reference metrics: `/REFSTA` or `/XREF` do not benefit from AMS speedup.
- Refer to the Starter Input and Engine Input *Reference Guides*.

Table 2: Main Advanced Mass Scaling Limitation Summary

	Elementary time step for AMS `/DT/AMS` in Radioss Engine Input
Lagrange Multipliers	Does not work
16-node thick shells and 20-node bricks	Free element time step is applied to these elements
Rigid bodies activated, deactivated with sensors	Does not work
Interfaces	Only TYPE2, TYPE7, TYPE10, TYPE11, TYPE19, TYPE20, TYPE21, TYPE24, and TYPE25 are supported

Recommended Checklist

True incompatible kinematic conditions (IKC) must be carefully checked and resolved when occurring. AMS may either fail or run with a very low target time step, leading to poor performances, whereas the classic time step control for the same model can run normally. In that sense, AMS can reveal modeling weaknesses.

Any targeted model for AMS application should run first in `/DT/NODA/CST` with a reasonable energy error (ERROR < +2%) and acceptable added mass (MAS.ER < 0.02) along its simulation time. A model unable to run with `/DT/NODA/CST` will not run with AMS either.

Too much mass added when using `/DT/NODA/CST` can cause result differences when compared to an AMS result where no mass is added. If necessary, rerun the `/DT/NODA/CST` model with a lower time step to reduce the amount of mass added and then compare the AMS results.

An AMS job may either fail or diverge, most likely with an error message. However, once an AMS run goes through, several aspects remain to be carefully checked in order to insure a maximized elapsed time saving, as well as a reliable result quality.

These aspects are:

- Time step value evolution and comparison with the targeted time step:
  - The time step should follow the targeted time step without time step drop (occasionally)

    - If the time step is at any time smaller than the targeted one, the target time step should be reduced
- Entity controlling the time step at cycle (`SOLID`, `SHELL`, `SPRING`, `BEAM`, `NODE`, `INTER`, and so on.)
- Energy Error evolution (should decrease slowly negatively - not increase)
- Output the number of AMS iterations per cycle via `/DT/AMS/Iflag` - I_{flag} =2 may help monitoring convergence quality at no extra CPU cost.

  Maximum allowed iterations before the divergence stops is 1000. 75 to 100 iteration per cycle is a sign of a poor convergence 50 still may provide some speedup

  30 iterations or less is considered a good convergence.

Basic Quality Indexes

With constant nodal time step control (`/DT/NODA/CST`), the energy error (ERROR %) and added mass (MASS.ER) are the first ones to check inside the output file (`_0001.out`).

With AMS, the added mass is irrelevant. Review the energy error, the time step variation, and the number of iteration per cycle remain.

Common to both approaches are the energy balance curves.

Results Accuracy

The numerical effect of AMS on results is similar to lowering the highest eigen frequencies of the structure. Since AMS affects high frequencies, it is suitable for quasi-static, drop test, and manufacturing (stamping) simulations. It is not recommended for highly deformed structures or high velocity phenomena, like explosion and ballistics simulation. Occasionally, AMS may be used in crash simulation, if selectively applied to a group of finely meshed structural parts, potentially penalizing the computing performance of the entire model. In such cases, the AMS target time step of these selected parts (excluding safety subsystems, such as: barriers, dummies, airbags, and restraint systems), must be set to the time step value of the rest of the model where a classical mass scaling is applied.

In most cases, the buckling behavior is not affected by eigen vectors (then by AMS). In specific cases like box beam crushing, the width and amplitude of peak forces may be altered since the peak force may not be detected, due to an AMS higher time step. The buckling pattern is correct, but the absorbed energy is not the same on the first peak.

Rigid bodies rotation: rotational inertia of small rigid bodies may be affected by AMS. Inertia of these small rigid bodies will be increased in order to achieve the target time step.

When comparing AMS results with reference run, make sure the added mass in the reference run is low enough so the results are not affected.

Lessons Learned

Starting from the constant nodal time step value (`/DT/NODA/CST` traditional in production) as a "reference" and applying a progressively increased target AMS time step (`/DT/AMS`) has shown four stages involving both numerical stability and result quality:

1. Too close to the traditional constant nodal time step value, the AMS numerical stability is excellent and results almost identical to the "reference". Of course, speedups are poor due to AMS iteration cost. In theory, a targeted x3 factor will return an "*effective speedup*" of x1 which is of no benefit.

2. Slightly above the previous case AMS numerical stability remains satisfactory mostly showing references to element related minimum time steps (possible references to interfaces). Results are still close to the "reference", AMS iteration cost becomes amortized and speedups start to rise. As an estimate targeted factors from x3 to x10, most likely return effective speedups ranging from x1.5 to x3, possibly more.
3. Above that, numerical stability may look satisfactory with possibly, higher numbers of iterations per cycle and more references to interface related minimum time steps. Results may either still look acceptable or significantly different compared to the "reference". Speedups continue to grow, unless penalty-based interfaces dictate the minimum time step causing iteration number increase and possible time step drops, what affects the speedup. However, targeted factors from x10 to x20 possibly x30 may return effective speedups ranging respectively from 4 to 6, possibly 9 (more was found in stamping simulation) that should NOT be considered a success as long as AMS results are not checked and compared to above defined "reference" results.
4. Above previous cases, computations either diverge then stop with an explicit AMS message or will stop due to Energy Error limit (no added mass) or the time step dramatically drops and the run needs to be interactively stopped.

The 1.out File Examples illustrates the AMS effect on the Radioss output file (`1.out`) depending on the entered target time step.

Hints

When models contain parts with very different mesh sizes it may be better to only apply AMS to the parts penalizing the time step. Then, for optimized computing time performances, it is advised to also apply a classical mass scaling to the parts not belonging to `/AMS` part group, otherwise, none AMS processed parts are computed, by default, in natural Element time step.

Example:

```
/DT/AMS/1
ΔT_sca ΔT_min
Tol_AMS
/DT/NODA/CST
ΔT_sca ΔT_min
```

It is pointless to use a higher time step in AMS than the one used in classic mass scaling, since the smaller time step limits performances. If no part group is specified (blank line) or is equal to 0, then AMS is applied to the model in its whole, and adding `/DT/NODA/CST` is irrelevant.

For AMS, like in standard mass scaling, it is recommended to not have friction in a `TYPE11` contact if a `TYPE7` (already handling friction) contact is already defined for the same parts. This avoids drops of time step and helps model convergence. This recommendation is obsolete, if `TYPE11` friction is using I_{form}=2 available for edge-to-edge contacts since 13.0.

Nodes that are secondary of both: a tied interface `TYPE2` and a contact interface (`TYPE7` or `TYPE11`) will have their contact stiffness removed a Starter message in the `0.out` file is then issued:

```
** WARNING SECONDARY NODE OF AN INTERFACE TYPE2 & AMS
   INTERFACE TYPE[7 or 11], ID=xxxxxx:
   SECONDARY NODE ID=yyyyyyy IS ALSO SECONDARY NODE OF AN INTERFACE TYPE2
THE NODE CONTACT STIFFNESS WILL BE DE-ACTIVATED CASE OF /DT/AMS
```

This contact deactivation can be avoided by using $Spot_{flag}$=25, 27 or 28 (`TYPE2` penalty formulation).

Kinematic formulation (TYPE2) for spotwelds may alter AMS performances when there are many spotwelds, particularly hexa spotwelds.

Some dynamic cases, often in elastic state, exhibit strange elastic vibrations (showing arlequin-like von Mises contours) forcing to lower the AMS target time step leading to poor AMS performances. These vibrations can be reduced and the AMS target time step re-increased for better performances by applying Rayleigh damping. The recommended damping value is:

$\alpha = 0$ and $\beta = 0.05\Delta t_{AMS}$ is AMS target time step.

> **Note:** As of version 13.0, the tolerance default value was changed from 1E-4 to 1E-3 (*Tol_AMS* = 0 # 0.001).

See Also
/AMS (Starter)
/DT/AMS (Engine)

Example: Automotive Application

For automotive applications, a target time step set to a value of 10 times the reference (nodal based constant) time step is a reasonable value to start with. However, greater than 20 times higher than the reference time step have been successfully utilized.

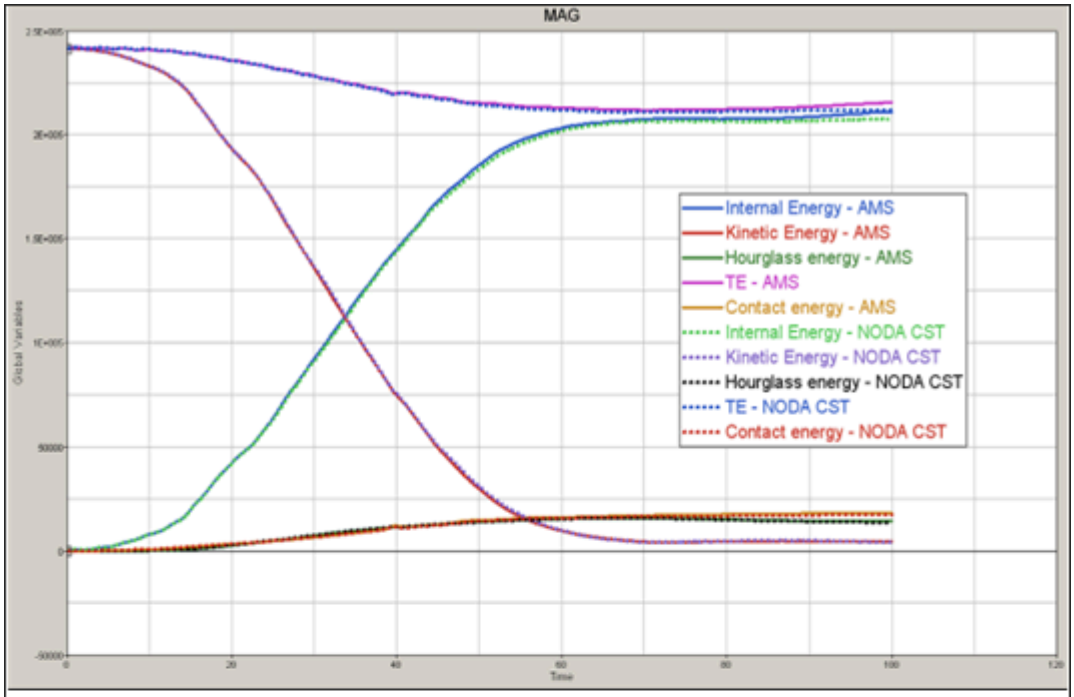
Table 3: Summary of Full Vehicle Tests Tried with AMS

Load Case	Vehicle Program	Radioss Version	Number of Cores	Model Zize	Original Nodal Time Step	Target Time Step	Speed-up v. Original Nodal Time Step	Comments	ODB Offset Deformable Barrier FFB Full Frontal Barrier (rigid)
Front crash	Neon public	10.0.2		1M	1μ s	10 μs	4 x		
	Taurus public FFB	11.0.201		0.8 M	0.5 μs	5 μs	2.83 x	Rigid wall replaced by rigid mesh and contact interface	
	OEM1 FFB	11.	24 x 1	1.3 M	0.5 μs	5 μs	3 x	Figure 7	
	OEM2 ODB	11.0.230	32 x 1	2 M	0.5 μs	5 μs	1.5 x	Figure 8	

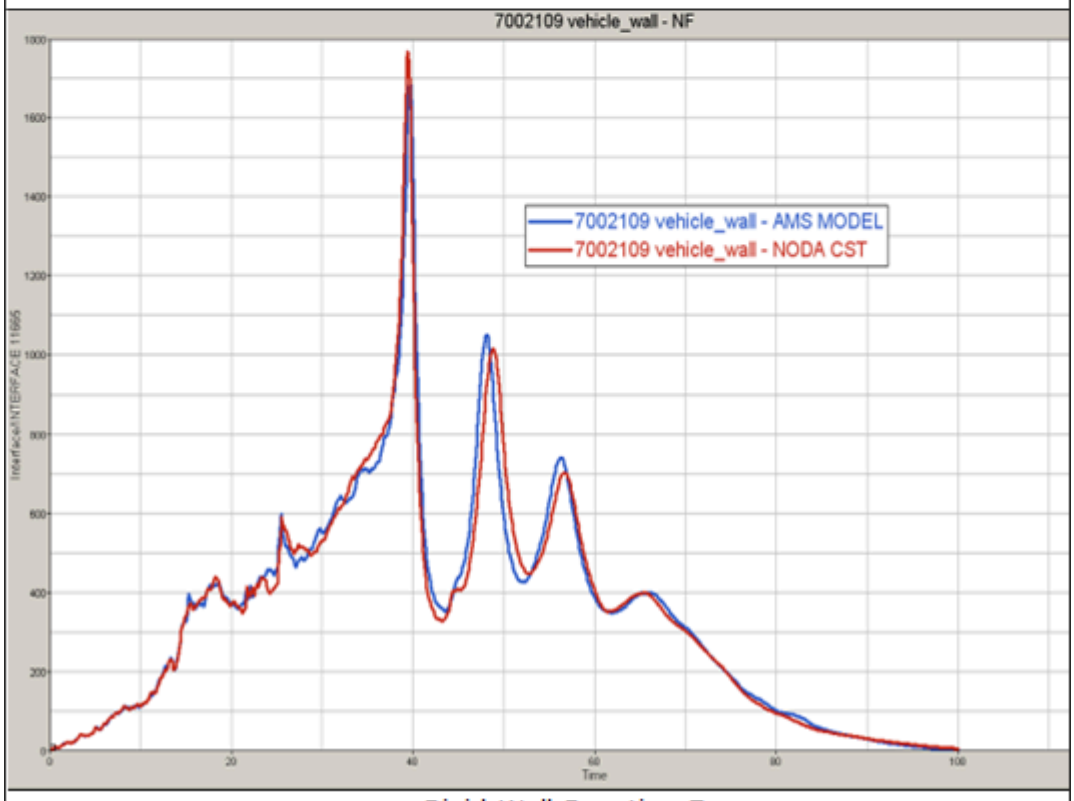
Load Case	Vehicle Program	Radioss Version	Number of Cores	Model Zize	Original Nodal Time Step	Target Time Step	Speed-up v. Original Nodal Time Step	Comments	ODB Offset Deformable Barrier FFB Full Frontal Barrier (rigid)
	OEM2 ODB	11.0.230	6 x 6	2 M	0.5 μs	5 μs	2 x		
	OEM4 FFB	11.0	32 x 1	1.5 M	0.5 μs	4 μs	1.9 x		
	OEM4 FFB	11.0.230	8 x 6	6 M	0.5 μs	5 μs	1.3 x	Rigid wall replaced by rigid mesh and contact interface	
Rear crash	OEM1	11.0_sp	16 x 1	1.1 M	0.1 μs	2 μs	5.8 x	Figure 9	
	OEM1	11.0	16 x 1	1.1 M	0.1 μs	2 μs	6.5 x		
Side impact	OEM4	11.0	32 x 1	1 M	0.5 μs	10 μs	2.1 x	Figure 10	
Roof Crush	OEM4	11.0	16 x 1	1.2 M	0.5 μs	10 μs	2.9 x	Figure 11	
	OEM4	11.0	32 x 1	1 M	0.5 μs	10 μs	3.4 x		
	Taurus public	11.0	16 x 1	0.8 M	0.5 μs	10 μs	4.6 x	Figure 12	

All speedup's compare original nodal time step double precision with AMS in single precision.

Automotive Related Results



Energy Balance



Rigid Wall Reaction Force

Figure 7: OEM1 Full Frontal

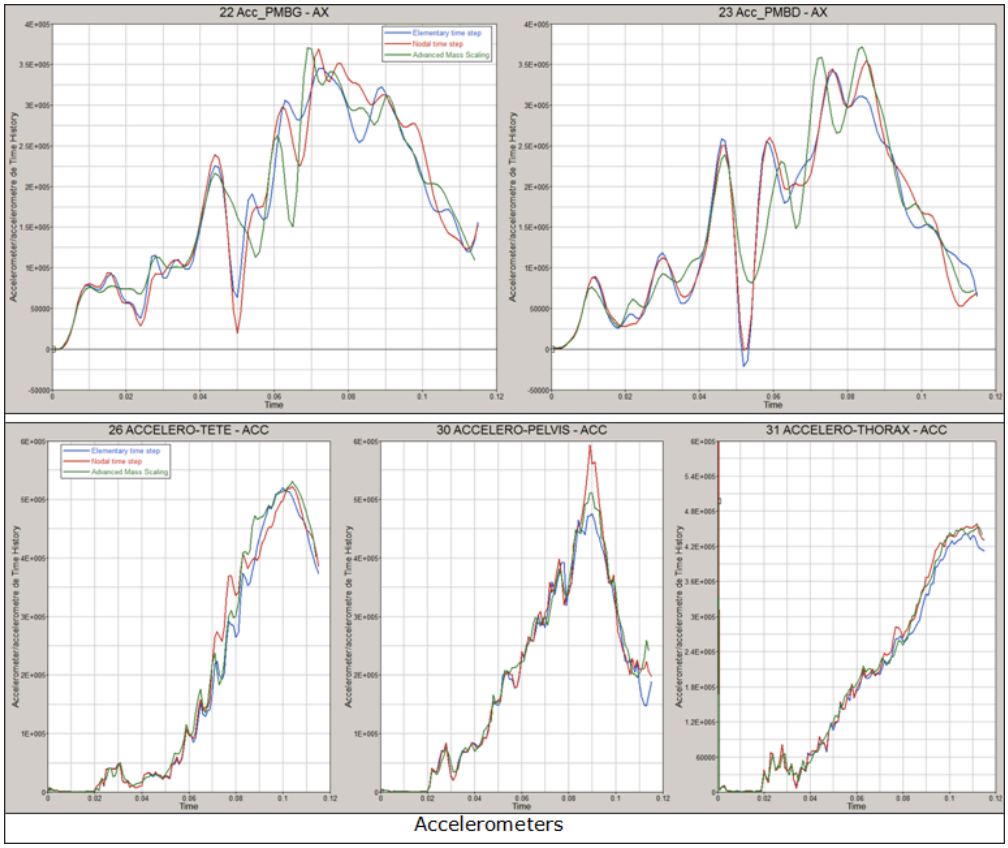
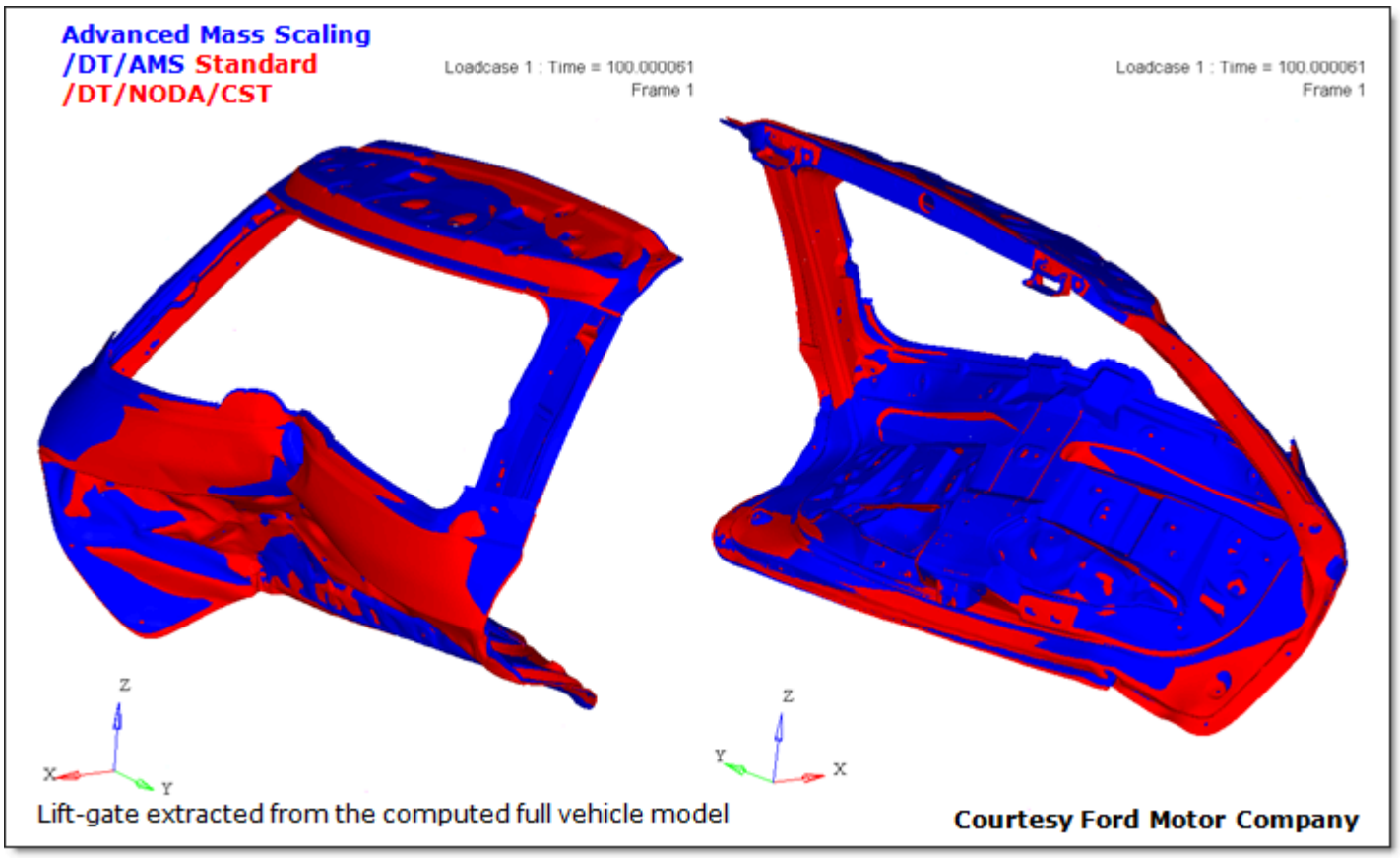


Figure 8: OEM2 Frontal Offset


Advanced Mass Scaling
/DT/AMS Standard
/DT/NODA/CST
Loadcase 1 : Time = 100.000061
Frame 1
Loadcase 1 : Time = 100.000061
Frame 1
Z
X
Y
Z
Y
X
Lift-gate extracted from the computed full vehicle model
Courtesy Ford Motor Company

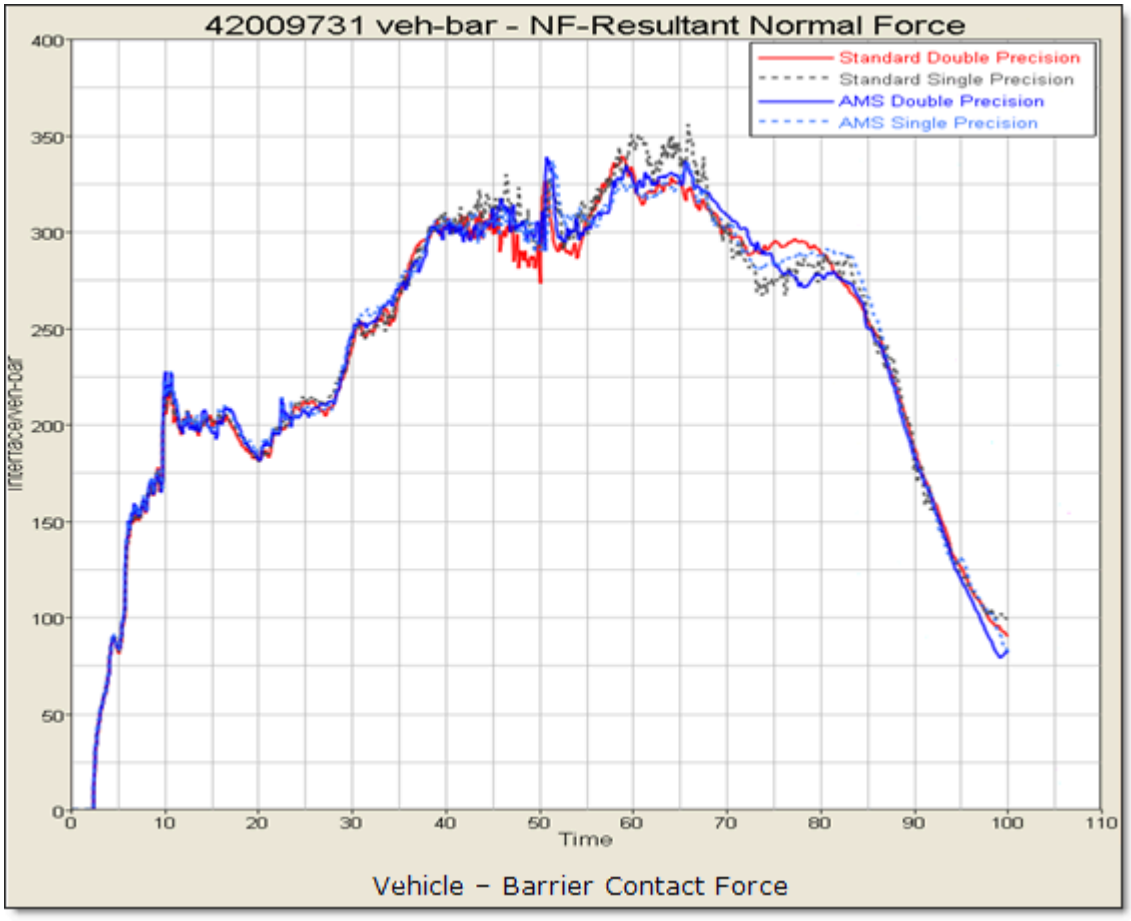


Figure 9: Front Rear Impact

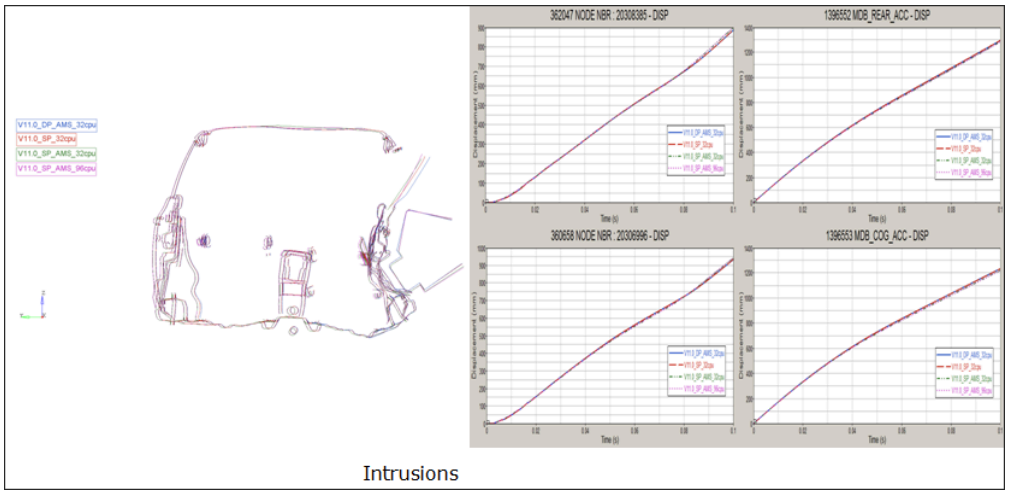


Figure 10: OEM4 Side Impact

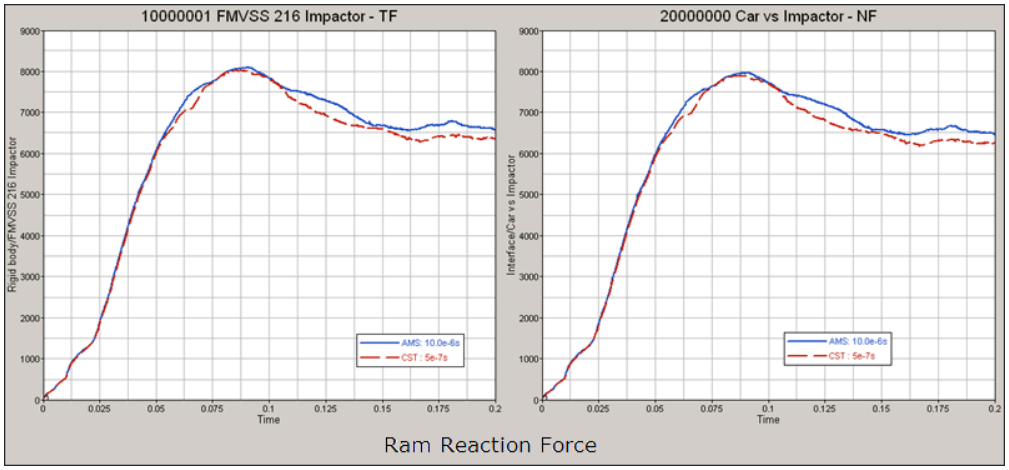


Figure 11: OEM4 Roof Crush

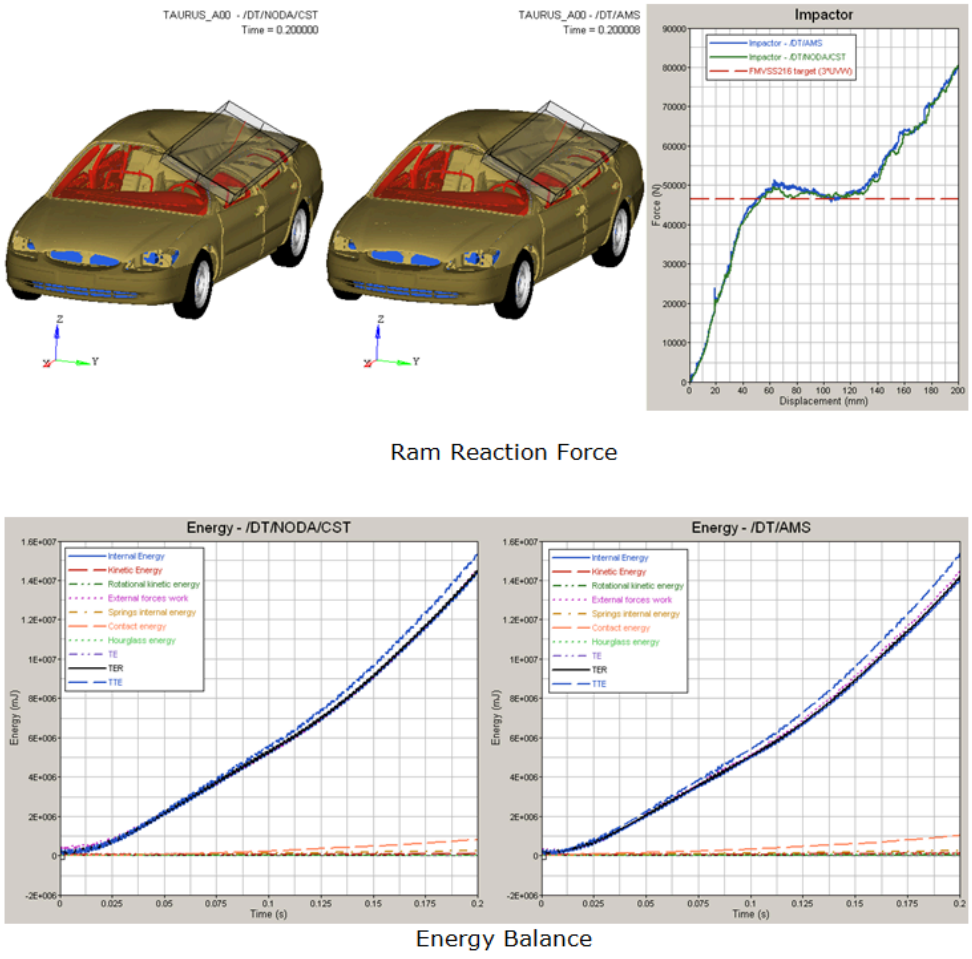
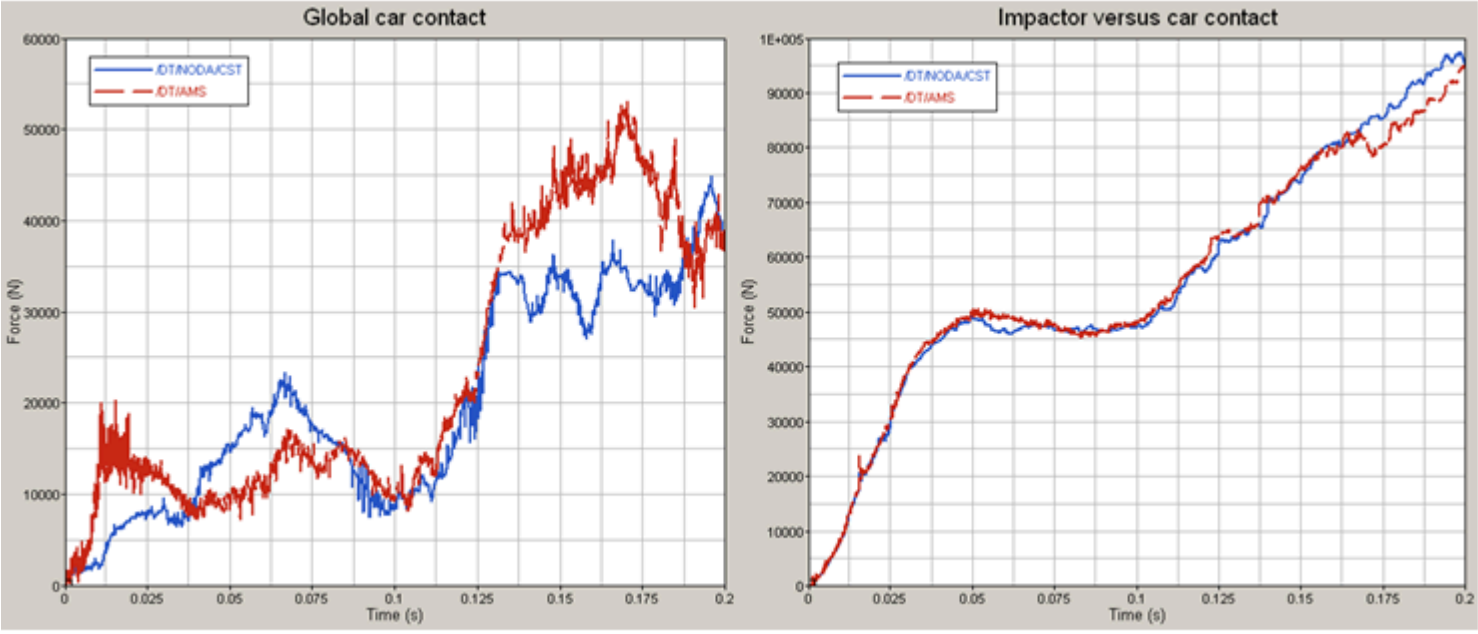


Figure 12: Ford Taurus Public Roof Crush



Contact Force Comparison

Figure 13:

Table 4: Summary of Full Vehicle Tests tried with AMS in Version 13.0

Load Case	Vehicle Program	Radioss Version	Number of Cores	Model Size	Original Nodal Time Step	Target Time Step	Speed-up v. Original Nodal Time Step	Comments	ODB Offset Deformable Barrier FFB Full Frontal
Front crash	Neon public	10.0.2		1M	1 ms	10 ms	4		Neon public
	Taurus public FFB	13.0	3 x 6 x 2	0.8 M	0.5 ms /dt/ inter/ del 0.4 ms	5 ms tol 0.001 /dt/ inter/ del 0.4 ms	1.87	Rigid wall replaced by rigid mesh and contact interface	
	OEM1 FFB	11.	24 x 1	1.3 M	0.5 ms	5 ms	3		
	OEM2 ODB	11.0.230	32 x 1	2 M	0.5 ms	5 ms	1.5		

Load Case	Vehicle Program	Radioss Version	Number of Cores	Model Size	Original Nodal Time Step	Target Time Step	Speed-up v. Original Nodal Time Step	Comments	ODB Offset Deformable Barrier FFB Full Frontal
	OEM2 ODB	13.0	3 x 6 x 2	2 M	0.5 ms /dt/ inter/ del 0.1 ms	5 ms tol 0.001 /dt/ inter/ del 0.1 ms	1.51		
	OEM4 FFB	13.0	3 x 6 x 2	1.5 M	0.5 ms /dt/ inter/ del 0.001 ms	4 ms tol 0.001 /dt/ inter/ del 0.001 ms	0.96		
	OEM4 FFB	11.0.230	3 x 6 x 2	6 M	0.5 ms /dt/ inter/ del 0.1 ms	5 ms /dt/ inter/ del 0.1 ms	1.3	Rigid wall replaced by rigid mesh and contact interface	
Rear crash	OEM1	13.0	3 x 6 x 2	1.1 M	0.1 ms /dt/ inter/ del 0.04 ms /dt/ brick/ cst 2.0 ms	2 ms tol 0.001 /dt/ inter/ del 0.04 ms /dt/ brick/ cst 2.0 ms	4.03		
Side impact	OEM4	13.0	3 x 6 x 2	1 M	0.5 ms	10 ms tol 0.001	2.11		

Load Case	Vehicle Program	Radioss Version	Number of Cores	Model Size	Original Nodal Time Step	Target Time Step	Speed-up v. Original Nodal Time Step	Comments	ODB Offset Deformable Barrier FFB Full Frontal
Roof Crush	OEM4 meshed	13.0	3 x 6 x 2	1.2 M	0.5 ms	10 ms tol 0.001	2.83	Rigid wall replaced by rigid mesh and contact interface	
	OEM4 meshed	13.0	3 x 6 x 2	1 M	0.5 ms	10 ms tol 0.001	1.13	Rigid wall replaced by rigid mesh and contact interface	
	Taurus public meshed rigid ram	13.0	3 x 6 x 2	0.8 M	0.5 ms	10 ms tol 0.001	3.80	Rigid wall replaced by rigid mesh and contact interface	
	Taurus public rigid wall ram	13.0	3 x 6 x 2	0.8 M	0.5 ms	10 ms tol 0.001	2.18	Rigid wall parallelepiped	

All speedup's compare original nodal time step with AMS both in double precision. The greyed rows relate to models that were not rerun in 13.0 and speedups belong to Table 4.

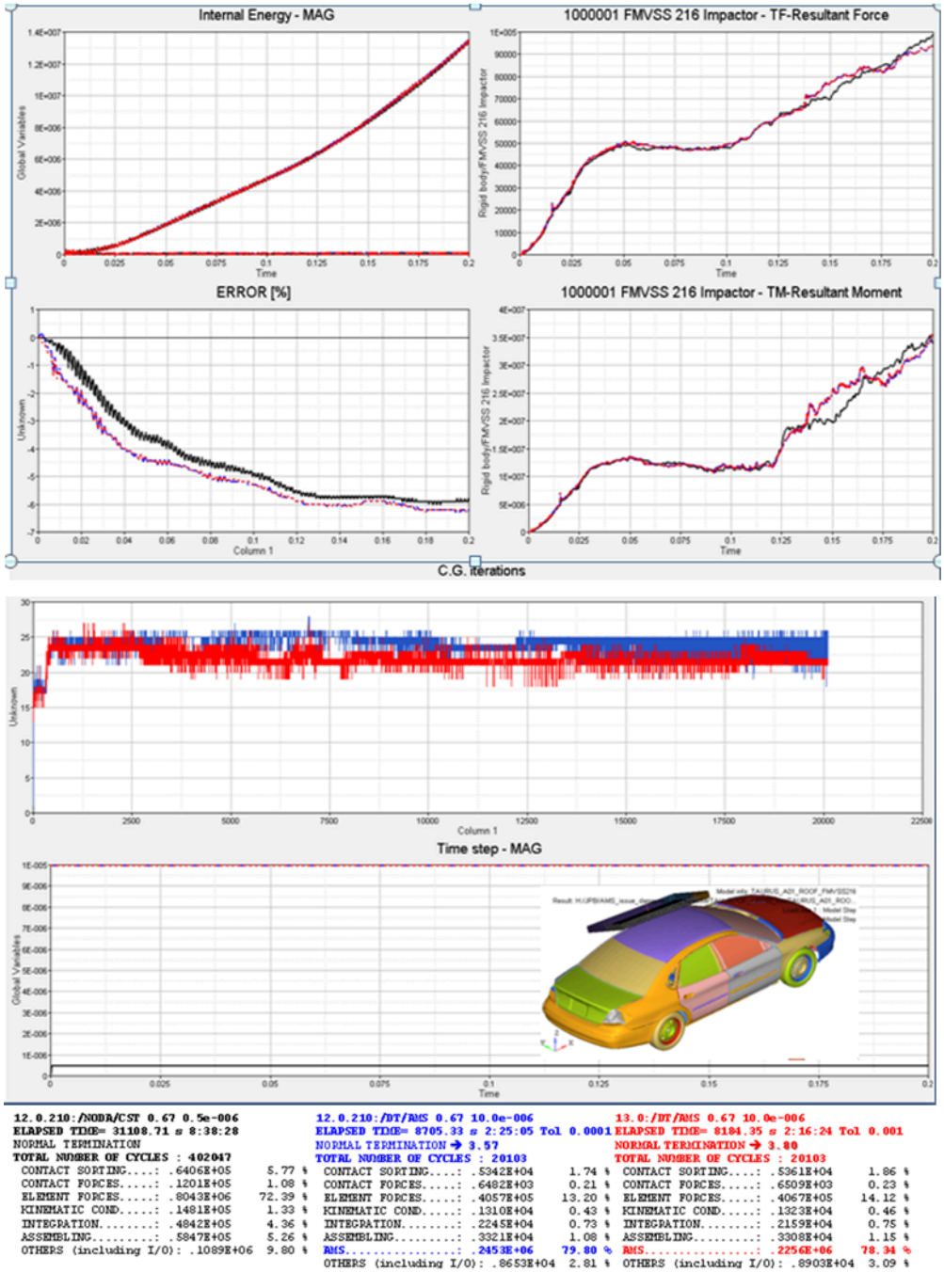


Figure 14: Ford Taurus Public Roof Crush

See Also

/AMS (Starter)

/DT/AMS (Engine)

Metal Forming Application Example

For metal forming and for quasi-static load cases, AMS allows obtaining the solution much faster and keep a high level of accuracy (of course, using the right targeted scale factor for the time step).

As an example, for stamping simulation, normally a scale factor of 20 provides an accurate and really close to reference results.

Other methods in metal forming simulations used to speed-up the solution are:

- Mesh adaptivity
- Dynamic condensation

In comparison with adaptive mesh, AMS is faster most of the time and is always more accurate than using adaptive meshing option (`/ADMESH`).

However, adaptivity is now compatible with AMS. To use an adaptive mesh and AMS together, dynamic condensation must be switched off.

Note:

1. AMS can be used with more than 2 levels of adaptivity. However, a scale factor of 20 might be too high, depending on the numbers of levels of adaptivity. This option must be used carefully.
2. When using HyperForm, there is no need for model changes, as optimized AMS parameters are automatically pre-defined.

See Also
/AMS (Starter)
/DT/AMS (Engine)

1.out File Examples

The following depicts, for one same model computed with Radioss version 11.0.201, the Engine output (`_0001.out`) file evolution from an over-estimated to an under-estimated imposed AMS time step.

See Also
/AMS (Starter)
/DT/AMS (Engine)

Over-Estimated /DT/AMS Target Time Step

20 Times the Reference /DT/NODA/CST

Note the progressive switch from element (`SOLID`) driven time step to interface (`INTER`) time step drop due to their spring reaction stiffness to finally end with an AMS divergence.

```
CYCLE    TIME        TIME-STEP ELEMENT           ERROR  I-ENERGY    K-ENERGY T  K-ENERGY R  EXT-WORK
 MAS.ERR
    0   0.000      0.3980E-02 SOLID    1026400   0.0%    0.000       0.4134E+06   0.000      -0.1719E-10
 0.7671E-02
    ANIMATION FILE: DFWDS734_AMS_FULL_REWELD_NOSECT_MAXA001 WRITTEN
   10  0.3980E-01 0.3980E-02 SOLID    1026953   0.0%  -4.176       0.4134E+06  0.2174       22.59
 0.7671E-02
```

```
   20  0.7960E-01 0.3980E-02 SOLID    1027171   0.0%   8.901      0.4135E+06  0.7490       81.41
 0.7671E-02
   30  0.1194     0.3980E-02 SOLID    1027865   0.0%   43.57      0.4135E+06   1.742       164.1
 0.7671E-02
   40  0.1592     0.3980E-02 SOLID    1026669   0.0%   94.41      0.4136E+06   4.155       261.0
 0.7671E-02
   50  0.1990     0.3980E-02 SOLID    1026928   0.0%   158.0      0.4136E+06   9.559       364.5
 0.7671E-02
   60  0.2388     0.3980E-02 SOLID    1027135   0.0%   230.8      0.4136E+06   17.50       470.2
 0.7671E-02
   70  0.2786     0.3980E-02 SOLID    1027670   0.0%   303.5      0.4137E+06   27.21       574.8
 0.7671E-02
   80  0.3184     0.3980E-02 SOLID    1027755   0.0%   366.8      0.4137E+06   36.79       674.5
 0.7671E-02
   90  0.3582     0.3980E-02 SOLID    1027323   0.0%   419.8      0.4137E+06   42.80       767.0
 0.7671E-02
  100  0.3980     0.3980E-02 SOLID    1026690   0.0%   464.3      0.4137E+06   46.51       850.7
 0.7671E-02
. . . . . .
```

```
CYCLE    TIME      TIME-STEP  ELEMENT          ERROR  I-ENERGY    K-ENERGY T   K-ENERGY R  EXT-WORK
 MAS.ERR
 1100   4.378    0.3980E-02 SOLID 1027754   0.0%   2558.      0.4116E+06   61.59       1016.
 0.7671E-02
 1110   4.418     0.3980E-02 SOLID  1027894   0.0%   2617.      0.4116E+06   63.80       1002.
 0.7671E-02
 1120   4.457     0.3980E-02 SOLID  1026231   0.0%   2680.      0.4115E+06   63.67       990.5
 0.7671E-02
 -- CONSTANT TIME STEP FOR SOLID ELEMENT NUMBER   7004937
 1130   4.497     0.3980E-02 SOLID   1027738   0.0%   2744.      0.4114E+06   61.73       980.9
 0.7671E-02
 1140   4.537     0.3980E-02 SOLID   1027858  -0.1%   2807.      0.4113E+06   59.50       973.5
 0.7671E-02
 -- CONSTANT TIME STEP FOR SOLID ELEMENT NUMBER   7004935
 1150   4.577     0.3980E-02 SOLID   1027880  -0.1%   2867.      0.4113E+06   58.56       968.7
 0.7671E-02
 1160   4.617     0.3980E-02 SOLID   1027278  -0.1%   2926.      0.4112E+06   59.27       966.8
 0.7671E-02
 -- CONSTANT TIME STEP FOR SOLID ELEMENT NUMBER   7004933
 1170   4.656      0.3980E-02 SOLID   1027327  -0.1%   2984.      0.4111E+06   62.11       968.2
 0.7671E-02
 -- CONSTANT TIME STEP FOR SOLID ELEMENT NUMBER   7004931
 1180   4.696     0.3916E-02 INTER  42009748  -0.1%   3047.      0.4111E+06   66.28       973.2
 0.7671E-02
 1190   4.735     0.3779E-02 INTER   42009748  -0.1%   3111.      0.4110E+06   72.50       982.0
 0.7671E-02
 1200   4.772     0.3709E-02 INTER   42009748  -0.1%   3178.      0.4109E+06   80.35       994.9
 0.7671E-02
 -- CONSTANT TIME STEP FOR SOLID ELEMENT NUMBER   7004929
 1210   4.809     0.3628E-02 INTER   42009748  -0.1%   3247.      0.4109E+06   89.95       1012.
 0.7671E-02
 1220   4.845     0.3591E-02 INTER   42009748  -0.1%   3319.      0.4108E+06   98.18       1033.
 0.7671E-02
 1230   4.881     0.3659E-02 INTER   42009748  -0.1%   3394.      0.4107E+06   104.3       1060.
 0.7671E-02
 -- CONSTANT TIME STEP FOR SOLID ELEMENT NUMBER   7004927
 1240   4.917     0.3621E-02 INTER   42009748  -0.1%   3474.      0.4107E+06   108.6       1090.
 0.7671E-02
 1250   4.954     0.3656E-02 INTER   42009748  -0.1%   3561.      0.4106E+06   113.7       1124.
 0.7671E-02
 1260   4.990     0.3622E-02 INTER   42009748  -0.1%   3656.      0.4105E+06   117.2       1159.
 0.7671E-02
     ANIMATION FILE: DFWDS734_AMS_FULL_REWELD_NOSECT_MAXA002 WRITTEN
 1270   5.026     0.3631E-02 INTER   42009748  -0.1%   3760.      0.4105E+06   122.4       1194.
 0.7671E-02
 1280   5.062     0.3585E-02 INTER   42009748  -0.1%   3865.      0.4104E+06   122.7       1226.
 0.7671E-02
 1290   5.098     0.3567E-02 INTER   42009748  -0.1%   3972.      0.4103E+06   125.0       1255.
 0.7671E-02
 1300   5.133     0.3578E-02 INTER   42009748  -0.1%   4074.      0.4102E+06   126.3       1280.
 0.7671E-02

. . . . . .

 39500   79.47     0.6686E-03 INTER   42009748  -9.4%  0.2141E+06  0.1554E+06   1465.      -4253.
 0.7671E-02
```

```
 -- RUPTURE OF SHELL ELEMENT NUMBER     873536
 -- RUPTURE OF SHELL ELEMENT NUMBER     873327
 39510   79.48     0.6686E-03 INTER   42009748  -9.4%  0.2141E+06  0.1554E+06   1465.      -4254.
 0.7671E-02
 -- RUPTURE OF SHELL ELEMENT NUMBER     873995
 39520   79.48     0.6685E-03 INTER   42009748  -9.4%  0.2141E+06  0.1554E+06   1466.      -4255.
 0.7671E-02
 -- RUPTURE OF SHELL ELEMENT NUMBER     872733
 39530   79.49     0.6684E-03 INTER   42009748  -9.4%  0.2141E+06  0.1554E+06   1466.      -4256.
 0.7671E-02
 -- RUPTURE OF SHELL ELEMENT NUMBER     876109
 -- RUPTURE OF SHELL ELEMENT NUMBER     876112
 39540   79.50     0.6684E-03 INTER   42009748  -9.4%  0.2141E+06  0.1554E+06   1467.      -4257.
 0.7671E-02
 -- RUPTURE OF SHELL ELEMENT NUMBER     874081
 -- RUPTURE OF SHELL ELEMENT NUMBER     874082
 ** ERROR : AMS IS LIKELY DIVERGING:
    TOTAL C.G. ITERATION NUMBER =     1000 AT CYCLE NUMBER    39550
39550   79.50     0.6685E-03 INTER   42009748   NaN%  0.2141E+06         NaN          NaN  -4258.
 0.7671E-02

                          ** CPU USER TIME **
```

Case: AMSmaxi_ x20 _ div 79.5stop_32cpu_14.5h_ 110201 sep30

Almost Acceptable /DT/AMS Target Time Step

13 Times the Reference /DT/NODA/CST

Note the late switch from element (SH3N and SOLID) driven time step to interface (INTER) time step drop due to their spring reaction stiffness and node (NODE) driven due to rotations whereas element translation was initially controlling the time step.

(No AMS divergence, the job successfully ended at the 37220th cycle for 100ms)

```
CYCLE     TIME       TIME-STEP ELEMENT            ERROR  I-ENERGY   K-ENERGY T   K-ENERGY R   EXT-WORK
 MAS.ERR
     0     0.000     0.2736E-02 SOLID    1026805    0.0%   0.000    0.4134E+06     0.000       .1719E-10
 0.7671E-02
     ANIMATION FILE: DFWDS734_AMS_FULL_REWELD_NOSECT_OPT1A001 WRITTEN
    10    0.2736E-01 0.2736E-02 SOLID    1026339    0.0%  -9.938    0.4134E+06     0.9492E-01     11.27
 0.7671E-02
    20    0.5472E-01 0.2736E-02 SH_3N     980467    0.05%  -2.140    0.4135E+06     0.2769         41.40
 0.7671E-02
    30    0.8209E-01 0.2736E-02 SH_3N     983904    0.0%   12.94    0.4135E+06     0.7553         86.61
 0.7671E-02
    40    0.1094     0.2736E-02 SOLID    1025858    0.0%   34.57    0.4135E+06     1.560          142.9
 0.7671E-02
    50    0.1368     0.2736E-02 SOLID    1026994    0.0%   65.14    0.4136E+06     2.777          207.0
 0.7671E-02
    60    0.1642     0.2736E-02 SH_3N     979591    0.0%   102.8    0.4136E+06      4.844         276.2
 0.7671E-02
    70    0.1915     0.2736E-02 SH_3N    1011870    0.0%   147.1    0.4136E+06     8.051          348.0
 0.7671E-02
    80    0.2189     0.2736E-02 SOLID    1027827    0.0%   197.0    0.4136E+06     12.51          421.2
 0.7671E-02
    90    0.2463     0.2736E-02 SOLID    1025763    0.0%   249.5    0.4136E+06     17.97          494.7
 0.7671E-02
   100    0.2736     0.2736E-02 SOLID    1026760    0.0%   300.5    0.4137E+06     24.45          567.3
 0.7671E-02

. . . . . .

-- RUPTURE OF SHELL ELEMENT NUMBER    9861056
 CYCLE    TIME      TIME-STEP  ELEMENT            ERROR   I-ENERGY    K-ENERGY T   K-ENERGY R   EXT-WORK
 MAS.ERR
  5500   15.04     0.2736E-02 SH_3N     980451    -1.0%  0.3598E+05  0.3741E+06    377.9          1060.
 0.7671E-02
  5510   15.06     0.2736E-02 SH_3N     980451    -1.0%  0.3607E+05  0.3740E+06    379.4          1048.
 0.7671E-02
  5520   15.09     0.2736E-02 SH_3N     980440    -1.0%  0.3614E+05  0.3739E+06    380.1          1035.
 0.7671E-02
```

```
  5530   15.12     0.2736E-02 SH_3N     983880    -1.0%  0.3623E+05  0.3738E+06    379.8         1021.
 0.7671E-02
  5540   15.15     0.2736E-02 SH_3N    1012368    -1.0%  0.3630E+05  0.3737E+06    379.0         1005.
 0.7671E-02
  5550   15.17     0.2736E-02 SH_3N     980451    -1.0%  0.3638E+05  0.3736E+06    377.9         987.8
 0.7671E-02
 -- RUPTURE OF SHELL ELEMENT NUMBER    9897179
  5560   15.20     0.2736E-02 SH_3N     980112    -1.0%  0.3645E+05  0.3735E+06    377.0         969.7
 0.7671E-02
 -- RUPTURE OF SHELL ELEMENT NUMBER    9896885
  5570   15.23     0.2736E-02 SH_3N    1015803    -1.0%  0.3652E+05  0.3734E+06    376.7         950.6
 0.7671E-02
 -- RUPTURE OF SHELL ELEMENT NUMBER    9897148
  5580   15.26     0.2736E-02 SH_3N    1012363    -1.0%  0.3659E+05  0.3733E+06    377.0         930.5
 0.7671E-02
  5590   15.28     0.2736E-02 SH_3N    1018445    -1.0%  0.3666E+05  0.3732E+06    377.9         909.5
 0.7671E-02
  5600   15.31     0.2736E-02 SH_3N     971703    -1.0%  0.3673E+05  0.3731E+06    378.8         887.8
 0.7671E-02

. . . . . .

 18200   48.77     0.2393E-02 INTER   42009748    -6.2%  0.1561E+06   0.2282E+06   888.8         -2873.
 0.7671E-02
 -- CONSTANT TIME STEP FOR SOLID ELEMENT NUMBER  7001813
 18210   48.79     0.2431E-02 INTER   42009748    -6.2%  0.1562E+06   0.2281E+06   889.2         -2873.
 0.7671E-02
 -- CONSTANT TIME STEP FOR SOLID ELEMENT NUMBER  7002310
 -- CONSTANT TIME STEP FOR SOLID ELEMENT NUMBER  7001709
 18220   48.82     0.2495E-02 INTER   42009748    -6.2%  0.1563E+06   0.2280E+06   888.3         -2872.
 0.7671E-02
 18230   48.84     0.2607E-02 INTER   42009748    -6.2%  0.1564E+06   0.2279E+06   891.0         -2870.
 0.7671E-02
 -- CONSTANT TIME STEP FOR SOLID ELEMENT NUMBER  7001821
 -- CONSTANT TIME STEP FOR SOLID ELEMENT NUMBER  7002666
 18240   48.87     0.2668E-02 NODE    10103704    -6.2%  0.1564E+06   0.2278E+06   893.2          -2869.
 0.7671E-02
 -- CONSTANT TIME STEP FOR SOLID ELEMENT NUMBER  7002205
 18250   48.90     0.2665E-02 NODE    10103704    -6.2%  0.1565E+06   0.2277E+06   896.1          -2867.
 0.7671E-02
 18260   48.92     0.2663E-02 NODE    10103704    -6.2%  0.1566E+06   0.2276E+06   897.3          -2865.
 0.7671E-02
 -- CONSTANT TIME STEP FOR SOLID ELEMENT NUMBER  7001710
 18270   48.95     0.2614E-02 INTER   42009748    -6.2%  0.1567E+06   0.2275E+06   897.7          -2863.
 0.7671E-02
 18280   48.98     0.2599E-02 INTER   42009748    -6.2%  0.1568E+06   0.2274E+06   895.9          -2861.
 0.7671E-02
 -- CONSTANT TIME STEP FOR SOLID ELEMENT NUMBER  7001702
 18290   49.00     0.2655E-02 NODE    10103704    -6.2%  0.1568E+06   0.2273E+06   895.6          -2860.
 0.7671E-02
 18300   49.03     0.2736E-02 SH_3N     979131    -6.2%  0.1569E+06   0.2272E+06   896.7          -2859.
 0.7671E-02

. . . . . .



37100   99.69     0.2612E-02 INTER   42009748     -9.4%  0.2150E+06   0.1548E+06    1712.         -3241.
 0.7671E-02
37110   99.71     0.2633E-02 INTER   42009748     -9.4%  0.2150E+06   0.1548E+06    1712.         -3239.
 0.7671E-02
 -- RUPTURE OF SHELL ELEMENT NUMBER     872981
37120   99.74     0.2631E-02 INTER   42009748     -9.4%  0.2150E+06   0.1549E+06    1711.         -3237.
 0.7671E-02
 -- RUPTURE OF SHELL ELEMENT NUMBER     876845
37130   99.77     0.2630E-02 INTER   42009748     -9.4%  0.2150E+06   0.1549E+06    1711.         -3235.
 0.7671E-02
 -- RUPTURE OF SHELL ELEMENT NUMBER     877165
37140   99.79     0.2627E-02 INTER   42009748     -9.4%  0.2150E+06   0.1549E+06    1710.         -3233.
 0.7671E-02
 -- RUPTURE OF SHELL ELEMENT NUMBER     872868
37150   99.82     0.2609E-02 INTER   42009748     -9.4%  0.2150E+06   0.1549E+06    1709.         -3231.
 0.7671E-02
 -- RUPTURE OF SHELL ELEMENT NUMBER     877277
 -- RUPTURE OF SHELL ELEMENT NUMBER     877278
37160   99.84     0.2669E-02 INTER   42009748     -9.4%  0.2150E+06   0.1549E+06    1709.         -3229.
 0.7671E-02
 -- RUPTURE OF SHELL ELEMENT NUMBER     877051
```

```
37170   99.87     0.2671E-02 INTER   42009748     -9.4%  0.2150E+06   0.1549E+06    1708.         -3227.
 0.7671E-02
 -- RUPTURE OF SHELL ELEMENT NUMBER     876266
37180   99.90     0.2671E-02 INTER   42009748     -9.4%  0.2150E+06   0.1549E+06    1706.         -3225.
 0.7671E-02
37190   99.92     0.2660E-02 INTER   42009748     -9.4%  0.2150E+06   0.1549E+06    1705.         -3224.
 0.7671E-02
37200   99.95
0.2643E-02 INTER   42009748     -9.4%  0.2150E+06   0.1549E+06    1704.         -3223.    0.7671E-02
37210   99.98     0.2643E-02 INTER   42009748     -9.4%  0.2150E+06   0.1549E+06    1703.         -3222.
 0.7671E-02
     ANIMATION FILE: DFWDS734_AMS_FULL_REWELD_NOSECT_OPT1A021 WRITTEN
37220   100.0     0.2623E-02 INTER   42009748     -9.4%  0.2150E+06   0.1549E+06    1703.         -3221.
 0.7671E-02
     RESTART FILES: DFWDS734_AMS_FULL_REWELD_NOSECT_OPT1_0001_[0001-0004].rst WRITTEN

     -------------
** CPU USER TIME **
```

Case: AMSopt1_x13_16cpu_26h_110201sep30

Acceptable /DT/AMS Target Time Step

12.5 Times the Reference /DT/NODA/CST

Note the essentially element (SH3N) driven time step with exceptional interface (INTER) time step influence and no node (NODE) driven time step.

(No AMS divergence, the job successfully ended at the 40230th cycle for 100ms)

```
    CYCLE    TIME       TIME-STEP ELEMENT            ERROR  I-ENERGY   K-ENERGY T  K-ENERGY R  EXT-WORK
      MAS.ERR      0    0.000      0.2487E-02 SH_3N     987177   0.0%    0.000      0.4134E+06   0.000
   -0.1719E-10  0.7671E-02      ANIMATION FILE: DFWDS734_AMS_FULL_REWELD_NOSECT_OPTA001 WRITTEN     10
  0.2487E-01  0.2487E-02 SH_3N     1004312   0.0%  -11.02      0.4134E+06  0.7503E-01   9.448      0.7671E-02
     20  0.4975E-01  0.2487E-02 SH_3N     994772   0.0%  -2.833      0.4135E+06  0.2556       34.67
   0.7671E-02     30  0.7462E-01  0.2487E-02 SH_3N     980477   0.0%   8.087      0.4135E+06  0.5928
    73.10      0.7671E-02     40  0.9950E-01  0.2487E-02 SH_3N     980465   0.0%   25.51      0.4135E+06
    1.253       121.5      0.7671E-02     50  0.1244      0.2487E-02 SH_3N     994783   0.0%   50.64
   0.4135E+06   2.161       177.3      0.7671E-02     60  0.1492      0.2487E-02 SH_3N     983873   0.0%
    81.66      0.4136E+06   3.542       238.2      0.7671E-02     70  0.1741      0.2487E-02 SH_3N
  983929   0.0%   118.5      0.4136E+06   5.751       302.3      0.7671E-02     80  0.1990      0.2487E-02
  SH_3N     980233   0.0%   160.6      0.4136E+06   9.149       368.2      0.7671E-02     90  0.2239
   0.2487E-02 SH_3N     980624   0.0%   206.9      0.4136E+06   13.47       434.9      0.7671E-02    100
  0.2487      0.2487E-02 SH_3N     985772   0.0%   254.8      0.4136E+06   18.63       501.6      0.7671E-02
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
    4980   12.39      0.2487E-02 SH_3N     983878  -0.7%  0.2624E+05  0.3855E+06   319.2       1427.
   0.7671E-02   4990   12.41      0.2487E-02 SH_3N     983854  -0.7%  0.2633E+05  0.3854E+06   317.9
   1422.      0.7671E-02   5000   12.44      0.2487E-02 SH_3N     980375  -0.7%  0.2642E+05  0.3852E+06
  317.3       1416.      0.7671E-02>
  -- RUPTURE OF SHELL ELEMENT NUMBER     9897228
   5010   12.46      0.2487E-02 SH_3N     986833  -0.7%  0.2651E+05  0.3851E+06   317.1       1409.
  0.7671E-02
   5020   12.49      0.2487E-02 SH_3N     980402  -0.7%  0.2660E+05  0.3850E+06   316.0       1402.
  0.7671E-02
   5030   12.51      0.2487E-02 SH_3N     978799  -0.7%  0.2669E+05  0.3849E+06   316.4       1393.
  0.7671E-02
  -- RUPTURE OF SHELL ELEMENT NUMBER     9897051
   5040   12.54     0.2440E-02 INTER    42009748  -0.7%  0.2679E+05  0.3848E+06   316.3       1384.
  0.7671E-02
  -- RUPTURE OF SHELL ELEMENT NUMBER     9897049
   5050   12.56     0.2473E-02 INTER     42009748  -0.7%  0.2687E+05  0.3847E+06   316.3       1375.
  0.7671E-02
  -- RUPTURE OF SHELL ELEMENT NUMBER     9897202
  -- RUPTURE OF SHELL ELEMENT NUMBER     9897044
   5060   12.59      0.2487E-02 SH_3N     985187  -0.7%  0.2696E+05  0.3846E+06   317.1       1364.
  0.7671E-02
   5070   12.61      0.2487E-02 SH_3N     983885  -0.7%  0.2705E+05  0.3845E+06   318.2       1353.
  0.7671E-02
   5080   12.64      0.2487E-02 SH_3N     980507  -0.7%  0.2713E+05  0.3844E+06   319.4       1342.
  0.7671E-02
. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
```

```
 40000   99.43      0.2487E-02 SH_3N     983740  -6.5%  0.2268E+06  0.1548E+06   1688.      -3544.
 0.7671E-02
 40010   99.46      0.2487E-02 SH_3N     981895  -6.5%  0.2268E+06  0.1548E+06   1691.      -3540.
 0.7671E-02
 40020   99.48      0.2487E-02 SH_3N     984028  -6.5%  0.2268E+06  0.1548E+06   1694.      -3536.
 0.7671E-02
 40030   99.51      0.2487E-02 SH_3N     982888  -6.5%  0.2268E+06  0.1549E+06   1697.      -3532.
 0.7671E-02
 40040   99.53      0.2487E-02 SH_3N     993482  -6.5%  0.2268E+06  0.1549E+06   1700.      -3528.
 0.7671E-02
 40050   99.55      0.2487E-02 SH_3N    1002607  -6.5%  0.2268E+06  0.1549E+06   1703.      -3525.
 0.7671E-02
 40060   99.58      0.2487E-02 SH_3N     986010  -6.5%  0.2268E+06  0.1549E+06   1706.      -3521.
 0.7671E-02
 40070   99.60      0.2487E-02 SH_3N     981287  -6.5%  0.2268E+06  0.1549E+06   1708.      -3518.
 0.7671E-02
 -- RUPTURE OF SHELL ELEMENT NUMBER      876247
 40080   99.63      0.2487E-02 SH_3N     993747  -6.5%  0.2268E+06  0.1549E+06   1711.      -3515.
 0.7671E-02
 -- RUPTURE OF SHELL ELEMENT NUMBER      876267
 40090   99.65      0.2487E-02 SH_3N     969592  -6.5%  0.2268E+06  0.1549E+06   1712.      -3512.
 0.7671E-02
 40100   99.68      0.2487E-02 SH_3N     985183  -6.5%  0.2268E+06  0.1549E+06   1713.      -3510.
 0.7671E-02
 -- RUPTURE OF SHELL ELEMENT NUMBER      876245
 40110   99.70      0.2487E-02 SH_3N     979126  -6.5%  0.2268E+06  0.1549E+06   1710.      -3507.
 0.7671E-02
 40120   99.73      0.2487E-02 SH_3N     981287  -6.4%  0.2269E+06  0.1549E+06   1705.      -3504.
 0.7671E-02
 -- RUPTURE OF SHELL ELEMENT NUMBER      876374
 40130   99.75      0.2487E-02 SH_3N     985787  -6.4%  0.2269E+06  0.1549E+06   1699.      -3501.
 0.7671E-02
 40140   99.78      0.2487E-02 SH_3N     981843  -6.4%  0.2269E+06  0.1549E+06   1694.      -3498.
 0.7671E-02
 40150   99.80      0.2487E-02 SH_3N     986153  -6.4%  0.2269E+06  0.1549E+06   1691.      -3495.
 0.7671E-02
 40160   99.83      0.2487E-02 SH_3N     993115  -6.4%  0.2269E+06  0.1549E+06   1689.      -3492.
 0.7671E-02
 40170   99.85      0.2487E-02 SH_3N     995959  -6.4%  0.2269E+06  0.1549E+06   1688.      -3490.
 0.7671E-02
 40180   99.88      0.2487E-02 SH_3N     979140  -6.4%  0.2269E+06  0.1549E+06   1688.      -3487.
 0.7671E-02
 40190   99.90      0.2487E-02 SH_3N     985244  -6.4%  0.2269E+06  0.1549E+06   1689.      -3485.
 0.7671E-02
 40200   99.93      0.2487E-02 SH_3N     993949  -6.4%  0.2269E+06  0.1549E+06   1691.      -3483.
 0.7671E-02
 40210   99.95      0.2487E-02 SH_3N     988714  -6.4%  0.2269E+06  0.1550E+06   1694.      -3481.
 0.7671E-02
 40220   99.98      0.2487E-02 SH_3N     986153  -6.4%  0.2269E+06  0.1550E+06   1697.      -3479.
 0.7671E-02
     ANIMATION FILE: DFWDS734_AMS_FULL_REWELD_NOSECT_OPTA021 WRITTEN
40230   100.0      0.2487E-02 SH_3N     980714  -6.4%  0.2269E+06  0.1550E+06   1698.      -3477.
 0.7671E-02
     RESTART FILES: DFWDS734_AMS_FULL_REWELD_NOSECT_OPT_0001_[0001-0004].rst WRITTEN
     -------------
 ** CPU USER TIME **
```

Case: AMSopt_x12.5_32cpu_17.8h_110201sep30

Under-estimated /DT/AMS Target Time Step

10 Times the Reference /DT/NODA/CST

Note the uniquely element (SH3N) driven time step. No interface (INTER) or node (NODE) driven time step (the job successfully ended at the 50250th cycle for 100ms)

```
CYCLE    TIME       TIME-STEP ELEMENT           ERROR  I-ENERGY   K-ENERGY T  K-ENERGY R  EXT-WORK
 MAS.ERR
     0   0.000      0.1990E-02 SOLID    1026809   0.0%   0.000      0.4134E+06   0.000      -0.1719E-10
 0.7671E-02
     ANIMATION FILE: DFWDS734_AMS_FULL_REWELD_NOSECTA001 WRITTEN
```

```
    10  0.1990E-01 0.1990E-02 SOLID    1027233   0.0%  -12.55      0.4134E+06  0.4509E-01   6.263
 0.7671E-02
    20  0.3980E-01 0.1990E-02 SOLID    1026793   0.0%  -3.404      0.4134E+06  0.1923       22.77
 0.7671E-02
    30  0.5970E-01 0.1990E-02 SOLID    1026919   0.0% -0.3391      0.4135E+06  0.3412       48.73
 0.7671E-02
    40  0.7960E-01 0.1990E-02 SOLID    1026987   0.0%   11.02      0.4135E+06  0.7014       82.15
 0.7671E-02
    50  0.9950E-01 0.1990E-02 SOLID    1027291   0.0%   25.55      0.4135E+06   1.259       121.6
 0.7671E-02
    60  0.1194     0.1990E-02 SOLID    1026789   0.0%   45.07      0.4135E+06   1.935       165.8
 0.7671E-02
    70  0.1393     0.1990E-02 SOLID    1026910   0.0%   68.49      0.4136E+06   2.833       213.5
 0.7671E-02
    80  0.1592     0.1990E-02 SOLID    1027497   0.0%   95.78      0.4136E+06   4.219       263.7
 0.7671E-02
    90  0.1791     0.1990E-02 SOLID    1027432   0.0%   126.7      0.4136E+06   6.282       315.5
 0.7671E-02
   100  0.1990     0.1990E-02 SOLID    1027197   0.0%   160.8      0.4136E+06   9.184       368.2
 0.7671E-02

. . . . . .

  6220   12.38     0.1990E-02 SOLID    1027341  -0.6%  0.2623E+05  0.3856E+06   304.8       1425.
 0.7671E-02
 -- RUPTURE OF SHELL ELEMENT NUMBER    9897042
  6230   12.40     0.1990E-02 SOLID    1026467  -0.6%  0.2630E+05  0.3856E+06   304.8       1422.
 0.7671E-02
  6240   12.42     0.1990E-02 SOLID    1027168  -0.7%  0.2637E+05  0.3855E+06   303.9       1418.
 0.7671E-02
  6250   12.44     0.1990E-02 SOLID    1026994  -0.7%  0.2645E+05  0.3854E+06   303.5       1413.
 0.7671E-02
  6260   12.46     0.1990E-02 SOLID    1026918  -0.7%  0.2652E+05  0.3853E+06   302.8       1408.
 0.7671E-02
  6270   12.48     0.1990E-02 SOLID    1027344  -0.7%  0.2659E+05  0.3852E+06   301.9       1402.
 0.7671E-02
 -- RUPTURE OF SHELL ELEMENT NUMBER    9897040
  6280   12.50     0.1990E-02 SOLID    1027487  -0.7%  0.2667E+05  0.3851E+06   300.8       1395.
 0.7671E-02
 -- RUPTURE OF SHELL ELEMENT NUMBER    9897152
  6290   12.52     0.1990E-02 SOLID    1027341  -0.7%  0.2674E+05  0.3850E+06   300.0       1388.
 0.7671E-02
  6300   12.54     0.1990E-02 SOLID    1027156  -0.7%  0.2681E+05  0.3849E+06   299.5       1381.
 0.7671E-02
  6310   12.56     0.1990E-02 SOLID    1027292  -0.7%  0.2688E+05  0.3849E+06   299.1       1372.
 0.7671E-02
 -- RUPTURE OF SHELL ELEMENT NUMBER    9897037
  6320   12.58     0.1990E-02 SOLID    1026512  -0.7%  0.2696E+05  0.3848E+06   299.1       1364.
 0.7671E-02
 -- CONSTANT TIME STEP FOR SOLID ELEMENT NUMBER  7004883
  6330   12.60     0.1990E-02 SOLID    1027473  -0.7%  0.2702E+05  0.3847E+06   299.6       1355.
 0.7671E-02
  6340   12.62     0.1990E-02 SOLID    1027340  -0.7%  0.2709E+05  0.3846E+06   300.6       1346.
 0.7671E-02
  6350   12.64     0.1990E-02 SOLID    1027345  -0.7%  0.2716E+05  0.3845E+06   301.1       1337.
 0.7671E-02

. . . . . .

 49970   99.44     0.1990E-02 SOLID    1026732  -7.5%  0.2226E+06  0.1550E+06   1683.      -3546.
 0.7671E-02
 -- RUPTURE OF SHELL ELEMENT NUMBER     878377
 49980   99.46     0.1990E-02 SOLID    1027394  -7.5%  0.2226E+06  0.1550E+06   1683.      -3543.
 0.7671E-02
 49990   99.48     0.1990E-02 SOLID    1026474  -7.5%  0.2226E+06  0.1550E+06   1683.      -3541.
 0.7671E-02
 50000   99.50     0.1990E-02 SOLID    1026918  -7.5%  0.2226E+06  0.1550E+06   1683.      -3538.
 0.7671E-02
 50010   99.52     0.1990E-02 SOLID    1026842  -7.5%  0.2226E+06  0.1550E+06   1684.      -3536.
 0.7671E-02
 -- RUPTURE OF SHELL ELEMENT NUMBER     877707
 50020   99.54     0.1990E-02 SOLID    1026846  -7.5%  0.2226E+06  0.1550E+06   1684.      -3533.
 0.7671E-02
 50030   99.56     0.1990E-02 SOLID    1026916  -7.5%  0.2226E+06  0.1550E+06   1685.      -3531.
 0.7671E-02
 50040   99.58     0.1990E-02 SOLID    1026846  -7.5%  0.2226E+06  0.1550E+06   1686.      -3529.
 0.7671E-02
 50050   99.60     0.1990E-02 SHELL     389951  -7.4%  0.2226E+06  0.1550E+06   1686.      -3526.
 0.7671E-02
```

```
 50060   99.62     0.1990E-02 SOLID    1026916  -7.4%  0.2226E+06  0.1550E+06   1686.      -3524.
 0.7671E-02
 50070   99.64     0.1990E-02 SOLID    1027698  -7.4%  0.2226E+06  0.1551E+06   1687.      -3522.
 0.7671E-02
 50080   99.66     0.1990E-02 SHELL     389982  -7.4%  0.2227E+06  0.1551E+06   1687.      -3520.
 0.7671E-02
 -- RUPTURE OF SHELL ELEMENT NUMBER     877600
 50090   99.68     0.1990E-02 SHELL     390246  -7.4%  0.2227E+06  0.1551E+06   1688.      -3518.
 0.7671E-02
 -- RUPTURE OF SHELL ELEMENT NUMBER     872853
 50100   99.70     0.1990E-02 SOLID    1026474  -7.4%  0.2227E+06  0.1551E+06   1688.      -3515.
 0.7671E-02
 50110   99.72     0.1990E-02 SOLID    1026911  -7.4%  0.2227E+06  0.1551E+06   1689.      -3513.
 0.7671E-02
 50120   99.74     0.1990E-02 SOLID    1026914  -7.4%  0.2227E+06  0.1551E+06   1689.      -3511.
 0.7671E-02
 -- RUPTURE OF SHELL ELEMENT NUMBER     877814
 50130   99.76     0.1990E-02 SOLID    1026500  -7.4%  0.2227E+06  0.1551E+06   1690.      -3508.
 0.7671E-02
 -- RUPTURE OF SHELL ELEMENT NUMBER     875485
 50140   99.78     0.1990E-02 SOLID    1027163  -7.4%  0.2227E+06  0.1551E+06   1690.      -3506.
 0.7671E-02
 -- RUPTURE OF SHELL ELEMENT NUMBER     880641
 50150   99.80     0.1990E-02 SOLID   1026107  -7.4%  0.2227E+06  0.1551E+06   1690.      -3504.
 0.7671E-02
 50160   99.82     0.1990E-02 SOLID    1027004  -7.4%  0.2227E+06  0.1551E+06   1691.      -3501.
 0.7671E-02
 -- RUPTURE OF SHELL ELEMENT NUMBER     877598
 50170   99.84     0.1990E-02 SOLID    1026998  -7.4%  0.2227E+06  0.1551E+06   1690.      -3499.
 0.7671E-02
 50180   99.86     0.1990E-02 SHELL     390073  -7.4%  0.2227E+06  0.1551E+06   1690.      -3497.
 0.7671E-02
 -- RUPTURE OF SHELL ELEMENT NUMBER     877149
 50190   99.88     0.1990E-02 SHELL     254450  -7.4%  0.2227E+06  0.1551E+06   1690.      -3495.
 0.7671E-02
 50200   99.90     0.1990E-02 SOLID    1026757  -7.4%  0.2227E+06  0.1551E+06   1690.      -3493.
 0.7671E-02
 -- RUPTURE OF SHELL ELEMENT NUMBER     880538
 50210   99.92     0.1990E-02 SOLID    1026603  -7.4%  0.2227E+06  0.1551E+06   1690.      -3491.
 0.7671E-02
 50220   99.94     0.1990E-02 SOLID    1027433  -7.4%  0.2227E+06  0.1551E+06   1690.      -3489.
 0.7671E-02
 -- RUPTURE OF SHELL ELEMENT NUMBER     877599
 50230   99.96     0.1990E-02 SOLID    1026799  -7.4%  0.2227E+06  0.1551E+06   1690.      -3488.
 0.7671E-02
 50240   99.98     0.1990E-02 SOLID    1027571  -7.4%  0.2227E+06  0.1551E+06   1690.      -3486.
 0.7671E-02
 50250   100.0     0.1990E-02 SOLID    1027536  -7.4%  0.2227E+06  0.1552E+06   1690.      -3485.
 0.7671E-02
 -- RUPTURE OF SHELL ELEMENT NUMBER     872739
     ANIMATION FILE: DFWDS734_AMS_FULL_REWELD_NOSECTA021 WRITTEN
 -- RUPTURE OF SHELL ELEMENT NUMBER     872967

     RESTART FILES: DFWDS734_AMS_FULL_REWELD_NOSECT_0001_[0001-0004].rst WRITTEN
    -------------

                           ** CPU USER TIME **
```

Case: AMSnorm_x10_16cpu_33.7h_110201sep30

Time Step Scale Factor

The theoretical stable time step for both elements and nodes is an approximation and may change during the following time increment.

To maintain simulation stability and prevent divergence, the calculated theoretical stable time step is multiplied by a time step scale factor ΔT_{sca}. If no time step control options are being used, then the minimum nodal or element time step of a model that is printed in the Starter output is multiplied by the time step scale factor with the result shown as `TIME-STEP` in the Engine output file.

Minimum time step listed in the Starter output:

```
      SOLID ELEMENTS TIME STEP
         ------------------------
      TIME STEP       ELEMENT NUMBER
  2.6322377948203E-04      11021
```

Default element time step activated in the Engine file:

```
/DT
0.9 0
```

Engine time step is then

```
TIME-STEP = 0.9 * 2.6322377948203E-04 = 0.2369E-03
```

Which matches the Engine output file cycle zero:

```
CYCLE    TIME      TIME-STEP  ELEMENT
     0          0.000        0.2369E-03     SOLID
```

When using any of the time step control methods, such as `/DT/NODA/CST` or `/DT/BRICK/CST`, the time step control is activated when the minimum time step of the mesh multiplied by the time the time step scale factor is less than the entered minimum time step, $\Delta T_{sca} * \min(\Delta t_{mesh}) \le \Delta T_{\min}$.

The minimum time step listed in the Starter output is:

```
       NODAL TIME STEP (estimation)
         ---------------
      TIME STEP         NODE NUMBER
    6.9475433E-07          10009
```

If the contant nodal time step option is used in the Engine file:

```
/DT/NODA/CST
0.9 7.0E-07
```

The initial Engine time step is:

$$\text{Initial Time Step} = 0.9 * 6.9475433E-07 = 0.6253E-06 \quad (11)$$

Since this initial time step is less than $\Delta T_{\min} = 7.0E-7$, mass is added to increase the theoretical minimum time step of the mesh. Enough mass must be added to increase the minimum mesh time step so that $\Delta T_{sca} * \min(\Delta t_{mesh}) \le \Delta T_{\min}$, which means:

$$\min(\Delta t_{mesh}) = \frac{\Delta T_{\min}}{\Delta T_{sca}} = \frac{7.0E-07}{0.9} = 0.7778E-06 \quad (12)$$

The Engine output shows the time step is the same as entered and there is a mass error (`MAS.ERR`), due to mass added to increase the time step.

```
CYCLE    TIME      TIME-STEP        ELEMENT          …  MAS.ERR
 0      0.000    0.7000E-06     NODE     10009         0.2887E-01
 1   0.7000E-06  0.7000E-06     NODE     10009         0.2887E-01
 2   0.1400E-06  0.7000E-06     NODE     10009         0.2887E-01
```

If $\Delta T_{sca} = 0.67$, then more mass must be added to make the theoretical time step of the mesh is higher.

$$\min(\Delta t_{mesh}) = \frac{\Delta T_{\min}}{\Delta T_{sca}} = \frac{7.0E-07}{0.67} = 1.0448E-06 \quad (13)$$

The default time step scale factor of 0.9 works well in most situations; however, in some cases, other values are recommended. One example would be models with foam materials, where there can be a sudden increase in stiffness, as shown in Figure 15.

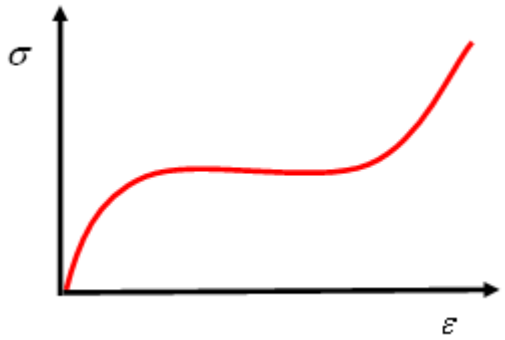


Figure 15: Typical Behavior of Foam

This increase in stiffness causes a decrease in the model's critical time step and can cause divergence, if the simulation time step is larger than the model's critical time step. Some common situations where a time step other than 0.9 should be used are:

1. Models that use advanced mass scaling, `/DT/AMS` to increase the time step: $\Delta T_{sca} = 0.67$
2. Models with foam materials: $\Delta T_{sca} = 0.66$
3. Model with one element: $\Delta T_{sca} = 0.1$
4. Model with two finite elements: $\Delta T_{sca} = 0.2$
5. Model with more than three finite elements: $\Delta T_{sca} = 0.9$
6. Never use a scale factor greater than 1.0

General Recommendations

Every time step control method has advantages and limitations.

Good engineering judgement should be used to evaluate if the method used applies to a simulation. When in doubt, a simulation without time step control can be compared to a simulation with time step control to make sure the results are acceptable.

Structural Dynamic Simulations

Nodal Time Step and Mass Scaling

For non-uniform meshes, the nodal time step method will lead to a slightly higher time step then the elementary method. This can be activated with the default options using:

```
/DT/NODA
```

```
0.9 0
```

Most simulations can benefit from a small amount of mass being added to the nodes with the lowest time step. The following option will activate the nodal time step with mass added to meet the minimum time step entered:

```
/DT/NODA/CST
0.9 ΔT_min
```

The process of finding a specific time step that adds a reasonable amount of mass can be done by running a model a short amount a time without any mass scaling to find the time step of the simulation. This is also a good time to see how long Radioss estimates the simulation will take to run. Next, increase the entered minimum time step and start running the model again to note the amount of mass added. Continue this process until a reasonable amount is added at the beginning of the simulation. Depending on the simulation, more mass may be added later, which would require another modification of the entered minimum time step.

Good engineering judgement must be used to determine how much mass is an acceptable amount to be added to a model to reduce the runtime. Adding too much mass can affect the physics of a drop or impact simulation. This is because the object being simulated weighs more than the real part. In general, it is recommended to keep the amount of mass added to be less than 5%, but more may be acceptable depending on a particular simulation.

Contact Interface Controlling the Simulation

When a contact has the minimum time step in a simulation, the Engine output lists `INTER` and the contact interface number that has the minimum time step. This indicates the time step was reduced to prevent a secondary node from passing through a main segment in one cycle. A contact can occasionally have the minimum time step during a simulation.

However, if the contact controls the time step for an extended amount of time, then the following possible causes should be investigated.

- Initial intersections in the model
- Incorrect penetrations in the model
- Incorrect material definition, which leads to too soft of a contact stiffness
- Small contact thickness or gap

Another possible solution is to use `/DT/INTER/DEL` to remove from the contact the secondary nodes that are causing the time step to drop. This is typically done by setting `/DT/INTER/DEL` $\Delta T_{\min}$ to 10 to 100 times less than the $\Delta T_{\min}$ used in `/DT/NODA/CST`.

Switch to Small Strain Formulation

If the default large strain formulation is used, large solid element deformation can give a cause a decrease in the time step. To prevent this from happening, the `/DT/BRICK/CST` can be used to switch to a small strain formulation at the entered minimum time step for elements whose property uses I_{smstr}= 2 or 12.

Example: /DT Engine Input

For a typical simulation, using a combination of the following time step control commands can help maintain a reasonable time step in the simulation.

```
/DT/NODA/CST
0.9 ΔT_min1
/DT/BRICK/CST
0.9 ΔT_min2
/DT/INTER/DEL
0.9 ΔT_min3
```

With ΔT_{min2} being two to four times smaller than $\Delta T_{\min 1}$ and ΔT_{min3} being ten to a hundrend times smaller than $\Delta T_{\min 1}$.

Quasi-static Simulations

For quasi-static simulations either traditional nodal mass scaling, `/DT/NODA/CST`, or Advanced Mass Scaling, `/DT/AMS` could be used. If the event is slow, then using larger amounts of nodal mass scaling often will not affect the results. Or using `/DT/AMS` per recommendations works well.

2. Courant, Richard, Kurt Friedrichs, and Hans Lewy. "*On the partial difference equations of mathematical physics.*" Mathematische Annalen 100 (1928): 32-74

Finite Elements

Based on mechanics of continuum, a three dimensional solid can be classified in a given category. For a solid with dimensions R, L and t, with t being the thickness, L is the working direction and R is the curvature radius, write:

- If L/t > 20 and R/t > 20: Thin shell assumptions are valid (Kirchhoff or Mindlin shell models)
- If 10 < L/t < 20 and 10 < R/t < 20: Moderately thick shell (Mindlin model)
- If 4 < L/t < 10 and 4 < R/t < 1: Thick shell
- If L/t < 4 or R/t < 4: Solids

An adapted finite element mesh is then applied to each case:

- Shell theory: 2D mesh of the mid-surface
- Thick shell: 3D mesh with constant normal stress assumption
- Solids: 3D mesh without constant normal stress assumption

In Radioss all of these formulations are available. For a given mesh, several kinds of formulations may be available based on:

- The number of integration points
- The stress updating method
- Large or small strain assumption
- Plasticity options for necking and material law integration
- Type of element and hourglass formulation

Solid Elements (/PROP/SOLID)

Solids hexahedron and tetrahedron with linear and quadratic interpolation functions are available in Radioss.

Linear elements are better in terms of time and memory consumption, especially due to the low number of integration points and a larger time step ($\Delta t = \frac{L}{c}$):

- TETRA4: $L = a\sqrt{\frac{2}{3}} = 0.816a$
- TETRA10: $L = a\frac{\sqrt{5/2}}{6} = 0.264a$
- BRICK8: $L = \frac{Volume}{\text{largest face area}}$
- BRICK20: $L \approx \frac{thickness}{2}$

Table 5 summarizes the differences between theses elements. For `BRICK8`, the use of co-rotational formulation is explained in Element Library the *Radioss Theory Manual*; which can avoid the accumulation of an error, due to the updating process, especially when elements undergo large shear deformation. The formulation is not used by default for this element and should be activated by you.

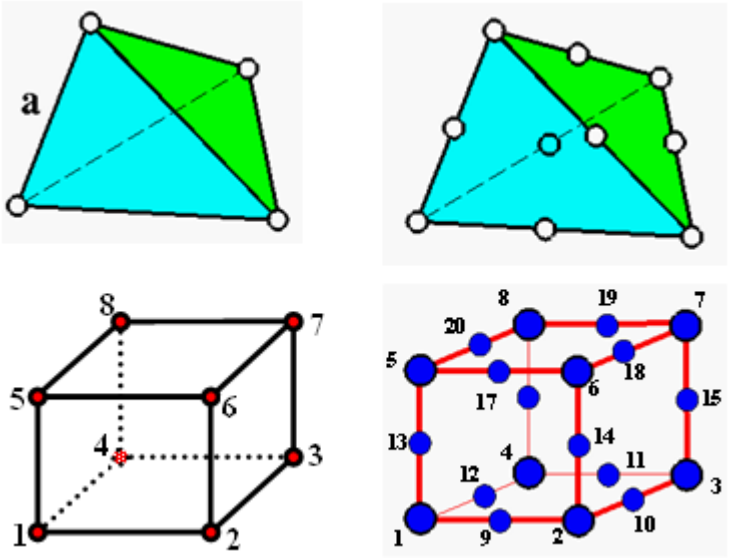


Figure 16: Solid Finite Element Meshes in Radioss

Solid elements can be degenerated to overpass some meshing problems. Degenerated elements can be obtained by merging nodes on a same edge (hexahedron) or suppressing a middle node in a `TETRA10`. The use of degenerated elements is not recommended, but if they cannot be avoided due to complex geometry, it is important to respect the element symmetry to keep a homogenous mass distribution. Some examples of degenerated solid elements are shown in Figure 17.

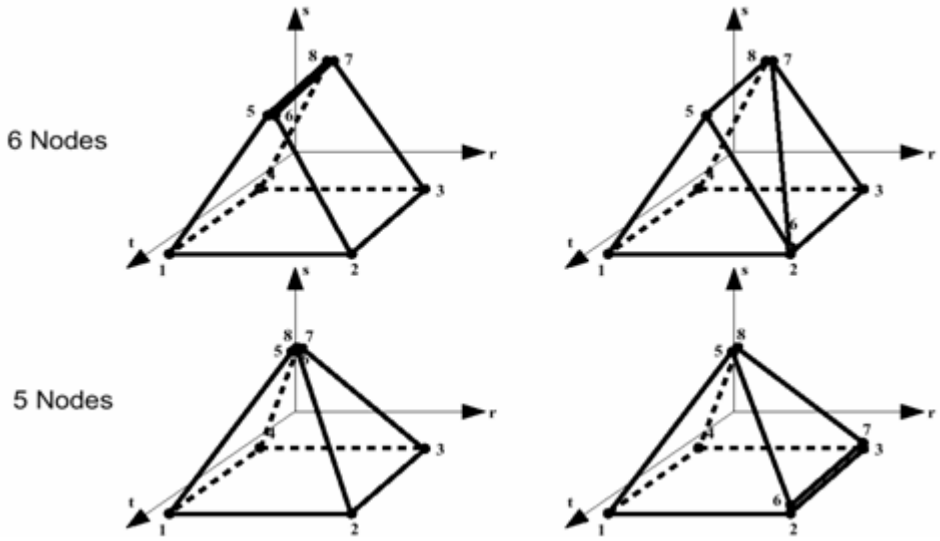
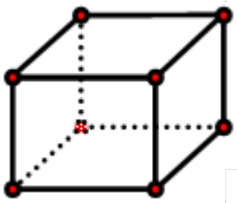
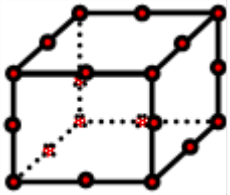
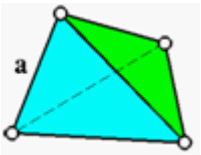
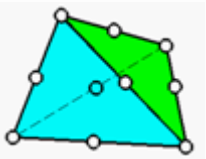


Figure 17: Solid Elements Degeneration Examples

Table 5: Solid Elements

Mesh	Element Name	Number of Integration Points	Hourglass Formulation	Comments
	BRICK8	1x1x1 2x2x2	Penalty for the case of 1 IP	Use co-rotational formulation. Avoid hourglass TYPE 2
	HA8	from 2x2x2 to 9x9x9	---	Note the value used for the flag, I_{cpre}
	HEPH	1x1x1	Physical stabilization	
	BRICK20	2x2x2 or 3x3x3	---	Too high cost element
	TETRA4	1	---	Shear locking in large deformation Low precision element
	TETRA10	4	---	High cost Good precision without shear locking

Solid-Shell Elements (/PROP/TSHELL)

The elements HA8, HEPH and BRICK20 can be transformed to solid-shell elements by setting constant the normal stress through the thickness.

This assumption allows obtaining solid elements suitable for thick-shell modeling, where only one element through a thickness can be used. In additional to these elements, SHELL16, PA6 and other solid-shell elements are available.

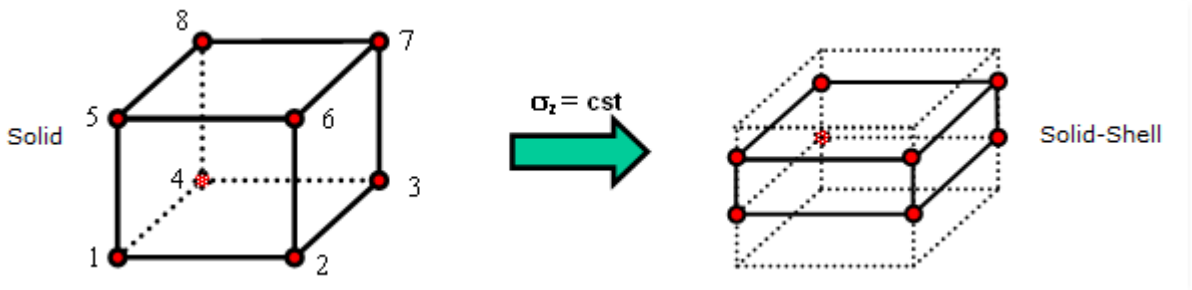
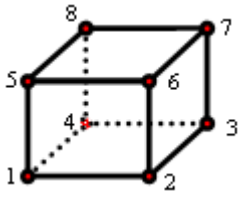
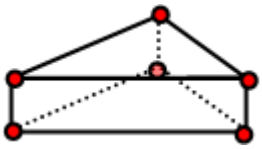
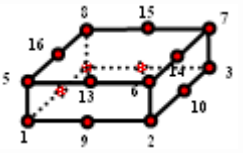


Figure 18: Solid versus Solid-Shell

Table 6: Solid-Shell Elements

Mesh	Element Name	Number of Integration Points	Hourglass Formulation
	HSEPH	Inplane: 1 In thickness: up to 9	Physical stabilization
	HA8	From 2x2x2 to 9x9x9	---
	PA6	Inplane: 1 In thickness: up to 9	---
	SHELL16	Inplane: 3x3 or 2x2 In thickness: up to 9	---

Solid-shell elements, especially SHELL16 can be useful to model thick shell structures with massive junctions (Figure 19). The time step proportion to the shortest distance between two nodes can be small depending on the thickness and material characteristics.

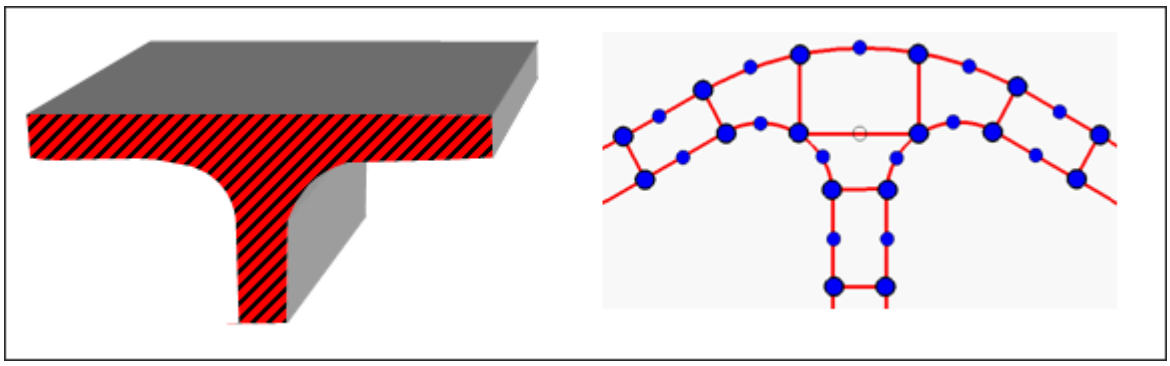


Figure 19: Application of Solid-Shell Elements

Shell Elements

Shell Elements (/PROP/SHELL)

A shell is defined by a curved mid-surface and a thickness h, which is supposed to be very small compared to the two other dimensions. A shell element is the most common element; a full car crash model is made of at least 90% shell elements.

They include membrane, bending and shear deformation and the normal stress, through the thickness, is always zero.

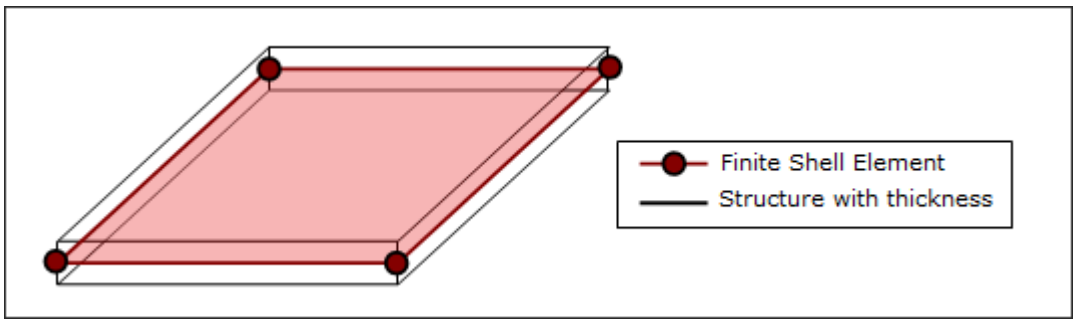


Figure 20: Shell Element

In Radioss, the standard shell elements are Mindlin plate elements with transverse shear deformation. These elements are more accurate than the standard Kirchhoff element; which does not take into account deformation; due to transverse shear, a plane orthogonal to the mid-plane remains orthogonal during the deformation.

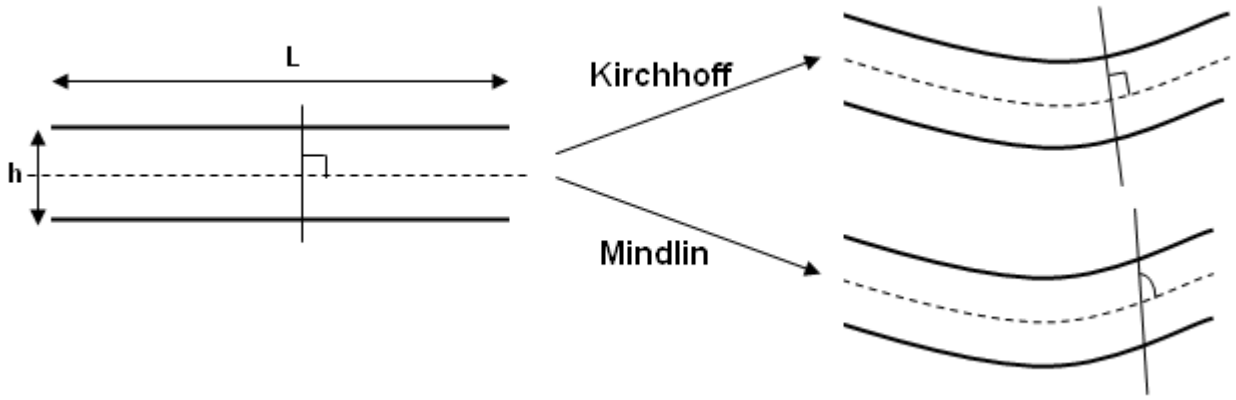
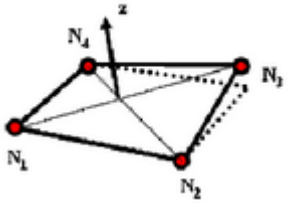


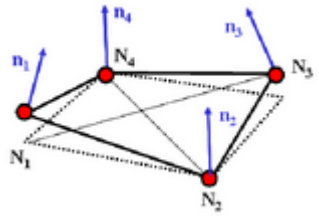
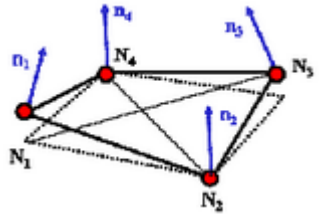
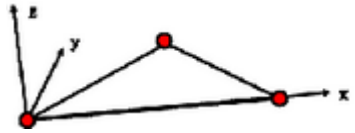
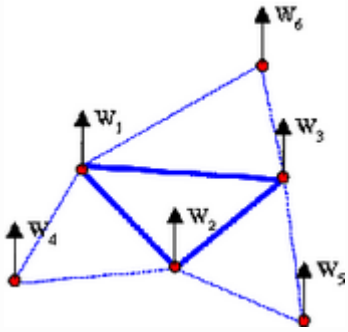
Figure 21: Shell Models

Even if the Kirchhoff model is less accurate, if the ratio L/h is greater than 20, the statements made by Kirchhoff are correct; but if the ratio L/h is between 10 and 20, the assumption stating that a plane orthogonal to the mid-plane remains orthogonal during the deformation is no longer valid, and the Mindlin plate model must be considered in which transverse shear strain is taken into account.
In Radioss, under-integrated shell elements (both 4-node and 3-node shell) are based on Mindlin assumptions. There is no specific formulation that enables to offset the mid-plane of the element away from the nodes; therefore, it is very important to discretize structures with thin walls on the mid-surface.

Shell element formulation plays a considerable role in crash worthiness simulations. Radioss shell elements are mainly based on Mindlin-Reissner shell theory. The formulation is then valid for moderately thick shells and for thin shells. The under-integrated elements are the most popular because of their low cost.

Table 7: Shell Element Library

Mesh	Element Name	Number of Integration Points	Hourglass Formulation	Comments
	*BT* (Classic Q4)	1	Four types based on penalty method	Constant normal vector Hourglass formulations types 3 and 4 are much better than type 1 - used by default

Mesh	Element Name	Number of Integration Points	Hourglass Formulation	Comments
	*QEPH*	1	Physical stabilization	Normal vectors at nodes No hourglass energy in output
	*QBAT*	2x2	Fully-integrated element	Normal vectors at nodes No hourglass energy
	C0	1	---	Flat facet element No hourglass energy
	*DKT18*	3	---	Kirchhoff shell (thin shell only) Higher t/L, lower the time step
	*S3N6*	1	---	Kirchhoff shell (thin shell only) No rotational DOF Rotation of the sides is determined with the help of vertical displacements in neighbors

The choice of a shell element for a given application needs experiences. The following general guidelines may be helpful:

- The `BT` element is simple, efficient and has a low cost. As it is under-integrated, the element is not very sensitive to the mesh quality and can be used for the case of coarse mesh.
- For cases of Quasi-Static Analysis, fine mesh, warped surface and buckling, it is advisable to use `QEPH` or `QBAT` elements.

- QBAT is the most accurate element in Radioss. However, as it is fully-integrated, it costs two to three times more than a *BT* element.
- QEPH is the best compromise between cost and quality. Generally, it costs no more than 15% of a *BT* element and the results obtained by this element are close to those of QBAT.
- Triangles are not recommended. A *C0* element is too stiff and DKT18 comes with a high cost. The number of total triangles in a mesh is generally limited to 5% to ensure a good quality.
- The S3N6 element has as good a behavior in bending as DKT18. It can also be used for some special applications, such as stamping simulations.

The relation between quality of results and mesh is schematically presented in Figure 22.

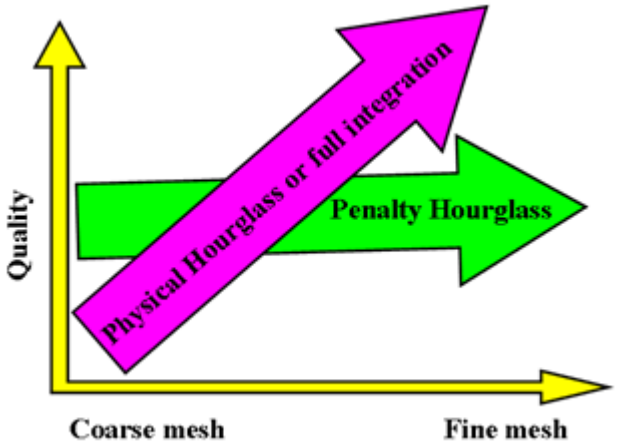


Figure 22: Use of Shell Formulations for Different Meshes

Integration through Thickness

In an elastic shell, the normal stress variation through the thickness is linear; therefore, the internal energy can be obtained by analytical integration. In case of plasticity, the stress distribution becomes nonlinear and a minimum of three integration points are required to take the nonlinearity into account. The nonlinear distribution of stress can be approached by measuring values at some added integration points. The quality of internal energy estimation depends on the number of integration points and the cost. A good compromise between cost and quality can be found, taking into account, material nonlinearity, thickness and bending rate. The number of integration points can be increased up to 10 in Radioss V5x. Using five integration points will provide better results; especially if the thickness is greater than 2mm, but the increase in CPU time is not negligible. To emulate a membrane element with no bending or transverse shear, it is enough to use only one integration point. For an elastic material (LAW1) due to the analytical computation, the option is ignored.

One way to get a membrane element with an elastic material is to use an elasto-plastic law (Johnson-Cook law) with a very high yield stress (example: 1e+30).

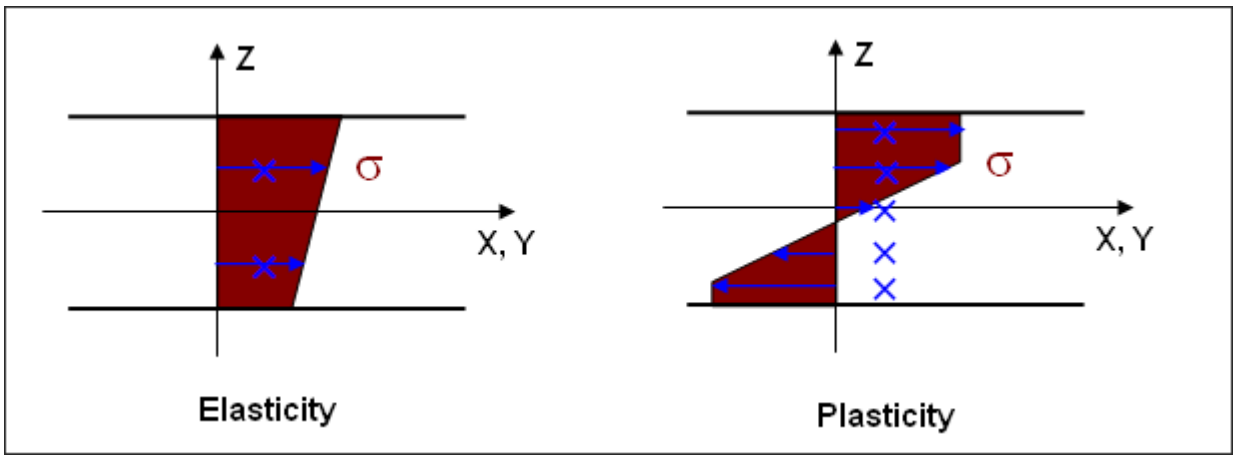


Figure 23: Normal Stress Distribution in a Section a Shell

One way to get accurate results with a low CPU cost is to use the global integration. It consists in the transformation of von Mises plasticity criteria, in a so-called Iluyshin criteria, where the stress components at integration points are replaced by internal forces (N, M, T, etc.).

For multi-layered shell element, one integration point is assigned at the center of each layer.

Note: For a composite made of only two layers, it is necessary to add extra layers (with respect of the overall thickness) in order to get a proper number of integration points.

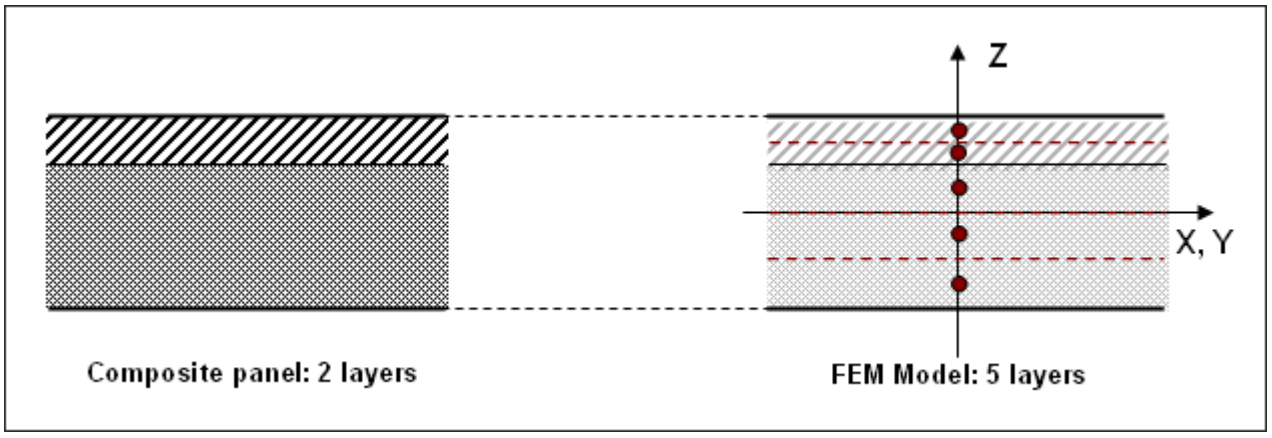


Figure 24: Layers Definition for Composite to Set the Number of Integration Points

Iterative Plastic Projection

In plasticity computation, two fundamental assumptions must be satisfied. First, the stress in the plastic region must verify the plasticity criteria (for example, von Mises criteria). Second, in the principal stresses space is the direction ($\Delta\sigma$), due to work hardening is normal to the yield surface.

By default, a radial return is applied on the stress tensor during the projection on yield surface. This method is better, but the plasticity is approximate, and the second assumption is not always satisfied (Figure 25). An iterative orthogonal projection is also available, this method enables to satisfy the second assumption using an iterative algorithm (see Shell Elements in the *Radioss Theory Manual*).

Use iterative projection if a high accuracy on stress distribution after projection is required. This is the case if residual stress tensor or spring back is required. The error in radial return is higher when the difference between principal stress components is large (pure shear case, for example).

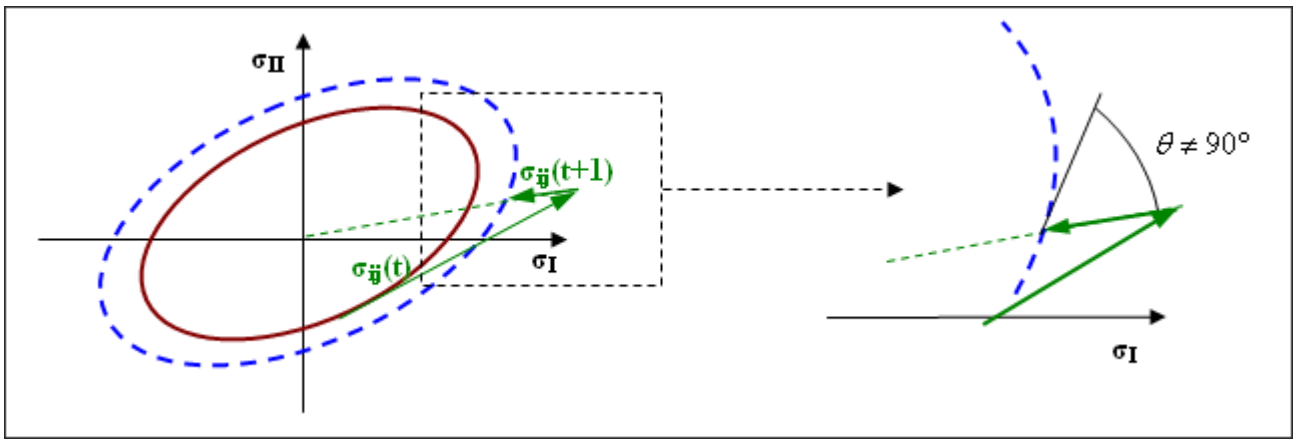


Figure 25: Radial Return

Thickness Variation

By default, shell thickness is supposed constant during shell deformation. Initial thickness is used to compute strains and to integrate stresses, but the thickness variation is still computed for post-processing reasons. If a variable thickness (I_{thick} =1) is used, true thickness is computed not only for post-processing, but also for strain computation and stresses integration.

Comments

1. For accurate results, especially when necking or spring back, it is strongly recommended to use iterative plastic projection and thickness variation.

Element Option Guidelines

The following table provides some general recommendations in the choice of element formulation options.

Applications	Material	Property	Hourglass	Number of Integration Points	Thickness	Plasticity
Basic crash	2	1	1	0 (global)	Constant	Radial
Crash with trapezoidal wrapped shells and with global rotation	2	1	3(c) or QEPH	0 (global)	Constant	Radial

Applications	Material	Property	Hourglass	Number of Integration Points	Thickness	Plasticity
Crash with spring back medium accuracy	2/36	1	1	3	Constant	Iteration
Crash with spring back high accuracy	2/36	1	3(c) or QEPH	3	Variable	Iteration
Crash with material failure ductile failure	2/36 (a)	1	1	5	Variable	Iteration
High quality crash	2/36 (a)	1	3(c) or QEPH	5	Variable	Iteration
Crash with material failure brittle failure	27	11	1	3/5	Variable	Iteration
Windshield	27	11	1	3+1+3(b)	Variable	Iteration
Membrane or Fabric	1/2/19/36	1	1	1	cst/var	rad/iter
Composite	25	9/10/11	1	1 to 30	not used	not used
Model with local hourglass excitation	2/...	1	3(c) or QEPH	0/3/5		
Model with low plasticity and low velocities	2/...	1	3(c) or QEPH	3/5		

1. With variable thickness and iterative plasticity it is possible to model necking failure. The material hardening must be accurate.

2. For glass, plastic, and glass windshield (3 glass layers, 1 plastic layer and 3 glass layers). Less accuracy, 2+1+2 can also be used. For more complex glass plastic windshields, more layers can be used.
3. If elastoplastic hourglass (3) is used, it is recommended to use 0.1 for *hm* and *hf* and default for *hr*.

Beam Elements (/PROP/BEAM, /PROP/INT_BEAM)

The two beam elements available in Radioss are used on one-dimensional structures and frames. It carries axial loads, shear forces, bending and torsion moments (contrary to the truss that supports only axial loads).

Classical Beam (/PROP/BEAM)

The default formulation is based on the Timoshenko formulation; therefore, transverse shear strain is taken into account. This formulation can degenerate into the standard Euler-Bernoulli formulation, where transverse shear energy is neglected.

Nodes 1 and 2 are used to define the local x-axis. Local y-axis is normal to the x-axis and is in the plane defined by node 1, 2 and 3 at time t=0. Then its position is corrected at each cycle, taking into account the mean x-rotation. Local z-axis is obtained using the right-hand rule.

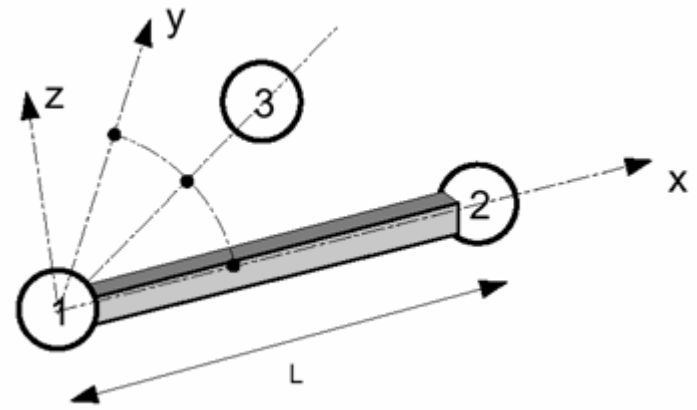


Figure 26: Beam Element and Local System

In Radioss, the beam geometry is defined by its cross-section area and by its three cross-section area moments of inertia. The area moments of inertia around local Y-axis and Z-axis are for bending and they can be calculated using:

$$I_y = \iint_A z^2 dydz \tag{14}$$

$$I_z = \iint_A y^2 dydz \tag{15}$$

The area moment of inertia regarding the local X-axis is for torsion. It can simply be obtained by the summation of I_y and I_z. The torsion model is only valid for full cross-section where the warping is neglected.

The minimum time step for a beam element is determined using:

$$\Delta t = \frac{aL}{c} \tag{16}$$

Where,

c Speed of sound

$$\sqrt{E/\rho}$$

$$a = \frac{1}{2}\min\left(\sqrt{\min\left(4,\ 1+\frac{b}{12}\right)} \cdot F_1, \sqrt{\frac{b}{3}} \cdot F_2\right)$$

$$F_1 = \sqrt{1+2d^2} - \sqrt{2}\,d$$

$$F_2 = \min\left(F_1, \sqrt{1+2d_s{}^2} - \sqrt{2}\,d_s\right)$$

$$b = \frac{AL^2}{\max(I_y, I_z)}$$

$$d = \max(d_m, d_f)$$

$$d_s = d \cdot \max\left(1, \sqrt{\frac{12}{b}} \cdot \sqrt{1+\frac{12E}{\frac{5}{6}Gb}(1-I_{shear})}\right)$$

User input parameters to define beam cross-section are the three area moments of inertia and the area. For accuracy and stability, it is recommended to respect the following limitations:

$$L > \sqrt{A} \tag{17}$$

$$0.01A^2 < I_y < 100A^2$$

$$0.01A^2 < I_z < 100A^2$$

$$0.1(I_y + I_z) < I_x < 10(I_y + I_z)$$

Only material LAW1 and LAW2 are available for this beam element. A global plasticity model in function of internal forces is used in LAW2. The main assumption is that the beam cross-section is full and rectangular. Optimal interaction between section and section inertia are:

- $12I_yI_z = A^4$
- $I_x = I_y + I_z$

This model also provides good results for circular or ellipsoidal cross-section. For thin-walled cross-sections, the global plasticity model may provide incorrect results. It is not recommended to use a single beam element per line of frame structure. The mass is lumped onto the nodes; therefore, to get a correct mass distribution, a fine mesh is required. This is especially true when dynamic effects are important.

Moreover, in Radioss beam element, the moment does not vary along the beam length. The moment is supposed constant and is evaluated at the beam center, as is the stress.

Consequently, a beam will yield at a slightly higher force in the case of a clamped cantilever beam, since the moment is calculated at the center, instead of at the root of the beam.

> **Note:** Output for beam elements are expressed in the local system. Some results may be confusing, due to the fact, that the local system is updated taking into account the mean *X* rotation. For instance, a beam with one node completely blocked, if an axial rotational velocity *V* is imposed to the other node, then the beam will rotate at a speed of *V*, but the local system will rotate at a speed of V/2. This may lead to a bad interpretation of results, especially the shear forces and bending moments.

New Beam (`/PROP/INT_BEAM`)

The cross-section of the element is defined using up to 100 integration points (Figure 27). The element properties of the cross-section, that is area moments of inertia and area, are computed by Radioss as:

$$A = \sum A_i = \sum \left(dy_i dz_i \right) \tag{18}$$

$$I_z = \sum A_i \left(y_i 2 + \frac{1}{12} dy_i 2 \right) \tag{19}$$

$$I_y = \sum A_i \left(z_i 2 + \frac{1}{12} dz_i 2 \right) \tag{20}$$

Beam model is based on the Timoshenko theory and takes into account transverse shear strain without warping in torsion. It can be used for deep beam cases (short beams). The use of several integration points in the section allows to get an elasto-plastic model, in which von Mises criteria is written on each integration point and the section can be partially plastified contrary to the classical beam element (TYPE3). Material LAW36 is also available, as well as LAW1 and LAW2. However, as the element has only one integration point in its length, it is not recommended to use a single beam element per line of frame structure in order to take into account the plasticity progress in length, as well as in depth.

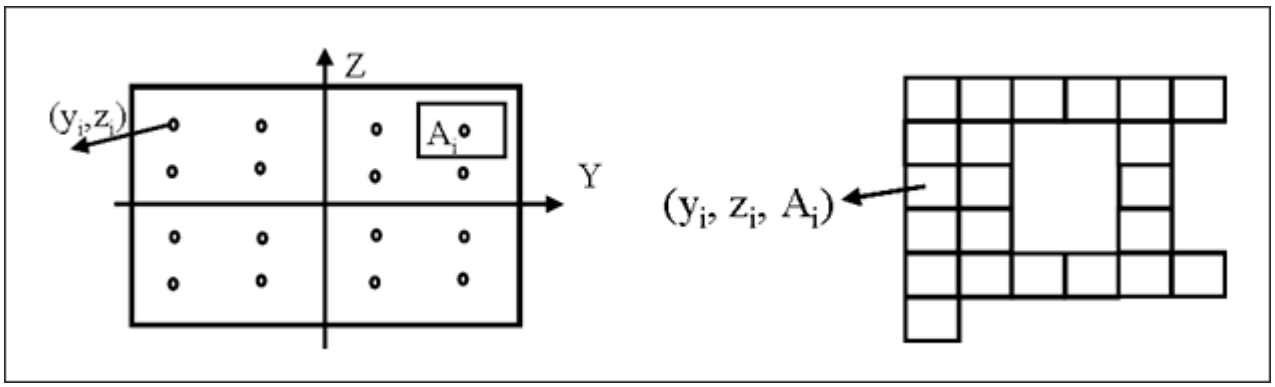


Figure 27: Cross-Section Definitions in the Integrated Beam

Springs

Stiffness Formulation

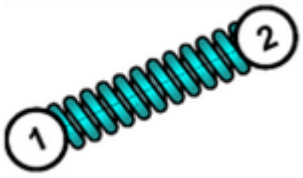
The stiffness of the spring can be defined multiple ways with each degree of freedom defined differently.

Force and Moment

Linear Stiffness and Damping Behavior

The simplest formulation is a linear elastic spring stiffness, where the internal force is proportional to the relative displacement. In this case, only the constant stiffness parameter K_i and optional damping parameter C_i are entered.

For linear stiffness, the force and moment are:



$$\mathrm{F}(\delta) = K_i \delta^i$$

$$\mathrm{M}(\theta) = K_i \theta^i$$

For linear dashpot, the force and moment are:

$$\mathrm{F}(\delta) = C_i \dot{\delta}^i$$

$$\mathrm{M}(\theta) = C_i \dot{\theta}^i$$

For linear stiffness and dashpot, the force and moment are:

$$\mathrm{F}(\delta) = K_i \delta^i + C_i \dot{\delta}^i$$

$$\mathrm{M}(\theta) = K_i \theta^i + C_i \dot{\theta}^i$$

Nonlinear Behavior

The force and moment in a spring is computed as:

$$F_i(\delta^i) = f\left(\frac{\delta^i}{Ascale_i}\right)\left[A_i + B_i \ln\left(\max\left(1, \left|\frac{\dot{\delta}^i}{D_i}\right|\right)\right) + E_i g\left(\frac{\dot{\delta}^i}{F_i}\right)\right] + C_i\dot{\delta}^i + Hscale_i h\left(\frac{\dot{\delta}^i}{F_i}\right) \quad (21)$$

Where, i is the translational degrees of freedom: 1,2,3

$$M_i(\theta^i) = f\left(\frac{\theta^i}{Ascale_i}\right)\left[A_i + B_i \ln\left(\max\left(1, \left|\frac{\dot{\theta}^i}{D_i}\right|\right)\right) + E_i g\left(\frac{\dot{\theta}^i}{F_i}\right)\right] + C_i\dot{\theta}^i + Hscale_i h\left(\frac{\dot{\theta}^i}{F_i}\right) \quad (22)$$

Where, i is the rotational degrees of freedom: 4,5,6

The variables in the force and moment equation represent:

$f\left(\frac{\delta^i}{Ascale_i}\right)$ Spring force versus displacement function input as *fct_ID*$_{1i}$

$f\left(\frac{\theta^i}{Ascale_i}\right)$ Spring force versus rotation function input as *fct_ID*$_{1i}$.

A_i, B_i, D_i, E_i and F_i Scaling coefficients

$\ln\left(\max\left(1, \left|\frac{\dot{\delta}^i}{D_i}\right|\right)\right)$ Logarithmic function that scales the spring stiffness as the velocity increases

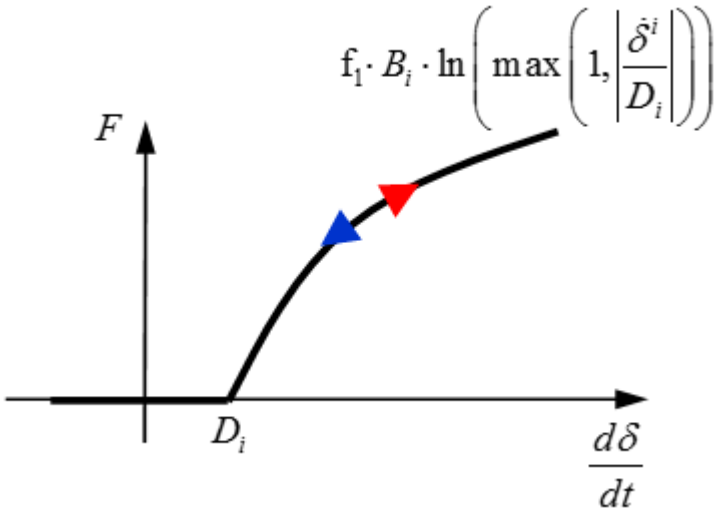


Figure 28: With Parameter Input

$g\left(\frac{\dot{\delta}^i}{F_i}\right)$ Scale the stiffness as a function of linear input as *fct_ID*$_{2i}$

$g\left(\frac{\dot{\theta}^i}{F_i}\right)$ Scale the moment as a function of rotational velocity input as *fct_ID*$_{2i}$

This input can be used to model nonlinear strain rate effects of the spring stiffness.

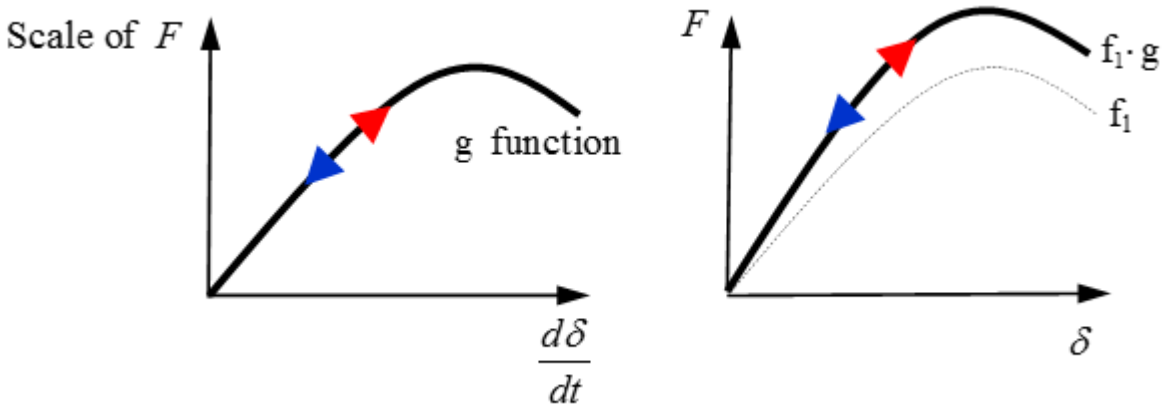


Figure 29: With g Function Input

C_i Linear damping coefficient used to increase the spring stiffness as a function of velocity

$h\left(\frac{\dot{\delta}^i}{F_i}\right)$ or $h\left(\frac{\dot{\theta}^i}{F_i}\right)$ Nonlinear damping function input as *fct_ID*$_{4i}$

Linear or nonlinear damping as a function of velocity can also be added to the spring force using either a linear damping coefficient or a user-defined function.

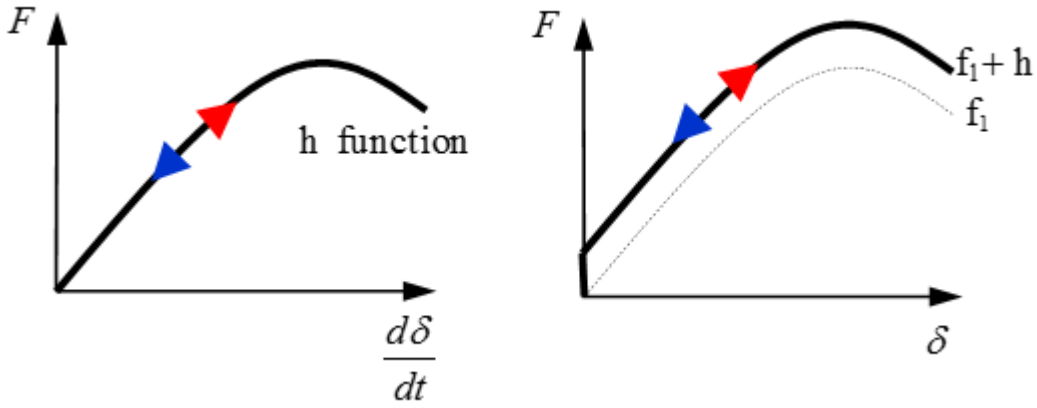


Figure 30: With h *Function Input*

The functions g and h both describe the damping behavior of the spring. However, the g function scales the spring stiffness function f, but the h function adds to the spring stiffness function f.

Time Step

Time step for spring:

$$\Delta t = \sqrt{\frac{M}{K}} \tag{23}$$

Time step for dashpot:

$$\Delta t = \frac{M}{2C} \tag{24}$$

Time step for spring and dashpot:

$$\Delta t = \frac{\sqrt{C^2 + K \cdot M} - C}{K} \tag{25}$$

Where,

M	Mass of the spring
K	Linear stiffness or $\max\left[\frac{dF}{d\delta}\right]$ for a nonlinear spring
C	Linear damping or $\max\left[\frac{\partial g\left(\frac{d\delta}{dt}\right)}{\partial\left(\frac{d\delta}{dt}\right)}\right]$ for nonlinear damping

For nonlinear springs, K is used for time step calculation and contact stiffness. If K is not defined in the spring property, the maximum slope of *fct_ID*$_{1i}$ ($\max\left[\frac{dF}{d\delta}\right]$) will then be automatically used to calculate the timestep. The behavior is the same for damping stiffness C.

Spring Failure

Spring failure in properties TYPE1, TYPE2, TYPE4, TYPE8, TYPE13, and TYPE25 may be considered in two ways.

- Uni-directional failure, or
- Multi-directional failure

This is controlled by the option I_{fail}. If the I_{fail} flag is not set/present in the property, then the default uni-directional failure is considered for the spring. For example, in TYPE4 there is no option I_{fail}, so uni-directional failure is used.

The failure model may consider displacement failure, force failure or internal energy failure. This is controlled by the option I_{fail2}. Similar to I_{fail}, if the I_{fail2} option is not set/present in the property, then displacement (or rotation) failure model is used.

Spring Type	Failure Criteria (Ifail)			Failure Model (Ifail2)			
	Uni-directional Failure	Multi-directional Failure	α^i, β^i in Multi-directional Failure	Displacemen (or Rotation) Criteria	Displacemen (or Rotation) Criteria Consider Velocity Effect	Force (or Moment) Criteria	Internal Energy Criteria
TYPE4	√			√			
TYPE8	√	√	$\alpha^i = 1$ $\beta^i = 2$	√		√	√
TYPE12	√			√			
TYPE13	√	√	arbitrary α^i, β^i (default $\alpha^i = 1,$ $\beta^i = 2$)	√	√	√	√
TYPE25	√	√	arbitrary α^i, β^i (default $\alpha^i = 1,$ $\beta^i = 2$)	√	√	√	√

Failure Criteria

- Uni-directional (I_{fail} = 0)

  If the criteria is uni-directional, the spring will fail as soon as the criteria is satisfied for one degree of freedom:

  $\left|\frac{\delta^i}{\delta^i_{\max}}\right| \geq 1$ or $\left|\frac{\delta^i}{\delta^i_{\min}}\right| \geq 1$ with $\delta^i_{\max}$ and $\delta^i_{\min}$ being the failure limits in direction i =1,2,3.

  $\left|\frac{\theta^i}{\theta^i_{\max}}\right| \geq 1$ or $\left|\frac{\theta^i}{\theta^i_{\min}}\right| \geq 1$ with $\theta^i_{\max}$ and $\theta^i_{\min}$ being the failure limits in direction i =4,5,6.

  Where, i is any degree of freedom. Its property type dependent.

For property TYPE4, there is only i=1, for translational X.

For property TYPE8, there are i=1,2,3,4,5,6 for translational X, Y, Z and rotational X,Y,Z.

For property TYPE13, there are i=1,2,3,4,5,6, but in this case, for tension/compression X, shear XY, shear XZ, torsion, bending Y, bending Z.

Examples of failure behaviors of uni-directional failure are:

If $\delta^1_{\max} = 0.04\text{m}$ in a tension only test, then there is spring failure, and the force goes to zero once the elongation reaches 0.04m. The same is true for rotation, if $\theta^4_{\max} = 0.035\text{rad}$, the spring fails and has zero force at 0.035rad.

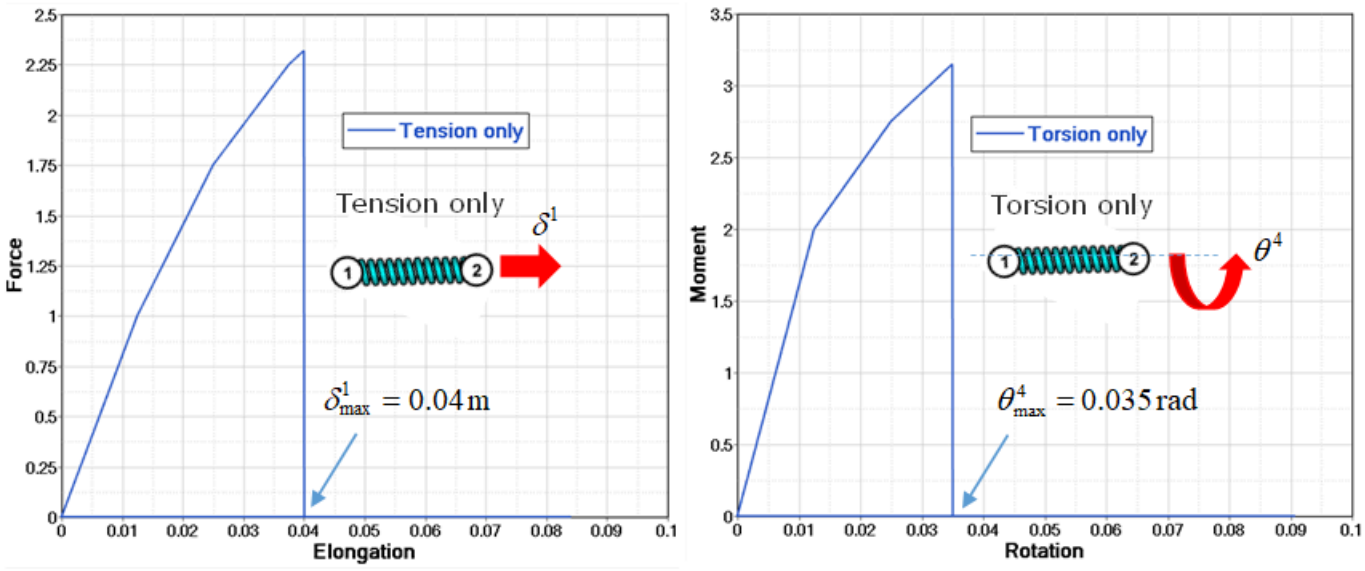


Figure 31:

If a spring is subject to two load cases, for example, tension and torsion and I_{fail} = 0 (uni-directional failure) is in use, then spring failure occurs if either one of the failure criteria is reached.

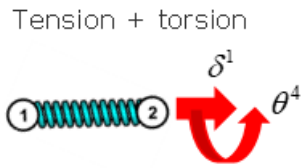


Figure 32:

Here the rotation criteria is reached first (at Time=0.58s), then the force and moment fall to zero at the same time.

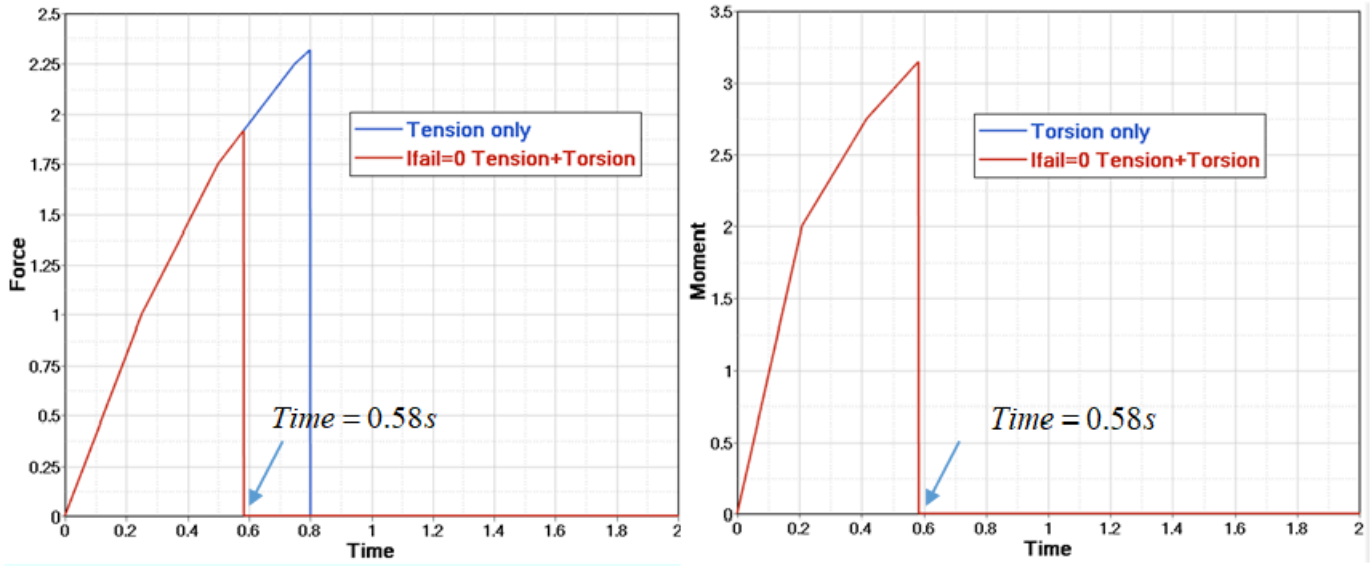


Figure 33:

- Muli-directional (I_{fail} = 1)

  If the criteria is multi-directional, all degrees of freedom are coupled and failure occurs when:

$$\sum_{i=1,2,3} \alpha^i \left(\frac{\delta^i}{\delta^i{}_{fail}} \right)^{\beta^i} + \sum_{i=4,5,6} \alpha^i \left(\frac{\theta^i}{\theta^i{}_{fail}} \right)^{\beta^i} \geq 1 \qquad (26)$$

  Where, $\delta^i{}_{fail}$ and $\theta^i{}_{fail}$ are failure criteria. Refer to Failure Criteria for more details.

  For property TYPE8, $\alpha^i = 1$ and $\beta^i = 2$ (failure criteria shown as blue curve in Figure 34).

  For properties TYPE13 and TYPE25, arbitrary α^i, β^i may be input with $\alpha^i > 0$ (default is $\alpha^i = 1$). Figure 34 shows failure criteria with different β^i.

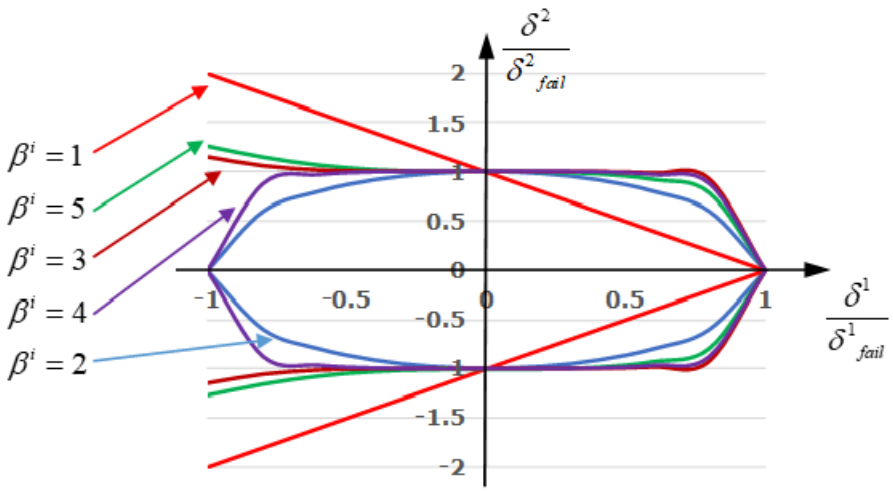


Figure 34:

In a test case of two load cases, tension + torsion and with I_{fail} = 1.

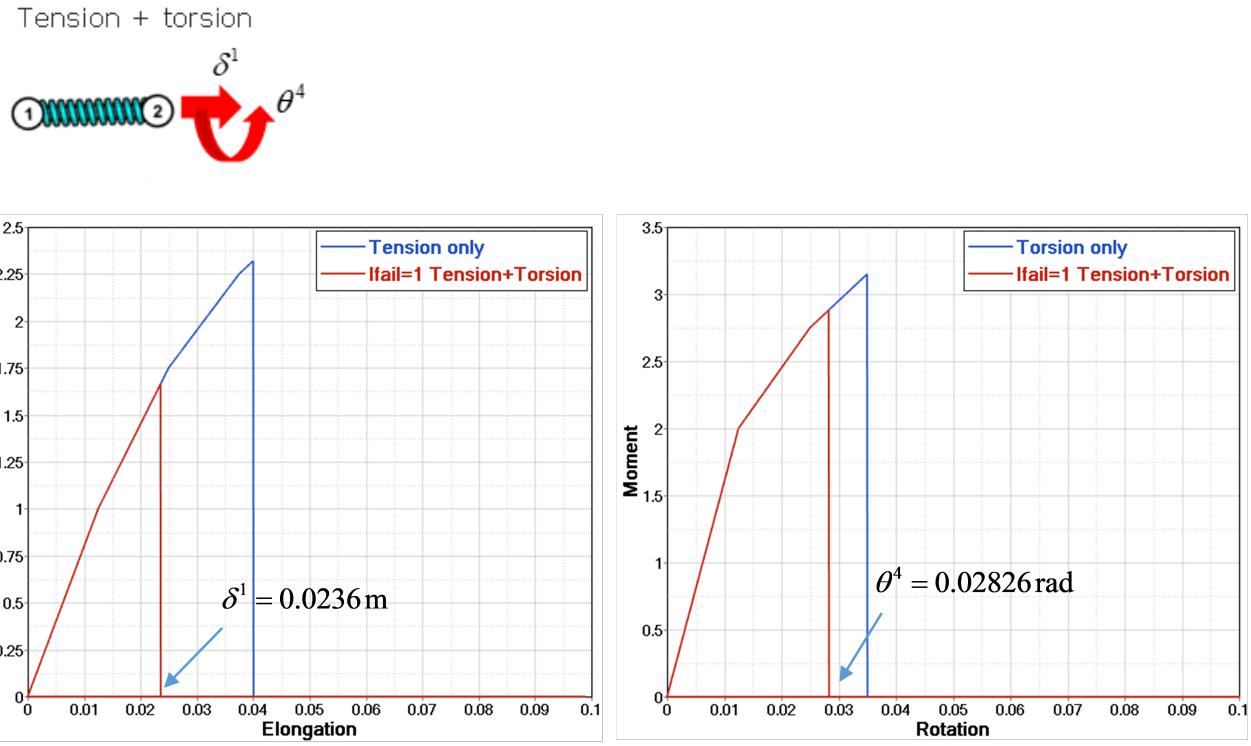


Figure 35:

Comparing the failure value against the limit set in just one direction, it usually is smaller than the defined limit. In this example case, the tension limit is set at 0.04m and torsion limit set at 0.035rad. The spring fails at an elongation 0.0236 < 0.04 and rotation 0.02826 < 0.035. This is

because the failure combination of tension and torsion reaches the failure circle (Figure 36) and; therefore, the spring failed (force and moment fall to zero).

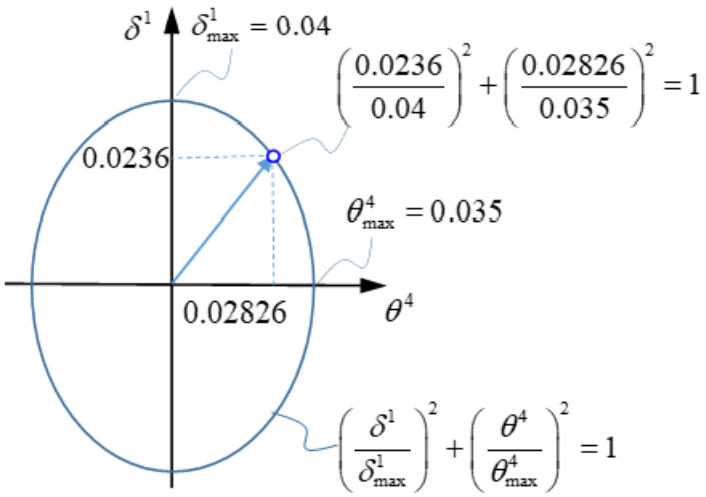


Figure 36:

Failure Model

Option I_{fail2} is available in properties TYPE8, TYPE13, and TYPE25.

- Displacement (or Rotation) Failure Criteria (I_{fail2} = 0)

$$\sum_{i=1,2,3}\left(\frac{\delta^i}{\delta^i{}_{fail}}\right)^2 + \sum_{i=4,5,6}\left(\frac{\theta^i}{\theta^i{}_{fail}}\right)^2 \geq 1 \qquad (27)$$

With,

$$\delta^i{}_{fail} = \begin{cases} \delta^i_{\max}, & if(\delta^i > 0) \\ \delta^i_{\min}, & if(\delta^i \leq 0) \end{cases} \text{ and } \theta^i{}_{fail} = \begin{cases} \theta^i_{\max}, & if(\theta^i > 0) \\ \theta^i_{\min}, & if(\theta^i \leq 0) \end{cases}$$

- Displacement (or Rotation) Failure Criteria considering velocity effect (I_{fail2} = 1)

This failure criteria will allow model velocity dependent failure limits, they are available with displacement, force and internal energy. Therefore, translational $\delta^i{}_{fail}$ and rotational $\theta^i{}_{fail}$ failure are modified to take into account velocity, as:

$$\sum_{i=1,2,3}\left(\frac{\delta^i}{\delta^i{}_{fail}}\right)^2 + \sum_{i=4,5,6}\left(\frac{\theta^i}{\theta^i{}_{fail}}\right)^2 \geq 1 \qquad (28)$$

$$\delta^i{}_{fail} = \begin{cases} \delta^i_{\max} + c_i \cdot \left|\frac{vi}{v_0}\right|^{ni}, & if(\delta^i > 0) \\ \delta^i_{\min} - c_i \cdot \left|\frac{vi}{v_0}\right|^{ni}, & if(\delta^i \leq 0) \end{cases} \text{ and } \theta^i{}_{fail} = \begin{cases} \theta^i_{\max} + c_i \cdot \left|\frac{\omega i}{\omega_0}\right|^{ni}, & if(\theta^i > 0) \\ \theta^i_{\min} - c_i \cdot \left|\frac{\omega i}{\omega_0}\right|^{ni}, & if(\theta^i \leq 0) \end{cases}$$

The parameter c_i is a scale of exponent function, and parameter *ni* effects the failure as in Figure 37.

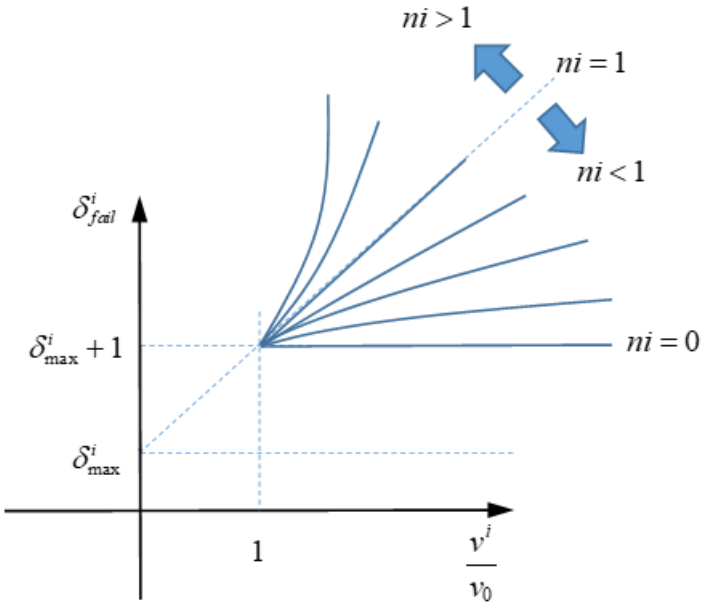


Figure 37:

The above formulas' are valid for displacement/rotation criteria and are also force/moment and energy criteria.

- Force (or Moment) Criteria (I_{fail2} = 2) and Internal Energy Criteria (I_{fail2} = 3)

  Translational $\delta^i{}_{fail}$ and rotational $\theta^i{}_{fail}$ failure are:

$$\sum_{i=1,2,3}\left(\frac{\delta^i}{\delta^i{}_{fail}}\right)^2+\sum_{i=4,5,6}\left(\frac{\theta^i}{\theta^i{}_{fail}}\right)^2\geq 1 \qquad (29)$$

$$\delta^i{}_{fail}=\begin{cases}\delta^i_{\max}+c_i\cdot\left|\frac{vi}{v_0}\right|^{ni}, & if(\delta^i>0)\\ \delta^i_{\min}-c_i\cdot\left|\frac{vi}{v_0}\right|^{ni}, & if(\delta^i\leq 0)\end{cases} \text{ and } \theta^i{}_{fail}=\begin{cases}\theta^i_{\max}+c_i\cdot\left|\frac{\omega i}{\omega_0}\right|^{ni}, & if(\theta^i>0)\\ \theta^i_{\min}-c_i\cdot\left|\frac{\omega i}{\omega_0}\right|^{ni}, & if(\theta^i\leq 0)\end{cases}$$

  The above formulas' are valid for displacement/rotation, force/moment and energy.

  Here $\delta^i{}_{\max}$, $\delta^i{}_{\min}$ ($\theta^i{}_{\max}$, $\theta^i{}_{\min}$) are not displacement (rotational angle) criterion, but maximum or minimum force(moment) for I_{fail2} = 2 and internal energy for I_{fail2} = 3.

  The influence of velocity is also taken into account and the relative velocity coefficient c_i is relative to force/moment (I_{fail2} = 2) or to internal energy (I_{fail2} = 3).

Spring Hardening

Isotropic, kinematic or uncoupled spring hardening options can be defined by the hardening flag H.

These examples only include the spring stiffness without any damping.

Linear Elastic Spring, H=0

A linear spring can be modeled by inputting only the linear stiffness as K_i and *fct_ID*$_{1i}$ =*fct_ID*$_{4i}$ =0. For linear spring, H is always 0.

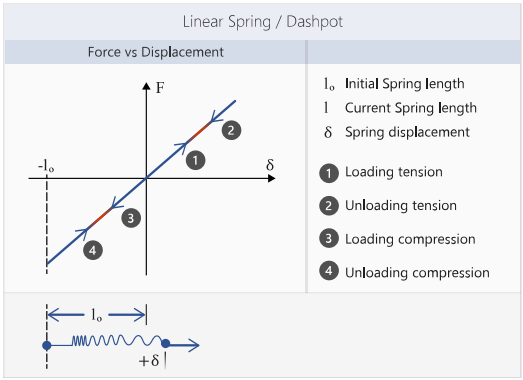


Figure 38: Linear elastic spring with H=0

Nonlinear Elastic Spring, H=0

A nonlinear elastic spring is modeled by defining a force versus displacement curve where f_1 in Figure 39 is defined in *fct_ID*$_{1i}$. Since the model is elastic, the loading and unloading follow the same path.

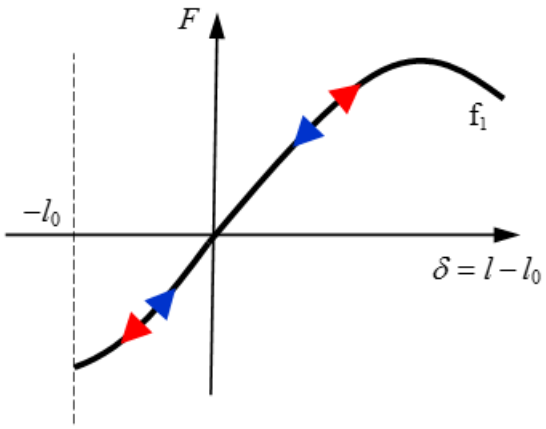


Figure 39: Nonlinear elastic spring with H=0

Nonlinear Elastic Plastic Spring with Isotropic Hardening, H=1

Figure 40 shows the behavior of a nonlinear elastic plastic spring with isotropic hardening where f_1 is defined in *fct_ID*$_{1i}$ and unloading stiffness K_u is input using K_i.

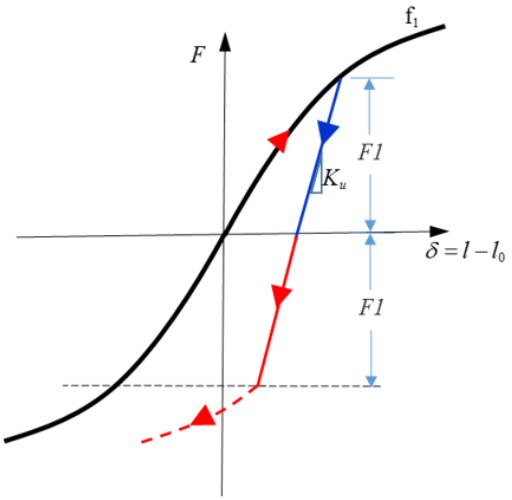


Figure 40: Isotropic hardening with H*=1*

To demonstrate istropic hardening, H=1, Figure 41 shows a spring loaded in tension and then unloads using the linear unloading stiffness, K_u. The unloading stiffness continues to be used in compressive loading until the loading force in compression matches the maximum loading force in tension. From this point, any additional compressive loading uses the input loading function.

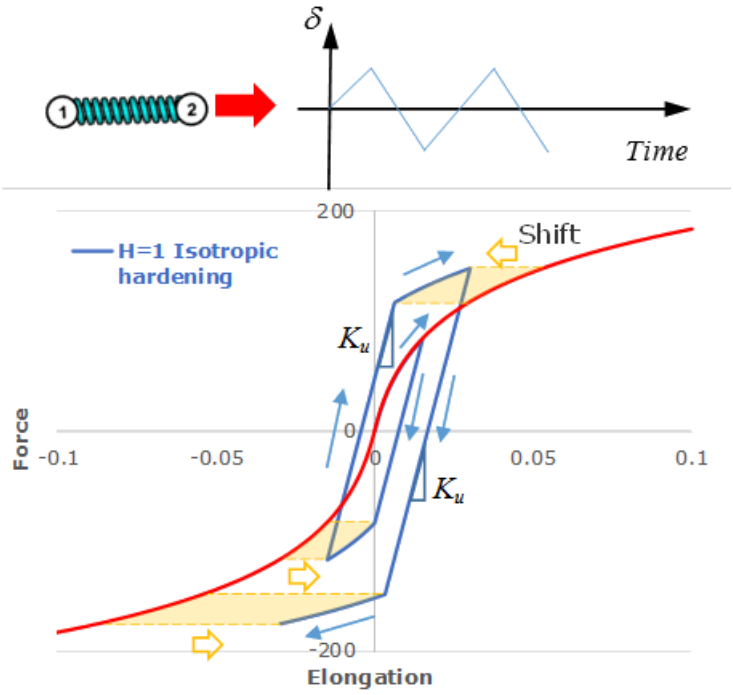


Figure 41: Cyclic loading applied on a spring

with H=*1*

Nonlinear Elastic Plastic Spring with Uncoupled Hardening, H=2

The force versus displacement curve f_1 in Figure 42 is defined in *fct_ID*$_{1i}$ and unloading stiffness K_u is input using K_i. When uncoupled harding H=2, is used, the tensile and compression behavior are uncoupled. Thus, once the unloading reaches zero force, there is no stiffness until zero displacement and then the compressive loading follows the force displacement curve.

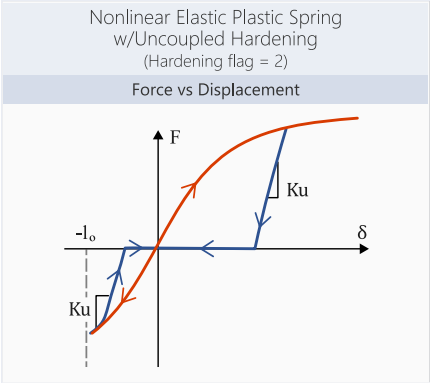


Figure 42: Isotropic hardening
with H=*2*

Nonlinear Elastic Plastic Spring with Kinematic Hardening, H=4

When H=4 is used, the loading function *fct_ID*$_{1i}$ and unloading *fct_ID*$_{3i}$ are mandatory and shown in Figure 43as f_1 and f_3. The loading curve should be positive for all values of abscissa. The unloading curve in this case should be negative for all values of abscissa. These curves represents upper and lower limits of yield force as function of current spring length variation or strain. The force follows K between function f_1 and f_3 and is input as K_i.

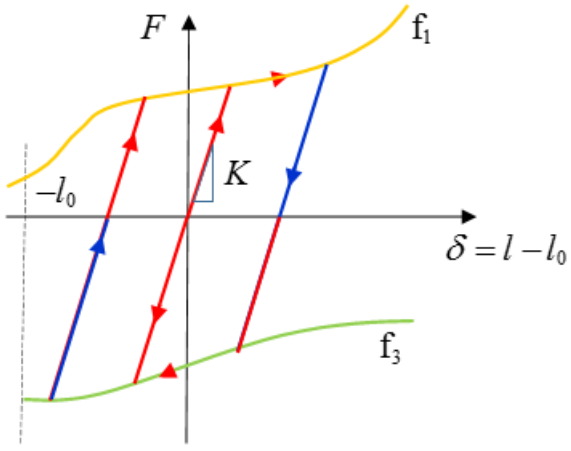


Figure 43: Kinematic Hardening
with H=*4*

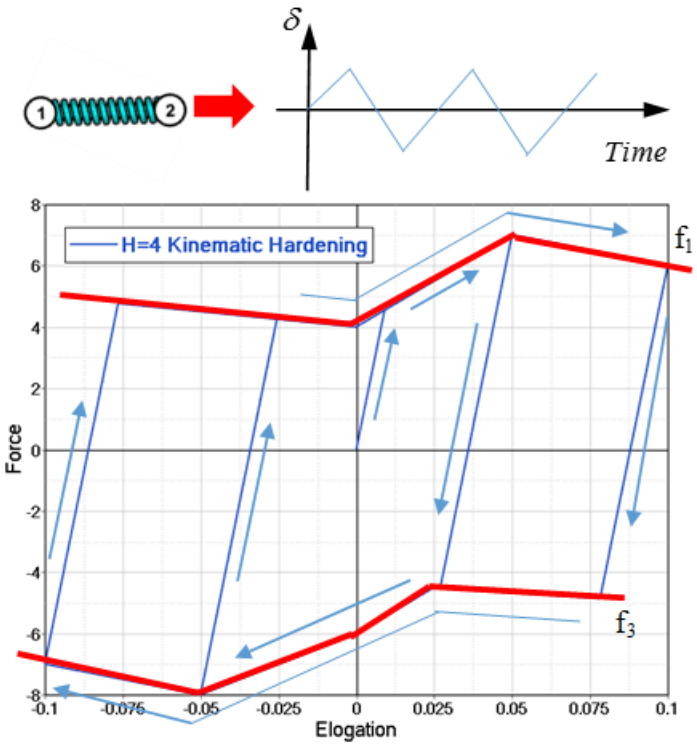


Figure 44: Cyclic loading applied on a spring with kinematic hardening H*=4*

If the minimum and maximum yield curves (f_1 and f_3) have identical shapes, the hardening is considered to be kinematic.

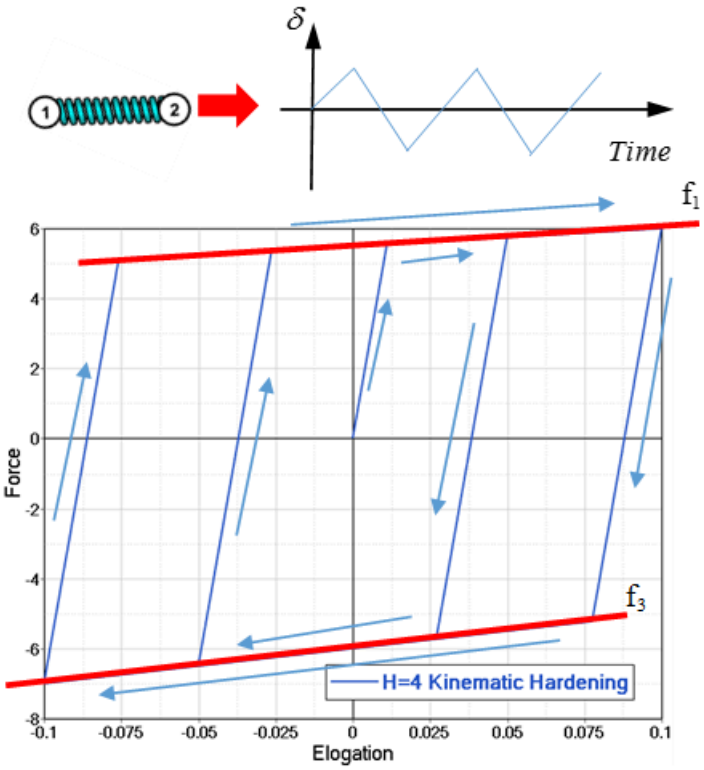


Figure 45: H=4, with the minimum and maximum yield curves (f_1 and f_3) input with identical shapes

Nonlinear Elastic Plastic Spring Nonlinear Unloading, H=5

When H=5, uncoupled hardening in compression and tensile with nonlinear unloading is modeled.

Function f_3 defines the residual displacement δ_{resid} related to displacement; where the unloading starts at δ_{peak}. The unloading is defined by:

$$F(K, f_3) = \alpha(\delta - \delta_{resid})^n \qquad (30)$$

with, $\delta_{resid} = f_3(\delta_{peak})$

Where, α and n being computed using K and $f_3(\delta_{peak})$. The loading function f_1 in Figure 46 is defined in *fct_ID*$_{1i}$ and residual deformation function f_3 input as *fct_ID*$_{3i}$.

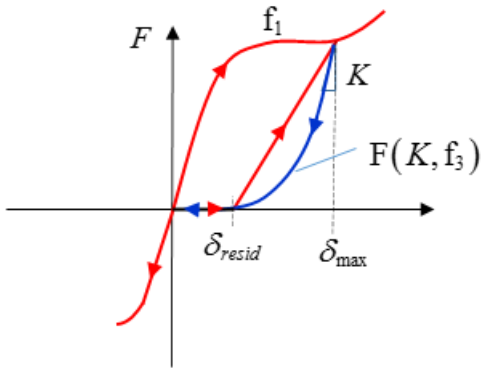


Figure 46: Nonlinear unloading with H=5

In Figure 47, a linear curve is defined for δ_{resid} and δ_{peak} in function f_3. δ_{resid} is 0.5 times δ_{peak}. In cycle loading, the first unloading started at $\delta_{peak1} = 0.05$ and then $\delta_{resid} = 0.5 \times 0.05 = 0.025$. The second unloading started at $\delta_{peak2} = 0.1$ and then $\delta_{resid} = 0.5 \times 0.1 = 0.05$.

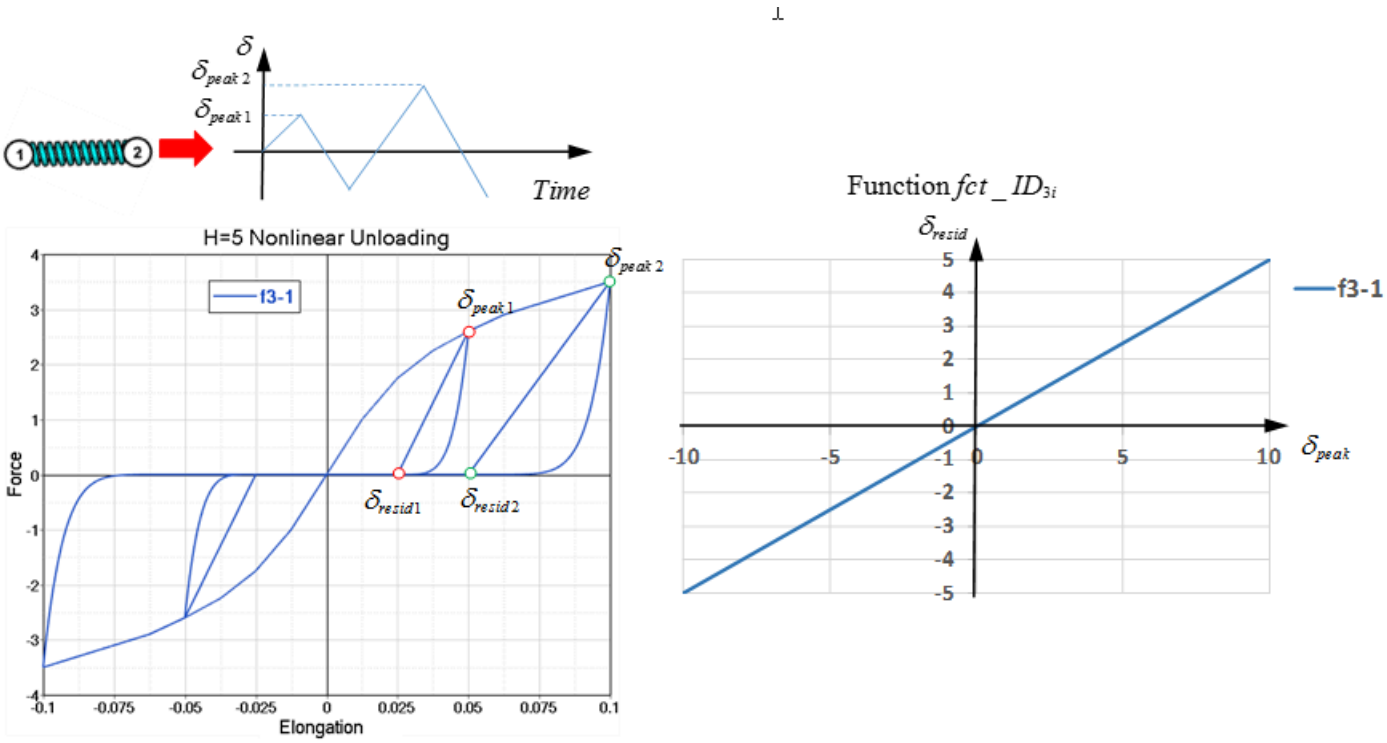


Figure 47: Linear residual versus maximum displacement curve with H=5

Figure 48 shows how increasing the slope of the residual versus maximum displacement curve changes the spring behavior.

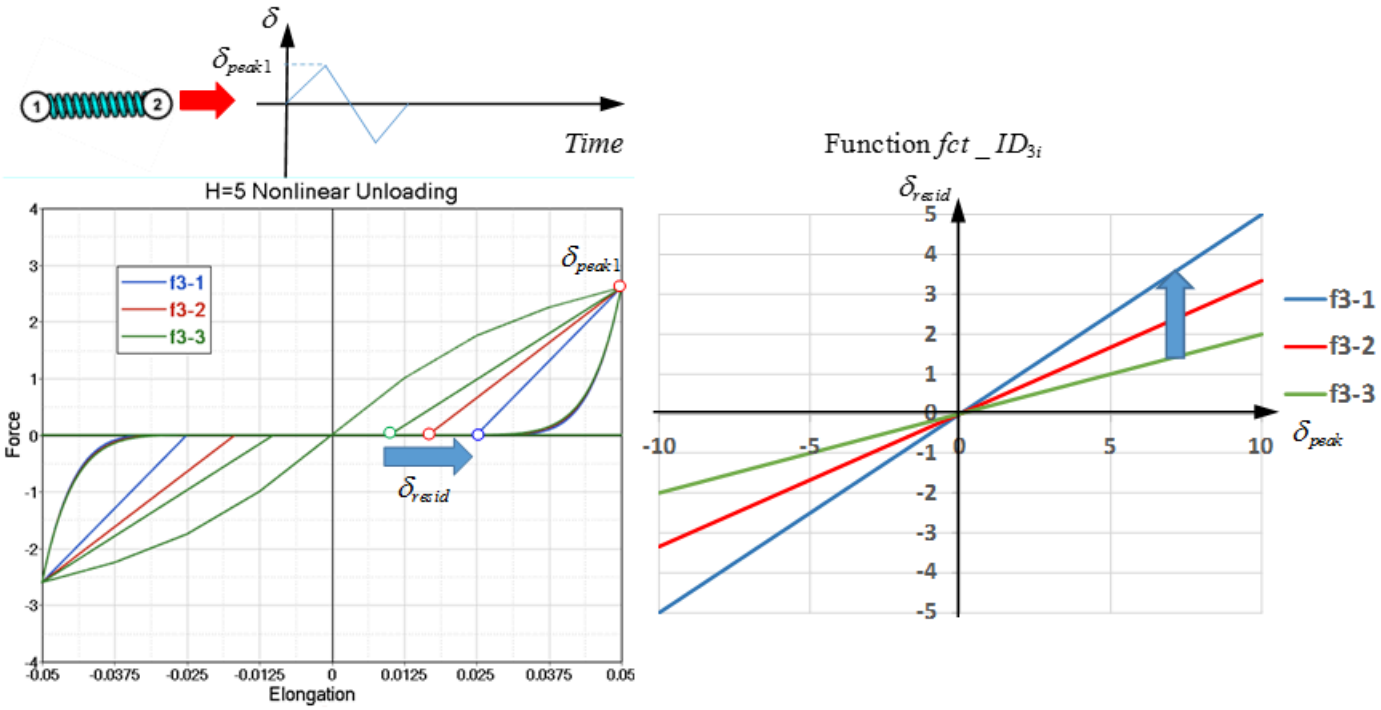


Figure 48: Different linear residual versus maximum displacement curves with H=5

Comparing Figure 47 and Figure 48, shows that the function f_3 only effects the residual displacement δ_{resid} and the shape of unloading curve. The shape of unloading curve is controlled by stiffness K and δ_{peak} (unloading start displacement).

If the same stiffness K and same δ_{peak} are used, then the unloading curve has the same shape.

If the same stiffness K but different δ_{peak} are used, then the unloading curve has a different shape.

If a different stiffness K and same δ_{peak} are used, then the unloading curve has a different shape, as shown in Figure 49.

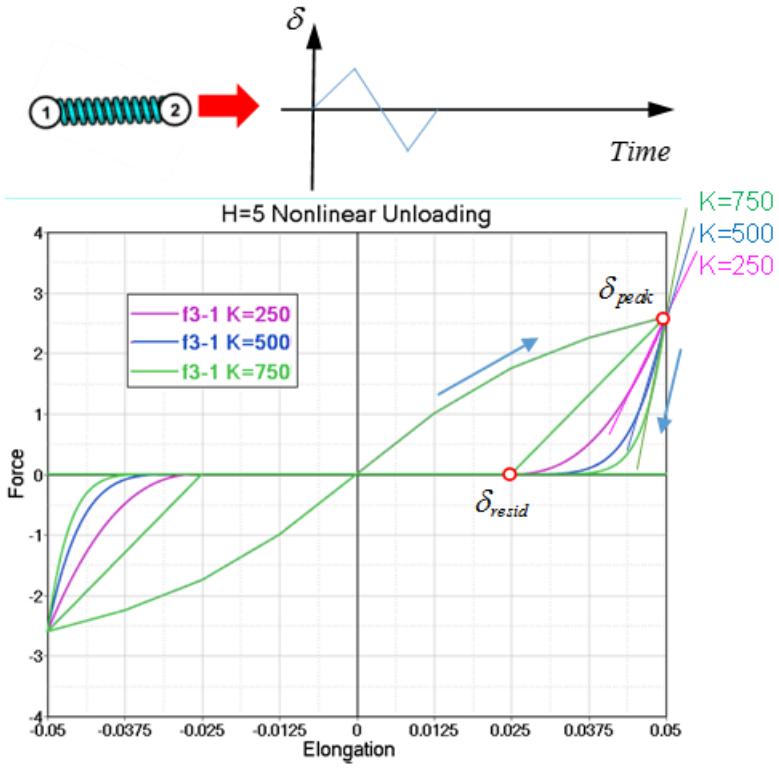


Figure 49: Different K values with H=5

Nonlinear Elastic Plastic Spring Istropic Hardening and Nonlinear Unloading, H=6

Both H=1 and H=6 represent isotropic hardening. In H=6, a nonlinear unloading with function f_3 is used while H=1 uses a constant K_u for linear unloading. When the spring is loaded in tension and then unloads, it follows the defined unloading curve. The unloading curve continues to be used in compressive loading until the loading force in compression matches the maximum loading force in tension. From this point, additional compressive loading uses the input loading function. The loading curve in f_1 is defined using *fct_ID*$_{1i}$ and unloading curve in f_3 is defined using *fct_ID*$_{3i}$.

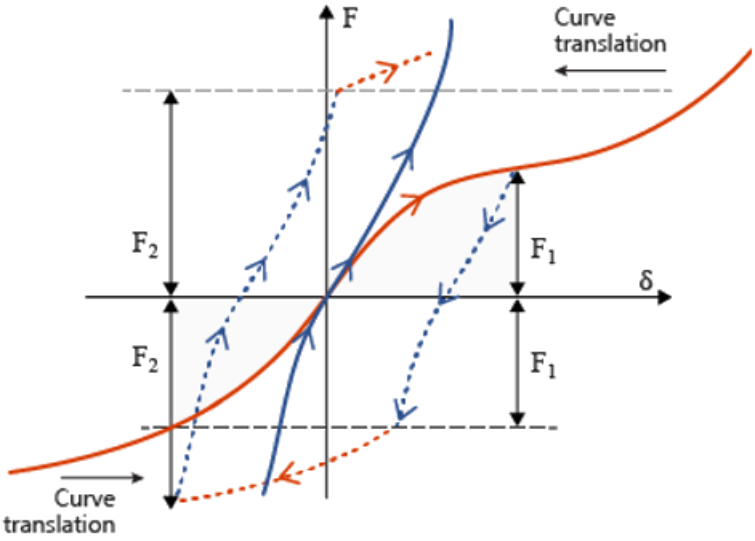


Figure 50: Istropic hardening and nonlinear unloading with H=6

Nonlinear Elastic Plastic Spring Elastic Hysteresis, H=7

With H=7, the spring unloading is initially linear using the input K value until it reach the unloading curve f_3. Additional unloading follows f_3. If reloading occurs, the stiffness K is used to reach the curve f_1, which is then followed. The curve f_3 must have ordinates smaller than curve f_1 at a defined abscissa value. The loading curve in f_1 is defined using *fct_ID*$_{1i}$ and unloading curve in f_3 is defined using *fct_ID*$_{3i}$.

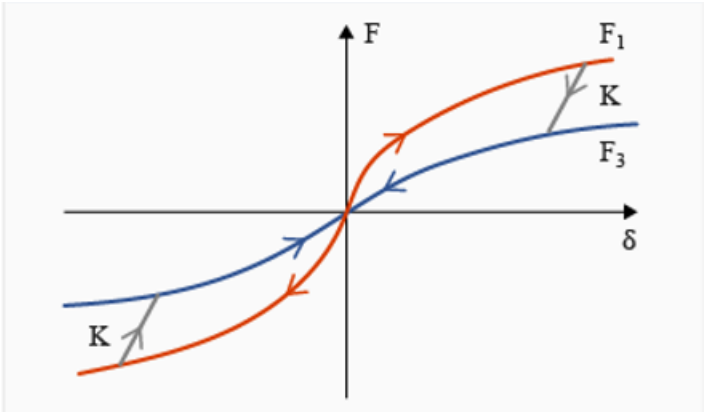


Figure 51: Nonlinear elastic plastic spring elastic hysteresis with H=7

A spring with H=7 could be used to describe hysteresis behavior. Figure 52 shows the difference between H=0 and H=7 under cycle loading. With H=0 (blue curve), it is nonlinear elastic. But with H=7 (red curve), more energy (yellow area in first loop) is absorbed, due to the hysteresis loop.

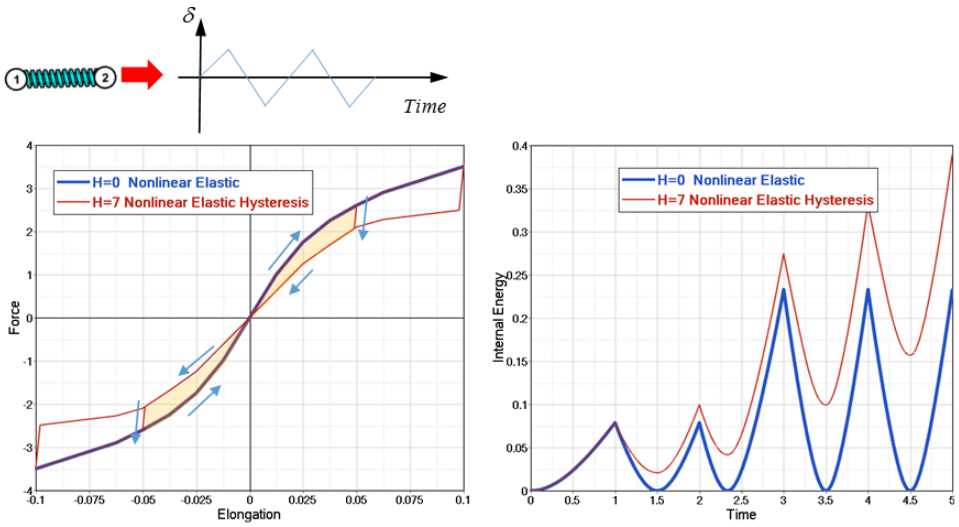


Figure 52: Comparison of nonlinear elastic with hysteresis H*=7 and Nonlinear Elastic* H*=0*

Nonlinear Elastic Total Length Function, H=8

The elastic total length spring H=8 is only available in `/PROP/TYPE4`. Unlike the other hardening options which use the change in spring length, this spring uses the total spring length when defining the spring stiffness. No stiffness occurs in compression. Input *fct_ID*$_{1i}$ to define the force versus total spring length.

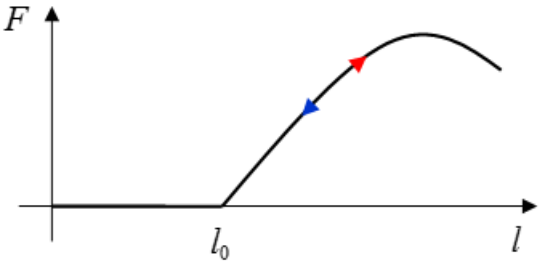


Figure 53: Nonlinear elastic total length function with H*=8*

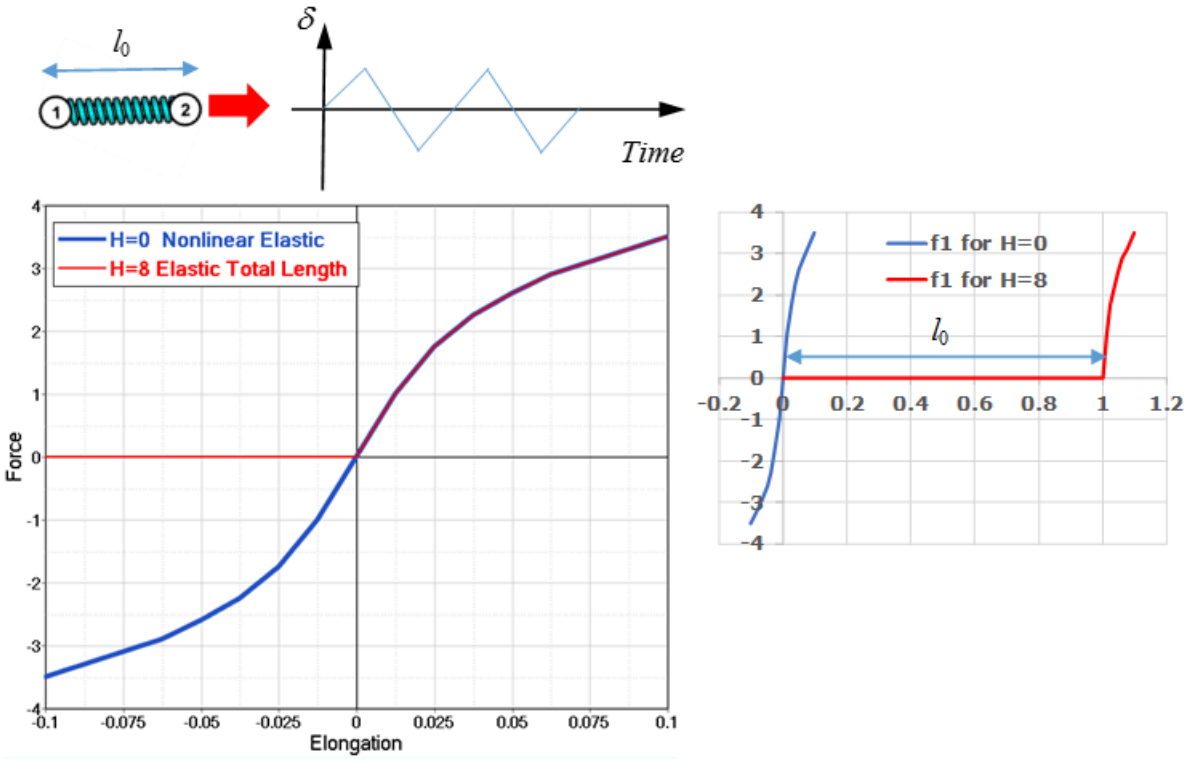


Figure 54: Comparison of H=0 *and* H=8 *with Cyclic Loading Applied*

Dashpot

A dashpot (damper) can be modeled by not defining any spring stiffness. Thus, with the first term in Equation 1 removed, the force becomes only a function of the constant damping coefficient C_i or a nonlinear force versus velocity damping function h as *fct_ID*$_4$.

$$F_i(\dot{\delta}^i) = C_i\dot{\delta}^i + Hscale_i\,h\left(\frac{\dot{\delta}^i}{F_i}\right) \tag{31}$$

Figure 55:

Damping Using a Function

Remembering that the g function scales the force are $f_1 \cdot g$, whereas the h function adds to the force $f_1 + h$. Figure 56 compares these two different methods.

A cyclic loading is applied to a nonlinear elastic plastic spring (H=1) and in two models one which uses the g function to scale the force and the other uses the h function to add to the force.

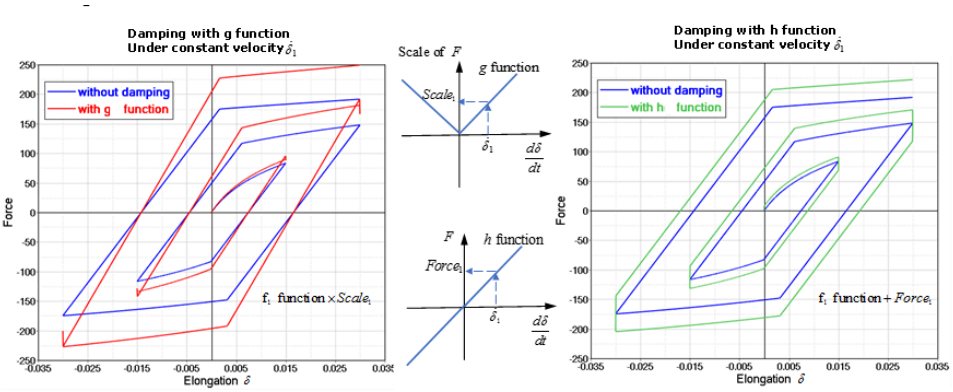


Figure 56:

Note: The function h should have thesame sign as velocity, but the function g should be always positive, due to it is a multiplier to force displacement curve f_1.

Inconsistent Stiffness

When creating a spring property with a user-defined curve "Force versus Displacement" for the stiffness, typically the end of the curve has a very high slope to deal with very high compression. In this case, the following warning is often received with Radioss Starter.

```
WARNING ID: 506
** WARNING IN SPRING PROPERTY
** WARNING IN SPRING PROPERTY SET ID=XXX
STIFFNESS VALUE 100 IS NOT CONSISTENT WITH THE MAXIMUM SLOPE (4550)
OF THE YIELD FUNCTION ID=X
THE STIFFNESS VALUE IS CHANGED TO 1000
```

This warning comes from the fact that the slope of the input curve (the stiffness) is not consistent with the initial stiffness. If the maximum slope of the curve (the maximum stiffness) is greater than the initial stiffness, unloading in the zone of maximum slope will be false (Figure 1). To obtain proper behavior, Radioss Starter modifies the initial stiffness according to the maximum slope.

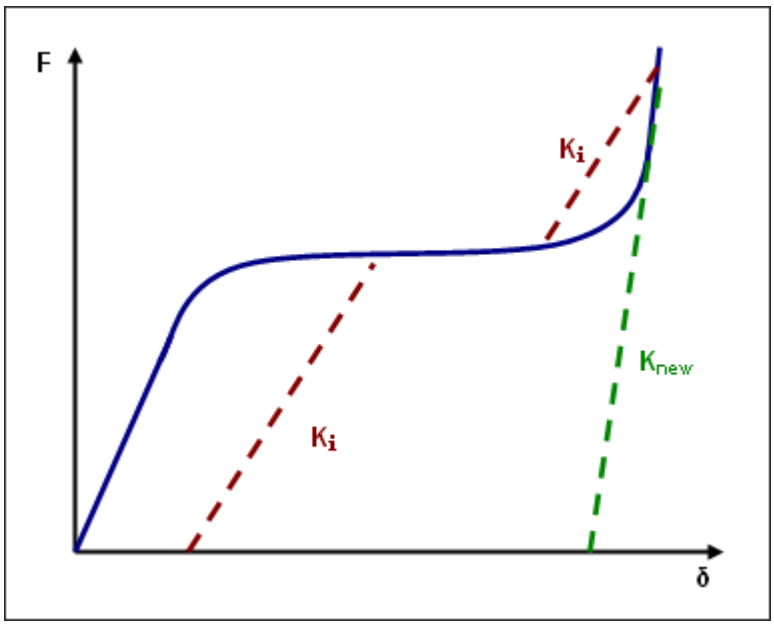


Figure 57: Stiffness not consistent

Spring Coordinate System

Spring Local Coordinate System

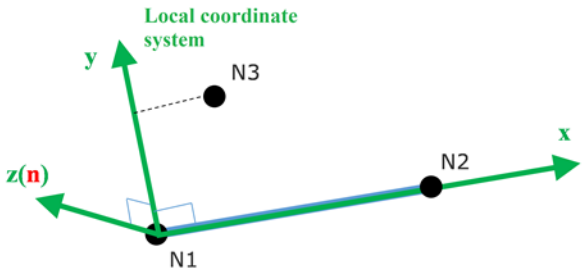


Figure 58:

A Spring element is defined using `/SPRING` with the local coordinate system defined (Figure 58).

- Local **x** direction: from Node N1 to Node N2
- Local **xy** plane: is the plane defined by N1, N2, N3. (N3 optional for 1 DOF springs, if N3 undefined for 3 DOF springs, an appropriate global axis is used).
- Local **z** direction: is normal to the **xy** plane starting at N1.
- Local **y** direction: $\mathbf{y} = \mathbf{z} \wedge \mathbf{x}$, cross product of the **z** and **x** axis.

The type of spring depends on the property assigned to the spring via the `/PART`.

Spring Coordinate System for Force Computation

- Property `/PROP/TYPE4 (SPRING)`, `/PROP/TYPE32 (SPR_PRE)` and `/PROP/TYPE35 (STITCH)`:

These springs all have one degree of freedom and the force is calculated in the local **x** direction of the spring.

- Property `/PROP/TYPE8 (SPR_GENE)`:

  Force and moment are calculated either from global system or in user specified skew. The spring has six DOF computed.

  - If no skew defined then global coordinate system is used

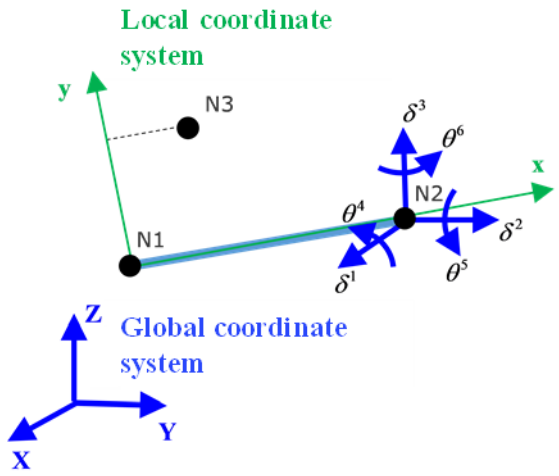


Figure 59: Global coordinate system used in property TYPE8

  - If skew defined then DOF use skew as,

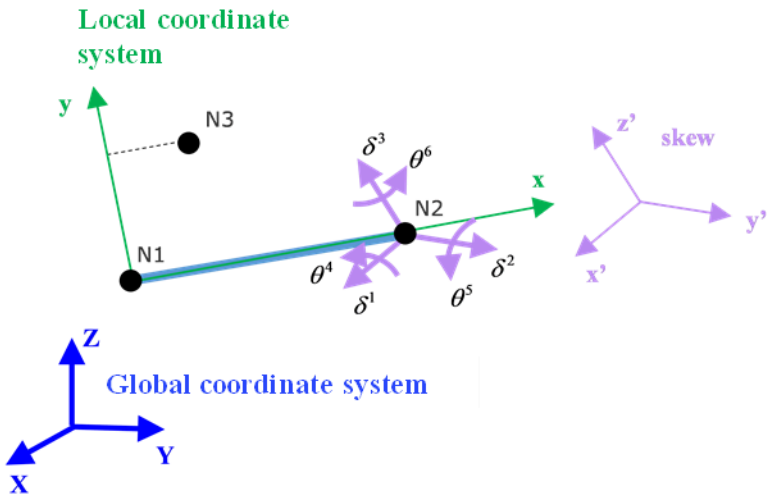


Figure 60: Skew used in property TYPE8

- Property `/PROP/TYPE13 (SPR_BEAM)`, `/PROP/TYPE25 (SPR_AXI)` and `/PROP/TYPE44 (SPR_CRUS)`:

  These properites may use either original local system or updated local system with skew or global system. Local **x** direction is always from Node N1 to N2. Depending on input, **z** direction may

be determined by Node N3, skew or global system. Finally, $\mathbf{y}$ direction is updated so that it is perpendicular to plane $\mathbf{xz}$.

- Local $\mathbf{x}$ direction: From Node N1 to Node N2.
- Local $\mathbf{z}$ direction:
  - If N3 defined, $\mathbf{z}$ direction is perpendicular to the plane of N1, N2, N3. These three nodes should not be in a line.
  - If N3 not defined but skew defined, then $\mathbf{z} = \mathbf{x} \wedge \mathbf{y}'$ with $\mathbf{y}'$ direction from skew.
  - If N3 and skew are not defined, then $\mathbf{z} = \mathbf{x} \wedge \mathbf{Y}$ with Y direction from global system.
- $\mathbf{y}$ direction: $\mathbf{y}$ direction is computed to be perpendicular to the plane $\mathbf{xz}$, $\mathbf{y} = \mathbf{x} \wedge \mathbf{z}$.

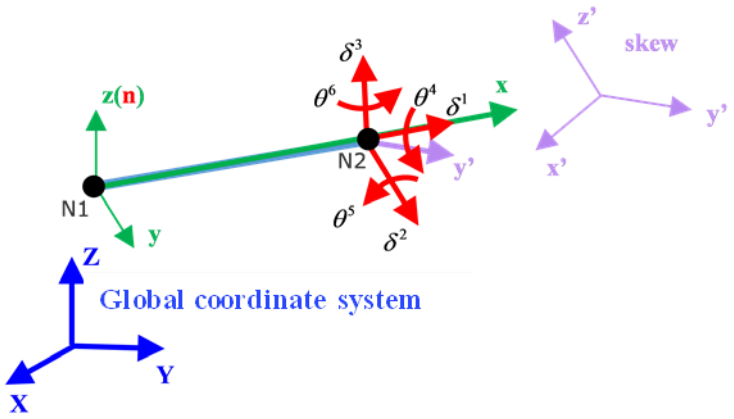


Figure 61: Using skew coordinate system

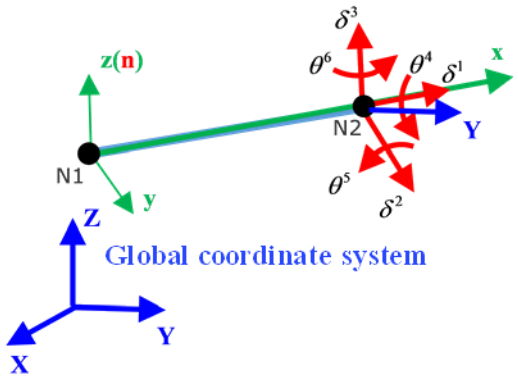


Figure 62: Using global coordinate system

- Property `/PROP/TYPE45 (KJOINT2)`:

  For each joint type, Node N1 and Node N2 are used to define the joint itself. Node N1 and Node N2 are no longer required to only connect to rigid bodies. Nodes N1, N2 can be non-coincident, but for better behavior of the joints, it is strongly advised to use initially coincident nodes. Optional nodes Node N3, Node N4 are used for the definition of local coordinate systems on two sides of the joint. Initially the coordinate systems are coincident. Behavior of the joint is determined by relative motion/rotation of these local coordinate systems.

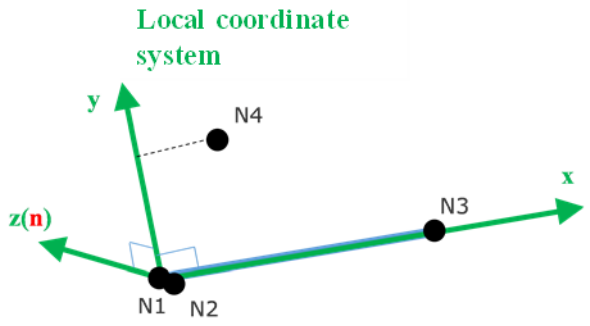


Figure 63:

Nodes of the spherical joints N1, N2 should be preferably coincident. Rigid joints may be described by nodes N1, N2. Global coordinate system is used to define initial orientation of the local coordinate system.

Type No.	Joint Type	dx	dy	dz	θ_X	θ_Y	θ_Z
1	Spherical	x	x	x	0	0	0
8	Rigid	x	x	x	x	x	x

x: denotes a fixed degree of freedom

0: denotes a free (user-defined) degree of freedom

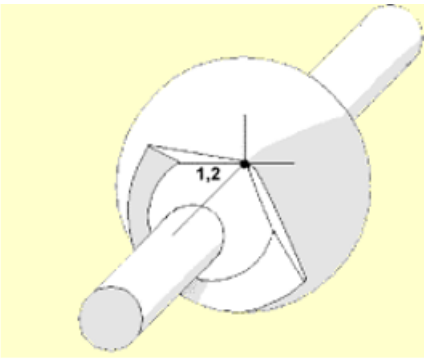


Figure 64: Spherical joint

For revolute, cylindrical and translational joints, Node N3 is used to define the first axis of the joint local coordinate system. If N3 is not specified, the axis of the joints is defined using the line between node N1 and N2. In this case, nodes N1 and N2 should not be coincident.

Type No.	Joint Type	dx	dy	dz	θ_X	θ_Y	θ_Z
2	Revolute	x	x	x	0	x	x
6	Translational	0	x	x	x	x	x
3	Cylindrical	0	x	x	0	x	x

x: denotes a fixed degree of freedom

0: denotes a free (user-defined) degree of freedom

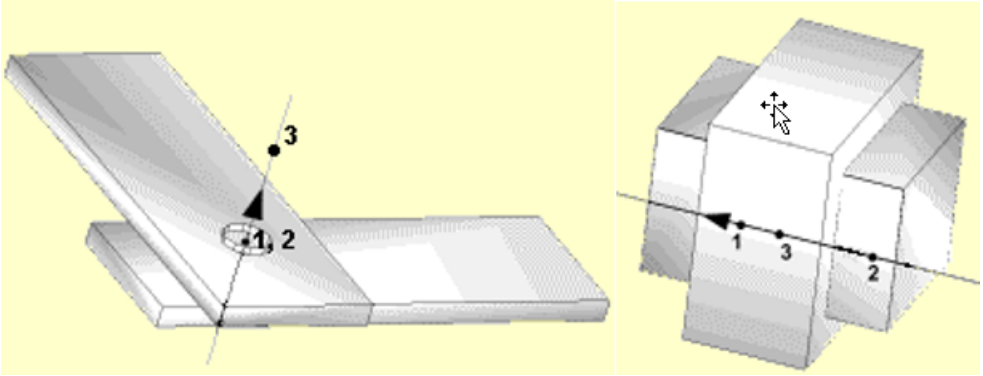


Figure 65: Spherical joint and Translational joint

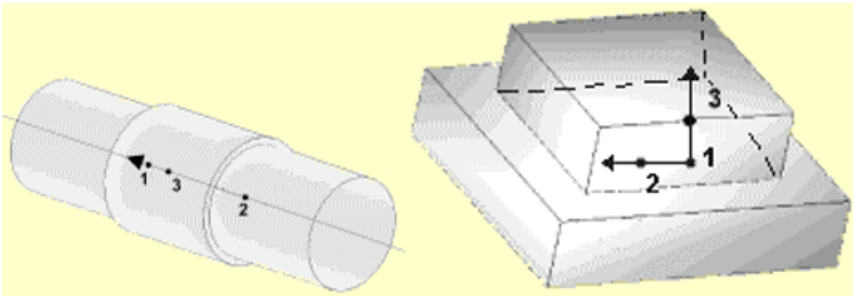


Figure 66: Cylindrical joint and Planar joint

For universal, free, and planar joints, nodes N3 and N4 are used to define the first and the second axes of the joint local coordinate system.

Type No.	Joint Type	dx	dy	dz	θ_X	θ_Y	θ_Z
4	Planar	x	0	0	0	x	x
5	Universal	x	x	x	x	0	0

Type No.	Joint Type	dx	dy	dz	θ_X	θ_Y	θ_Z
	(developmen source only)						
9	Free	0	0	0	0	0	0

x: denotes a fixed degree of freedom

0: denotes a free (user-defined) degree of freedom

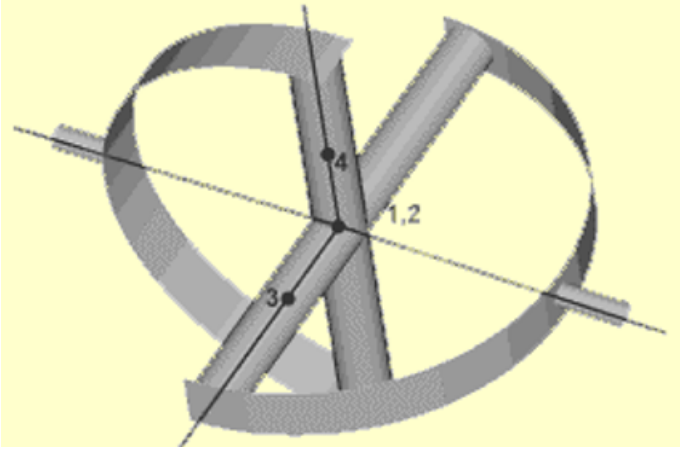


Figure 67: Universal joint

When joint properties are non-isotropic (different stiffness, friction or damping for each non-blocked DOF), a full definition of the local coordinate systems is required. In this case the first axis of the local coordinate system is defined by Node N3, the second axis of the local coordinate system is defined by Node N4, and the third axis is computed automatically.

Depending on skew definition, `/PROP/TYPE45 (KJOINT2)` calculates force and moment.

$Skew\_ID_1 = 0;\ Skew\_ID_2 = 0$	Take spring local coordinate system.
$Skew\_ID_1 \neq 0;\ Skew\_ID_2 = 0$	Take $Skew\_ID_1$
$Skew\_ID_1 \neq 0;\ Skew\_ID_2 \neq 0$	Take mean local coordinate system. The x-axis of $Skew\_ID_1$ and $Skew\_ID_2$ must be parallel.
$Skew\_ID_1 = 0;\ Skew\_ID_2 \neq 0$	First set global coordinate system in $Skew\_ID_1$. Then take mean local coordinate system.

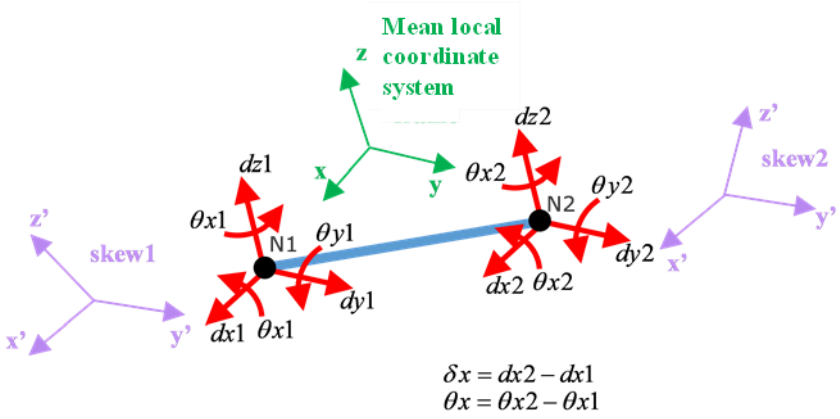


Figure 68: DOF of KJOINT2 properties

See Also
/PROP/TYPE4 (SPRING) (Starter)
/PROP/TYPE8 (SPR_GENE) (Starter)
/PROP/TYPE12 (SPR_PUL) (Starter)
/PROP/TYPE13 (SPR_BEAM) (Starter)
/PROP/TYPE25 (SPR_AXI) (Starter)
/TH/SPRING (Starter)

Spring TYPE4 - 1 DOF Spring (/PROP/SPRING)

Spring TYPE4 is a simple physical spring, physical dashpot or parallel spring and dashpot.

Time step depends on spring mass, stiffness and damping.

$$\Delta t = \sqrt{\frac{M}{K}} \qquad \Delta t = \frac{M}{2C} \qquad \Delta t = \frac{\sqrt{KM + C^2} - C}{K}$$

Figure 69: Spring TYPE4 - Time Step

Critical spring time step ensures the stability of explicit time integration scheme, but it does not ensure a high accuracy on spring vibration behavior. Only two time steps are available during one vibration period of a free spring. To reproduce the true sinusoidal behavior, reduce the time step by a factor of at

least 5. If the spring is used to connect two parts together, the spring vibration period increases and the default spring time step ensures stability and accuracy.

Spring TYPE4 has only one degree of freedom, it can only resist to axial traction and axial compression. The forces applied on the two nodes are always co-linear with local x direction.

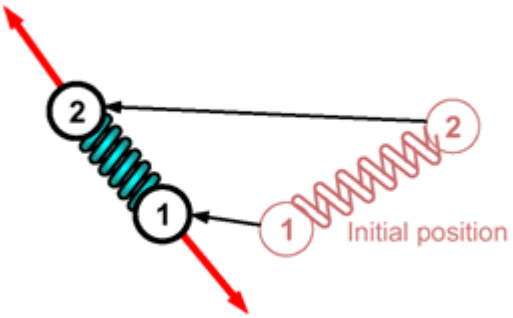


Figure 70: Co-linear Forces in Spring TYPE4

If the spring works under compression, it can reach a zero length situation in which the orientation of internal forces can be arbitrary (Figure 71). That can result in stability problems and should be avoided.

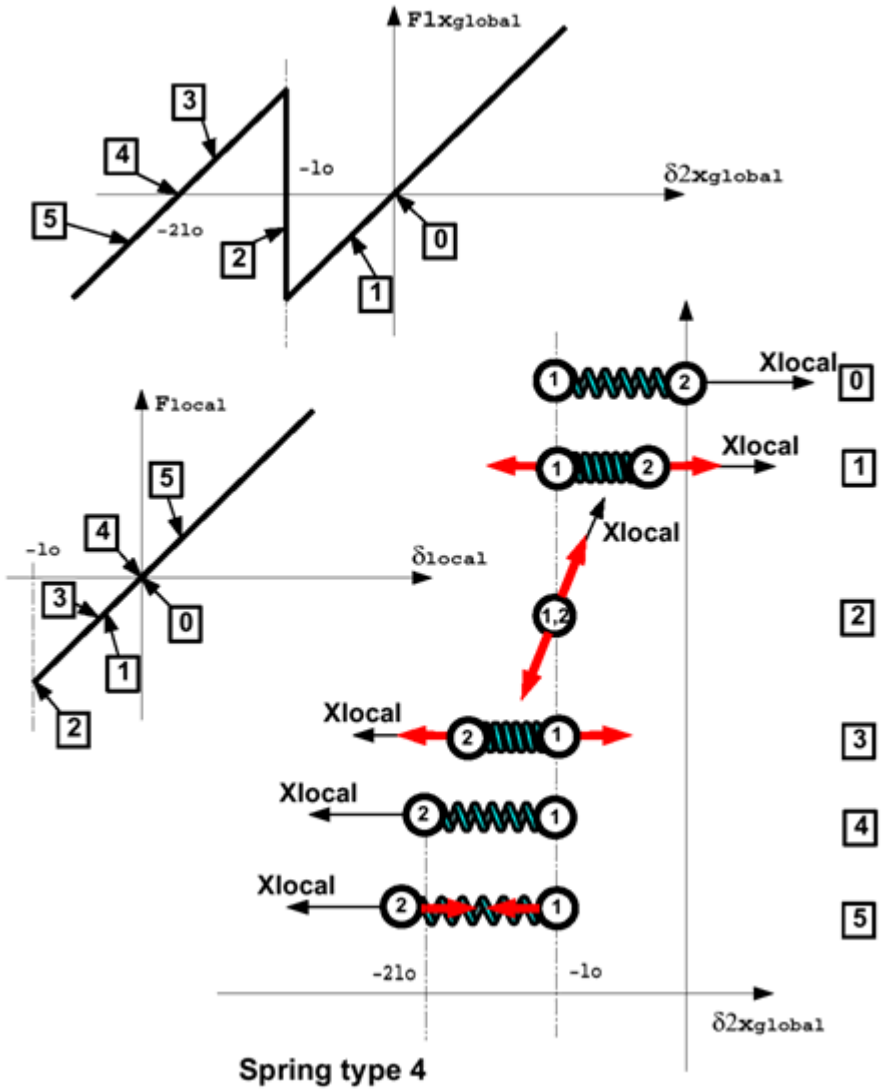


Figure 71: Instability of a Zero Length Spring

For further information, refer to the *Radioss Theory Manual.*

General TYPE8 Spring (/PROP/SPR_GENE)

Spring TYPE8 is a six degrees of freedom spring: three translations and three rotations.

Each degree of freedom is independent from the other, and is associated with a specific stiffness (see Stiffness Formulation for more details). To compute the critical time step, the minimum over all degrees of freedom is kept using the following formula:

$$dt = \min\left(\sqrt{\frac{M}{K}}, \sqrt{\frac{I}{K_\theta}}\right) \quad (32)$$

Where,

I	Moment of inertia
K_θ	The related stiffness

The local coordinate system is not associated with the nodes positions, if no local skew is specified (moving or fixed), the global coordinate system is used.

To understand the physical behavior of this mathematical spring, imagine that the two nodes of the spring are chained together. Therefore, only the relative translation/rotation between nodes can be considered as independent degrees of freedom. Figure 72 is a schematic illustration.

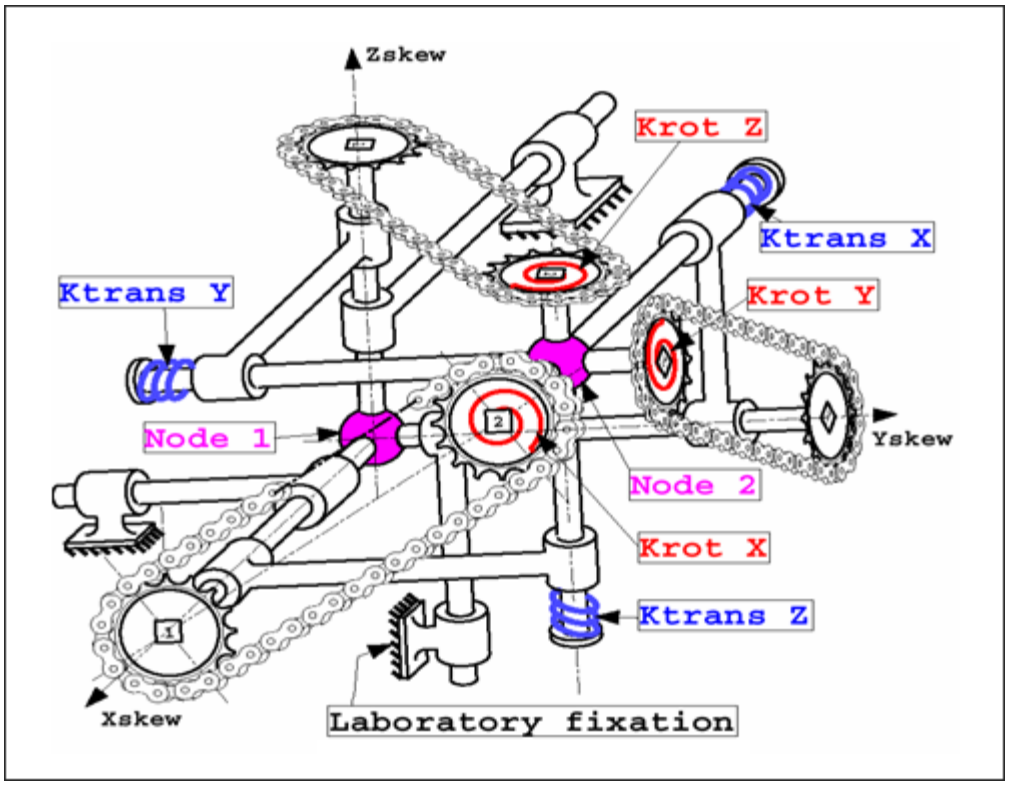


Figure 72: Spring TYPE8

Note: If the local coordinate system is defined through the use of a moving skew, it is possible to emulate the local X-behavior of a spring TYPE4, providing the first two nodes defining the moving skew are node 1 and 2 of the spring (Figure 73).

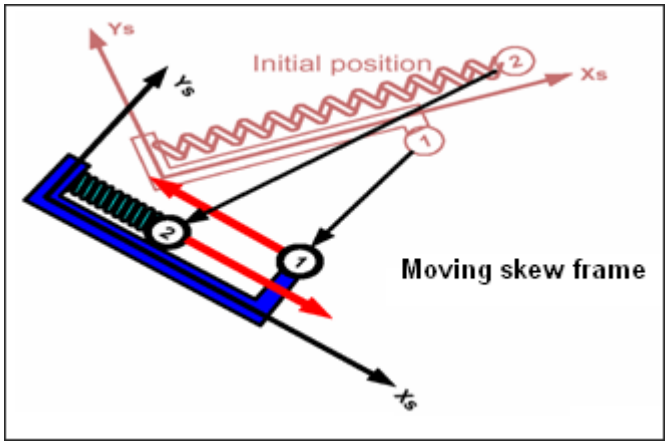


Figure 73: Deformation in Local X-Direction

It can also be noted that the global force equilibrium is fully respected but not the global moment equilibrium. Therefore, the spring is connected to the laboratory which applies the missing moments. It is generally recommended to use spring TYPE8, only if one node is fixed in all directions or if two nodes are coincident (if the two nodes are coincident, the translational stiffness' has to be large enough to ensure that the nodes remain near coincident during the simulation). Doing so, the global moment equilibrium is respected.

Note: An increase in energy may be observed if a non-zero length spring TYPE8 undergoes rigid body motion. The spring TYPE8 when their lengths are not zero.

Spring TYPE8 is generally used to model connection between two parts. For example, pivot points in the steering system are modeled with spring TYPE8. To properly model this type of connection, translational degrees of freedom must be blocked through the use of rather high stiffness' and all rotational degrees of freedom must be set free. Below is an example of such a connection, note that the stiffness values are arbitrary and that they should be adjusted according to a relevant unit system.

Note: It is possible to limit the movement by using a nonlinear stiffness in rotation with a sudden increase (see Stiffness Formulation for mode details).

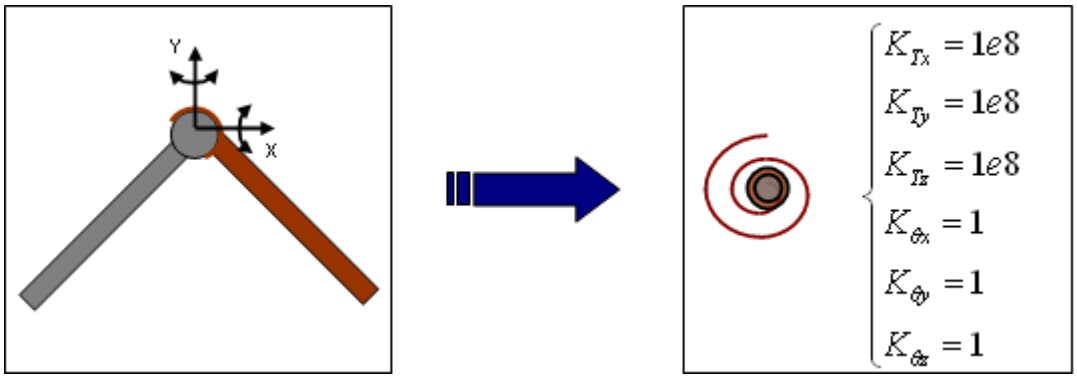


Figure 74: Modeling of Hinge (Use of Spring TYPE8)

For further information, refer to General Spring Elements (TYPE8) in the *Radioss Theory Manual*.

Spring TYPE12 - Pulley (/PROP/SPR_PUL)

Spring TYPE12 is used to model a pulley. When used in a seat belt model, it is defined with three nodes.

Node 2 is located at the pulley, and a deformable rope is joining the three nodes (Figure 75). The spring mass is distributed on the three nodes with ¼ at node 1 and node 3 and ½ at node 2.

A Coulomb friction can be applied at node 2, taking into account the angle between the two strands. Without friction, forces are computed as:

$$|\mathbf{F}_1| = |\mathbf{F}_2| = K\delta \tag{33}$$

With,

δ	Total rope elongation
K	Stiffness

If the Coulomb friction is used, forces are computed as:

$$\mathbf{F}_{fr} = \min\left(|\Delta\mathbf{F}|, \max\left[0, (|\mathbf{F}_1| + |\mathbf{F}_2|) \cdot \tanh\left(\frac{\beta \cdot \mu}{2}\right)\right]\right) \cdot sig(\Delta\mathbf{F}) \tag{34}$$

Where,

$$\mu = \mathrm{f}_{fr}\left(\frac{\Delta F}{Xscale\_F}\right) \cdot Yscale\_F$$

β	Angle (radians unit)
f_{fr}	Function of *fct_ID*$_{fr}$

- I_{fr} =0 (symmetrical behavior)

$$\Delta\mathbf{F} = |\mathbf{F}_1 - \mathbf{F}_2| \tag{35}$$

- I_{fr} =1 (non-symmetrical behavior)

$$\Delta\mathbf{F} = \mathbf{F}_1 - \mathbf{F}_2 \tag{36}$$

δ_1 is the elongation of strand 1-2 and δ_2 of strand 2-3.

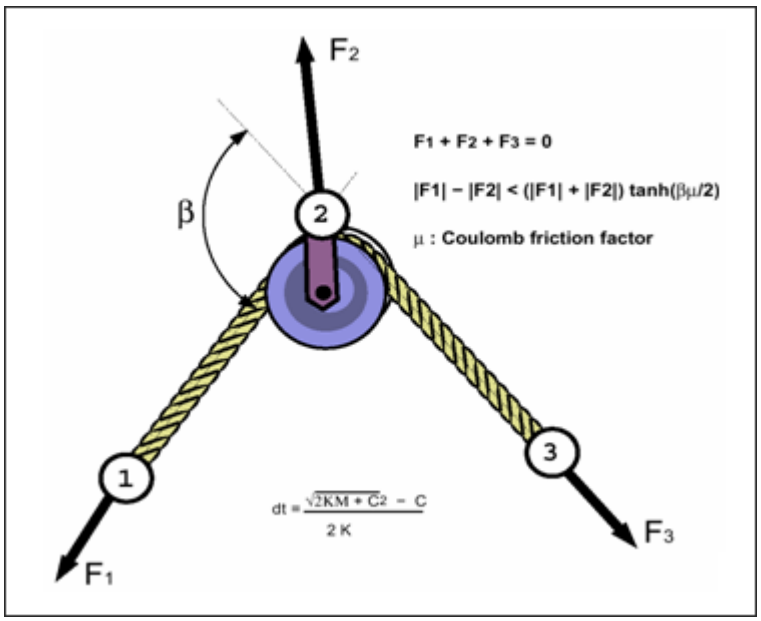


Figure 75: Spring TYPE12, Pulley

Time step is computed with the same equation that for spring TYPE4, but the stiffness is replaced with twice the stiffness to ensure stability with high friction coefficients.

Note: The two strands have to be long enough to avoid node 1, or node 3 slides up to node 2. Nodes 1 and 3 will be stopped at node 2, if there is a knot at nodes 1 and 3.

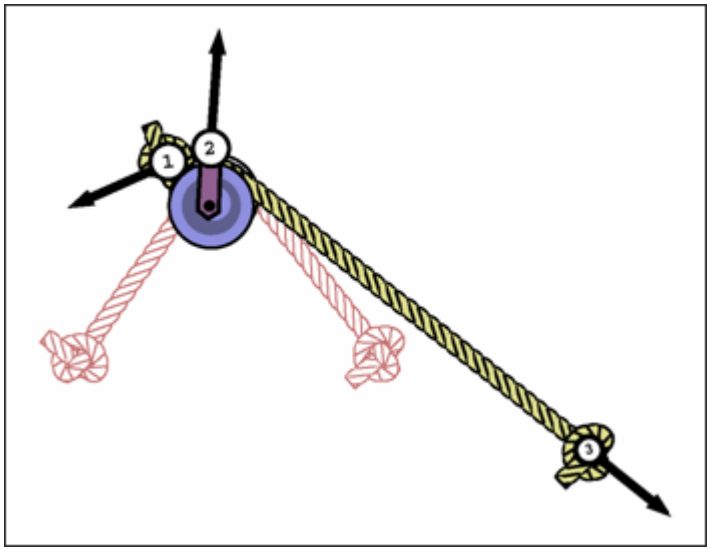


Figure 76: Spring TYPE12, Locking

For further information, refer to the *Radioss Theory Manual*.

Beam TYPE13 Spring (/PROP/SPR_BEAM)

Spring TYPE13 functions like a beam element with several independent modes of deformation:

- Traction / Compression
- Torsion
- Bending (2 modes)
- Shear (2 modes)

Contrary to a beam element, relations between displacements (resp. rotations) and forces (resp. moments) are not deducted from the geometry (area, length, and moment of inertia) and the material properties (Young's modulus, shear modulus). Instead, they are user-defined through different stiffness formulations (see Stiffness Formulation for more details).

Spring TYPE13 works only if the length is not equal to zero. Nodes 1 and 2 are always used to define the local X-axis. Local Y direction is defined at time t=0 and its position is updated at each cycle, taking into account the mean X rotation. Initially, the Y-axis can be defined in different ways. A third node can be used for beam elements. It is also possible to use the local Y-axis of a skew frame. If a skew frame and third node are not defined, global Y-axis replaces the Y skew axis. If the Y skew axis is co-linear with the local X-axis, the local Y-axis and Z-axis are put in an arbitrary position. The Z-axis is finally computed as the cross product of X-axis and Y-axis.

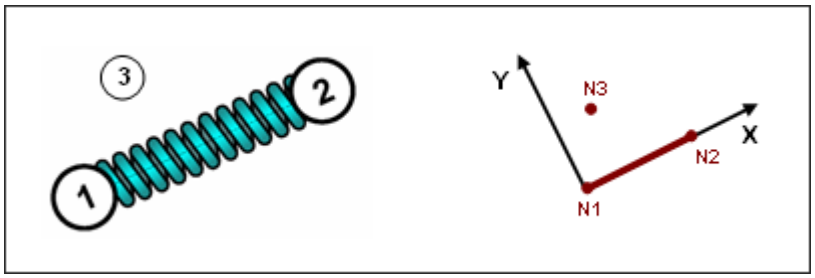


Figure 77: Spring TYPE13

An illustration of a beam type spring is shown below (Figure 78).

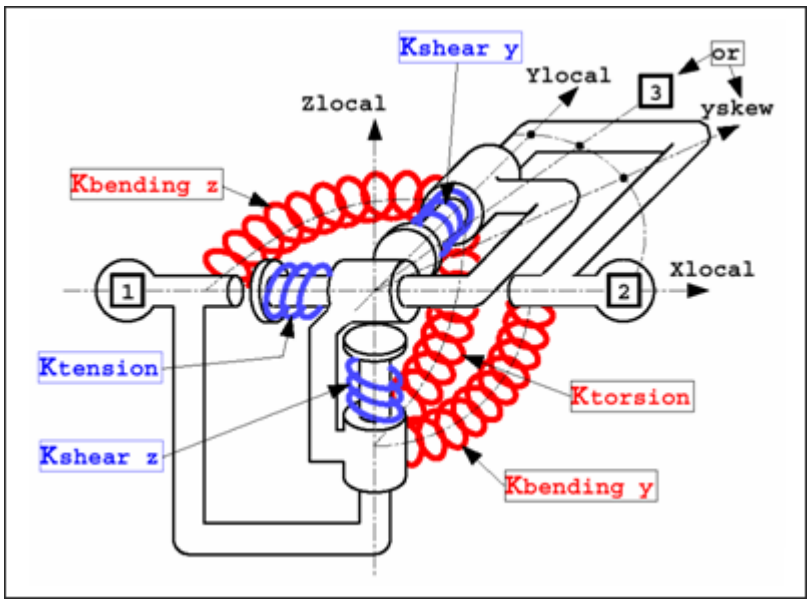


Figure 78: Spring TYPE13

A bending deformation is illustrated in Figure 79. This deformation only takes into account the difference of the two nodes rotations. Beam double bending does generate bending deformation; but shear deformation is shown below.

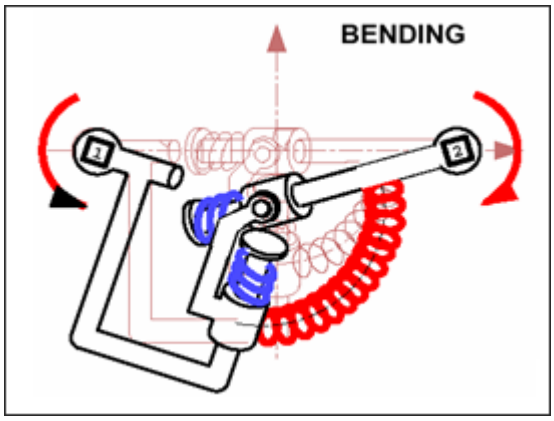


Figure 79: Spring TYPE13 Under Bending

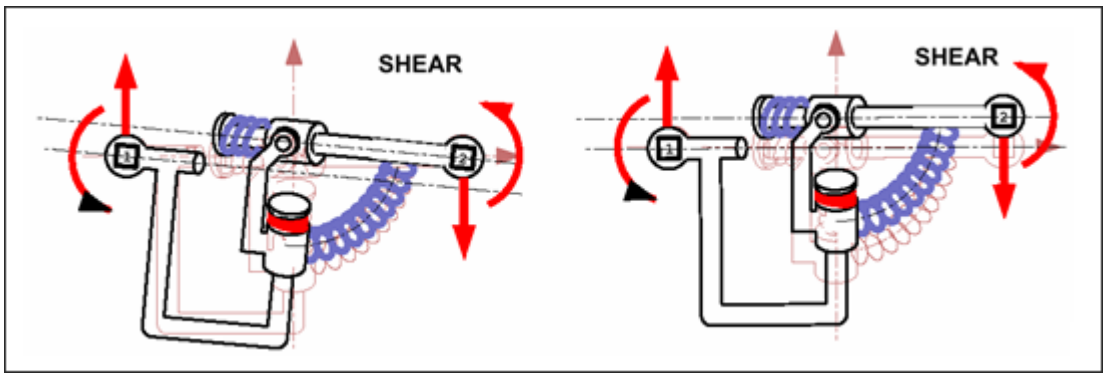


Figure 80: Spring TYPE13 Under Shear-Bending

The beam type spring behaves as a physical beam, in which the variation of bending moment in length generates the increase in shear. The shear force in the spring implies the change of bending moment.

Note: When defining the spring properties, it is strongly recommended to introduce values with physical meanings. A spring with high shear stiffness and a zero bending stiffness is not recommended and may lead to incorrect results.

Contrary to spring TYPE8, rigid body rotation is possible without any artificial force and moment generation for spring TYPE13. Behaviors of spring TYPE8 and TYPE13 under a rigid body rotation are compared in Figure 81.

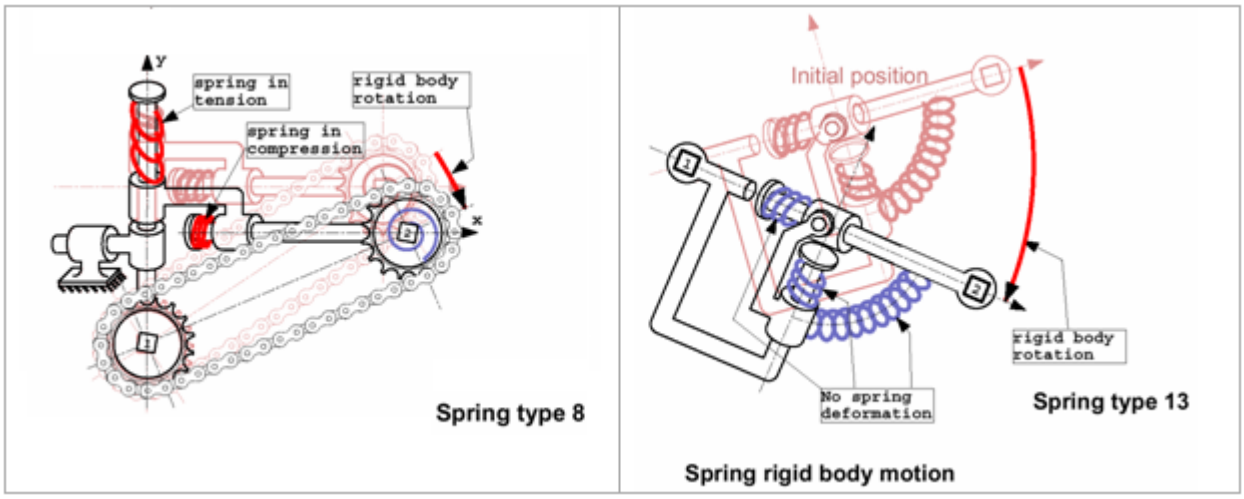


Figure 81: Rigid Body Motions for Spring TYPE8 and TYPE13

To compute the critical time step, the same formula as spring TYPE4 is applied; the minimum overall degree of freedom is finally kept. Therefore to take into account the coupling between bending and shear, and the stiffness' in bending are modified:

$$K_{yy} \approx K_{yy} + l^2 K_z \quad (37)$$

$$K_{zz} \approx K_{zz} + l^2 K_y \quad (38)$$

Between spring TYPE8 and TYPE13, the sign conventions is not the same, therefore results may be rather confusing when trying to compare both spring. For spring TYPE13, the deformation sign is based on the variation of initial length.

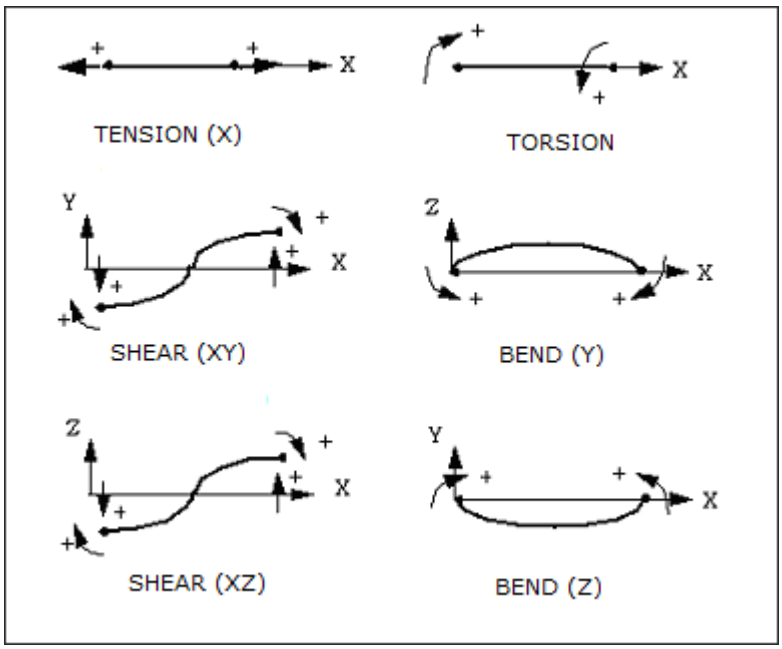


Figure 82: Spring TYPE13 Sign Conventions

For a spring subjected to axial tension, the deformation will always be positive (Figure 82). This is not true any longer for spring TYPE8, since this spring can have a zero length for one (or all) direction, positive and negative spring deformations cannot be defined with the variation of initial length. The sign convention chosen for all degrees of freedom is that a deformation is positive (resp. negative), if displacement (or rotation) of node 2 minus displacement (or rotation) of node 1 is positive (resp. negative).

Figure 83 illustrates the difference between spring TYPE8 and TYPE13, resulting from the sign conventions in Radioss.

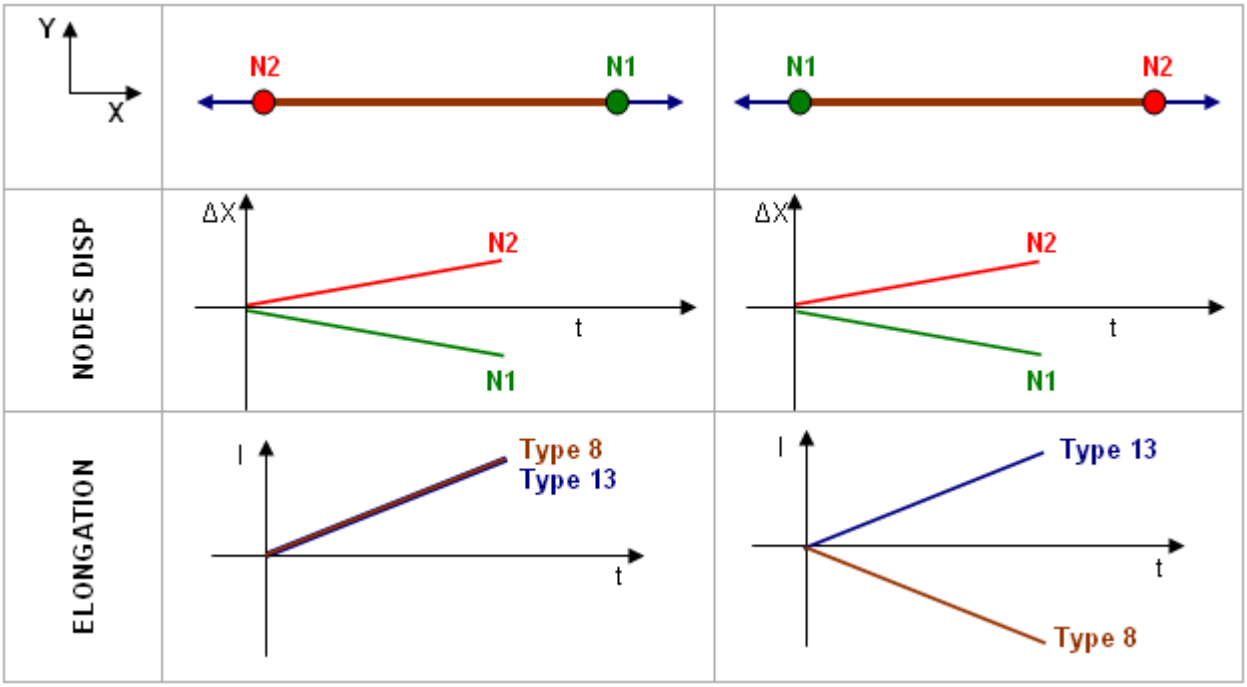


Figure 83: Difference between Spring TYPE8 and TYPE13

For further information, refer to Beam Type Spring Elements (TYPE13) in the *Radioss Theory Manual*.

Kinematic Joint TYPE33 (/PROP/KJOINT)

Kinematic joints are declared by `/PROP/KJOINT`. Joints are defined by a spring and two local coordinate axes, which belong to connected bodies.

Assume that the connected bodies are rigid to ensure the orthogonality of their local axis. However, deformable bodies may also be connected with a joint. If the axis becomes non-orthogonal during deformation, the stability of the joint cannot be ensured.

There are several kinds of kinematic joints available in Radioss, which are listed in Kinematic Joint Types.

Kinematic Joint Types

Type No.	Joint Type	dx	dy	dz	θx	θy	θz
1	Spherical	x	x	x	0	0	0
2	Revolute	x	x	x	0	x	x
3	Cylindrical	0	x	x	0	x	x
4	Planar	x	0	0	0	x	x

Type No.	Joint Type	dx	dy	dz	θx	θy	θz
5	Universal	x	x	x	x	0	0
6	Translation	0	x	x	x	x	x
7	Oldham	x	0	0	x	x	x
8	Rigid	x	x	x	x	x	x
9	Free	0	0	0	0	0	0

x: denotes a blocked degree of freedom

0: denotes a free (user-defined) degrees of freedom

Joint properties are defined in a local frame computed with respect to two connected coordinate systems. They do not need to be initially coincident. If the initial position of the local coordinate axis coincides at any time, the joint local frames are defined at a mean position. Then the joint local frame will be computed with respect to these rotated axes.

There are a total of six joint degrees of freedom: $\delta X', \delta Y', \delta Z', \theta X', \theta Y'$ and $\theta Z'$. They are computed in the local skew frame.

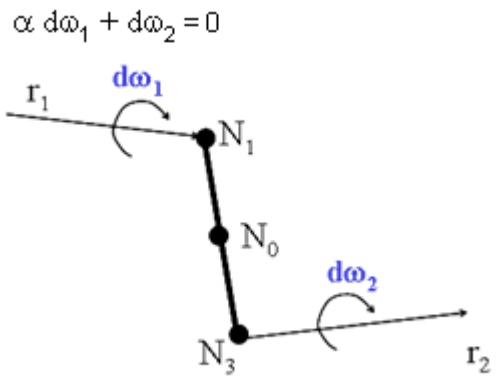


Figure 84: Kinematic Joint Definition

In each type of joint you distinguish the blocked degrees of freedom and the free degrees of freedom. The blocked degrees of freedom are characterized by a constant stiffness. Selecting a high value with respect to the free degrees of freedom stiffness is recommended. The free degrees of freedom have user-defined characteristics, which can be linear or nonlinear elastic, combined with a sub-critical viscous damping.

The translational and rotational degrees of freedom are defined as:

$$\delta = dx_2 - dx_1 \tag{39}$$

Where, dx_1 and dx_2 are total displacement of two joint nodes in the local coordinate system.

$$\theta = \theta_2 - \theta_1 \qquad (40)$$

Where, θ_1 and θ_2 are total relative rotation of two connected body axes, with respect to the local joint coordinate frame.

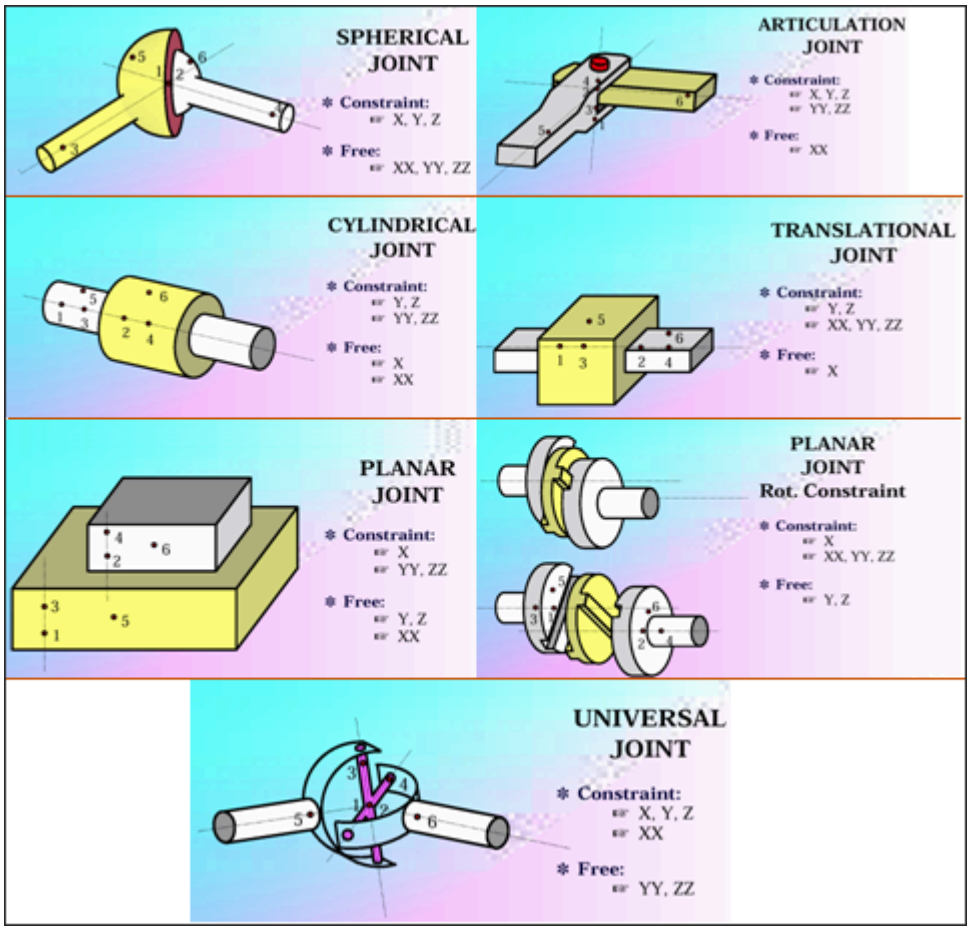


Figure 85: Schematic Representation of Kinematic Joints

Forces and Moments Calculation

The force in direction δ is computed as:

Linear spring: $\mathbf{F} = \mathbf{K}_t\boldsymbol{\delta} + \mathbf{C}_t\dot{\boldsymbol{\delta}}$

$\mathbf{K}_t$: Translational stiffness (K_{tx}, K_{ty}, and K_{tz})

$\mathbf{C}_t$: Translational viscosity (C_{tx}, C_{ty}, and C_{tz})

Nonlinear spring: $\mathbf{F} = \mathbf{K}_t\mathrm{f}(\boldsymbol{\delta}) + \mathbf{C}_t\mathrm{g}(\dot{\boldsymbol{\delta}})$

The moment in direction θ is computed as:

Linear spring: $\mathbf{M} = \mathbf{K}_r\boldsymbol{\theta} + \mathbf{C}_r\dot{\boldsymbol{\theta}}$

$\mathbf{K}_r$: Rotational stiffness (K_{rx}, K_{ry}, and K_{rz})

$\mathbf{C}_r$: Rotational viscosity (C_{rx}, C_{ry}, and C_{rz})

Nonlinear spring: $\mathbf{M} = \mathbf{K}_r f(\boldsymbol{\theta}) + \mathbf{C}_r g(\dot{\boldsymbol{\theta}})$

The joint length may be equal to 0. It is recommended to use a zero length spring to define a spherical joint or a universal joint. To satisfy the global balance of moments in a general case, correction terms in the rotational degrees of freedom are calculated as:

$$M_{\theta x} = M_{\theta x} + L_y \times F_z - L_z \times F_y \quad (41)$$

$$M_{\theta y} = M_{\theta y} + L_z \times F_x - L_x \times F_z \quad (42)$$

$$M_{\theta z} = M_{\theta z} + L_x \times F_y - L_y \times F_x \quad (43)$$

Joints do not have user-defined mass or inertia, so the nodal time step is always used.

Spring Joint TYPE45 (/PROP/KJOINT2)

In Radioss, property TYPE45 is the joint type spring.

DOF of KJOINT2

The DOF (degrees of freedom) of KJOINT2 are defined with option "Type." Different types of joint made by different DOF combinations. The following types are available in Radioss:

Table 8: DOF Only Free in Rotation

Type No.	Joint Type	dx	dy	dz	θ_x	θ_y	θ_z
1	Spherical	x	x	x	0	0	0
5	Universal (developmer source only)	x	x	x	x	0	0
2	Revolute	x	x	x	0	x	x

x: denotes a fixed degree of freedom

0: denotes a free (user-defined) degree of freedom

Table 9: DOF Only Free in Translation

Type No.	Joint Type	dx	dy	dz	θ_x	θ_y	θ_z
6	Translational	0	x	x	x	x	x
6	Oldham	x	0	0	x	x	x

x: denotes a fixed degree of freedom

0: denotes a free (user-defined) degree of freedom

Table 10: DOF Free in Translation and Rotation

Type No.	**Joint Type**	**dx**	**dy**	**dz**	θ_x	θ_y	θ_z
3	Cylindrical	0	x	x	0	x	x
4	Planar	x	0	0	0	x	x

x: denotes a fixed degree of freedom

0: denotes a free (user-defined) degree of freedom

Table 11: Specific DOF

Type No.	**Joint Type**	**dx**	**dy**	**dz**	θ_x	θ_y	θ_z
8	Rigid	x	x	x	x	x	x
9	Free	0	0	0	0	0	0

x: denotes a fixed degree of freedom

0: denotes a free (user-defined) degree of freedom

Table 12:

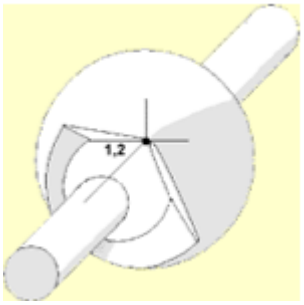


Figure 86: Spherical

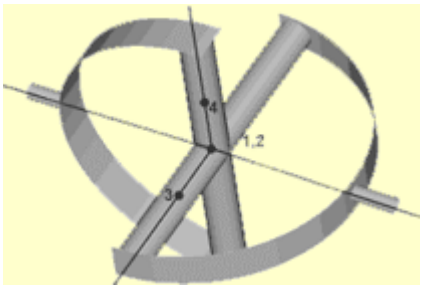


Figure 87: Universal

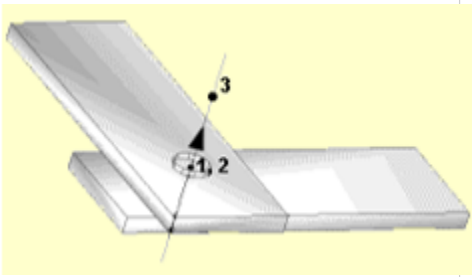


Figure 88: Revolute

Table 13:

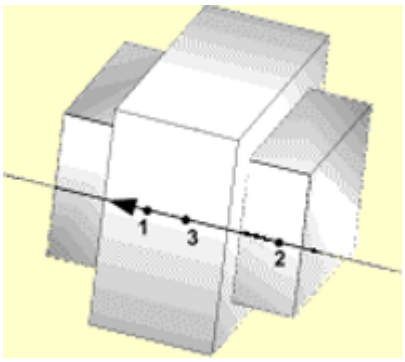


Figure 89: Translational

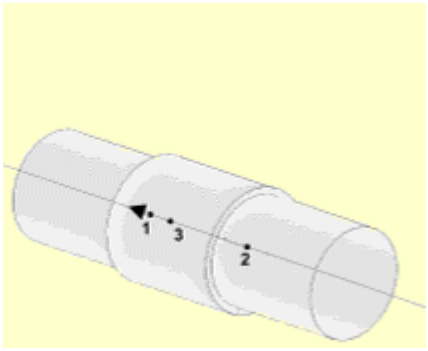


Figure 90: Cylindrical

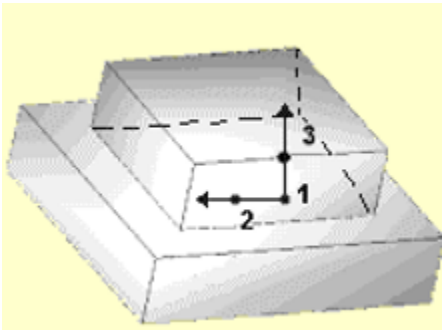


Figure 91: Planar

The coordinate for DOF is the one used for spring force computation. It is not necessarily the spring local coordinate system. Depending on the different types of KJOINT2 and the definition of skew, different coordinate systems are used for spring force computations.

Stiffness and Damping in KJOINT2

There are four types of stiffness used in the KJOINT2 property.

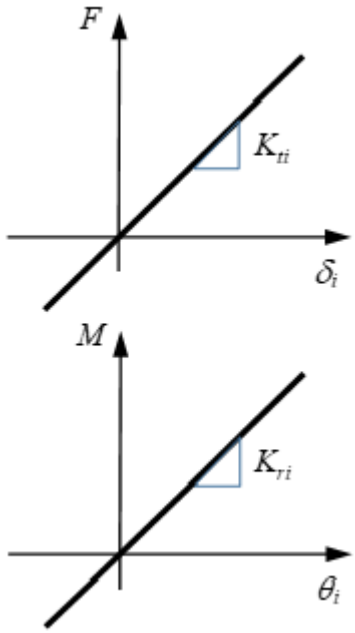
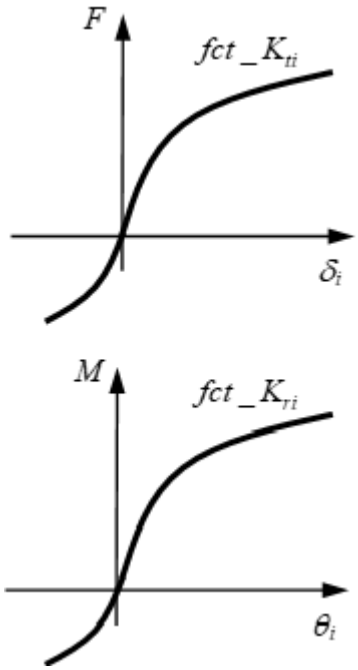
1. Blocked stiffness K_n. For example, for blocked stiffness = Yes:
   - If Type = 1 (Spherical), then K_n are blocked stiffnesses in translational directions d_x, d_y, d_z.
   - If Type = 6 (Translational), then K_n are blocked stiffnesses in translational directions d_y, d_z and in rotational directions $\theta_X, \theta_Y, \theta_Z$.

     Table 14:

Type No.	Joint Type	dx	dy	dz	θ_x	θ_y	θ_z
1	Spherical	x	x	x	0	0	0
6	Translationa	0	x	x	x	x	x

     If K_n = 0 (default), Radioss computes blocked stiffness internally. If K_n > 0, user-defined blocked stiffness is used.
2. Translational stiffness K_{ti} and rotational stiffness K_{ri}. These stiffnesses are used in free DOF direction and used for force and moment computation:

Table 15:

Linear Spring	Nonlinear Spring
$\mathbf{F} = \mathbf{K}_{ti}\boldsymbol{\delta}$ $\mathbf{M} = \mathbf{K}_{ri}\boldsymbol{\theta}$ K_{ti}: translational stiffness (K_{tx}, K_{ty}, K_{tz}) K_{ri}: rotational stiffness (K_{rx}, K_{ry}, K_{rz})  *Figure 92:*	$\mathbf{F} = \mathbf{K}_{ti}\mathrm{f}(\boldsymbol{\delta})$ $\mathbf{M} = \mathbf{K}_{ri}\mathrm{f}(\boldsymbol{\theta})$ K_{ti}: translational stiffness factor (K_{tx}, K_{ty}, K_{tz}) K_{ri}: rotational stiffness factor (K_{rx}, K_{ry}, K_{rz}) $\mathrm{f}(\boldsymbol{\delta})$ is translational stiffness function fct_K_{ti} $\mathrm{f}(\boldsymbol{\theta})$ is rotational stiffness function fct_K_{ti}  *Figure 93:*

3. Translational viscosity coefficient C_{ti} and rotational viscosity coefficient C_{ri} are for damping. These viscosity coefficients are used in free DOF direction and used for force and moment computation.

Table 16:

Linear Spring	Nonlinear Spring
$\mathbf{F} = \mathbf{C}_{ti}\dot{\boldsymbol{\delta}}$ $\mathbf{M} = \mathbf{C}_{ri}\dot{\boldsymbol{\theta}}$ C_{ti}: translational stiffness (C_{tx}, C_{ty}, C_{tz})	$\mathbf{F} = \mathbf{C}_{ti}\mathrm{g}(\dot{\boldsymbol{\delta}})$ $\mathbf{M} = \mathbf{C}_{ri}\mathrm{g}(\dot{\boldsymbol{\theta}})$ C_{ti}: translational stiffness factor (C_{tx}, C_{ty}, C_{tz})

Linear Spring	**Nonlinear Spring**
C_{ri}: rotational stiffness (C_{rx}, C_{ry}, C_{rz})	C_{ri}: rotational stiffness factor (C_{rx}, C_{ry}, C_{rz}) $g(\dot{\delta})$ is translational stiffness function fct_C_{ti} $g(\dot{\theta})$ is rotational stiffness function fct_C_{ri}

For example, tensile spring (which used KJOINT2) with constant velocity = 1.33:

- If only translational stiffness is set to $K_t = 100$, then force (green curve in figure below) increases with K_t.
- If only translational viscosity is set to $C_t = 50$, then force (orange curve in figure below) is constant since tensile velocity is constant. The force is $Force = C_t\dot{\delta} = 50 \times 1.33 = 66.5$.
- If both translational stiffness $K_t = 100$ and translational viscosity $C_t = 50$ are set, then force (blue curve in figure below) is the combination of the above two cases with $Force = K_t\delta + C_t\dot{\delta}$.

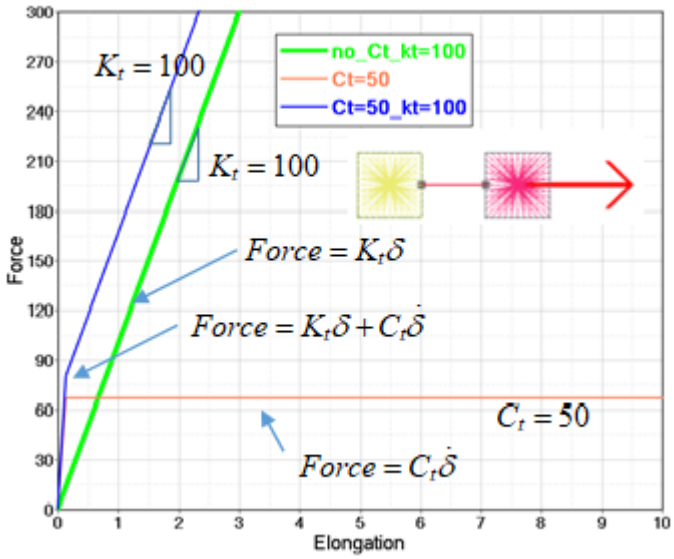


Figure 94:

4. Frictional stiffness in translational direction K_{fxi} and in rotational direction K_{fri}. These are used to compute penalty force to prevent displacement exceeding the limit SD_{i+}, SD_{i-} or rotation exceeding the limit SA_{i+}, SA_{i-}.

If K_{fxi}, K_{fri} are not set, Radioss will compute it automatically with very high stiffness.

For example, in the following tensile test, setting $SD_{i+} = 100$, then:

- If K_{fxi} are not set (figure below on the left), then very high stiffness is used after reaching limit $SD_{i+} = 100$.

- If $K_{fxi} = 1000$ is set (figure below on the right), then $K_{fxi} = 1000$ is used after reaching limit $SD_{i+} = 100$.

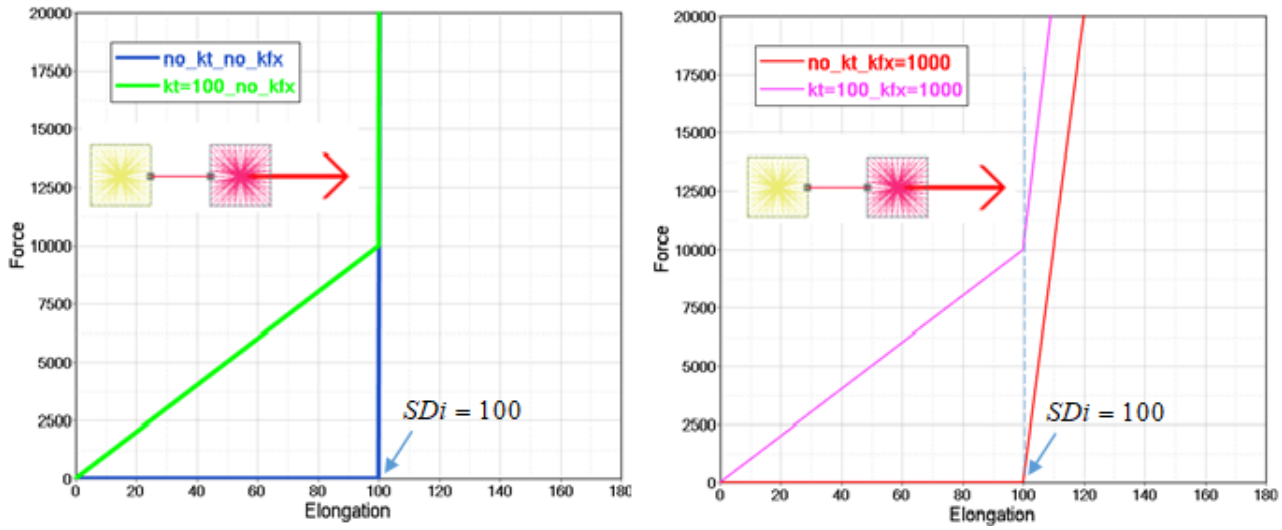


Figure 95:

The above tests also show translational stiffness K_{ti}. This only affects force before reaching the limit SD_{i+}. The value of K_{fxi} should not be set too small. This value is used to simulate blocking behavior after reaching the limit. Too small a value for K_{fxi} leads to the following error message:

```
WARNING ID :           979
** WARNING IN FRICTION DEFINITION FOR KJOINT2
DESCRIPTION :
   ELASTIC STIFFNESS 0.1000000000000 IS SMALLER THAN THE MAX DERIVATIVE
 720.0000000000 OF THE
   FRICTION FUNCTION 0
```

- If K_{ti} is not set but $K_{fxi} = 1000$. The figure below on the left shows how constant friction force $FF_i = 15000$ affects the spring force. With $FF_i = 15000$, force increases with K_{fxi} to 15,000 and then remains constant until reaching the limit SD_{i+}. After reaching SD_{i+}, force increases with stiffness K_{fxi} again.
- If, as in the previous case, an additionally set translation stiffness $K_{ti} = 100$ (figure below on the right), then force increases with K_{fxi} to $FF_i + K_{ti}\delta_i = 15000 + 100 \times 15 = 16500$ and then increases with translation stiffness $K_{ti} = 100$ until reaching the limit SD_{i+}. After reaching SD_{i+}, force increases with stiffness K_{fxi} again.

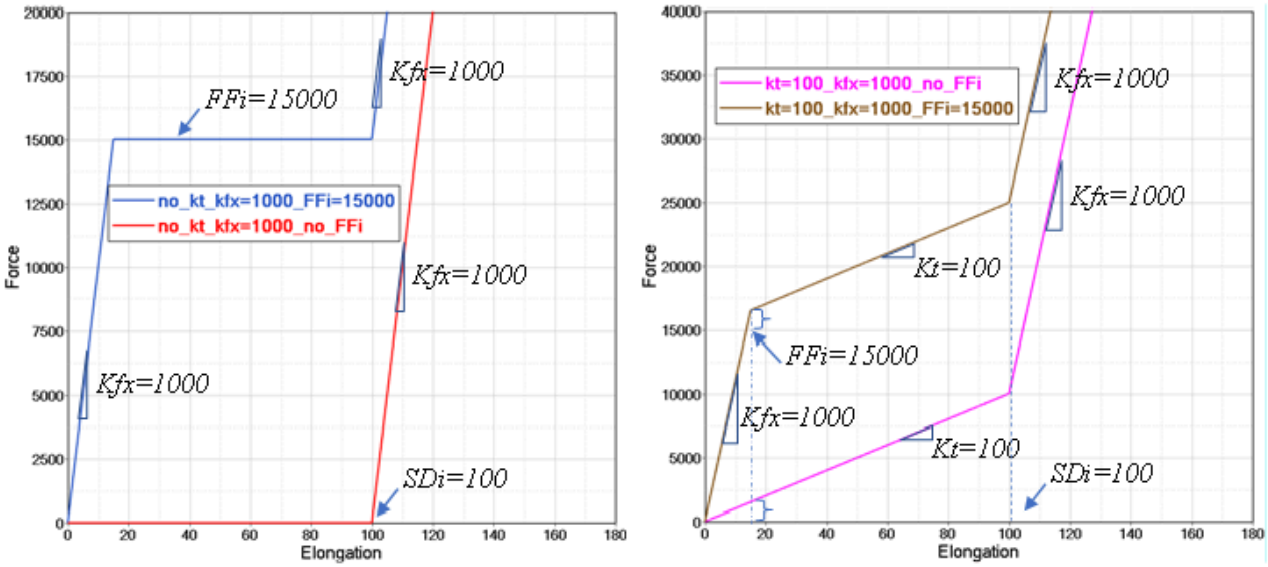


Figure 96:

K_{fxi}, K_{fri} must be defined if it is required to consider friction behavior after reaching the stop limit. Otherwise, even if friction (constant friction FF_i, FM_i or friction curve fct_FF_i, fct_FM_i) has been defined, Radioss will still take the internally computed (very high) stiffness to treat the force (or moment) behavior after reaching the limit.

Axisymmetric Spring TYPE25 (/PROP/SPR_AXI)

This spring is a simplification of spring TYPE13; in which the properties of the spring cross-section are considered to be invariable with respect to Y and Z.

That allows you to simplify input formats to avoid errors. For further information about this spring, refer to *Radioss Starter Input Manual* or the *Radioss Theory Manual*.

Mesh Recommendations

Mesh recommendations in crash worthiness and in Implicit Analysis are covered.

Number of Elements Recommended

It is recommended to set a minimum of five to 10 elements in a structural working length. For buckling situation, five to 10 elements along one buckle wave length is also recommended (Figure 97).

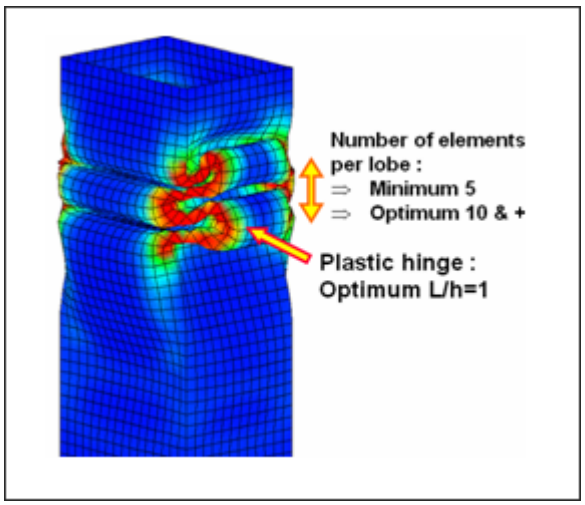


Figure 97: Number of Elements Along a Buckle Wave

On the other hand, due to the direct relation between the size of the smallest element in the mesh and the time step, it is valuable to suppress geometry details, if they are smaller than the average size of elements (Figure 98). A uniform mesh allows you to optimize time step and to ensure the consistency of the momentum transmission during shock wave propagation to avoid parasite reflections. A minimum of three elements along the width is required for better elasto-plastic behavior.

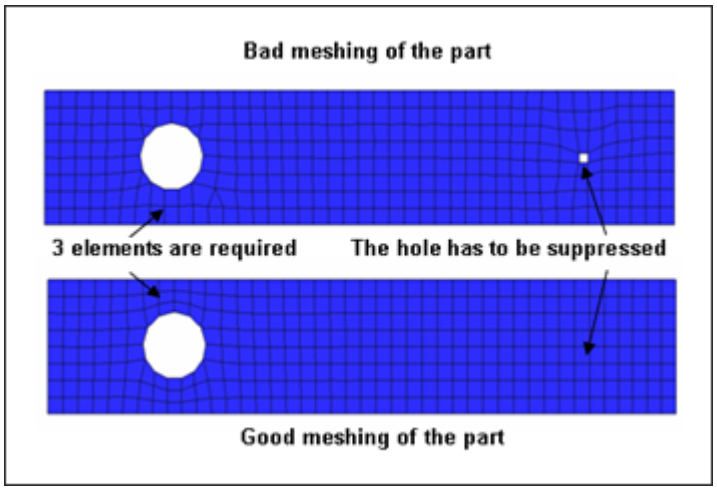


Figure 98: Surface Mesh Generation for Part with Holes

Observe the minimum number of elements, taking into account the mechanical behavior. The minimum number of elements in the width is:

- 1 for the parts working with quasi-uniform stress distribution as pressure and shear loading without bending
- 2 for elastic behavior, including bending
- 3 for low accuracy elastoplastic behavior
- 5 for good accuracy elastoplastic behavior
- 10 for good accuracy elastoplastic behavior, including local loading or local discontinuities

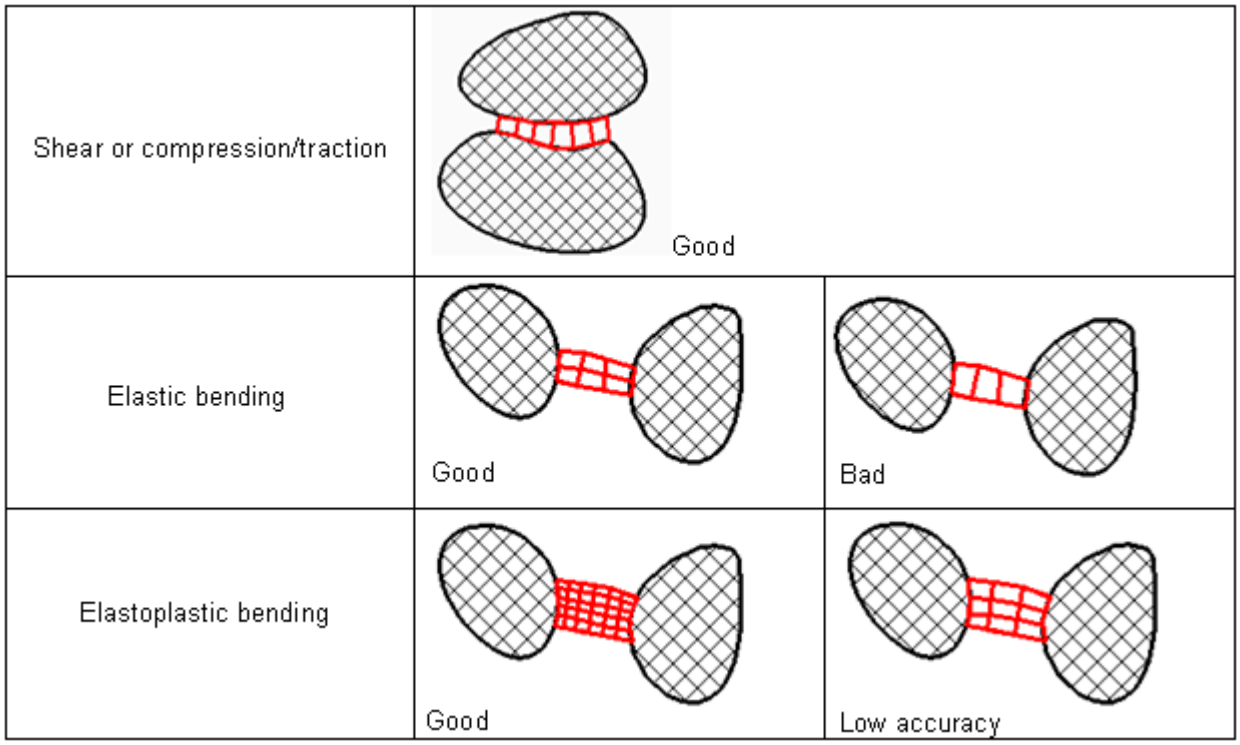


Figure 99: Minimum Number of Elements

Mesh Transition

It is not recommended to use different kinds of element formulation in a given physical part. Fully-integrated and under-integrated elements do not have the same stiffness matrices; but they do have the same mass matrix. The transition of momentum between these two kinds of elements may be a problem.

To reduce the number of triangles and to improve the consistency of the mesh, the transition patterns illustrated in Figure 100 can be used.

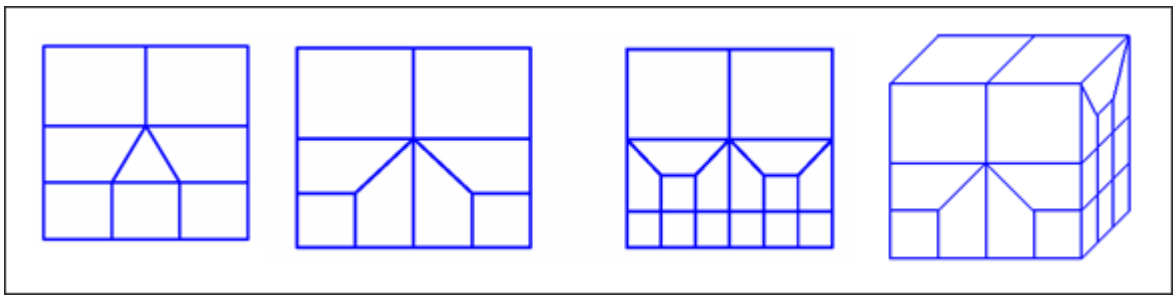


Figure 100: Mesh Transition Examples

Mesh Patterns

With one integration point, there are no element shapes for which the element formulation becomes completely wrong (elements with angles larger than 180 degrees are still working with bad accuracy, but without numerical problems). It is recommended to use elements as regular as possible with an angle typically between 45 and 120 degrees. The ratio between the largest and smallest edge is

not critical for explicit formulation. A uniform mesh is recommended to distribute mass uniformly over the structure. It is possible to avoid triangle creation, even for a triangular geometry or a circle (Figure 101).

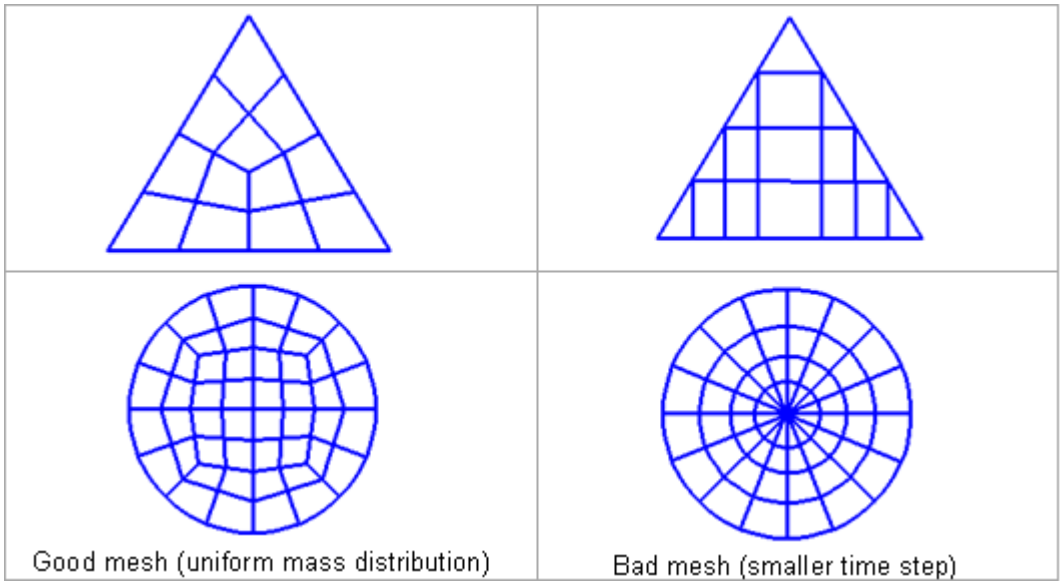


Figure 101: Mesh Transition Examples

In Radioss what kind of mesh quality given often depends on the robustness of the elements. Often, an explicit run has different exigencies than an implicit one, for example, homogeneous mesh is more important for explicit (due to time step), than implicit and positive Jacobi warpage, skewness and aspect ratio are more important for implicit (due to convergence), than explicit. Initially, getting a good mesh quality condition is necessary for a good simulation result.

Hourglass Formulations

Under-integrated elements are very familiar in crash worthiness. In these elements, a reduced number of integration points are used to decrease the computation time. This simplification generates zero energy deformation modes, called hourglass modes.

As shown in Figure 102 for an anti-symmetric deformation of a shell element with one integration point at the center, a zero energy mode can be found for a physical deformation mode. That cannot be true from a physical point of view and must be corrected. For a shell mesh, the two main hourglass modes are in-plane (membrane mode) and out-of-plane (*W* mode).

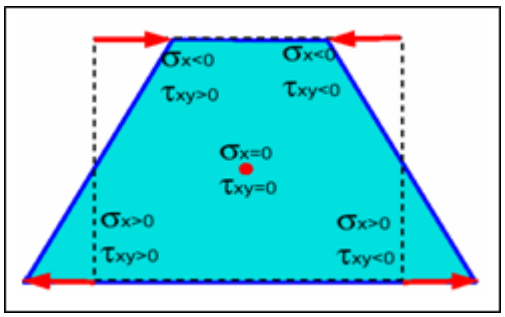


Figure 102: Zero Energy Mode Generation for a Reduced Integrated Shell

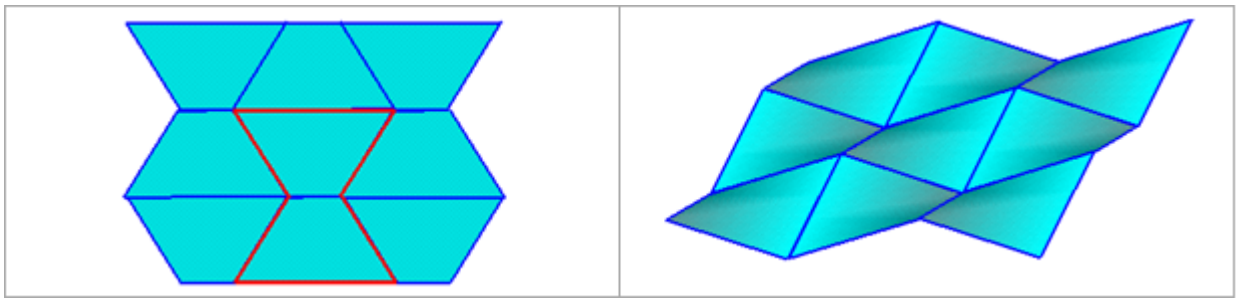


Figure 103: In-plane and Out-of-plane Hourglass Modes

The two correcting methods available are:

- Perturbation or penalty method: Hourglass control by application of anti-hourglass forces to maintain deformation stability of the element (historical method).
- Physical stabilization method: Hourglass control by the correction of internal energy in an analytical way.

For the first method, four formulations are available in Radioss:

- Type 1 (Default option): Only in-plane modes are corrected, valid for shells and solids.
- Type 2: Old formulation no longer recommended.
- Type 3: Plasticity is taken into account in the computation of anti-hourglass forces; in-plane and out-of-plane hourglass modes are corrected; available only for shells.
- Type 4: Like Type 1 with additional higher order terms in the computation of anti-hourglass forces and with out-of-plane hourglass mode correction; available only for shells.

The second method is powerful and can be used when the mesh quality is good. The physical stabilization method is incorporated in the following Radioss elements:

- `QEPH` shell
- `HEPH` solid
- `HSEPH` solid-shell

The choice of hourglass formulation plays an essential role on the quality of results. If the mesh is fine enough, the physical stabilization method can be used. Otherwise, Type 3 and Type 4 formulations are recommended for shell meshes. When the material undergoes elastic-plastic behavior, the Type 3 formulation is much more efficient. The hourglass coefficient, *hm* and *hf* are recommended to be set to 0.1; however, the default value, *hf*=0.01 can be used for *hr*.

Stress-Strain Computation Options (/PROP)

Usually a small stress formulation is used to simulate problems for small deformations. Typical application covers linear elastic studies. For nonlinear problems with elasto-plastic behavior and large deformation, it is recommended to use a large strain and large displacement formulation.

For some problems with very large mesh distortion, a large strain formulation has some limitations. If the time step decreases too much, the CPU time will increase highly and if a brick element reaches a negative volume, the computation will stop, or this element may have to be removed.

With a small strain formulation these limitations disappear; however, it seems surprising to use a small strain formulation in case of large deformation; but this can be a better solution than element deletion. For specific materials, like honeycomb, for which there is no Poisson's effect, the small strain limitation can be corrected just by using the appropriated stress strain curve.

Large Strain Formulation

By default, Radioss uses a large strain, large displacement formulation with explicit time integration. By computing the derivative of shape functions at each cycle, large displacement formulation is obtained. The large strain formulation results from incremental strain computation. Stresses and strains are therefore true stresses and true strains.

Time integration of Cauchy stress (true stress) can be performed as:

$$\sigma_{ij}(t+\delta t) = \sigma_{ij}(t) + \frac{d\sigma_{ij}(t)}{dt} dt \tag{44}$$

The objectivity of stress rate means that the stress tensor follows the rigid body rotation of the material. Stress rate is the function of element average rigid body rotation and of strain rate. Stability of explicit scheme is given by Courant condition:

$$\Delta t < \frac{l}{c} \tag{45}$$

With l being the element characteristic length and c is the sound speed. The time step is computed at each cycle. Large element deformation can provide a large time step decrease. For very large deformations, a negative volume can be reached and then becomes impossible to integrate the stresses over the volume.

Small Strain Formulation

Assuming a constant Jacobian matrix during time and also a constant volume, previous equations degenerate into a small strain formulation. All special variables are then values defined at time t=0 (or at the time the small strain formulation is initiated).

Time step then becomes constant:

$$\Delta t = \frac{l_0}{c} \tag{46}$$

Then the effective negative volume has no effect on the computation (only the initial volume is used), but the small strain assumption is acceptable for some specific material behavior, like honeycomb; so the small displacement formulation is obviously not recommended for crash analysis.

For stress objectivity, the Jacobian matrix time transformation depends upon the element deformation and the element rigid body rotation. Rigid body translation has no effect on the Jacobian matrix. If element deformation is not taken into account, a small strain formulation is obtained. If element rigid body rotation is not taken into account, a small displacement formulation is obtained.

To switch to a small strain formulation after reaching $\Delta T_{\min}$, the following options can be used in the Engine file:

`/DT/BRICK/CST`

ΔT_{sca} $\Delta T_{\min}$

`/DT/SHELL/CST`

ΔT_{sca} $\Delta T_{\min}$

`/DT/SH_3N/CST`

ΔT_{sca} $\Delta T_{\min}$

Note: For some brick material laws, the sound speed is pressure dependent. Therefore, it possible that time step becomes slightly smaller than $\Delta T_{\min}$.

Update Stress and Strains (/PROP)

In Radioss two formulations are available (Figure 104):

- Co-rotational Formulation (CR): The FEM equations of each element are referred to two systems. A fixed or base configuration is used to compute the rigid body motion of the element. Then, the deformed current state is referred to the co-rotated configuration, which is obtained by the rigid body motion of the initial reference.
- Updated Lagrangian Formulation (ULF): The reference is the last known (accepted) solution. It remains fixed over a step and updated at the end of each step.

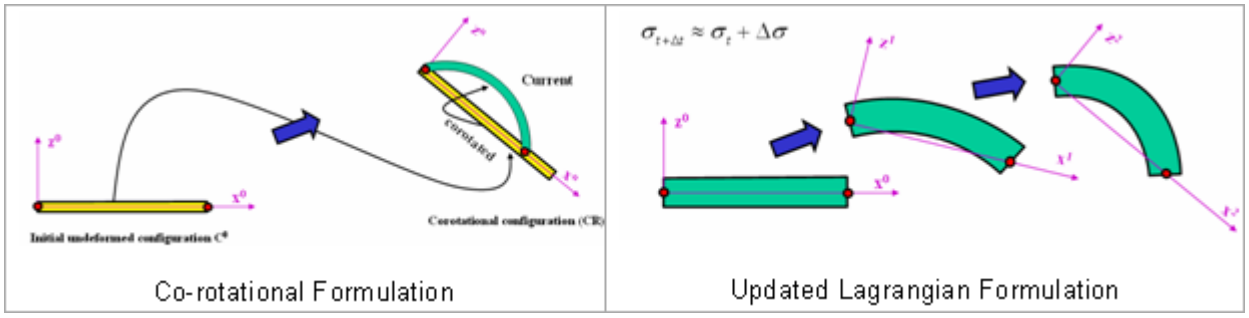


Figure 104: Updating Methods in Radioss

The co-rotational formulation is the most recent of the formulations in geometrically Nonlinear Structural Analysis. It decouples small strain material nonlinearities from geometric nonlinearities and naturally handles the question of frame indifference of anisotropic behavior, due to material nonlinearities. For historical Radioss solid elements, the co-rotational formulation is optional and should be activated before the computations (refer to `/PROP/SOLID`).

Note: An updated Lagrangian formulation can handle incorrect results for solids in large shear deformation, due to the accumulation of updating errors at each cycle. The use of co-rotational formulation is strongly recommended, especially when elements undergo large shear deformation or large rigid body motions.

Radioss Coordinate System

Shell and solid (thick shell) elements are introduced for the coordinate systems.

- Global system ($\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Z}$)
- Natural system (isoparametric frame) (ξ, η, ζ)
- Local element coordinate system ($\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$)

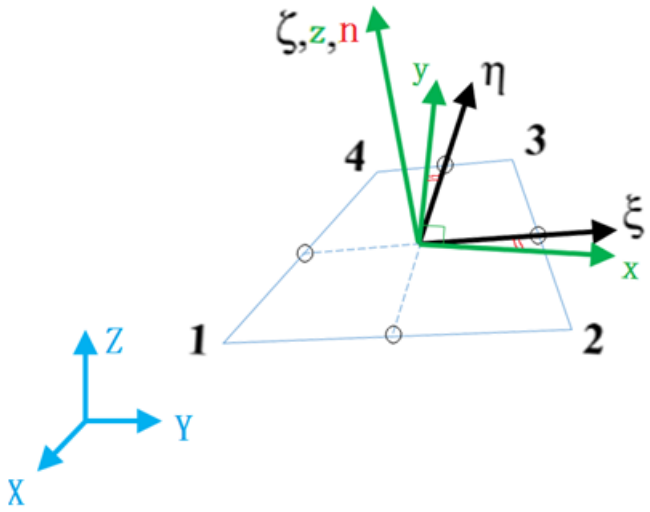


Figure 105: 4-Node Shell Element

($\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Z}$) - Global Cartesian fixed system:

(ξ, η, ζ) - Natural system (non-normalized coordinate system).

- ξ is from middle point of Line 14 to middle point of Line 23.
- η is from middle point of Line 12 to middle point of Line 34.
- Plane (ξ, η) is in the middle surface of shell element and ζ is normal of the middle surface.

($\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$) - Local coordinate system (orthogonal, normalized elemental coordinate system):

- $\mathbf{z}$ is normal of middle surface.
- ($\mathbf{x}$ and $\mathbf{y}$) are in the middle surface
- $\mathbf{x}$ and $\mathbf{y}$ are positioned so that they have same angle between $\mathbf{x}$ and ξ, $\mathbf{y}$ and η

The origin of (ξ, η, ζ) and ($\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$) are the same as it is at the intersection point of middle point line.

3-node Shell Element

($\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Z}$) - Global Cartesian fixed system

(ξ, η, ζ) - Natural system (non-normalized coordinate system).

- ξ is from Node 1 to Node 2.
- η is from Node 1 to Node 3.
- Plane (ξ, η) is in the middle surface of shell element and ζ is normal of the middle surface.

($\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$) - Local coordinate system (orthogonal, normalized elemental coordinate system).

- $\mathbf{z}$ is normal of middle surface.
- $\mathbf{x}$ is from Node 1 to Node 2.
- $\mathbf{y}$ is orthogonal to $\mathbf{x}$ and ($\mathbf{x}$ and $\mathbf{y}$) are in the middle surface.

The origin of (ξ, η, ζ) and ($\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$) are the same as it is at Node 1.

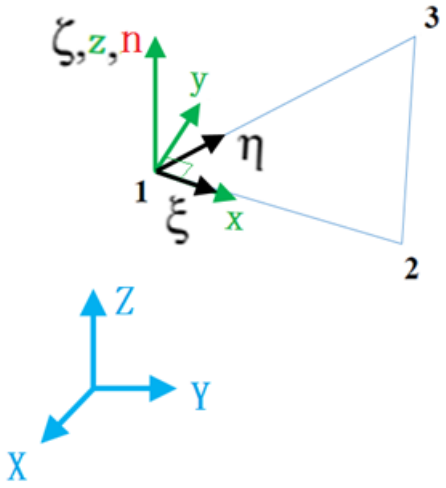


Figure 106:

Solids and Thick Shells (hexa)

- Global system ($\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Z}$)
- Natural system ($\mathbf{r}$, $\mathbf{s}$, and $\mathbf{t}$)
- Local element coordinate system ($\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$)
- Material system

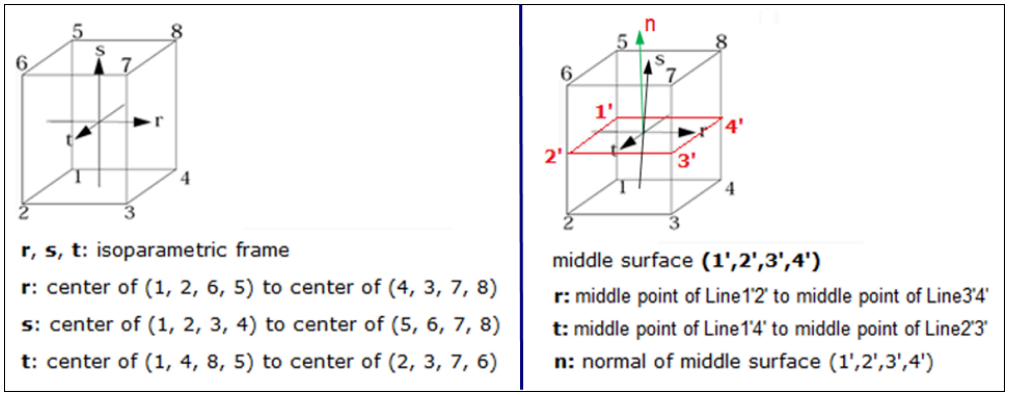


Figure 107: Solids and Thick Shells (hexa)

(**X**, **Y**, and **Z**) - Global Cartesian fixed system

(**r**, **s**, and **t**) - Natural system (non-normalized coordinate system).

- **r** is from the center of surface (1, 2, 6, and 5) to center of surface (4, 3, 7, and 8)
- **s** is from the center of surface (1, 2, 3, and 4) to center of surface (5, 6, 7, and 8)
- **t** is from the center of surface (1, 4, 8, and 5) to center of surface (2, 3, 7, and 6)

Plane (**r**, **t**) is also in the middle surface (1', 2', 3', and 4').

- **r** is also from middle point of Line 1' and 2' to middle point of Line 3' and 4'.
- **t** is also from middle point of Line 1' and 2' to middle point of Line 3' and 4'
- **n** is normal of middle surface (1', 2', 3', and 4')

(**x**, **y**, and **z**) - Local coordinate system (orthogonal, normalized elemental coordinate system).

Local coordinate system in middle surface (1', 2', 3', and 4') is the same as the local coordinate system in middle surface (1, 2, 3, and 4) for shell element. r in solid is the same as ξ in shell element.

Tetra Elements

(**r**, **s**, and **t**) - Natural system (non-normalized coordinate system).

- **r** is from node 4 to node 1
- **s** is from node 4 to node 2
- **t** is from node 4 to node 3

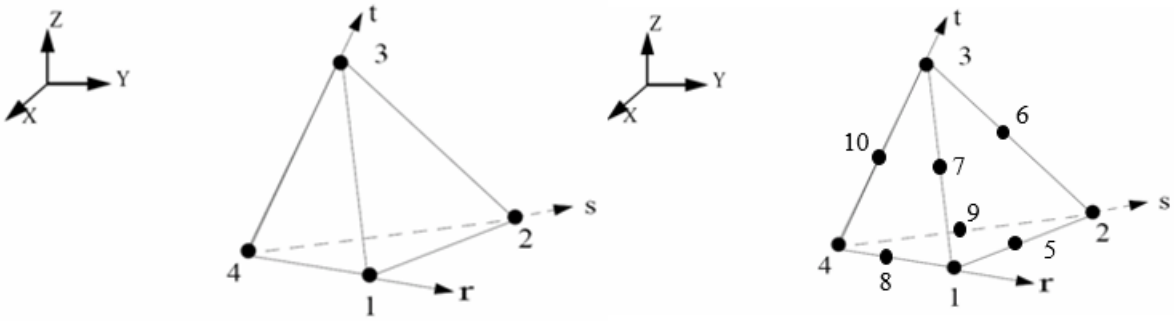


Figure 108: Tetra Elements

Material System

For shell element, anisotropic can be defined with property type 9, 10, 11, 16, 17, and 19, using a material system to describe the anisotropic. Vector $\mathbf{V}$ and angle ϕ requested to define material system. (See `/PROP/TYPE9` Format below). Material direction ($\mathbf{m1}$ and $\mathbf{m2}$) presents the direction of different mechanic characters (Example: E-Modulus, shear Modulus, stress-strain behavior, damage, ...) for anisotropic.

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
V_X		V_Y		V_Z		ϕ			

Use vector $\mathbf{V}$ and angle ϕ material direction 1 ($\mathbf{m1}$) can be defined (along normal n project vector $\mathbf{V}$ to middle surface and get vector $\mathbf{V'}$. Rotate angle ϕ of vector $\mathbf{V'}$ then get material direction $\mathbf{m1}$. Material direction $\mathbf{m1}$ is normally the fiber direction. For composite, a different ply could be defined with one vector $\mathbf{V}$ and different ϕ.

Second material axis $\mathbf{m2}$ is perpendicular to $\mathbf{m1}$ (except for `/PROP/TYPE16`, angle between $\mathbf{m1}$ and $\mathbf{m2}$ could be defined with α).

- $\mathbf{n}$ is normal of shell middle surface

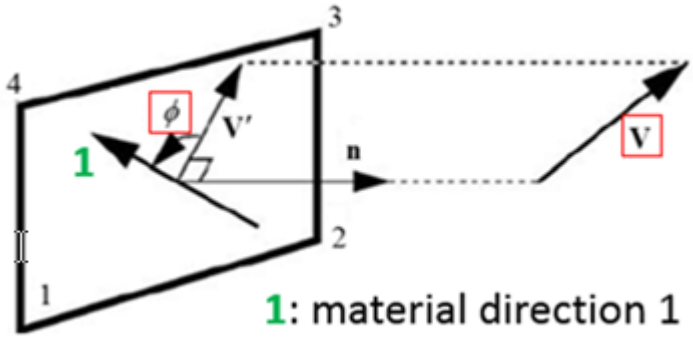


Figure 109:

In `/PROP/TYPE11`, I_{orth} can determine the relative orientation of the material system.

- I_{orth}=0 (default): The orthotropic direction follows the local co-rotational reference. The angle between $\mathbf{x}$ and $\mathbf{m1}$ is constant during the simulation. Internal force is computed in local frame and then rotated to the global system. This formulation is more accurate, if a large rotation occurs.
- I_{orth}=1: The orthotropic direction is attached to the local isoparametric frame. The angle between ξ and $\mathbf{m1}$ is updated during the simulation. It is updated in a way that projection of vector $\mathbf{m1}$ to ξ and $\boldsymbol{\eta}$ is always constant during the simulation. Pure shear could not well descripted with this method, but traction could well be described. So, this method usually is used to defined fiber direction in airbag.

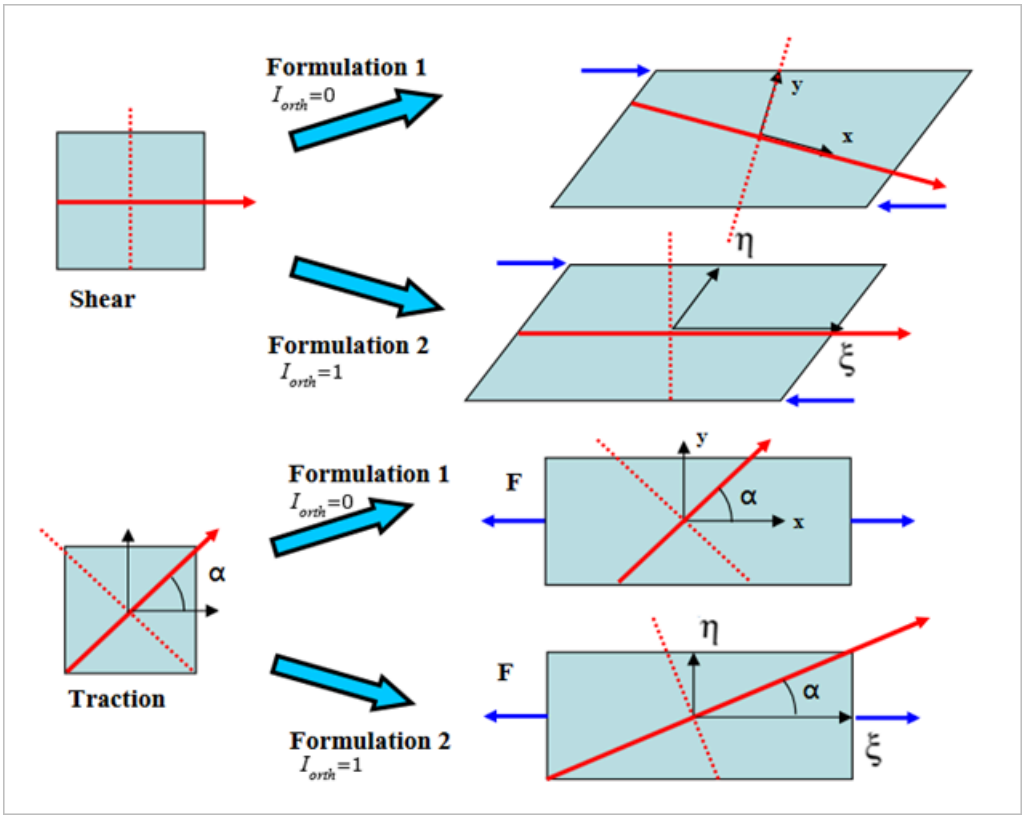


Figure 110:

For brick and thick shell elements, use the same process to determine the material direction and orthotropic direction (I_{orth}), like shell elements. In `/PROP/TYPE6` (`SOL_ORTH`) use the option I_P to determine the reference plane.

***I_P*=0**	Use *Skew_ID*
***I_P*=1**	Plane ($\mathbf{r}$,$\mathbf{s}$) + angle ϕ
***I_P*=2**	Plane ($\mathbf{s}$,$\mathbf{t}$) + angle ϕ
***I_P*=3**	Plane ($\mathbf{t}$,$\mathbf{r}$) + angle ϕ
***I_P*=11**	Plane ($\mathbf{r}$,$\mathbf{s}$) + orthogonal projection of references vector $\mathbf{V}$ on plane ($\mathbf{r}$,$\mathbf{s}$)
***I_P*=12**	Plane ($\mathbf{s}$,$\mathbf{t}$) + orthogonal projection of references vector $\mathbf{V}$ on plane ($\mathbf{s}$,$\mathbf{t}$)
***I_P*=13**	Plane ($\mathbf{t}$,$\mathbf{r}$) + orthogonal projection of references vector $\mathbf{V}$ on plane ($\mathbf{t}$,$\mathbf{r}$)

- Definition is the same for any I_{solid} and I_{frame} parameters

- In the simplest case, material directions **m1**, **m2** and **m3** directly with skew (I_P=0) are recommended
- For I_P > 0 the isoparametric, non-orthogonal system **r**, **s**, and **t**, is used to determine material directions.
  - First material axis **m1** is determined according to I_P.
  - For example, for I_P=1
  - The first material axis **m1** and **m2** is orthogonal and rotated by angle ϕ in the (**r'** and **s'**) plane.
  - The third material axis **m3** is normal of **m1** and **m2** plane (vector product of **m1** and **m2**).

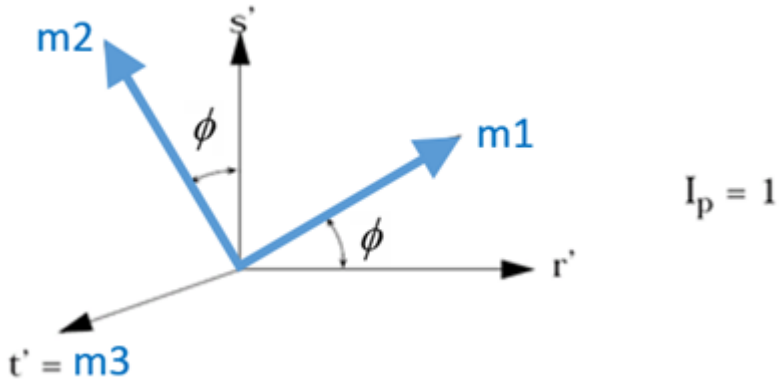


Figure 111:

  - The (**r'**, **s'**, and **t'**) system is orthogonal and it is generated from non-orthogonal isoparametric system (**r**, **s**, and **t**).

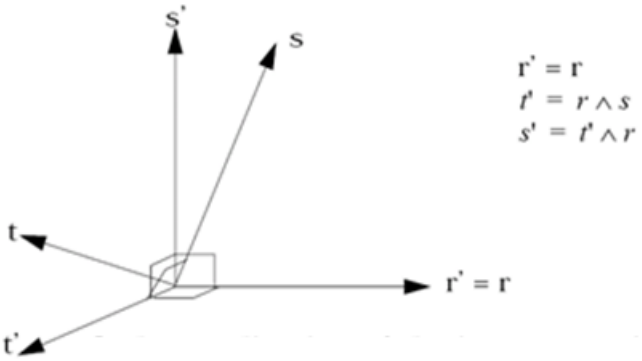


Figure 112: Orthogonalization of the Isoparametric System

Depending on I_{solid} and I_{frame} parameters, three definitions of systems are used in Radioss for hexa elements (8-noded bricks) using `/PROP/TYPE6` (`SOl_ORTH`).

Global System Definition

- Definition 1: Solids, I_{solid}=1, 2, 17 + I_{frame}=0, 1 (default)

  Global system is used, no element system (non-co-rotational formulation) available.

Element System Definition

- Definition 2: Solids, I_{solid}=1, 2, 17 + I_{frame}=2

  Element system (with I_{frame}=2 co-rotational formulation) is used.

- Definition 3: Solids, I_{solid}=14 or 24

  I_{frame} parameter has no effect. Element system is used, and co-rotational formulation defined already.

> **Note:** If the co-rotational formulation is used, the orthotropic frame (defined with I_{orth}) keeps the same orientation with respect to the local (co-rotating) frame and is therefore also co-rotating.

Modeling Tools

Skew and Frame (/SKEW & /FRAME)

Skews and frames are used to define local directions.

These directions can be used to apply:

- Boundary conditions
- Concentrated load
- Fixed velocity

To define:

- Rigid link orientation
- Rigid body added inertia frame
- General spring reference frame
- Beam type spring initial reference frame
- Nodal time history output frame

Two reference definitions are available in Radioss:

Skew reference

It is a projection reference to define the local quantities with respect to the global reference. In fact, the origin of the skew remains at the initial position during the motion even though a moving skew is defined. In this case, a simple projection matrix is used to compute the kinematic quantities in the reference.
In Figure 113, imposed velocity is applied in Y direction. In `/IMPVEL`, skew is used. Then the imposed velocity is computed in the Y axis of global coordinate system and then projected onto the Y’ axis of local skew reference.

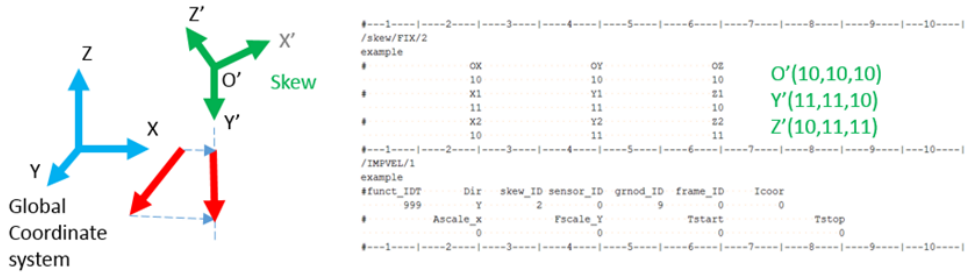


Figure 113: Skew Example

Frame reference

It is a mobile or fixed reference. The quantities are computed with respect to the origin of the frame which may be in motion or not depending on the kind of reference frame. For a moving reference frame, the position and the orientation of the reference vary in time during the motion. The origin of the frame defined by a node position is tied to the node.
Frame measures relative motion whereas skew measures global motion and projects it to skew. Only few options use frame like imposed velocity, `TH/NODE` and others use skew.

In Figure 114, rotational velocity is applied around Z axis. In /LOAD/CENTRI, frame is used. Then rotational velocity is around Z' axis of frame reference not in Z axis of global coordinate system anymore.

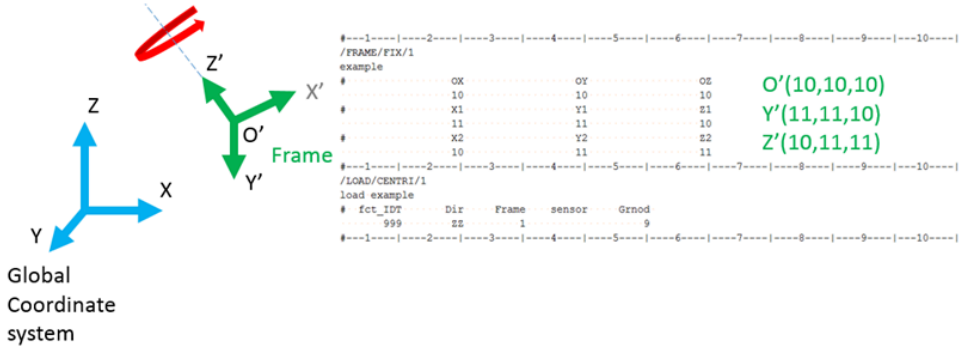


Figure 114: Frame Example

Sections (/SECT)

A section is used to measure the force, moment, and energy which are passing through a set of elements and nodes. The section displacements can also be saved to a file and used as imposed displacements in a smaller cut section model.

To define the section, the following are required:

- A group of nodes and groups of elements. The nodes and elements can be selected by a user in /SECT. They can also be automatically selected by defining a section cut using /SECT with *frame_ID*, /SECT/PARAL, or /SECT/CIRCLE.
- A local output system defined by selecting 3 points
- A reference point to compute forces and moment

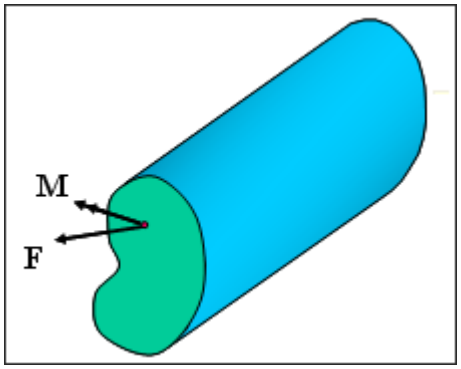


Figure 115: Definition of a Section for an Oriented Solid

Section Cutting Plane

In /SECT, the cutting plane is infinite and is defined by a group of elements and a group of nodes. The elements and nodes can be user-defined and should be along one row, if possible.

It is recommended to select all the nodes on one side of the element section.

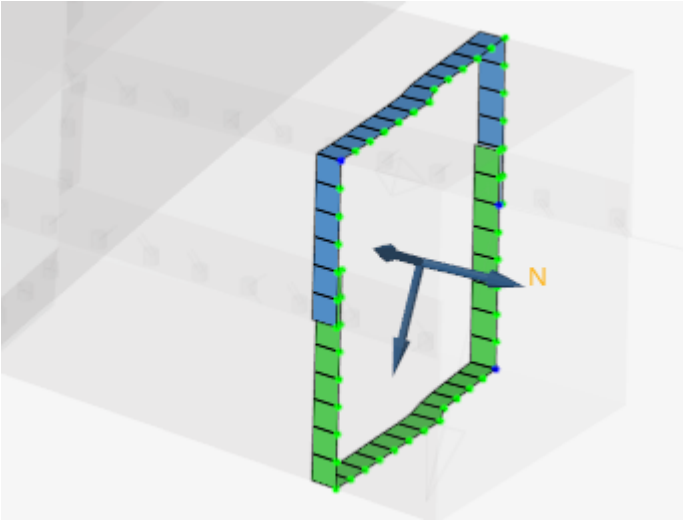


Figure 116: Elements and Nodes Selected Manually

Alternatively, the section cut can be defined using a local system. The local system /FRAME/MOV is defined by picking three nodes on the section cut. In this case, the element groups contain all the elements in the area of the section, and the node group in /SECT is not used. The nodes and elements on the xy plane of the local system are automatically selected for the section calculation. The cutting plane is the xy plane of the local system.

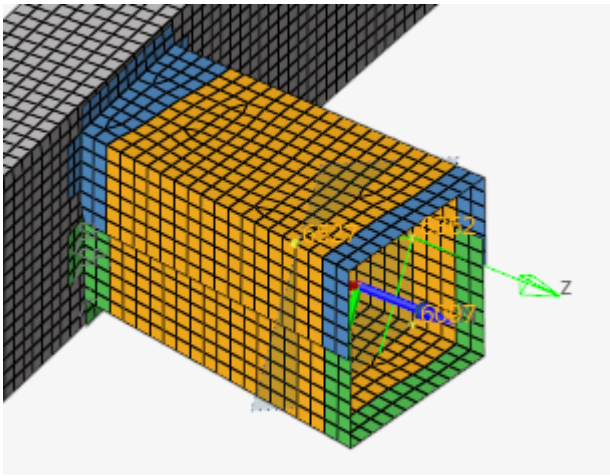


Figure 117: Element Group (in orange) defined for a section cut defined using a local system

The nodes and elements that lie on the section cut are then output in the Starter output file.

```
      NUMBER OF NODES. . . . . . . . . .        40
       NODES:
    6062      6064      6055      6074      6076      5895      6078      6173      6166      6136
    6174      6181      6182      6219      6210      6220      6227      6228      6359      6358
```

```
...

      NUMBER OF SHELL ELEMENTS . . . . .        43
 SHELL      N1      N2      N3      N4
    5819         0         1         1         0
    5820         0         1         1         0
    5826         0         0         1         0
    5839         0         1         1         0
...
```

In /SECT/PARAL, the cutting parallelogram is defined by three points *M*, *M1*, and *M2* which define the two lines of the parallelogram. The element groups in the defined section should contain all the elements in the area of the parallelogram section. The nodes and elements used in the section calculation are automatically selected and output in the Starter output file, as shown previously for `/SECT` when a local system is used.

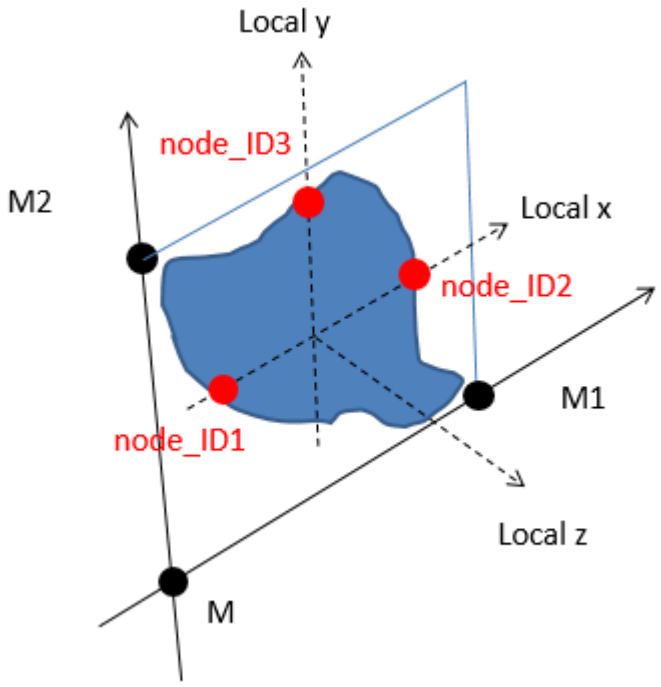


Figure 118: /SECT/PARAL Definition

In /SECT/CIRCLE, the cutting disk is defined by a point *M*, radius, and a normal vector. The element groups in the defined section should contain all the elements in the area of the circle section. The nodes and elements used in the section calculation are automatically selected and output in the Starter output file, as shown previously.

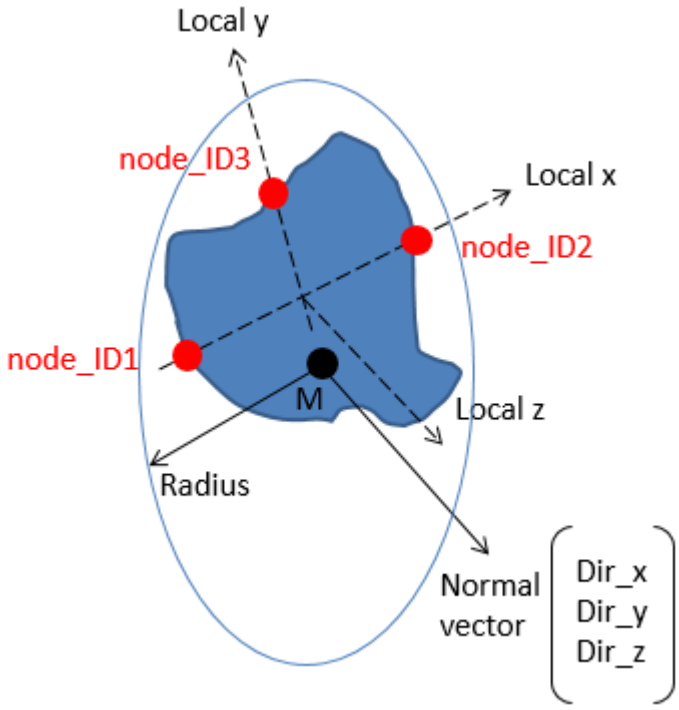


Figure 119: /SECT/CIRCLE Definition

All section types can cut solid, shell, truss, beam, and spring elements. Contact interfaces can also be selected.

When using `/SECT` with *frame_ID*, `/SECT/PARAL`, or `/SECT/CIRCLE` the nodes used for the section calculation will on the +z side of the selected elements. Since these nodes are automatically defined, the element groups can be defined by part and; thus, will not need to be redefined, if the part is re-meshed.

Local System of Cutting Plane

A local system must be defined to compute the force and moment from the section.

In all three section types (`/SECT`, `/SECT/PARAL`, and `/SECT/CIRCLE`), three nodes must be selected to define the local system used for the section output. These three nodes should be nodes on the section plane so their position is updated when the section moves. The nodes *node_ID*$_1$, *node_ID*$_2$, and *node_ID*$_3$ define the local system as:

- Nodes *node_ID*$_1$ and *node_ID*$_2$ define the local x-axis of the section.
- Nodes *node_ID*$_1$, *node_ID*$_2$, and *node_ID*$_3$ define the local plane xy of the section.
- The local y-axis is defined by projecting *node_ID*$_3$ perpendicular to the local x-axis.
- The intersection of the local x- and y-axis is then the origin of the system.
- The section normal is the local z-axis which is perpendicular to the xy plane.

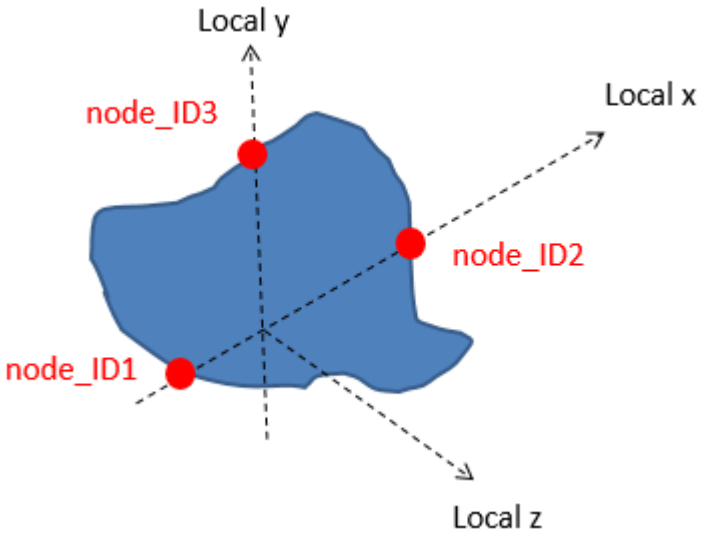


Figure 120: Section local system defined using nodes

When creating a `/SECT` in HyperMesh, these three nodes are automatically selected for the user. If they are manually selected, it is recommended to select nodes that belong to the group of nodes used in the section calculation. This will allow the local system to move as the section deforms.

Alternatively, in `/SECT` if the 3 nodes are not defined and instead a *frame_ID* is defined then the xy plane of the `/FRAME/MOV` is used as the local system. When defining the `/FRAME/MOV` it is recommended to use nodes that belong to the group of nodes used in the section calculation.

Force and Moment Computation

The section force is the sum of nodal force coming from the selected elements.

$$F = \sum f_i \tag{47}$$

The normal Force FN of the section is the component of section force in the normal direction and tangent force FT of the section is the component of section force in cutting plane.

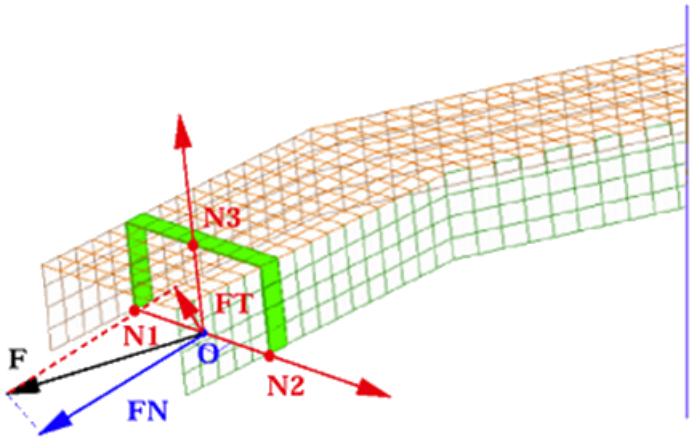


Figure 121: Normal section force and tangent section force in section

The section moment is the sum of nodal forces coming from the selected element multiplied by the distance to the local coordinate system center.

$$M = \sum m_i + \sum ON_i \times f_i \quad (48)$$

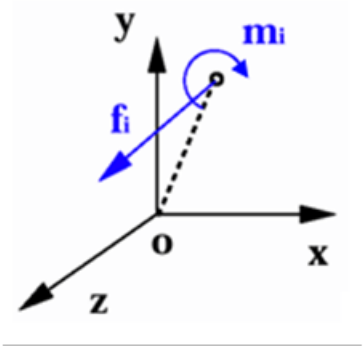


Figure 122: Resultant of force and moment for a node

Moments are computed with respect to the section center defined by the parameter I_{frame}. Options for the output system center include local system origin, geometric center, center of gravity, and global system origin. The output system can be either the local system or the global system.

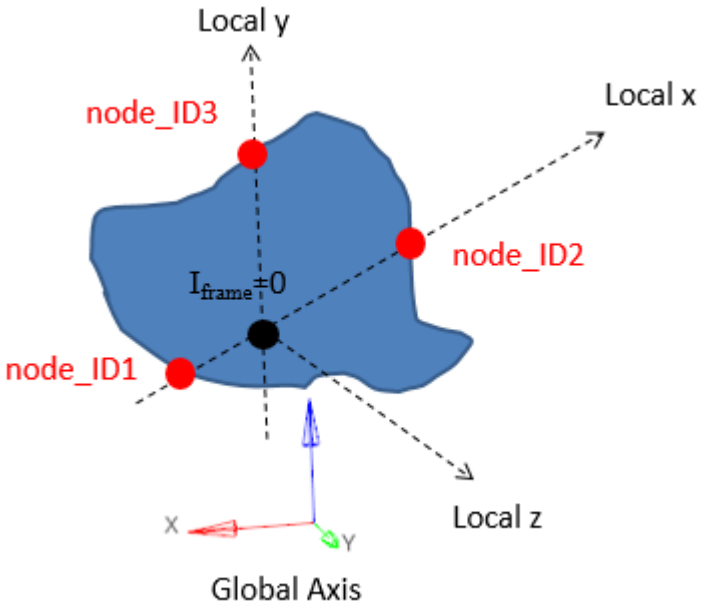


Figure 123: I_{frame}=0 Local system and origin used for section output

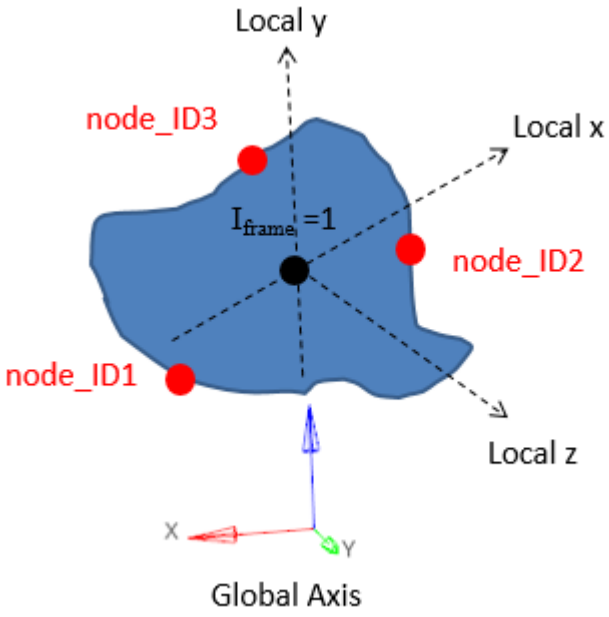


Figure 124: I_{frame}=1 Local system with the origin as the geometric center of the section

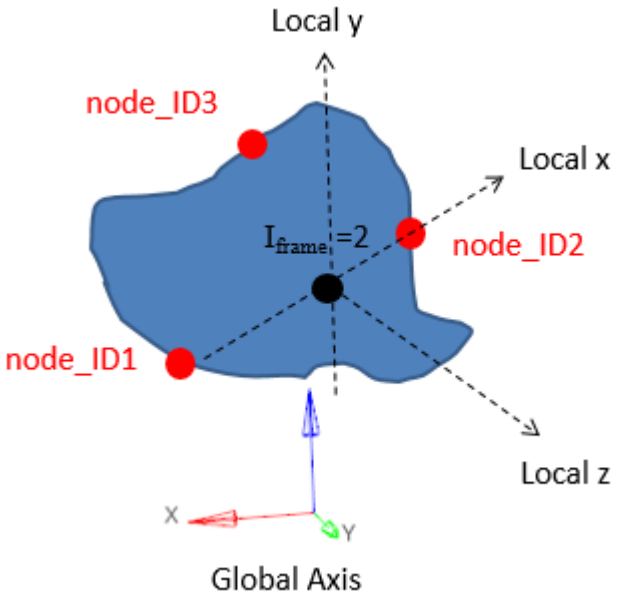


Figure 125: I_{frame}=2 Local system with the origin as the center of gravity of the section

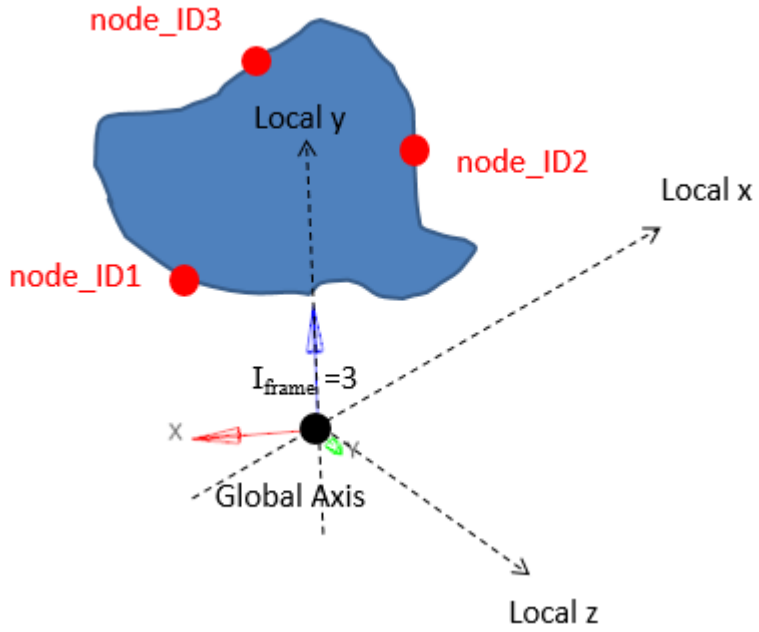


Figure 126: I_{frame}=3 Local system with the global origin as the center

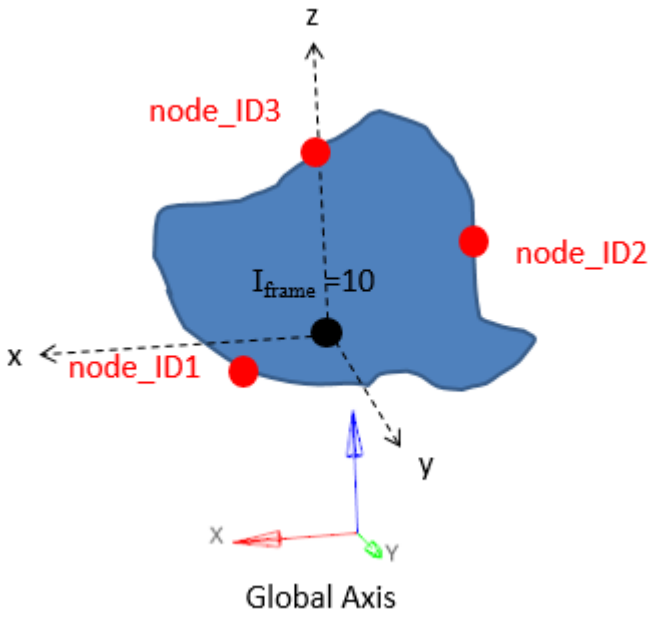


Figure 127: I_{frame}=10 Global system with the center as the local system origin

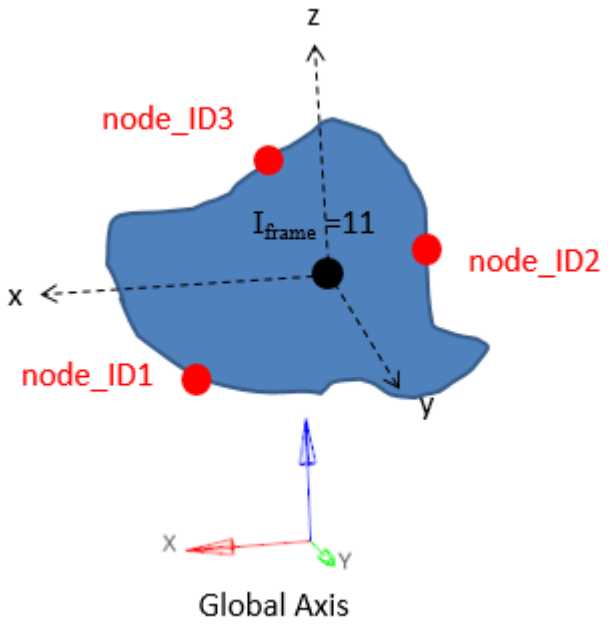


Figure 128: I_{frame}=11 Global system with the origin as the geometric center of the section

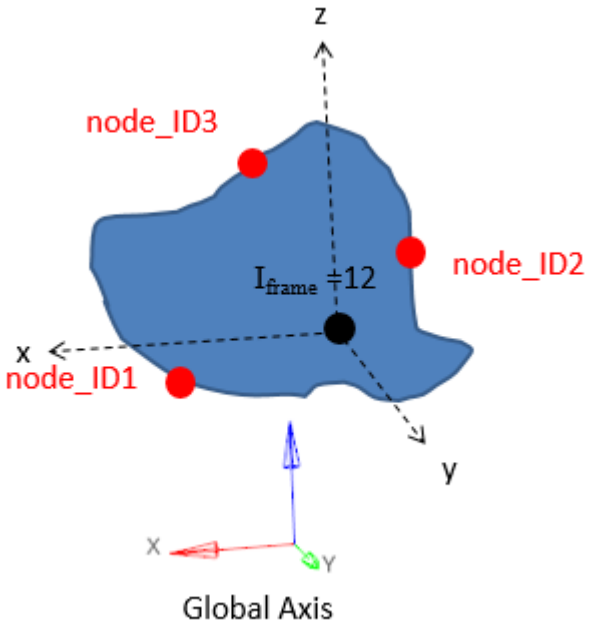


Figure 129: I_{frame}=12 Global system with the origin as the center of gravity of the section

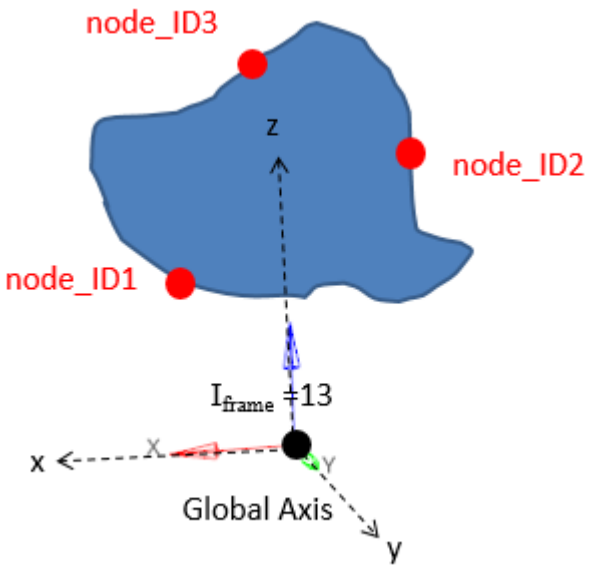


Figure 130: I_{frame}=13 Global system with the global origin as the center

Output of Section

Two types of section output are available.

First is the time history output `/TH/SECTIO` consisting of the sum of the force and moments acting on the section. The output can be in the global system or the local system with the most commons output being the variable groups, GLOBAL, LOCAL, and CENTER.

The second type of output is displacements and optional force and moment for every node in the section written to the section output data `SC01` file. This file can then be read as an imposed displacement applied to a second cut section model. RD-E: 5400 Cut Methodology is an example of the cut section methodology that can be used. In this case, a full model is run with I_{SAVE}=1 or 2 to save displacement and optionally the resultant section forces/moments in the file `file_nameSC01`.

Next, the second cut model (submodel) is created with a section defined using the option I_{SAVE}=100 or 101 to read the displacement of the nodes the section file. The section's local system defined by the three nodes *node_ID*$_1$, *node_ID*$_2$, and *node_ID*$_3$ or *frame_ID* must be the same one used when saving the data (I_{SAVE}=1 or 2) and reading the data (I_{SAVE}=100 or 101).

If I_{SAVE}=2 is used in the full model and I_{SAVE}=101 is used in the cut model, then Radioss outputs the difference between resultant section forces and moments in full and cut model in the section (`/TH/SECTIO`).

To improve solution times and decrease memory needed, it is recommended to set I_{SAVE} =0 if no cut modeling methodology is needed.

Filter

When reading and applying the displacements to a cut section model, the displacement can be filtered using an exponential moving average filter. For example, the filtered displacement in the x-direction would be:

$$x_f = \alpha x(t) + (1-\alpha)x(t-dt) \tag{49}$$

Where,

x_f	Filtered displacement
α	Exponential moving average filtering constant
$x(t)$	Unfiltered displacement at the current time
$x(t-dt)$	Displacement at the previous time step

Recommendations are:

$\alpha = \frac{2\pi dt}{T}$ for filtering -3dB

$\alpha = \frac{2\pi dt}{\sqrt{3T}}$ for filtering -6dB

Where,

T	Filtering period
dt	Model time step

The filtering period $T = 10\Delta t$ is often used making $\alpha = \frac{2\pi}{10} = 0.62832$ for a -3dB filter.

Materials

Different material tests could result in different material mechanic character.

The typical material test for metal is tensile test. With this test strain-stress curve, yield point, necking point and failure point of material could be observed.

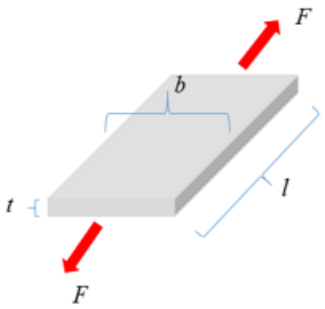


Figure 131: Force (F) and Length (l) are Measured

Engineer strain-stress curve could be generated by:

$$\sigma_e = \frac{F}{S_0} \quad (50)$$

$$\varepsilon_e = \frac{\Delta l}{l_0} \quad (51)$$

Where,

S_0	Section area in the initial state
l_0	Initial length

In this Force-elongation curve or engineer stress-strain curve, three points are important.

1. Yield point: where material begin to yield. Before yield you can assume material is in elastic state (the Young's modulus *E* could be measured) and after yield, material plastic strain which is non-reversible.
   - Some material in this test will first reach the upper yield point (R_{eH}) and then drop to the lower yield point (R_{eL}). In engineer stress-strain curve, lower yield stress (conservative value) could be taken.
   - Some material can not easily find yield point. Take the stress of 0.1 or 0.2% plastic strain as yield stress.
2. Necking point: where the material reaches the maximum stress in engineer stress-strain curve. After this point, the material begins to soften.
3. Failure point: where material failed.

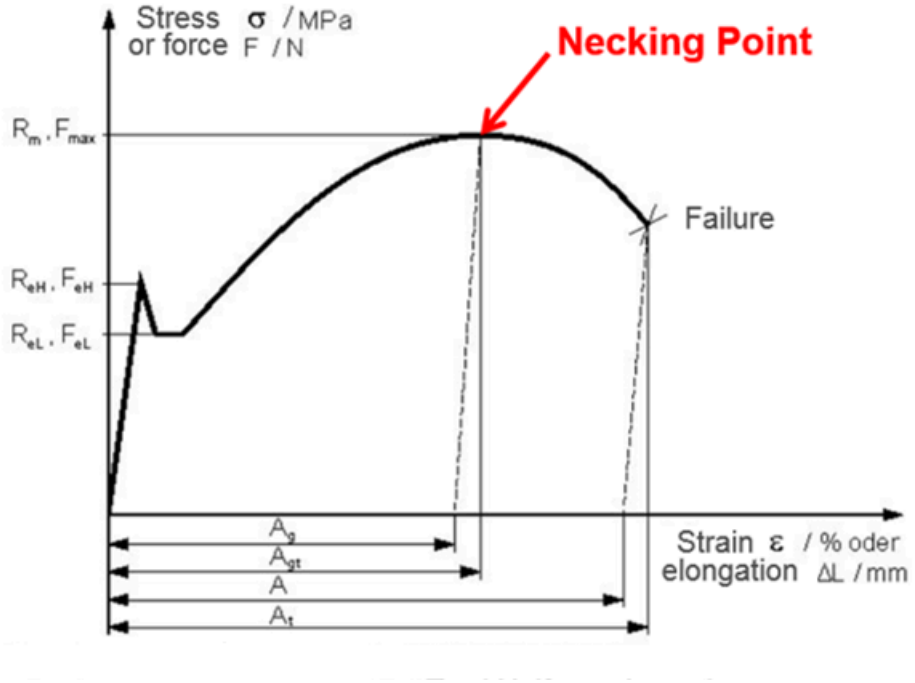


Figure 132:

R_m	Maximum resistance
F_{max}	Maximum force
R_{eH}	Upper yield level
R_{eL}	Lower yield level
A_g	Uniform elongation
A_{gt}	Total uniform elongation
A_t	Total failure strain

True stress-strain curve which is requested in most materials in Radioss, except in `LAW2`, where both engineer stress-strain and true stress-strain are possible to input material data.

In Figure 133, find engineer stress-strain curve (blue) by using:

$$\sigma_{tr} = \sigma_e \exp(\varepsilon_{tr}) \quad (52)$$

$$\varepsilon_{tr} = \ln(1+\varepsilon_e) \quad (53)$$

The result is true stress-strain curve (red). Plastic true stress-strain curve is shown in green, which plastic strain begin from 0. This green plastic true stress-strain curve is what you need, as in `LAW36`, `LAW60`, `LAW63`, and so on.

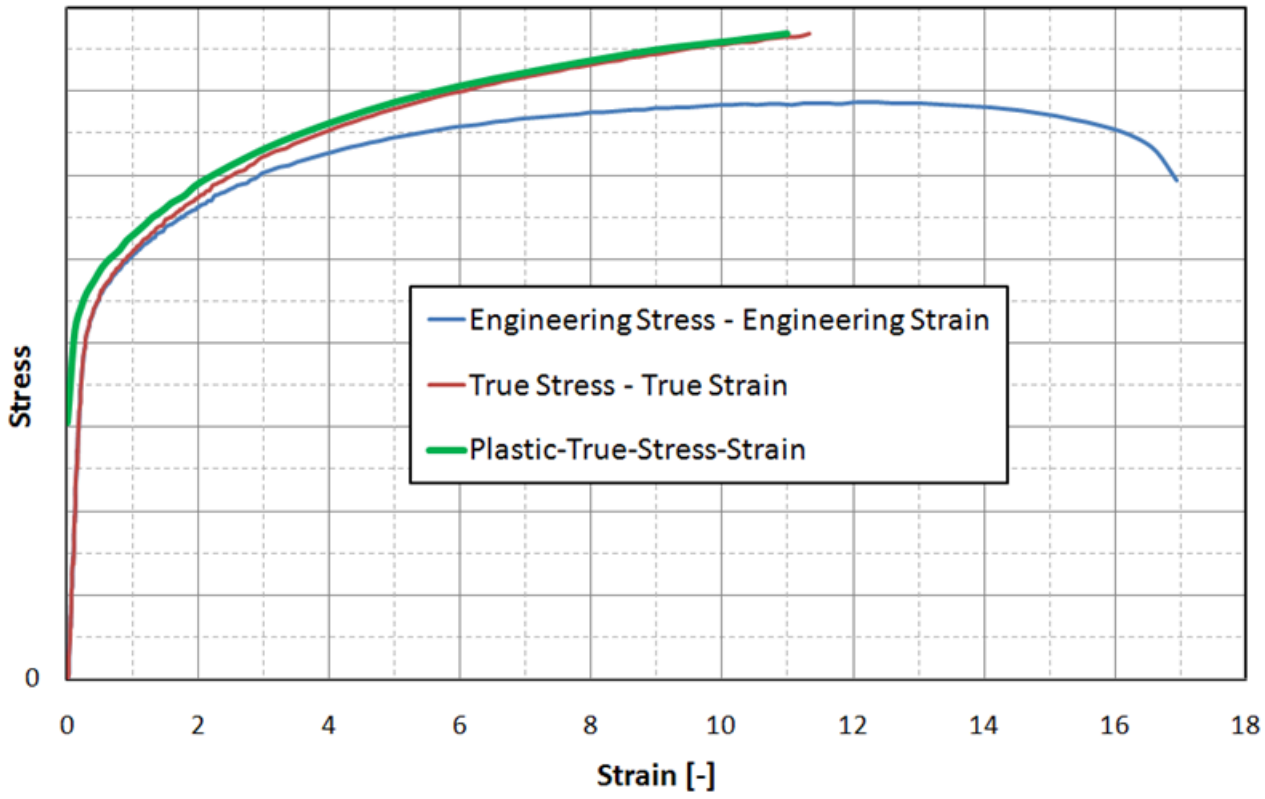


Figure 133:

The true stress-strain curve is valid until the necking point of the material. After the necking point, the material curve has to be defined manually for hardening. Using a different material law, Radioss will extrapolation the true stress-strain curve to 100%.

- Linear extrapolation: If stress-strain curve is as function input (`LAW36`), then stress-strain curve is linearly extrapolated with a slope defined by the last two points of the curve. It is recommended that the list of abscissa value be increased to a value greater than the previous abscissa value.
- Johnson-Cook: After necking point, Johnson-Cook hardening is one of the most commonly used to extrapolate the true stress-strain curve.

  $$\sigma_y = a + b\varepsilon_p{}^n \qquad (54)$$

  However, it may overestimate strain hardening for automotive steel, In this case, combination of swift-voce hardening is more accurate.
- Swift and Voce: After necking point, use one of the following equations to extrapolate the true stress-strain curve.

  Swift model

  $$\sigma_y = A\left(\bar{\varepsilon}_p + \varepsilon_0\right)^n$$

  A and n are positive.

  Voce model

  $$\sigma_y = k_0 + Q\left[1 - \exp\left(-B\bar{\varepsilon}_p\right)\right]$$

k_0, Q and B are positive.

Combination of Swift and Voce model (LAW84 and LAW87)

$$\sigma_y = \alpha \underbrace{\left[A\left(\bar{\varepsilon}_p + \varepsilon_0\right)^n\right]}_{Swift\,hardening} + (1-\alpha)\underbrace{\left\{k_0 + Q\left[1 - \exp\left(-B\bar{\varepsilon}_p\right)\right]\right\}}_{Voce\,hardening}$$

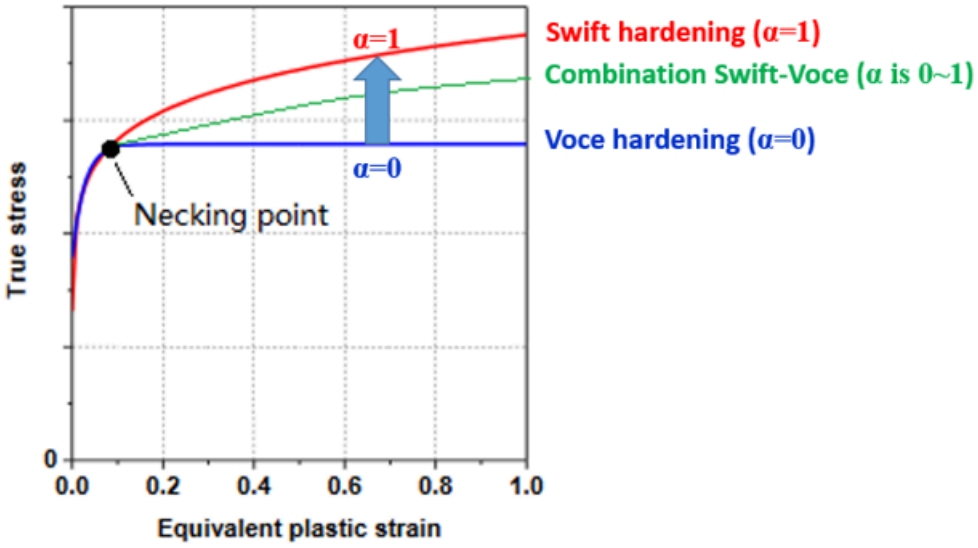


Figure 134:

Here, α is weight of Swift hardening and Voce hardening. Here one Compose script as example to fit the Swift hardening parameters A, ε_0, n and Voce hardening parameters k_0, Q, B with input stress-strain curve.

See Also

/MAT/LAW84 (Starter)

/MAT/LAW87 (BARLAT2000) (Starter)

Hyperelastic Materials

Hyperelastic materials are used to model materials that respond elastically under very large strains. These materials normally show a nonlinear elastic, incompressible stress strain response which returns to its initial state when unloaded.

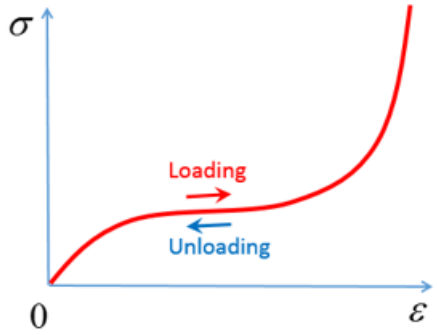


Figure 135:

Hyperelastic materials are a specific case of Cauchy material where the stress only depends on the current deformation. Meyer's kinetic theory describes how these materials consist of flexible chain like structure which rotates and straightens when deformed. This led to the theory that a strain energy density function could be written as a function of the deformation. [3] The strain energy density can then be differentiated to obtain the stress strain behavior of the material.

The most common hyperelastic materials are elastomers or rubbers. Some properties of elastomers include. [4]

- The material is nearly ideally elastic and deformation is reversible with stress being a function only of current strain and independent of history or rate of loading, if deformed at constant temperature or adiabatically.
- The material incompressible and strongly resists volume changes. The bulk modulus which is a ratio of volume change to hydrostatic component of stress is comparable to that of metals.
- The material is very weak in shear with a shear modulus 10^{-5} times small than most metals.
- The material is isotropic its stress-strain response is independent of material orientation.

Element Property Recommendations

When using hyperelastic material laws, there are some recommended element property settings. When using solid elements, it is always better to mesh with 8 node `/BRICK` elements, if possible. If not, then `/TETRA4` or `/TETRA10` elements can be used. Recommended `/PROP/SOLID` for 8 nodes brick are, I_{smstr}=10, I_{cpre}=1, with I_{solid}=24. If hourglassing occurs, then I_{solid}=17 with I_{frame}=2 can be used.

3. Treloar, L. R. G. "The elasticity and related properties of rubbers." *Reports on progress in physics* 36, no. 7 (1973): 755
4. Bower, Allan F. *Applied mechanics of solids*. CRC press, 2009

Ogden Materials

Hyperelastic materials can be used to model the isotropic, nonlinear elastic behavior of rubber, polymers, and similar materials. These materials are nearly incompressible in their behavior and can be stretched to very large strains.

In Radioss, material laws LAW42, LAW62, LAW69, LAW82, and LAW88 utilize different strain energy density functions of the Ogden material model [5] to model hyperelastic materials. [6]

Material Definition

Stretch (also called *stretch ratio*) λ is the ratio of final length and initial length. It is used for materials with large deformations. For a cube in tension:

$\varepsilon_1 = \frac{\Delta l}{l_{01}}$ Engineering strain (also called *nominal strain*) in direction 1

$\lambda_1 = \frac{l_1}{l_{01}}$ Stretch in direction 1

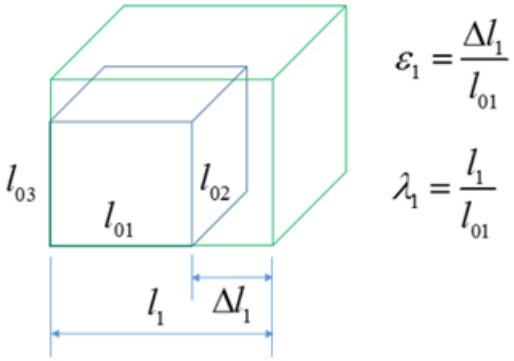


Figure 136:

Thus, strain and stretch are related as:

$$\lambda = 1 + \varepsilon \tag{55}$$

Principal stretch λ_i can be used to describe the volumetric deformation by calculating the relative volume, J, computed as:

$$J = \frac{V}{V_0} = \frac{l_1 \cdot l_2 \cdot l_3}{l_{01} \cdot l_{02} \cdot l_{03}} = \lambda_1 \cdot \lambda_2 \cdot \lambda_3 \tag{56}$$

For an incompressible material the volume should not change and thus J=1 and thus, the stretch can be calculated for the following material tests.

- Uniaxial test:

  $\lambda_1 = \lambda$ and $\lambda_2 2 = \lambda_3 2 = \frac{1}{\lambda}$

- Biaxial test:

$\lambda_1 = \lambda_2 = \lambda$ and $\lambda_3 = \lambda^{-2}$

- Planar (shear) test:

  $\lambda_1 = \lambda;\ \lambda_3 = 1$ and $\lambda_2 = \lambda^{-1}$

/MAT/LAW42 (Ogden)

This material model defines a hyperelastic, viscous, and incompressible material specified using the Ogden, Neo-Hookean, or Mooney-Rivlin material models. This law is generally used to model incompressible rubbers, polymers, foams, and elastomers. This material can be used with shell and solid elements.

LAW42 uses the following strain energy density representation of the Ogden material model.

$$W(\lambda_1, \lambda_2, \lambda_3) = \sum_{p=1}^{5} \frac{\mu_p}{\alpha_p}\left(\bar{\lambda}_1^{\alpha_p} + \bar{\lambda}_2^{\alpha_p} + \bar{\lambda}_3^{\alpha_p} - 3\right) + \frac{K}{2}(J-1)^2 \quad (57)$$

Where,

W	Strain energy density
λ_i	i^{th} principal engineering stretch
J	Relative volume defined as: $J = \lambda_1 \cdot \lambda_2 \cdot \lambda_3 = \frac{\rho_0}{\rho}$
$\bar{\lambda}_i = J^{-\frac{1}{3}}\lambda_i$	Deviatoric stretch
α_p **and** μ_p	Material constants coefficient pairs. Up to 5 material constant pairs can be defined.

The initial shear modulus μ and bulk modulus (K) are given by:

$$\mu = \frac{\sum_{p=1}^{5} \mu_p \cdot \alpha_p}{2} \quad (58)$$

and

$$K = \mu \cdot \frac{2(1+\nu)}{3(1-2\nu)} \quad (59)$$

Where, ν is the Poisson's ratio and is only used for computing the bulk modulus.

Material Parameters

Parameters α_p and μ_p must be chosen so that initial shear modulus is:

$$\mu = \frac{\sum_{p=1}^{5} \mu_p \cdot \alpha_p}{2} > 0 \quad (60)$$

For material stability, it is required that each material constant pair

$$\mu_p \cdot \alpha_p > 0 \quad (61)$$

In general, the Ogden model can be used for strains up to 700%. The number of terms material pairs, α_p and μ_p, needed depends on the range of experimental data that is fit and curve fitting accuracy desired. In practice, 3 material pairs fit most data. If the material pairs are not known for a particular material, then a curve fit of uniaxial test data can be done in Radioss using LAW69 or via separate fitting software.

Neo-Hookean Model

A simple case of the Ogden material model is the Neo-Hooken model represented using the following equation for the strain energy density function:

$$W = C_{10}(I_1 - 3) \quad (62)$$

Where,

I_1	The first invariants of the right Cauchy-Green Tensor
C_{10}	Material constant

This representation can be derived from the LAW42 Ogden strain energy density function when:

$\mu_1 = 2 \cdot C_{10}$; $\alpha_1 = 2$, and $\mu_2 = \alpha_2 = 0$

The Neo-Hookean model is a simple model that is typically only accurate for strains less than 20%.

Mooney-Rivlin Model

A slightly more complex case of the LAW42 Ogden material model is the Mooney-Rivlin model, which can be represented using the following equation for the strain energy density function:

$$W = C_{10}(I_1 - 3) + C_{01}(I_2 - 3) \quad (63)$$

Where,

I_1 **and** I_2	The first and second invariants of the right Cauchy-Green Tensor
C_{10} **and** C_{01}	Material constants

This representation can be derived from the LAW42 Ogden strain energy density function when:

$\mu_1 = 2 \cdot C_{10}$, $\mu_2 = -2 \cdot C_{01}$, $\alpha_1 = 2$, and $\alpha_2 = -2$

Mooney-Rivlin constants are available from a material supplier or testing company. If they are not available, then a curve fit of uniaxial test data can be done in Radioss using LAW69 or via separate fitting software. The Mooney-Rivlin material law is accurate for strains up to 100%.

Poisson's Ratio and Material Incompressibility

If a material is truly incompressible, then $v = 0.5$. However, in practice is not possible to use because that would result in an infinite bulk modulus, an infinite speed of sound, and thus an infinitely small solid element Time Step.

$$K = \mu \cdot \frac{2(1+v)}{3(1-2v)} = \mu \cdot \frac{2(1+v)}{3(1-2*0.5)} = \infty \qquad (64)$$

The effect of different Poisson's ratio input can be seen in Figure 137. The largest difference in the results is at higher amounts of strain. The results will match the test data better when $v = 0.4997$ but this results in a time step that is 4 times lower than $v = 0.495$. Thus, to balance the computation time and accuracy it is recommended to use $v = 0.495$ for incompressible rubber material.

The effect of Poisson's ratio and Bulk modulus are similar in other Ogden material law.

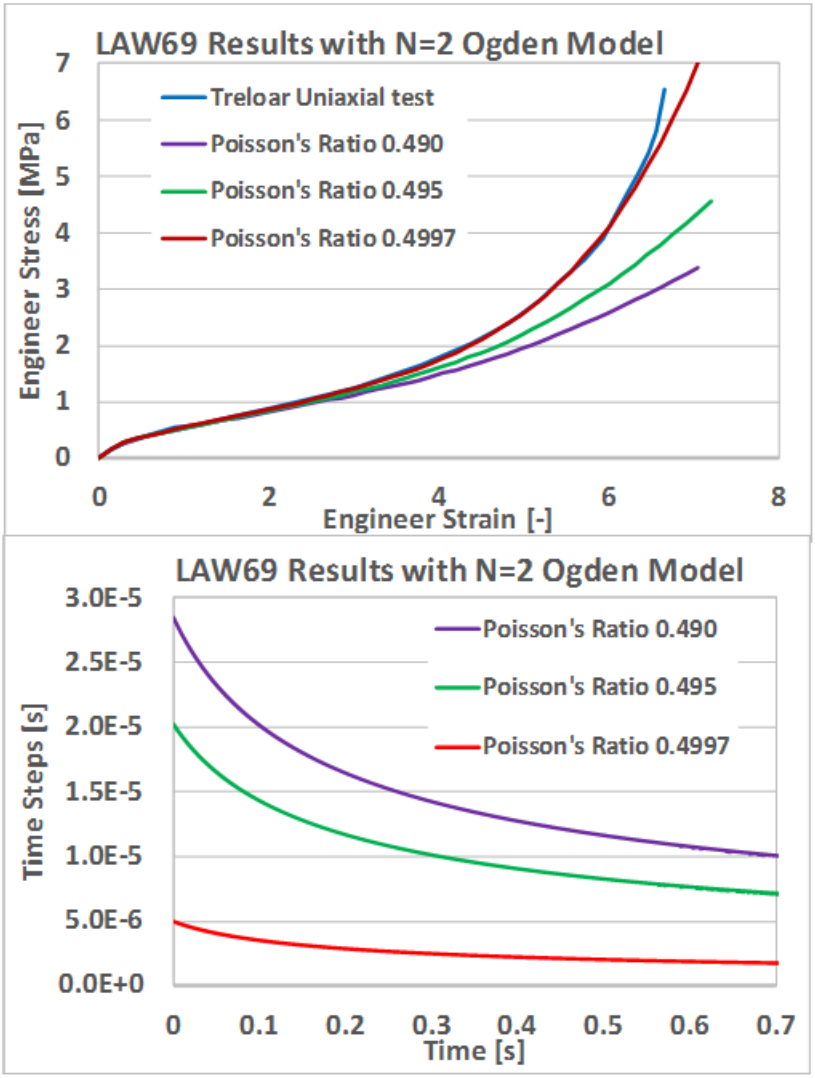


Figure 137:

Higher values of the Poisson's ratio may lead to a very small time step or divergence for explicit simulations.

In LAW42, material incompressibility is provided by using a penalty approach, which calculates the pressure proportional to a change in density.

$$P = K \cdot Fscale_{blk} \cdot \mathrm{f}_{blk}(J) \cdot (J - 1) \tag{65}$$

Where,

K	Bulk modulus
$J = \frac{V}{V_0} = \frac{m\rho_0}{m\rho} = \frac{\rho_0}{\rho}$	Relative volume which simplifies to relative density if mass is constant
$\mathrm{f}_{blk}(J)$	Bulk coefficient scale factor versus relative volume function
$Fscale_{blk}$	Abscissa scale factor for function $\mathrm{f}_{blk}(J)$

The bulk modulus (K) of hyperelastic materials is generally a very high value which provides the needed pressure-resistance to maintain the incompressibility condition (J =1). But if a material starts to compress (J < 1) then the bulk modulus can be increased by including the *fct_ID$_{blk}$* input function which allows the scaling of the bulk coefficient value as a function of J. By default, there is no scaling and; thus, if the function identifier is zero and the value of the bulk scaling function is equal to 1. It is advisable to output (`/ANIM/BRICK/DENS`) and review the material density of LAW42 components to make sure that the density variation is small, that is the value of J is close to 1 and the material is incompressible.

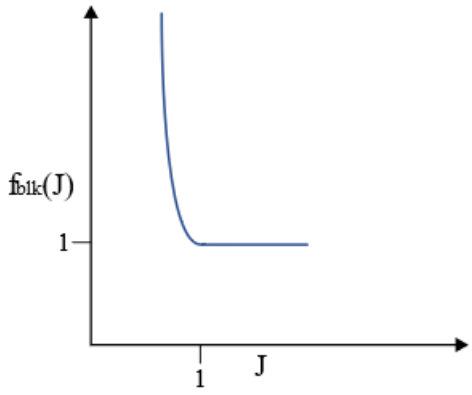


Figure 138: Bulk Modulus Scale Factor Function fct_ID$_{blk}$

Viscous (Rate) Effects

Viscous (rate) effects are modeled in LAW42 using a Maxwell model, which can be described in a simplified manner as a system of η springs with stiffness' G_i and dampers η_i:

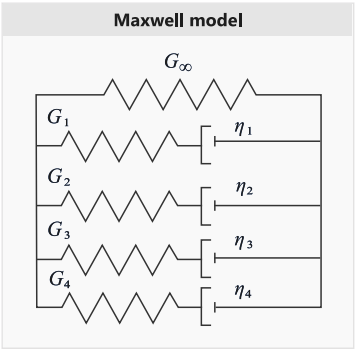


Figure 139: Maxwell Model

The Maxwell model is represented using Prony series inputs (G_i, τ_i). The hyperelastic initial shear modulus μ is the same as the long-term shear modulus G_∞ in the Maxwell model, and τ_i is the relaxation time:

$$\tau_i = \frac{\eta_i}{G_i} \tag{66}$$

The G_i and τ_i values must be positive.

/MAT/LAW62 (VISC_HYP)

A hyper visco-elastic material law in Radioss that can be used to model polymers and elastomers.

The hyperelastic behavior in this material law is defined using the following strain energy density function:

$$W(\lambda_1, \lambda_2, \lambda_3) = \sum_{i=1}^{N} \frac{2\mu_i}{\alpha_i^2}\left(\lambda_1^{\alpha_i} + \lambda_2^{\alpha_i} + \lambda_3^{\alpha_i} - 3 + \frac{1}{\beta}(J^{-\alpha_i\beta} - 1)\right) \tag{67}$$

Where,

W	Strain energy density
λ_i	i_{th} principal stretch
J	Relative volume defined in Equation 67
$\beta = \frac{\nu}{(1-2\nu)}$	
α_i **and** μ_i	Material constants coefficient pairs. Up to 5 material constant pairs can be defined.

Poisson's ratio must be $0 < \nu < 0.5$. This law can be used to model compressible or sometimes called *hyperfoam materials* by defining a low Poisson's ratio value.

Note: The μ_i material coefficients are different, but can be converted using:

$$\mu_i^{LAW62} = \frac{\mu_i^{LAW42} \cdot \alpha_i^{LAW42}}{2} \tag{68}$$

Viscous (Rate) Effects

Viscous (rate) effects are modeled in LAW62 using a Maxwell model which can be described in a simplified manner as a system of n springs with stiffness' G_i and dampers η_i .

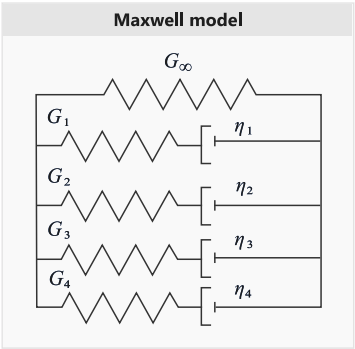


Figure 140: Maxwell Model

The Maxwell model is represented using a Prony series with inputs. The initial shear modulus is:

$$G_0 = \sum_{i=1}^{N} \mu_i \tag{69}$$

The sum of μ_i should be greater than 0.

$$G_0 = G_\infty + \sum_i G_i \tag{70}$$

The stiffness ratio is:

$$\gamma_\infty = \frac{G_\infty}{G_0} = 1 - \sum_i \gamma_i \tag{71}$$

$$\gamma_i = \frac{G_i}{G_0} \tag{72}$$

With,

$$\gamma_i \in [0,\ 1],\ \sum_i \gamma_i < 1 \tag{73}$$

and the ground shear modulus

$$G_0 = G_\infty + \sum_i G_i \tag{74}$$

The relative time, τ_i must be positive:

$$\tau_i = \frac{\eta_i}{G_i} \quad (75)$$

Note: When viscosity is included, the shear modulus in LAW62 is the initial shear modulus $G_0 = \sum_{i=1}^{N} \mu_i$ which includes viscosity, but in LAW42 the shear modulus is the long-term shear modulus, which does not include viscosity $G_\infty = \frac{\sum_{p=1}^{5} \mu_p \cdot \alpha_p}{2}$.

/MAT/LAW69

This law, like `/MAT/LAW42 (OGDEN)` defines a hyperelastic and incompressible material specified using the Ogden or Mooney-Rivlin material models. Unlike LAW42, where the material parameters are input, this law computes the material parameters using test data from a uniaxial engineering stress-strain curve.

This material can be used with shell and solid elements.

The strain energy density formulation used depends on the *law_ID*:

- *law_ID* = 1 (Ogden law):

$$W(\lambda_1, \lambda_2, \lambda_3) = \sum_{p=1}^{5} \frac{\mu_p}{\alpha_p}\left(\bar{\lambda}_1^{\alpha_p} + \bar{\lambda}_2^{\alpha_p} + \bar{\lambda}_3^{\alpha_p} - 3\right) + \frac{K}{2}(J-1)^2 \quad (76)$$

- *law_ID* = 2 (Mooney-Rivlin law):

$$W = C_{10}(I_1 - 3) + C_{01}(I_2 - 3) \quad (77)$$

Material Parameters

After reading the stress-strain curve (*fct_ID*$_1$), Radioss calculates the corresponding material parameter pairs using a nonlinear least-square fitting algorithm. For the classic Ogden law, (*law_ID*=1), the calculated material parameter pairs are μ_p and α_p where the value of *p* is defined via the *N_pair* input. The maximum value is *N_pair*=5 with a default value of 2. Usually no more than *N_pair*=3 is needed for a good fit.

For the Mooney-Rivlin law (*law_ID* =2), the material parameter C_{10} and C_{01} are calculated remembering that μ_p and α_p for the LAW42 Ogden law can be calculated using this conversion:

$\mu_1 = 2 \cdot C_{10}$, $\mu_2 = -2 \cdot C_{01}$, $\alpha_1 = 2$, and $\alpha_2 = -2$

The minimum test data input should be a uniaxial tension engineering stress strain curve. If uniaxial compression data is available, the engineering strain should increase monotonically from a negative value in compression to a positive value in tension. In compression, the engineering strain should not be less than -1.0 since -100% strain is physically not possible.

To improve the quality of the nonlinear least square fit, it is recommended that:

- The experimental data curve represents a smooth monotonically increasing function with uniform distribution of abscissa points. The number of data points in the experimental data curve should be greater than the number of parameter pairs (*N_pair*).
- The engineering strain is negative in compression and positive in tension. For compression test data, the engineering strain should be greater than -1.0 (100% compression maximum) but tension only stress strain data can also be used.
- If *N_pair* ≥ 3, then the test data should cover at least 100% of the tensile strain and/or 50% of the compressive strain.
- *N_pair* should not be set to a very large value to avoid instabilities in the fitting procedure.

This material law is stable when $\mu_p\alpha_p > 0$ (with p=1,…5) is satisfied for parameter pairs for all loading conditions. By default, Radioss tries to fit the curve by accounting for these conditions (I_{check}=2). If a proper fit cannot be found, then Radioss uses a weaker condition (I_{check}=1:), which ensures that the initial shear hyperelastic modulus (μ) is positive.

To determine how well the calculated material parameters represent the input test data, the Radioss Starter outputs an "averaged error of fitting" value which is recommended to not exceed 10%. For visual comparison, the stress-strain curve calculated from the strain energy density and calculated material parameters is also output by the Radioss Starter.

Due to the friction involved in a uniaxial compression test, it is usually more accurate to take equal biaxial tension test data and convert it to uniaxial compressive data using these formulas [7] which are valid for incompressible materials.

$$\varepsilon_c = \frac{1}{(\varepsilon_b + 1)^2} - 1 \tag{78}$$

$$\sigma_c = \sigma_b(1 + \varepsilon_b)^3 \tag{79}$$

Where,

ε_c	Uniaxial engineering compressive stain
ε_b	Equal biaxial engineering tension strain
σ_c	Uniaxial engineering compressive stress
σ_b	Equal biaxial engineering tension stress

Material Incompressibility

Material LAW69 uses the same method to maintain incompressibility as LAW42. For additional information, refer to Poisson's Ratio and Material Incompressibility in LAW42.

Viscous (Rate) Effects

`/VISC/PRONY` must be used with LAW69 to include viscous effects. Alternatively, LAW69 could be used to extract the Ogden or Mooney-Rivlin parameters and then those parameters can be used in LAW42 with viscosity added.

/MAT/LAW82

This material model defines a hyperelastic, and incompressible material specified using the Ogden, Neo-Hookean, or Mooney-Rivlin material models. This law is generally used to model incompressible rubbers, polymers, foams, and elastomers.

This material can be used with shell and solid elements. As compared to LAW42 or LAW62, this law uses a different Ogden strain energy density formulation given in Equation 80. The LAW82 strain energy density formulation matches what is used in some other finite elements solver's hyperelastic model and; thus, the material parameters for this form of the Ogden strain energy density are sometimes available from material suppliers or other sources.

$$W = \sum_{i=1}^{N} \frac{2\mu_i}{\alpha_i^2}\left(\bar{\lambda}_1^{\alpha_i} + \bar{\lambda}_2^{\alpha_i} + \bar{\lambda}_3^{\alpha_i} - 3\right) + \sum_{i=1}^{N} \frac{1}{D_i}(J-1)^{2i} \tag{80}$$

Where,

W	Strain energy density
N	Number of material constants α_i, μ_i and D_i
$\bar{\lambda}_i = J^{-\frac{1}{3}}\lambda_i$	Deviatoric stretch
J	Relative volume as defined in Equation 56

The initial shear modulus:

$$\mu = \sum_{i=1}^{N} \mu_i \tag{81}$$

The Bulk Modulus is calculated as $K = \frac{2}{D_1}$ based on these rules:

- If $\nu = 0$, D_1 should be entered.
- If $\nu \neq 0$, D_1 input is ignored and will be recalculated and output in the Starter output using:

$$D_1 = \frac{3(1-2\nu)}{\mu(1+\nu)} \tag{82}$$

- If $\nu = 0$ and D_1=0, a default value of $\nu = 0.475$ is used and D_1 is calculated using Equation 82

Neo-Hookean Model

Like LAW42, LAW82 can also be simplified to a Neo-Hooken model by using:

$\mu_1 = 2 \cdot C_{10}$, $\alpha_1 = 2$ and $\mu_2 = \alpha_2 = 0$

Mooney-Rivlin Model

Like LAW42, LAW82 can also be simplified to a Mooney-Rivlin model by using:

$\mu_1 = 2 \cdot C_{10}$, $\mu_2 = 2 \cdot C_{01}$, $\alpha_1 = 2$ and $\alpha_2 = -2$

Viscous (Rate) Effects

`/VISC/PRONY` must be used with LAW82 to include viscous effects.

Drücker Condition Stability Check

In LAW42 and LAW69, the Drücker stability is automatically calculated by the Radioss Starter.

The Drücker stability condition checks if the change in the Kirchhoff stress corresponding to the infinitesimal change in the logarithmic strain (true strain) satisfies the following inequality.

$$\sum_{i=1}^{3} d\tau_i d\varepsilon_i > 0 \tag{83}$$

Where, i=1,2,3 principal direction.

With the change in logarithmic strain

$$d\varepsilon_i = \frac{d\lambda_i}{\lambda_i} \tag{84}$$

$d\tau_i = J \cdot d\sigma_i$ The change of Kirchhoff stress

$d\boldsymbol{\tau} = \mathbf{D} : d\boldsymbol{\varepsilon}$ Relationship between Kirchhoff stress and logarithmic strain

The Drücker stability condition will be:

$$\sum d\boldsymbol{\varepsilon} : \mathbf{D} : d\boldsymbol{\varepsilon} > 0 \tag{85}$$

Here $\mathbf{D}$ is tangential material stiffness matrix and it is also the slope of stress-strain curve:

$$\mathbf{D} = \begin{bmatrix} D_{11} & D_{12} & D_{13} \\ D_{21} & D_{22} & D_{23} \\ D_{31} & D_{32} & D_{33} \end{bmatrix} \tag{86}$$

For a stable material, it requests tangential material stiffness $\mathbf{D}$ be positive (slope of stress-strain curve is positive). The tangential material matrix $\mathbf{D}$ is positive if following conditions satisfied:

$$I_1 = tr(\mathbf{D}) = D_{11} + D_{22} + D_{33} > 0 \tag{87}$$

$$I_2 = D_{11}D_{22} + D_{22}D_{33} + D_{33}D_{11} - D_{23}^2 - D_{13}^2 - D_{12}^2 > 0 \tag{88}$$

$$I_3 = \det(\mathbf{D}) > 0 \tag{89}$$

The Kirchhoff stress for Ogden model is:

$$\tau_i = \sum_p \mu_p \left[\bar{\lambda}_i^{\alpha_p} - \frac{1}{3}\left(\bar{\lambda}_1^{\alpha_p} + \bar{\lambda}_2^{\alpha_p} + \bar{\lambda}_3^{\alpha_p} \right) \right] + K(J^2 - J) \tag{90}$$

Since $D_{ij} = \frac{\partial \tau_i}{\partial \lambda_j}$, then for a given Ogden parameter α_p, μ_p with conditions $I_1 > 0$, $I_2 > 0$ and $I_3 > 0$, the strain range of material in Drücker stability could then be calculated.

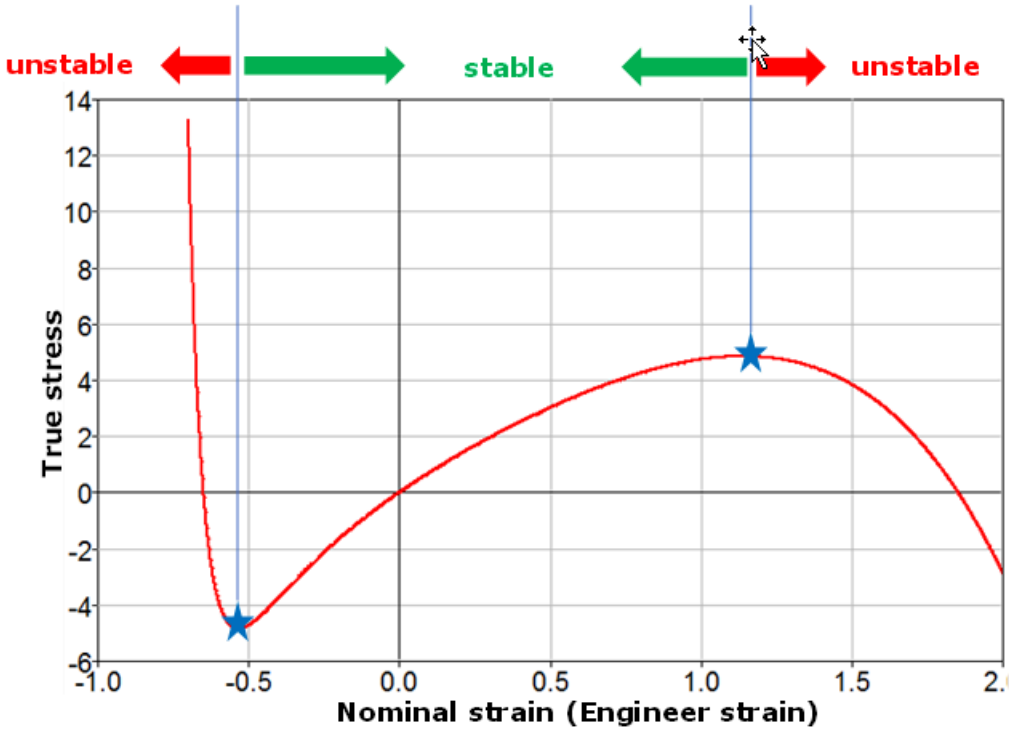


Figure 141:

The Drücker stability criterion calculates the strain range where the material model will remain stable given a set of material parameters. This stability check cannot be made for every deformation but instead is commonly used to check material stability under uniaxial, biaxial and planar strain loading.

For example, using the following Ogden parameters:

$\mu_1 = 13.99077258830 \quad \alpha_1 = 3.788192935039$

$\mu_2 = -9.13454532223 \quad \alpha_2 = -7.17617341059$

$\mu_3 = 8.904655103235 \quad \alpha_3 = -7.27028137148$

Then the Drücker stability will be automatically checked in Radioss Starter, and results printed in Starter output file `*0.out`. This shows the strain at which instability can occur for the given Ogden parameters:

```
CHECK THE DRUCKER PRAGER STABILITY CONDITIONS
     ---------------------------------------------
     MATERIAL LAW = OGDEN (LAW42)
     MATERIAL NUMBER =         1
       TEST TYPE = UNIXIAL
        COMPRESSION:    UNSTABLE AT A NOMINAL STRAIN LESS THAN -0.3880000000000
        TENSION:        UNSTABLE AT A NOMINAL STRAIN LARGER THAN  0.9709999999999
       TEST TYPE = BIAXIAL
        COMPRESSION:    UNSTABLE AT A NOMINAL STRAIN LESS THAN -0.2880000000000
        TENSION:        UNSTABLE AT A NOMINAL STRAIN LARGER THAN  0.2780000000000
       TEST TYPE = PLANAR (SHEAR)
        COMPRESSION:    UNSTABLE AT A NOMINAL STRAIN LESS THAN -0.3680000000000
        TENSION:        UNSTABLE AT A NOMINAL STRAIN LARGER THAN  0.5829999999999
```

Note: For a Neo-Hookean material with $C_{10} > 0$ (or $\mu_1 > 0$), the material is always stable and thus no critical value is found by the Drücker stability check.

For a Mooney-Rivlin material, the Drücker stability should be checked since C_{01} or μ_2 could be negative, which leads material instability.

/MAT/LAW88

This law utilizes a tabulated uniaxial tension and compression engineering stress and strain test data at different strain rates to model incompressible materials. It is only compatible with solid elements.

The material is based on the following Ogden's strain energy density function but does not require curve fitting to extract material constants like most other hyperelastic material models. [8]

$$W = \underbrace{\sum_{i=1}^{3}\sum_{j=1}^{m}\frac{\mu_j}{\alpha_j}\left(\bar{\lambda}_i^{\alpha_j} - 1\right)}_{deviatoric\,part} + \underbrace{K\left(J - 1 - \ln J\right)}_{spherical\,part} \tag{91}$$

Instead, this law determines the Ogden function directly from the uniaxial engineering stress strain curve tabulated data.

Unlike other Ogden material laws, the Bulk Modulus must be input from either test data or extracted from Starter output of the LAW69 Ogden curve fit. When comparing results between LAW42 or LAW69 to LAW88, the same bulk modulus must be used.

Unloading Behavior

Unloading can be represented using an unloading function or by providing hysteresis and shape factor inputs to a damage model based on energy.

If using the damage model, the loading curves are used for both loading and unloading and the unloading stress tensor is reduced by:

$$\boldsymbol{\sigma} = \left(1 - D\right)\boldsymbol{\sigma} \tag{92}$$

with

$$D = \left(1 - Hys\right)\left(1 - \left(\frac{W_{cur}}{W_{\max}}\right)^{Shape}\right) \tag{93}$$

Where,

W_{cur}	Current energy
$W_{\max}$	Maximum energy corresponding to the quasi-static behavior
Hys **and** $Shape$	Input by user

If an unloading curve is provided, these options are available:

Tension

Loading and Unloading

= 0

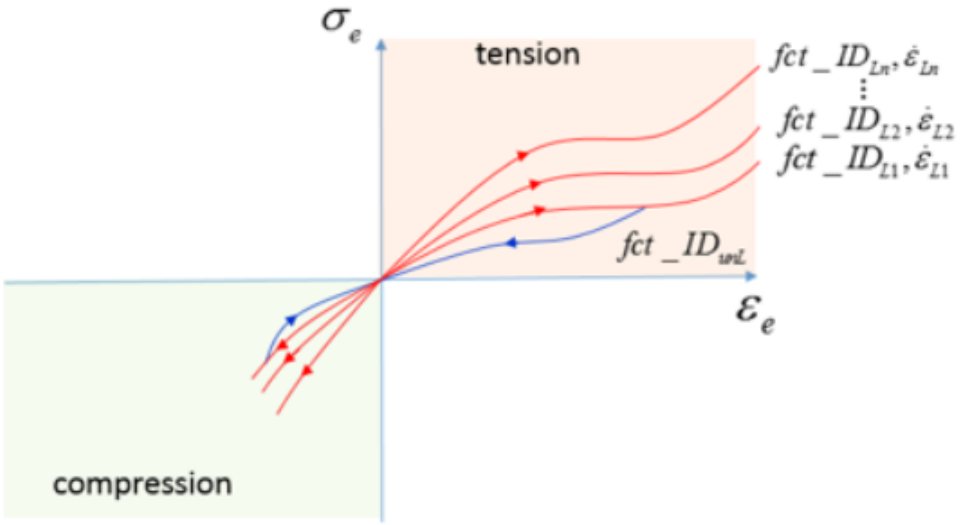


Figure 142:

Loading use loading function *fct_ID*$_{Li}$ and $\dot{\varepsilon}_{Li}$

Unloading use unloading function *fct_ID*$_{unL}$

= 1

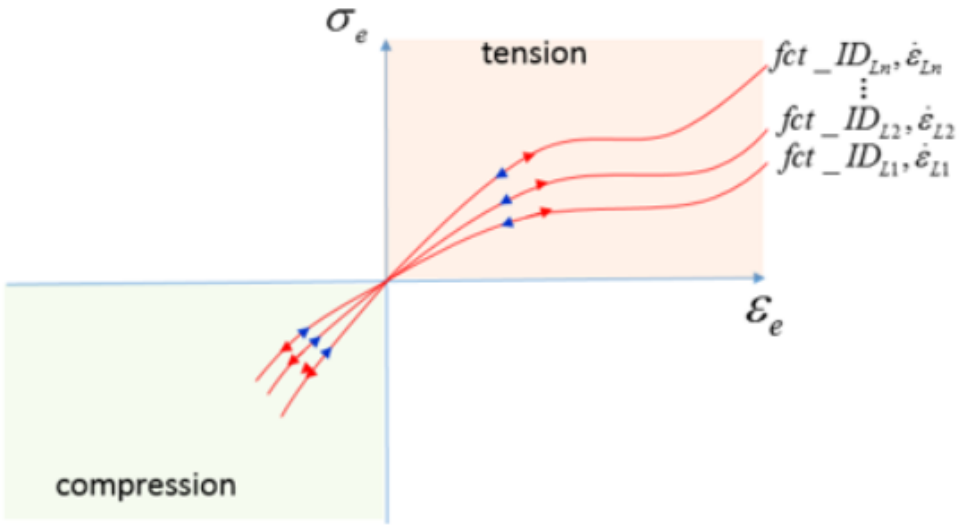


Figure 143:

Loading and unloading all use loading function *fct_ID*$_{Li}$ and $\dot{\varepsilon}_{Li}$

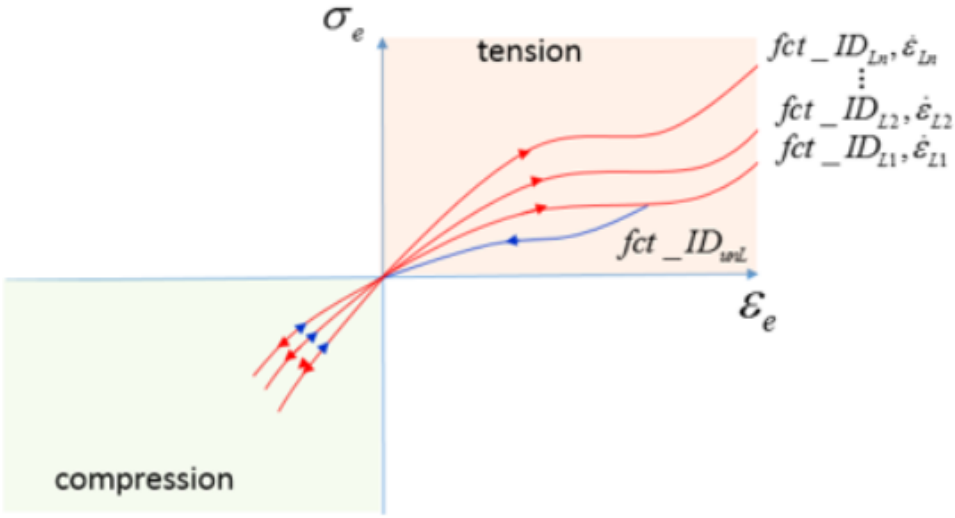
Tension	Loading and Unloading
= -1	

Figure 144:

Loading use loading function *fct_ID*$_{Li}$ and $\dot{\varepsilon}_{Li}$

- Tension

  Unloading use unloading function *fct_ID*$_{unL}$
- Compression

  Unloading use loading *fct_ID*$_{Li}$ and $\dot{\varepsilon}_{Li}$

Viscous (Rate Effects

Strain rate effects can be modeled by including engineering stress strain test data at different strain rates. This can be easier than calculating viscous parameters for traditional hyperelastic material models.

Conclusion

Make sure to use the material law that best fits the test data available.

For example, if minimal test data is available and the strains are not too large than the LAW42 Neo-Hookean Model could be used. If the loading state is known, then it is important to have test data that represents that stress state and make sure the material model fits that test data.

5. Ogden, R. W., and Non-linear Elastic Deformations. "Ellis Horwood." New York (1984)
6. Miller, Kurt. "Testing Elastomers for Hyperelastic Material Models in Finite Element Analysis" Axel Products, Inc., Ann Arbor, MI (2017). Last modified April 5, 2017

Arruda-Boyce (/MAT/LAW92)

LAW92 describes the Arruda-Boyce material model, which can be used to model hyperelastic behavior. This model is based on the statistical mechanics of a material with a cubic representative volume element containing eight chains along the diagonal directions.

It assumes that the chain molecules are located on the average along the diagonals of the cubic in principal stretch space.

Material Parameters

The strain energy density function is:

$$W = \underbrace{\mu \sum_{i=1}^{5} \frac{c_i}{(\lambda_m)^{2i-2}} \left(\bar{I}_1^i - 3^i\right)}_{W(\bar{I}_1)} + \underbrace{\frac{1}{D}\left(\frac{J^2-1}{2} + \ln(J)\right)}_{U(J)} \tag{94}$$

The material constant, c_i are:

$c_1 = \frac{1}{2},\ c_2 = \frac{1}{20},\ c_3 = \frac{11}{1050},\ c_4 = \frac{19}{7000},\ c_5 = \frac{519}{673750}$

$\bar{I}_1 = \bar{\lambda}_1^2 + \bar{\lambda}_2^2 + \bar{\lambda}_3^2$	First strain invariant
λ_i	i_{th} principal engineering stretch

A material with LAW92 can be defined in two different ways:

- Parameter Input

  Shear modulus, bulk modulus and strain stretch (μ, D, λ_m)

  Where, only the above 3 parameters with clear physical meaning are necessary to define the material.

  μ is shear modulus at zero strain.

  $$D = \frac{2}{K} \tag{95}$$

  Where,

K	Bulk coefficient at zero strain
λ_m	Defines the limit of stretch

http://www.axelproducts.com/downloads/TestingForHyperelastic.pdf

7. Axel Products, Inc. "Compression or Biaxial Extension", Ann Arbor, MI (2017). Last modified November 12, 2008

   http://www.axelproducts.com/downloads/CompressionOrBiax.pdf

8. Kolling, S., P. A. Du Bois, D. J. Benson, and W. W. Feng. "A tabulated formulation of hyperelasticity with rate effects and damage." Computational Mechanics 40, no. 5 (2007): 885-899

Also called *locking stretch*. It specifies the beginning of the hardening phase in tension (locking strain in tension). Default = 7.0.

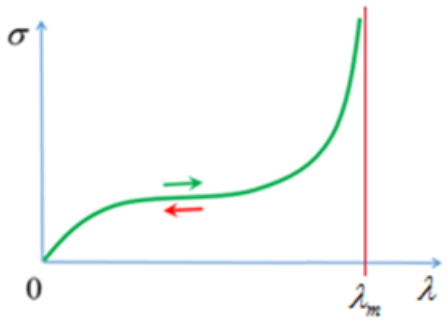


Figure 145:Locking Stretch λ_m

In parametric input, Poisson's ratio is computed as:

$$v = \frac{3K - 2\mu}{6K + 2\mu} \quad (96)$$

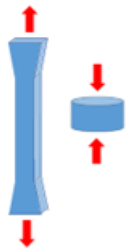
- When using function input, Poisson ratio v and *Itype* must be defined. *Itype* defines which type of engineering stress strain test data that is being used as input.
  - 

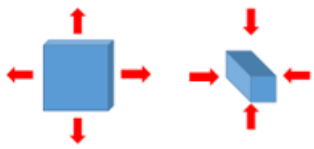
    Figure 146: Itype = 1: Uniaxial data test
  - 

    Figure 147: Itype = 2: Equibiaxial data test

Figure 148: Itype = 3: Planar data test

Poisson's Ratio and Material Incompressibility

If function input is defined, then parameters μ, D, λ_m are ignored and Radioss will calculate the material constant by fitting the input function. A nonlinear least squares algorithm is used to fit the Arruda-Boyce parameters by Radioss. The curve fitting is performed using the assumption that Poisson's value is close to 0.5, which means the material is incompressible. Similar to the other hyperelastic material models, Poisson ratio values closer to 0.5 result in high bulk modulus and a lower timestep. For a good balance between incompressibility and a reasonable timestep, a Poisson's ratio value of 0.495 is recommended.

The material fitting information can be found in the Starter output file (`*0000.out`).

```
FITTING RESULT COMPARISON:

UNIAXIAL TEST DATA

NOMINAL STRAIN     NOMINAL STRESS(TEST)       NOMINAL STRESS(RADIOSS)
        0.0000              0.0040                     0.0000
        0.0300              0.3000                     0.2387
        0.0600              0.5500                     0.4643
        0.1000              0.8000                     0.7472
        0.2000              1.4000                     1.3809
        0.3000              2.0000                     1.9346
        0.5000              2.7000                     2.8831
        0.7000              3.4000                     3.6982
        1.0000              4.0000                     4.7799

-----------------------------------------
AVERAGED ERROR OF FITTING :      20.25%
```

Figure 149: LAW92 function example

The fitting error and fitted material parameters are printed in the Starter output file.

```
FITTED PARAMETERS FOR HYPERELASTIC_MATERIAL LAW
 ---------------------------------------
 ARRUDA-BOYCE  LAW
MU . . . . . .= 2.731364005395
D. . . . . .= 7.3468374874591E-03
LAM. . . . . .= 2358.028306182
GROUND-STATE SHEAR MODULE . . . . . = 2.731364300131
BULK MODULUS. . . . . = 272.2259752463
```

Figure 150:

Viscous (Rate Effects

`/VISC/PRONY` must be used with LAW92 to include viscous effects.

Yeoh (/MAT/LAW94)

LAW94 is a hyperelastic material model that can be used to describe incompressible materials.

The strain energy density function of LAW94 only depends on the first strain invariant and is computed as:

$$W = \sum_{i=1}^{3}\left[\underbrace{C_{i0}(\bar{I}_1 - 3)^i}_{W(\bar{I}_1)} + \underbrace{\frac{1}{D_i}(J-1)^{2i}}_{U(J)}\right] \quad (97)$$

Where,

$\bar{I}_1 = \bar{\lambda}_1^2 + \bar{\lambda}_2^2 + \bar{\lambda}_3^2$ First strain invariant

$\bar{\lambda}_i = J^{-\frac{1}{3}}\lambda_i$ Deviatoric stretch

The Cauchy stress is:

$$\sigma_i = \frac{\lambda_i}{J}\frac{\partial W}{\partial \lambda_i} \quad (98)$$

Material Parameters

For incompressible materials with i=1 only and D_1 are input and the Yeoh model is reduced to a Neo-Hookean model.

9. Arruda, E. M. and Boyce, M. C., 1993, "A three-dimensional model for the large stretch behavior of rubber elastic materials", J. Mech. Phys. Solids, 41(2), pp. 389–412

C_{10}, C_{20}, C_{30}	Material constants specify the deviatoric part (shape change) of the material
D_1, D_2, D_3	Parameters specify the volumetric change of the material

These six material constants need to be calculated by curve fitting material test data. RD-E: 5600 Hyperelastic Material with Curve Input includes a Yeoh fitting Compose script for uniaxial test data. The Yeoh material model has been shown to model all deformation models, even if the curve fit was obtained using only uniaxial test data.

The initial shear modulus and the bulk modulus are:

$$\mu = 2 \cdot C_{10} \tag{99}$$

and

$$K = \frac{2}{D_1} \tag{100}$$

Poisson's Ratio and Material Incompressibility

LAW94 is available only as an incompressible material model.

If D_1 = 0, an incompressible material is considered where, $v = 0.495$ and D_1 is calculated as:

$$D_1 = \frac{3(1-2v)}{\mu(1+v)} \tag{101}$$

Bergstrom-Boyce (/MAT/LAW95)

This law is a constitutive model for predicting the nonlinear time dependency of elastomer like materials. It uses a polynomial material model for the hyperelastic material response and the Bergstrom-Boyce material model to represent the nonlinear viscoelastic time dependent material response.

This law is only compatible with solid elements.

The response of the material can be represented using two parallel networks A and B. Network A is the equilibrium network with a nonlinear hyperelastic component. In Network B, a nonlinear hyperelastic component is in series with a nonlinear viscoelastic flow element, and hence, is time-dependent network.

10. Yeoh, O. H. "Some forms of the strain energy function for rubber." Rubber Chemistry and technology 66, no. 5 (1993): 754-771

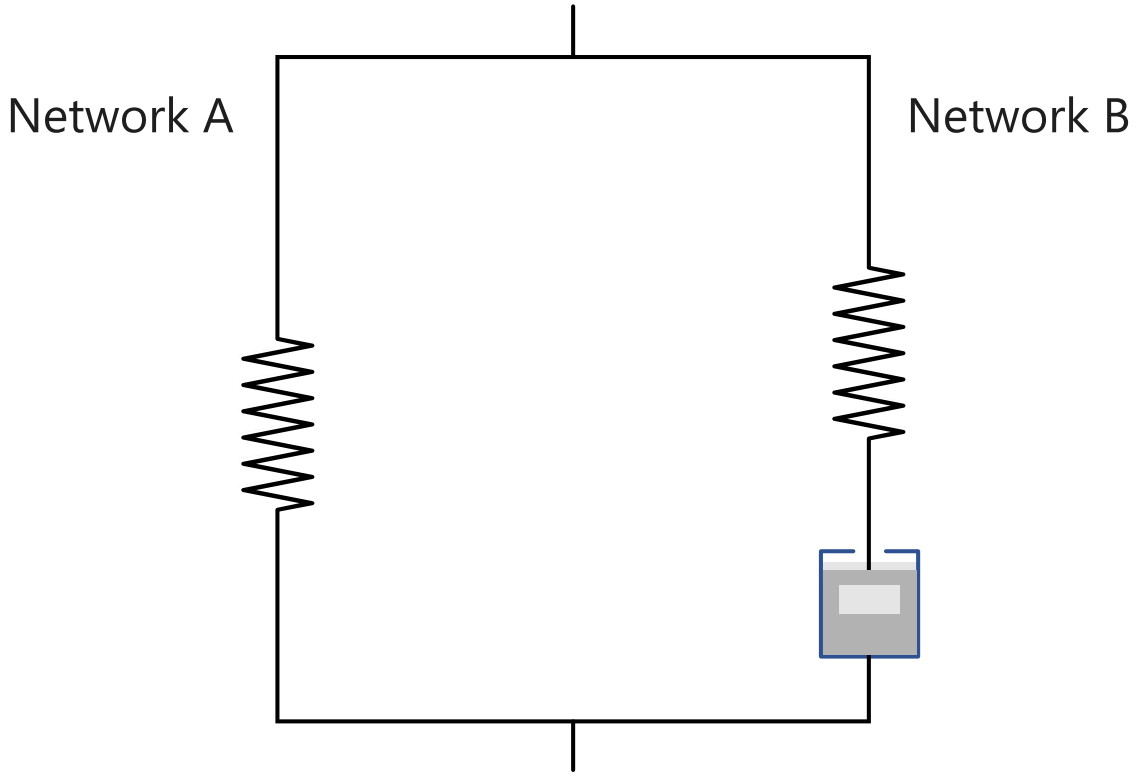


Figure 151:

Material Parameters

The same polynomial strain energy density formulation is used for the hyperelastic components in both networks. In Network B, it is scaled by a factor S_b. The strain energy density is then written for the hyperelastic component of the network.

$$W_A = \sum_{i+j=1}^{3} C_{ij}\left(\bar{I}_1 - 3\right)^i \cdot \left(\bar{I}_2 - 3\right)^j + \sum_{i=1}^{3} \frac{1}{D_i}\left(J - 1\right)^{2i} \quad (102)$$

and

$$W_B = S_b \cdot W_A \quad (103)$$

Where,

$\bar{I}_1 = \bar{\lambda}_1^2 + \bar{\lambda}_2^2 + \bar{\lambda}_3^2$

$\bar{I}_2 = \bar{\lambda}_1^{-2} + \bar{\lambda}_2^{-2} + \bar{\lambda}_3^{-2}$

$\bar{\lambda}_i = J^{-\frac{1}{3}}\lambda_i$

C_{ij} **and** D_i Material parameters

The hyperelastic component the Cauchy stress is computed as:

$$\sigma_i = \frac{\lambda_i}{J}\frac{\partial W}{\partial \lambda_i} \quad (104)$$

The total stress is the summer of stress in network A and network B.

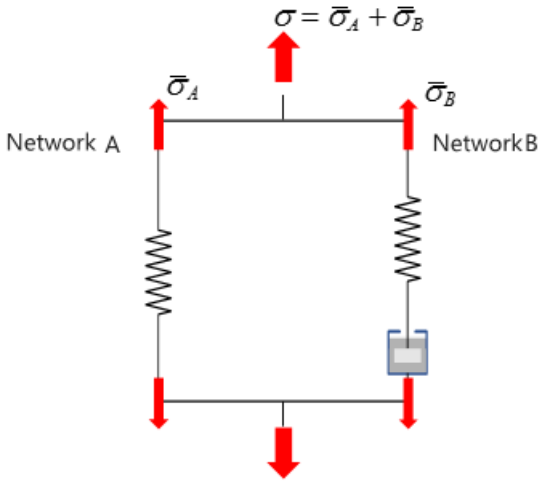


Figure 152:

$\sigma = \bar{\sigma}_A + \bar{\sigma}_B$

Since $W_B = S_b \cdot W_A$, then $\bar{\sigma}_B = S_b \cdot \bar{\sigma}_A$ and total stress is $\sigma = (1 + S_b) \cdot \bar{\sigma}_A$.

For example, in one tensile test. If use $S_b = 2$, then the stress is 3 times of the one without considering viscous (which means only considered hyperelastic).

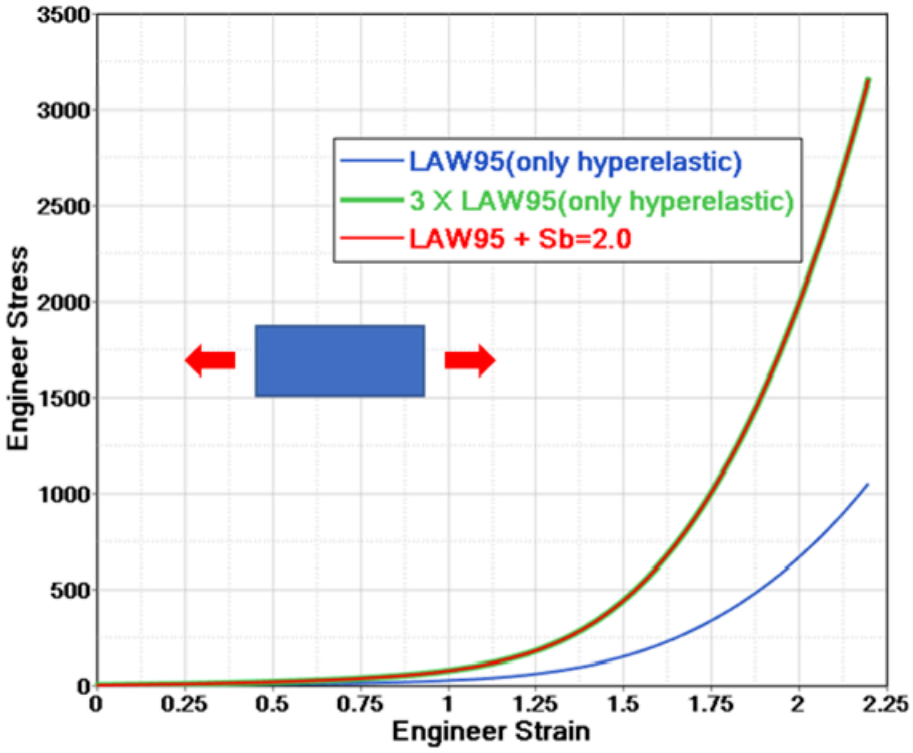


Figure 153:

For special values of C_{ij}, the polynomial model can be reduced to the following material models.

- Yeoh:

  $j = 0$

  Where, C_{10}, C_{20}, C_{30} are not zero.

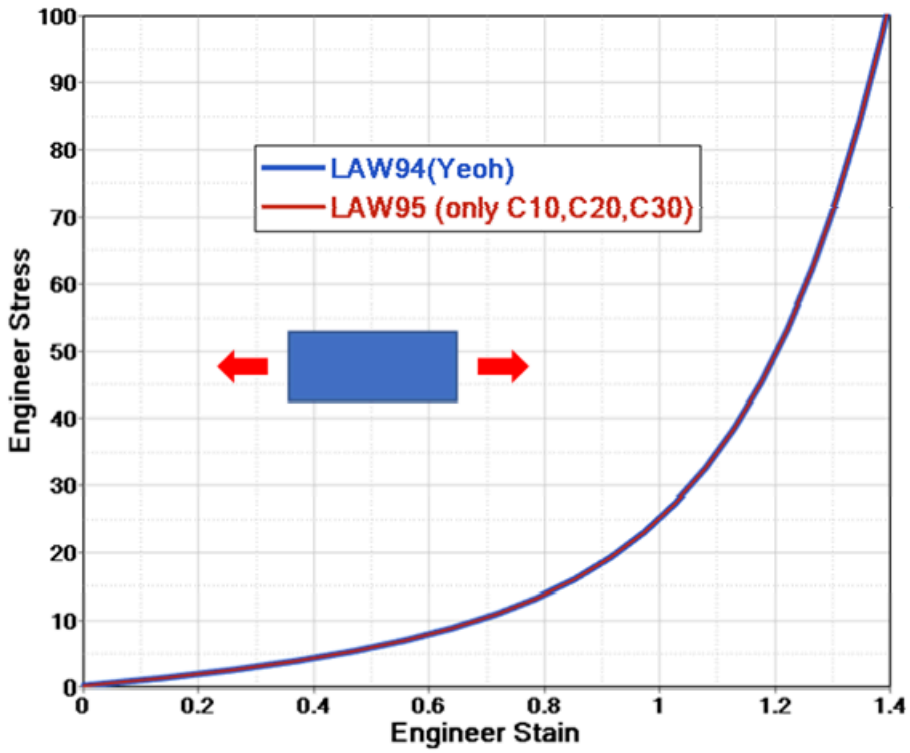


Figure 154:

- Mooney-Rivlin:

  $i+j=1$

  Where, C_{10} and C_{01} are not zero and $D_2=D_3=0$.
- Neo-Hookean:

  Only C_{10} and D_1 are not zero.

Where,

C_{ij} **and** D_i	Material parameters which can be calculated by completing a curve fit for quasi-static material test data.

RD-E: 5600 Hyperelastic Material with Curve Input, contains a curve fit example for Mooney-Rivlin and Yeoh material models. D_1 can be calculated from the bulk modulus or left blank.

The initial shear modulus and the bulk modulus are computed as:

$$\mu=2(S_b+1)(C_{10}+C_{01}) \tag{105}$$

and

$$K=\frac{2}{D_1}(1+S_b) \tag{106}$$

If the bulk modulus of the material is known, D_1 can be calculated, or if D_1=0, an incompressible material is assumed.

Viscous (Rate) Effects

The effective creep strain rate in Network B is given by:

$$\dot{\varepsilon}_B^v = A\left(\tilde{\lambda} - 1 + \xi\right)^C \left(\frac{\bar{\sigma}_B}{\tau_{ref}}\right)^M \quad (107)$$

Where,

$$\tilde{\lambda} = \sqrt{\frac{\bar{I}_1}{3}}$$

$\bar{\sigma}_B$ Effective stress in Network B.

A, ξ, M, C**, and** τ_{ref} Input material parameters.

The material constants A, M and C are limited to a specific range of real values as defined in the *Reference Guide*. If limited data is available, a trial and error method [11] could be used to determine these constants. Start with the default values of ξ, M, C, S_b=1.6; and A=5. Next, compare model predictions with experimental data for at least one strain rate and adjust A to get a fit for the strain rate data.

11. Bergström, J. S., and M. C. Boyce. "Constitutive modeling of the large strain time-dependent behavior of elastomers." Journal of the Mechanics and Physics of Solids 46, no. 5 (1998): 931-954

Elasto-plastic Materials

Johnson-Cook (/MAT/LAW2)

In LAW2 there are three parts to the stress calculation.

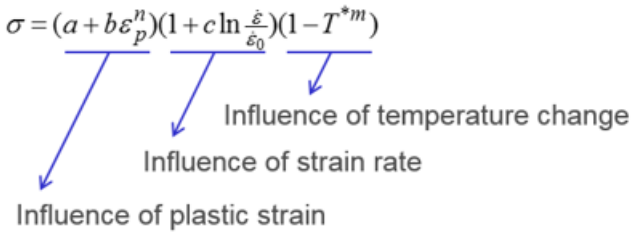


Figure 155:

- Influence of plastic strain
- Influence of strain rate
- Influence of temperature change

Material Parameters

There are two ways to input material parameter for LAW2.

- *Iflag*=0: Classic input for Johnson-Cook parameter a, b, n is active
- *Iflag*=1: New, simplified input with yield stress, UTS (engineering stress), or strain at UTS

Iflag= 0

$$\sigma = a + b \cdot \varepsilon_{pn} \qquad (108)$$

Where,

a	The yield stress which could be read from material test and converted to true stress.
b **and** n	The material parameters. Fitting the material stress-strain curve (for example, Altair Compose script) can result in these two parameter.
***Iflag* = 1**	With this new input, you will need yield stress (σ_y), Ultimate tensile engineer stress (UTS) and engineer strain (ε_{UTS}) at necking point. With this new input, Radioss automatically calculates the equivalent value for a, b and n.

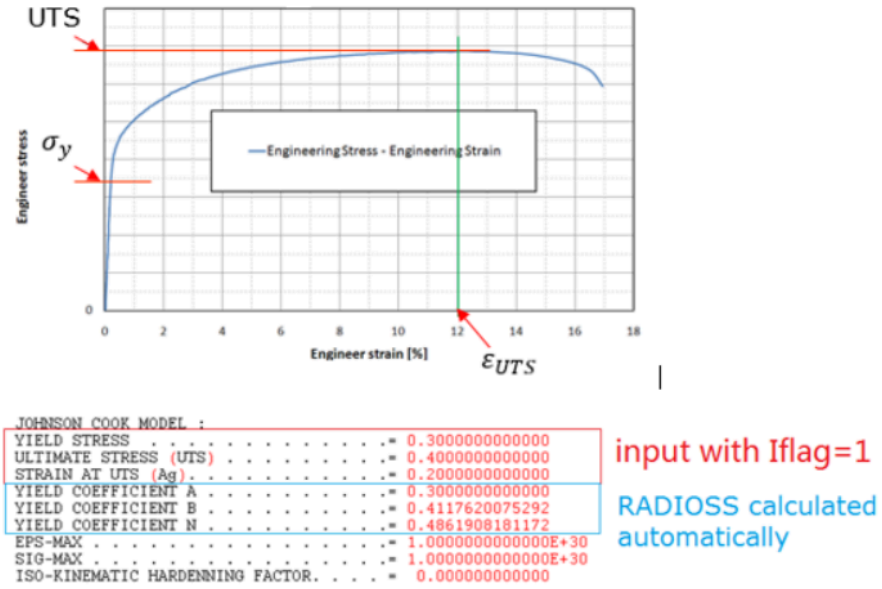


Figure 156: Tension Test

Strain Rate

Strain rate has a major effect of material character on crash performance in tensile or in fracture. In Johnson-Cook theory, the yield stress is affected directly by the strain rate and is described as:

$$\sigma = \left(a + b \cdot \varepsilon_{pn}\right)\left(1 + c\ln\frac{\dot{\varepsilon}}{\dot{\varepsilon}_0}\right) \quad (109)$$

Generally, yield stress increases with increasing the test strain rate. With the strain rate coefficient, c, you can scale the factor of yield stress increase. No effect of strain rate could also be defined, if c=0; or with $\dot{\varepsilon}_0 = 10^{30}$ or $\dot{\varepsilon} \le \dot{\varepsilon}_0$.

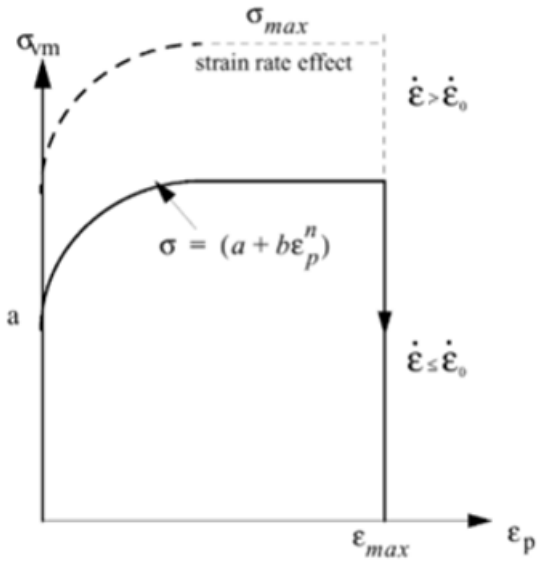


Figure 157:

Temperature Change

Yield stress decreases with increasing temperature. In LAW2 influence is considered with $(1-T^{*m})$.

$$\sigma=(a+b\cdot\varepsilon_{pn})\left(1+c\ln\frac{\dot{\varepsilon}}{\dot{\varepsilon}_0}\right)(1-T^{*m}) \quad (110)$$

with

$$T^*=\frac{T-T_r}{T_{melt}-T_r} \quad (111)$$

Where,

T_{melt}	Melt temperature in unit Kelvin.
T_r	Room temperature in unit Kelvin.

With T computed as:

$$T=T_i+\frac{E_{\text{int}}}{\rho C_p(Volume)} \quad (112)$$

Where,

E_{int}	Internal energy. Change of internal energy will affect the yield stress in Johnson-Cook law.

Hardening Coefficient

Metal deformed up to yield and then generally hardened (yield stress increased). Different materials show different ways of hardening (isotropic hardening, kinematic hardening, etc.). This is also a very important material character (for spring-back).

In LAW2, use option C_{hard} (hardening coefficient) to describe which hardening model is used for the material. This feature is also available in material LAW36, 43, 44, 57, 60, 66, 73 and 74.

The value of C_{hard} is from 1 to 0. C_{hard} =0 for isotropic model, C_{hard}=1 for kinematic Prager-Ziegler model, or between 1 and 0 for hardening between the above two models.

C_{hard} = 0: Isotropic Model

In a one dimension case, material strengthens after yield stress. The maximum stress of the last tension is the yield in the subsequent loading, and this new yield stress is the same in subsequent tension and compression.

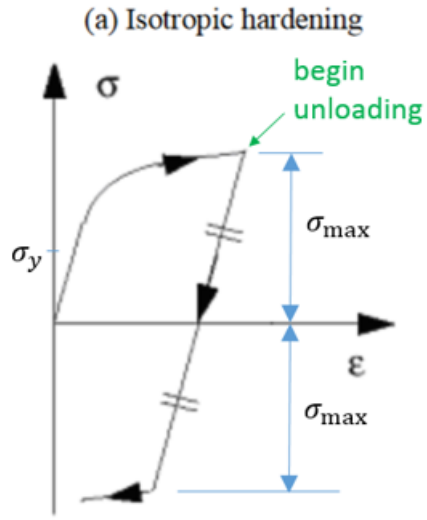


Figure 158:

C_{hard} = 1: Kinematic Prager-Ziegler Model

To model the Bauschinger effect (after hardening in tension, there is softening in a subsequent compression which mean yield in compression is decreased), use kinematic hardening.

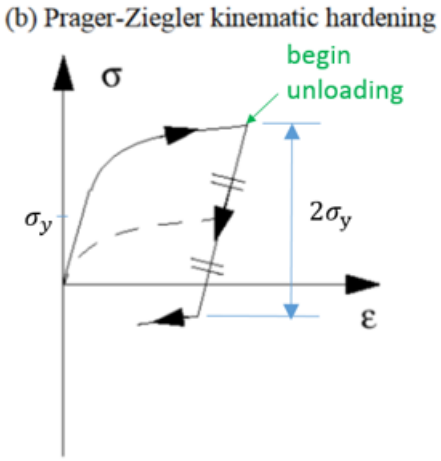


Figure 159:

Elastic Plastic Piecewise Linear Material (/MAT/LAW36)

In LAW36, the numbers of plastic stress-strain curves can be directly defined for different strain rates.

Plastic stress-strain of high strain rate should always be above the lower plastic stress-stain curve.

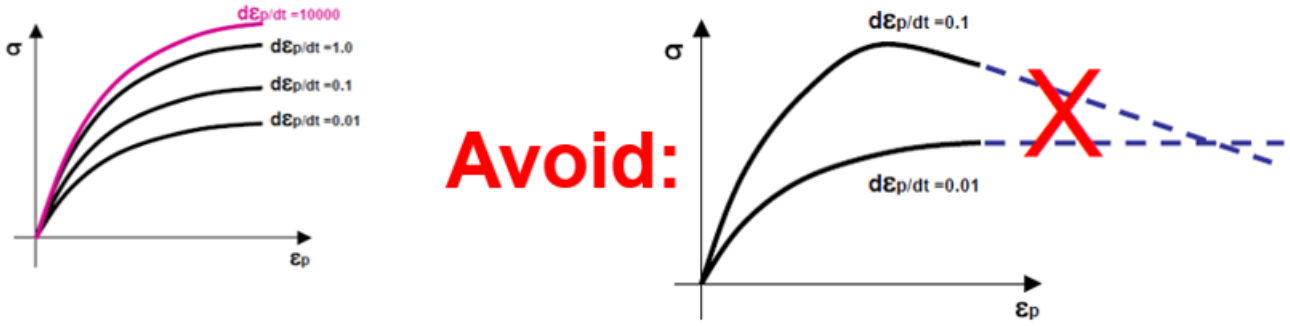


Figure 160:

Young's Modulus

Young's modulus can be updated (decreased) in unloading with options *fct_ID*$_E$, E_{inf} and C_E. Using this feature improves the accuracy of spring-back (in unloading phase) for high strength steel. This feature is also available in material LAW43, LAW57, LAW60, LAW74 and LAW78.

- Use *fct_ID*$_E$ to update the Young's modulus (*fct_ID*$_E \neq 0$):

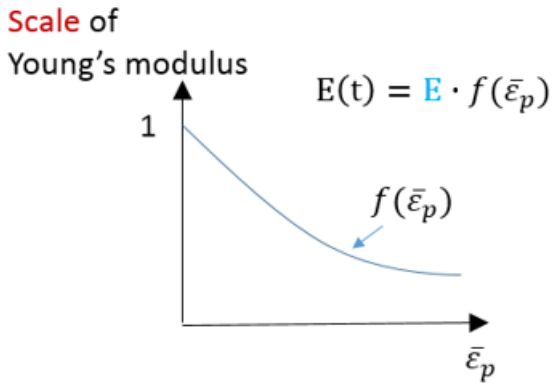


Figure 161:

- Use E_{inf} and C_E to update the Young's modulus (fct_ID_E = 0):

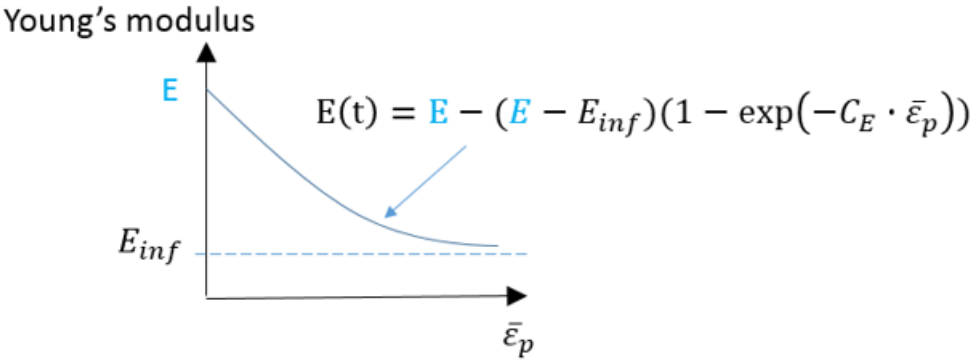


Figure 162:

Material Behavior

fct_ID_p is used to distinguish the behavior in tension and compression for certain materials (pressure dependent yield). The effective yield stress is then obtained by multiplying the nominal yield stress by the yield factor corresponding to the actual pressure.

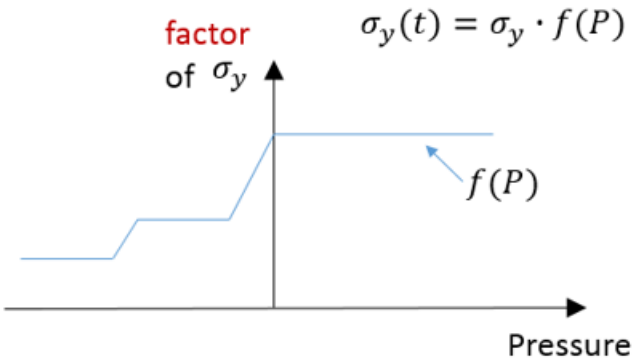


Figure 163:

See Also
/MAT/LAW2 (PLAS_JOHNS) (Starter)
RD-E: 1101 Elasto-plastic Material Law Characterization

HILL Materials

In Radioss material laws LAW32, LAW43, LAW72, LAW73, LAW74, LAW78 and LAW93 use HILL criteria.

HILL Criteria

The typical HILL criteria is:

- 3D equivalent HILL stress:

$$f = \sqrt{F(\sigma_{yy}-\sigma_{zz})^2 + G(\sigma_{zz}-\sigma_{xx})^2 + H(\sigma_{xx}-\sigma_{yy})^2 + 2L\sigma_{yz}^2 + 2M\sigma_{zx}^2 + 2N\sigma_{xy}^2} \quad (113)$$
$$= \sqrt{\underbrace{(G+H)}\sigma_{xx}^2 + \underbrace{(F+H)}\sigma_{yy}^2 + \underbrace{(F+G)}\sigma_{zz}^2 - \underbrace{2H}\sigma_{xx}\sigma_{yy} - \underbrace{2F}\sigma_{yy}\sigma_{zz} - \underbrace{2G}\sigma_{zz}\sigma_{xx} + \underbrace{2L}\sigma_{yz}^2 + \underbrace{2M}\sigma_{zx}^2 + \underbrace{2N}\sigma_{xy}^2}$$

- Shell element:

$$f = \sqrt{F\sigma_{yy}^2 + G\sigma_{xx}^2 + H(\sigma_{xx}-\sigma_{yy})^2 + 2N\sigma_{xy}^2} = \sqrt{\underbrace{(G+H)}\sigma_{xx}^2 + \underbrace{(F+H)}\sigma_{yy}^2 - \underbrace{2H}\sigma_{xx}\sigma_{yy} + \underbrace{2N}\sigma_{xy}^2} \quad (114)$$

  Where, F, G, H, L, M and N are six HILL anisotropic parameters. For shell elements, only F, G, H and N are the four HILL parameters needed.

  In LAW78, the HILL criteria is:

$$\varphi(A) = \underbrace{\frac{1}{G+H}} \cdot A_{xx}^2 - \underbrace{\frac{2r_0}{1+r_0}}_{2H} A_{xx}A_{yy} + \underbrace{\frac{r_0(1+r_{90})}{r_{90}(1+r_0)}}_{F+H} A_{yy}^2 + \underbrace{\frac{r_0+r_{90}}{r_{90}(1+r_0)}(2r_{45}+1)}_{2N} A_{xy}^2 \quad (115)$$

  There are two ways to determine HILL parameters by using Lankford parameters.

  - Strain ratio r_{00}, r_{45}, r_{90} (LAW32, LAW43, LAW72, LAW73)
  - Yield stress ratio R_{11}, R_{22}, R_{33}, R_{12}, R_{13}, R_{23} (LAW74, LAW93)

Strain Ratio

The Lankford parameters r_α is the ratio of plastic strain in plane and plastic strain in thickness direction ε_{33}.

$$r_\alpha = \frac{d\varepsilon_{\alpha+\pi/2}}{d\varepsilon_{33}} \tag{116}$$

Where, α is the angle to the orthotropic direction 1.

r_α could be measured with different samples which cut in different angle with orthotropic direction 1. Like r_{00} measured from tensile test in which the loading direction is along the orthotropic direction 1. r_{90} measured from tensile test in which the loading is perpendicular to orthotropic direction 1.

The strain ratio is the strain in width direction of sample to strain in thickness direction of sample.

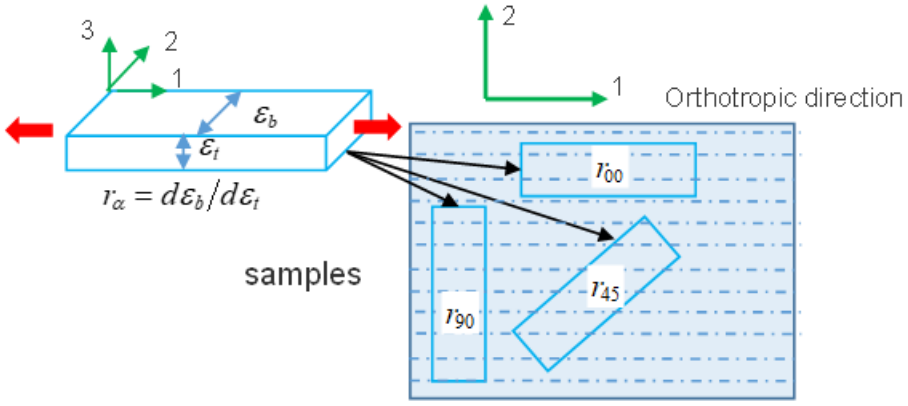


Figure 164:

In this case, the HILL parameters are:

$$F = \frac{r_{00}}{r_{90}(r_{00}+1)} \tag{117}$$

$$G = \frac{1}{(r_{00}+1)} \tag{118}$$

$$H = \frac{r_{00}}{(r_{00}+1)} \tag{119}$$

$$N = \frac{(1+2r_{45})(r_{00}+r_{90})}{2r_{90}(r_{00}+1)} \tag{120}$$

Here, $G+H=1$.

In LAW32, LAW43, and LAW73, the HILL criteria is:

$$\sigma_{eq} = \sqrt{A_1\sigma_1^2 + A_2\sigma_2^2 - A_3\sigma_1\sigma_2 + A_{12}\sigma_{12}^2} \tag{121}$$

$$R = \frac{r_{00} + 2r_{45} + r_{90}}{4}$$

$$H = \frac{R}{1+R}$$

$$A_1 = H\left(1 + \frac{1}{r_{00}}\right)$$

$$A_2 = H\left(1 + \frac{1}{r_{90}}\right)$$

$$A_3 = 2H$$

$$A_{12} = 2H(r_{45} + 0.5)\left(\frac{1}{r_{00}} + \frac{1}{r_{90}}\right)$$

They all request Lankford parameter (strain ratio) r_{00}, r_{45}, r_{90} and the HILL parameter A_i is automatically computed by Radioss.

Yield Stress Ratio

In LAW93, the yield stress ratio used is:

$$R_{ij} = \frac{\sigma_{ij}}{\sigma_0} \tag{122}$$

To get yield stress ratio R_{ij}, yield stress in two loading cases need to be measured.

- Yield stress $\sigma_{11}, \sigma_{22}, \sigma_{33}$ from tensile test
- Yield shear stress $\sigma_{12}, \sigma_{13}, \sigma_{23}$ from shear test

In LAW93, if parameter input is used, then take initial stress parameter σ_y as reference yield stress σ_0. If curve input is used, then take the yield stress from curve as reference yield stress σ_0.

Four HILL parameters for shell are automatically computed by Radioss.

$$F = \frac{1}{2}\left(\frac{1}{R_{22}^2} + \frac{1}{R_{33}^2} - \frac{1}{R_{11}^2}\right) \tag{123}$$

(124)

$$H = \frac{1}{2}\left(\frac{1}{R_{22}^2} + \frac{1}{R_{11}^2} - \frac{1}{R_{33}^2}\right) \tag{125}$$

$$N = \frac{3}{2R_{12}^2} \tag{126}$$

In LAW74, yield stress ratio R_{ij} is used with yield stress $\sigma_{11}, \sigma_{22}, \sigma_{33}$ and $\sigma_{12}, \sigma_{13}, \sigma_{23}$ input directly, and then six HILL parameters for solid are automatically computed by Radioss.

$$F = \frac{1}{2}\left(\frac{1}{\sigma_{22}^2} + \frac{1}{\sigma_{33}^2} - \frac{1}{\sigma_{11}^2}\right)$$

$$G = \frac{1}{2}\left(\frac{1}{\sigma_{22}^2} + \frac{1}{\sigma_{33}^2} - \frac{1}{\sigma_{11}^2}\right)$$

$$H = \frac{1}{2}\left(\frac{1}{\sigma_{22}^2} + \frac{1}{\sigma_{33}^2} - \frac{1}{\sigma_{11}^2}\right)$$

$$L = \frac{1}{2\sigma_{23}^2}$$

$$M = \frac{1}{2\sigma_{31}^2} \qquad N = \frac{1}{2\sigma_{12}^2}$$

For shell element, take $M = N$ and $L = N$.

Concrete and Rock Materials

In Radioss these materials can be used to represent rock or concrete materials.

These materials use a Drücker–Prager yield criterion[12], which is a pressure-dependent model for determining whether a material has failed or undergone plastic yielding.

Concrete Material (/MAT/LAW10 and /MAT/LAW21)

Drücker-Prager Yield Criteria

The material has failed or undergone plastic yielding is determined by pressure using:

$$F = \underbrace{J_2}_{J_2 part} - \underbrace{(A_0 + A_1 P + A_2 P^2)}_{I_1 part} \quad (127)$$

Where,

J_2 Second stress invariant (von Mises stress) of the deviatoric part of the stress and $P = -\frac{I_1}{3}$.

I_1 First stress invariant (hydrostatic pressure).

$I_1 = \sigma_1 + \sigma_2 + \sigma_3 = -3P$

$J_2 = \frac{1}{6}\left[(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2\right] = \frac{1}{3}\sigma_{VM}^2$ In a uniaxial test.

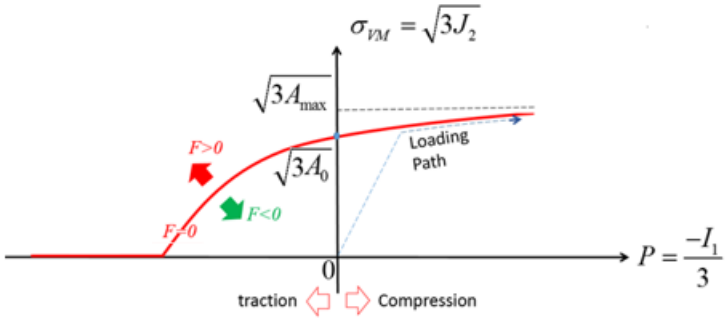


Figure 165: Drücker-Prager yield criteria

A polynomial equation is used to describe the pressure $A_0 + A_1 P + A_2 P^2$ at the Drücker-Prager yield surface of the material:

$$\sigma_{VM} = \sqrt{3(A_0 + A_1 P + A_2 P^2)} \quad (128)$$

The constants of the polynomial A_0, A_1, A_2 are determined by:

- If $F < 0$, $J_2 < A_0 + A_1 P + A_2 P^2$ the material is under yield surface and is in the elastic region.

- If $F = 0$, $J_2 = A_0 + A_1 P + A_2 P^2$ and the material is at the yield surface.
- If $F > 0$, $J_2 > A_0 + A_1 P + A_2 P^2$ and the material is past the yield surface and has failed.
- If $A_1 = A_2 = 0$, $\sigma_{VM} = \sqrt{3J_2} = \sqrt{3A_0}$, which is the von Mises criterion.

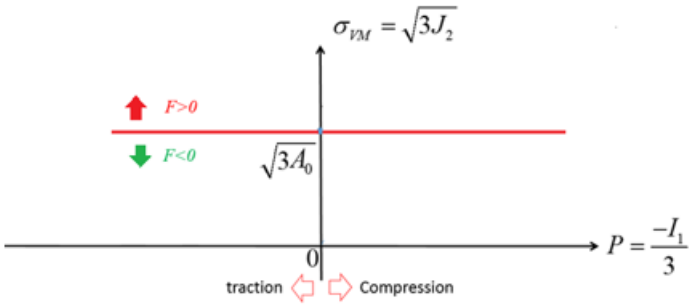


Figure 166:

Pressure Computation

In LAW10, a polynomial equation with input parameters C_0, C_1, C_2, C_3 is used to describe the pressure. The pressure can be plotted as a function of volumetric strain.

$$\mu = \frac{\rho}{\rho_0} - 1 \tag{129}$$

- If $P_{ext} = 0$, the pressure is $P = \Delta P$ and the pressure limit is $P_{\min} = \Delta P_{\min}$.

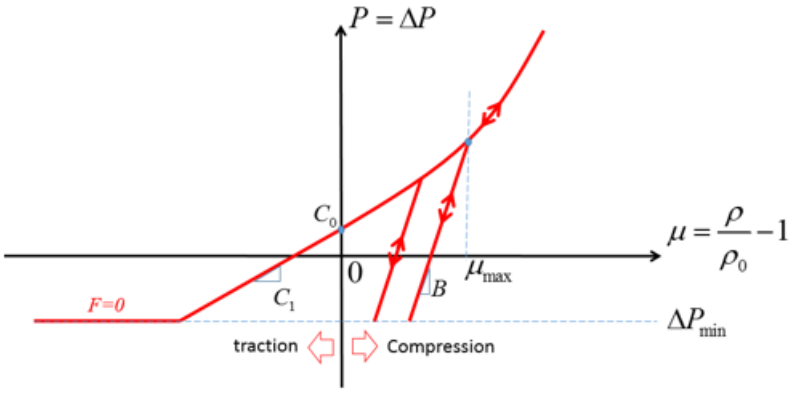


Figure 167: Pressure curve without external pressure

- If $P_{ext} \neq 0$, the pressure is shifted by P_{ext}, then $P = P_{ext} + \Delta P$ and the pressure limit is $P_{\min} = P_{ext} + \Delta P_{\min}$.

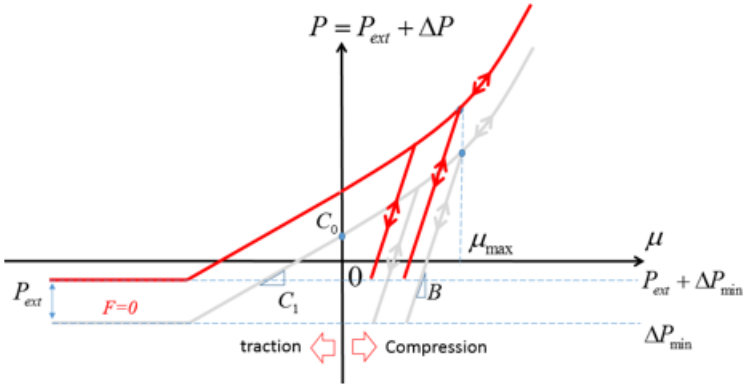


Figure 168: Pressure curve with external pressure

Here,

$$\Delta P = \begin{cases} \max\{\Delta P_{min}, C_0 + C_1\mu + C_2\mu^2 + C_3\mu^3\} & if \quad \mu \geq 0 \; compression \\ \max\{\Delta P_{min}, C_0 + C_1\mu\} & if \quad \mu < 0 \; traction \end{cases} \tag{130}$$

- In traction or tension the pressure is linear and limited by ΔP_{min}.
- In compression the pressure is nonlinear also limited by ΔP_{min}.

The only difference between the material laws is that in LAW10 the material constants C_0, C_1, C_2, C_3 are used to describe the pressure versus volumetric strain ($P - \mu$ curve). In LAW21 you can describe this curve via function input *fct_ID$_f$*.

Load and Unload

In LAW10 and LAW21 different loading and unloading paths of the $P - \mu$ curve can be considered by using the parameters μ_{max} and B.

- In Tension ($\mu < 0$)
  - For LAW10, linear loading and unloading with $P = C_1\mu$ (Figure 167).
  - For LAW21, loading is defined using the input function *fct_ID$_f$* and linear unloading with $P = K_t\mu$.
- In Compression ($\mu > 0$), for both LAW10 and LAW21:
  - If neither B and μ_{max} are defined, the loading and unloading path are identical.

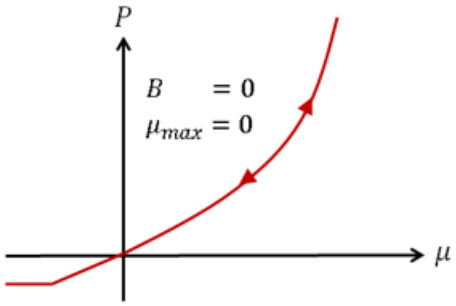


Figure 169: Identical loading and unloading for LAW10 and LAW21

- If either B or μ_{max} is defined:
  1. If only B is defined, μ_{max} is the volumetric strain where the tangent of $P-\mu$ curve is equal to B with $B=\left.\frac{dP}{d\mu}\right|_{\mu_{max}}$.
  2. If only μ_{max} is defined, then B is the tangent of $P-\mu$ curve at μ_{max}. The loading and unloading in compression is:
     - If $\mu > \mu_{max}$, loading and unloading path are identical.
     - If $\mu < \mu_{max}$, loading and unloading path are different, it is linear unloading with slope B.

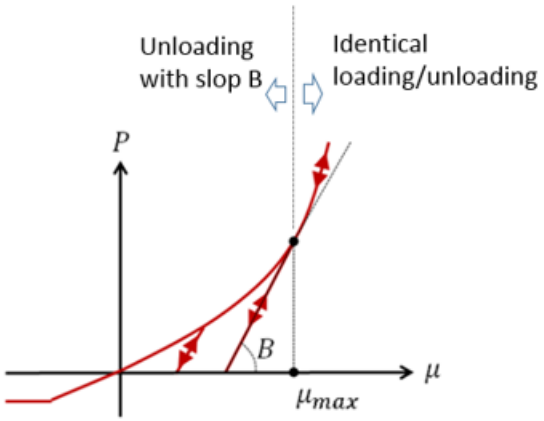


Figure 170: Different loading and unloading treatment for LAW10 and LAW21

Concrete Material (/MAT/LAW24)

LAW24 uses a Drücker-Prager criteria with or without a cap in yield to model a reinforced concrete material. This material law assumes that the two failure mechanisms of the concrete material are tensile cracking and compressive crushing.

Concrete Tensile Behavior

In LAW24, the options H_t, D_{sup}, and ε_{max} can be used to describe tensile cracking and failure in tension.

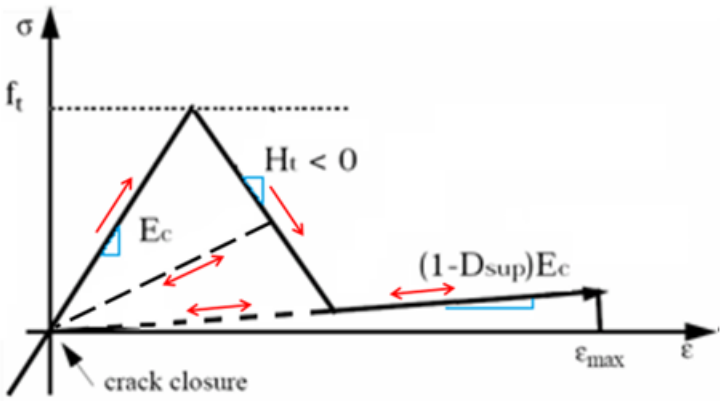


Figure 171: LAW24 Tensile Loading

In the initial very small elastic phase, the material has an elastic modulus E_c.

Once tensile strength, f_t is reached, the concrete starts to soften with the slope H_t. The maximum damage factor, D_{sup}, is significant because it enables the modeling of residual stiffness during and after a crack.

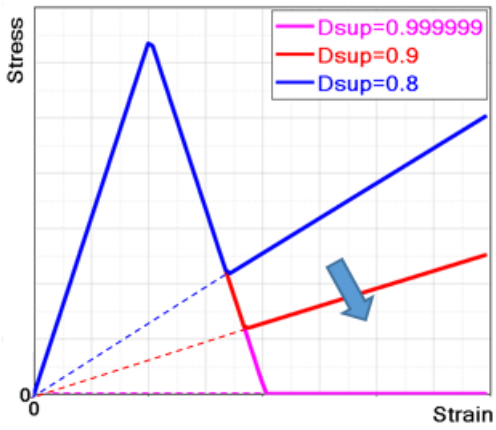


Figure 172: Maximum Damage Factor Effects

The residual stiffness is computed as:

$$E = \left(1 - D_{\text{sup}}\right) \cdot E_c \qquad (131)$$

When there is crack closure, the concrete becomes elastic again, and the damage factor (for each direction) is conserved.

The bearing capacity of concrete in tensile is much lower than in compression. It is normally considered elastic when in tension.

It is recommended to choose a D_{sup} value close to 1 (default is 0.99999) in order to minimize the current stiffness at the end of the damage and consequently avoid residual stress in tension, which can become very high if the element is highly deformed due to tension. This will happen if the force causing the damage remains.

It is possible to adjust the D_{sup} (and H_t) in order to simulate and fit the behavior of concrete reinforced by fibers. The concrete material fails once it reaches the total failure strain $\varepsilon_{\max}$.

Concrete Yield Surface in Compression

For concrete, the yield surface is the beginning of the plastic hardening zone which is between the failure surface, r_f, and the yield surface.

The yield surface is assumed to be the same as the failure surface in the tension zone. In compression, the yield surface is a scaled down failure surface using the factor $\mathrm{k}(\sigma_m, k_0)$. The yield in LAW24 for concrete is:

$$f = \underbrace{r}_{\substack{J_2 \\ part}} - \underbrace{\mathrm{k}(\sigma_m, k_0) \cdot r_f}_{I_1 part} = 0 \qquad (132)$$

- For I_{cap} =0 or 1 (without a cap in yield) the yield curve is:

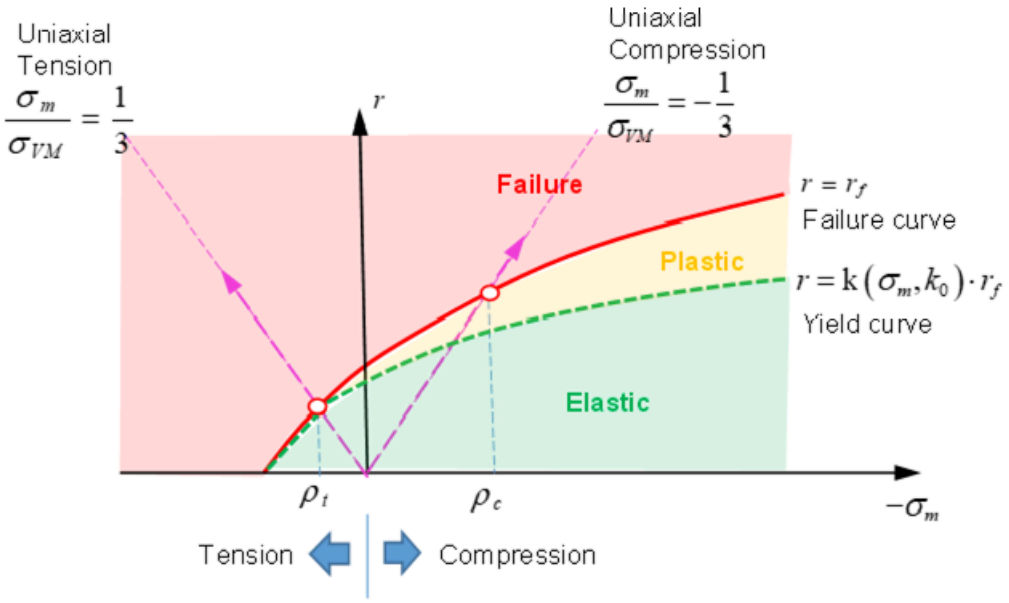


Figure 173: Drücker-Prager Criteria without a Cap in Yield

- For I_{cap} =2 (with cap in yield) the yield is:

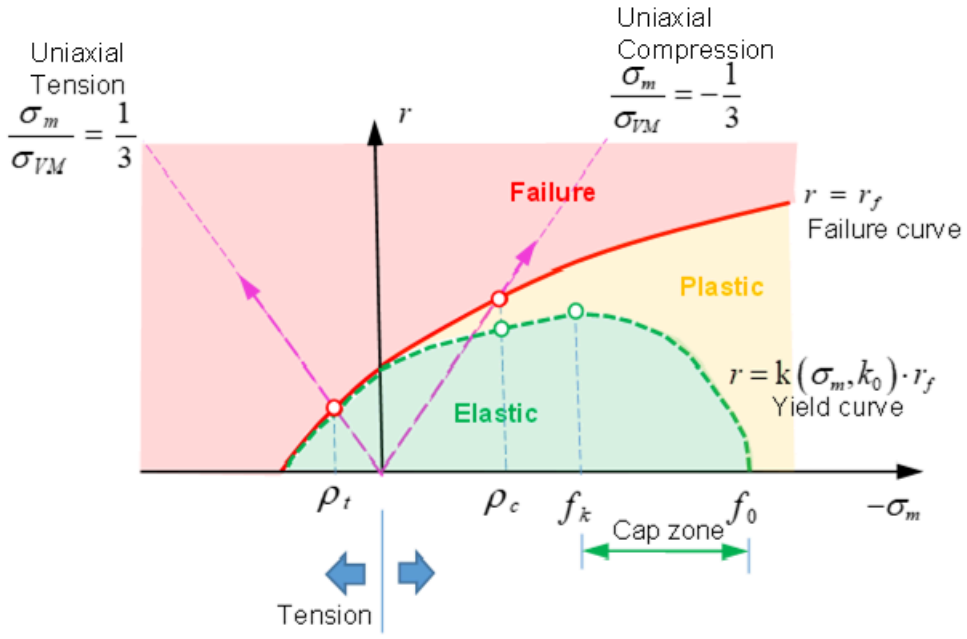


Figure 174: Drücker-Prager Criteria with Cap in Yield

$r < k(\sigma_m, k_0) \cdot r_f$ *(green area in* Figure 174*)*

The material is under yield in the elastic phase.

$r \geq r_f$ *(red area in* Figure 174*)*

The material has failed.

$k(\sigma_m, k_0) \cdot r_f < r < r_f$ *(yellow area in* Figure 174*)*

The material is above yield and below the failure surface which is the plastic hardening phase.

The input parameter ρ_t is the hydrostatic failure pressure in a uniaxial tension test and ρ_c is the hydrostatic pressure by failure in a uniaxial compression test.

The scale factor $k(\sigma_m, k_0)$ is a function of mean stress σ_m and can be described as:

- When $\sigma_m \geq \rho_t$ (in tension) the scale factor $k(\sigma_m, k_0) = 1$. In this case, the yield surface equals the failure surface, $r = r_f$.

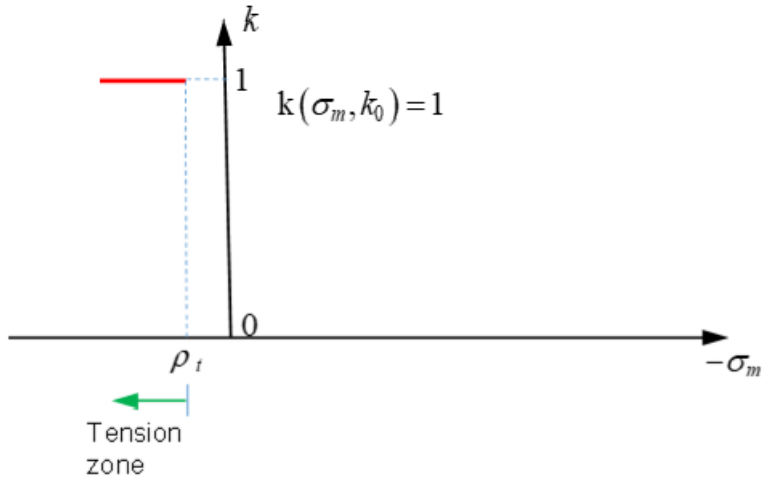


Figure 175: k Function in the Tension Zone

- In the tension-compression region, $\rho_t > \sigma_m \geq \rho_c$, then

$$k(\sigma_m, k_0) = 1 + \frac{(1-k_0)\cdot\left[\rho_t(2\rho_c - \rho_t) - 2\rho_c\sigma_m + \sigma_m 2\right]}{(\rho_c - \rho_t)^2} \text{ with } k_y \leq k_0 \leq 1$$

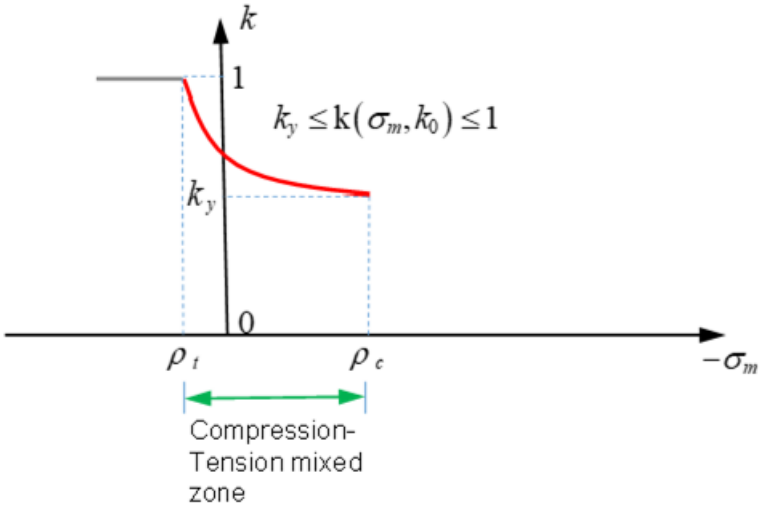


Figure 176: k Function in the Compression-Tension Mixed Zone

- The rest of the curve depends on the I_{cap} option and the different scale factors $k(\sigma_m, k_0)$ used.
  - For I_{cap} =0 or 1 and $\sigma_m < \rho_c$ (in compression), then $k(\sigma_m, k_0) = k_y$

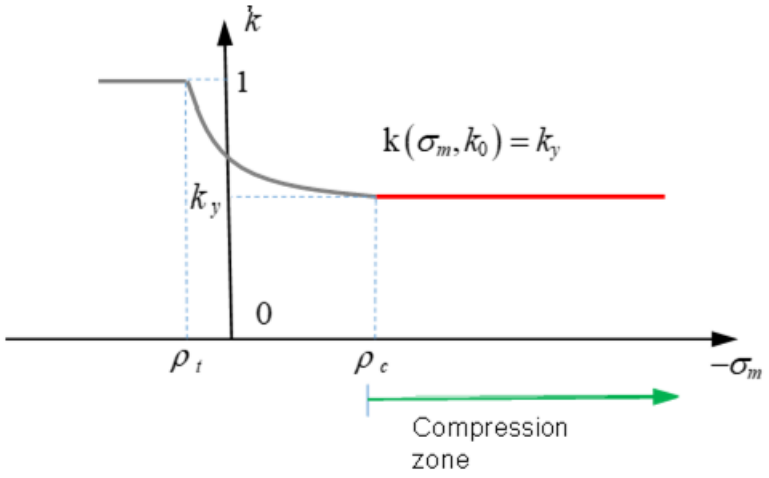


Figure 177: k Function in the Compression Zone

- For I_{cap} =2 (with cap in yield) and $\rho_c < \sigma_m < f_k$ (in compression), then $k(\sigma_m, k_0) = k_y$

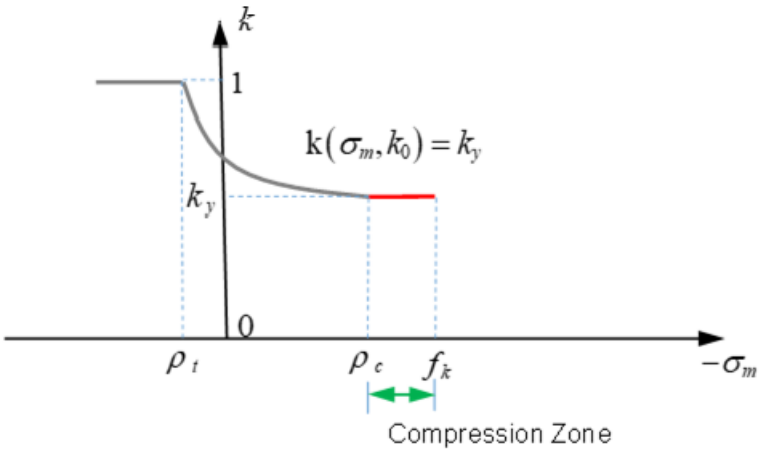


Figure 178: k Function in Drücker-Prager Criteria without a Cap

- In $f_k < \sigma_m < f_0$ (in cap zone)

$$k(\sigma_m, k_0) = k_0\left[1 - \left(\frac{\sigma_m - f_k}{f_0 - f_k}\right)^2\right] \text{ with } 0 \le k_0 \le k_y$$

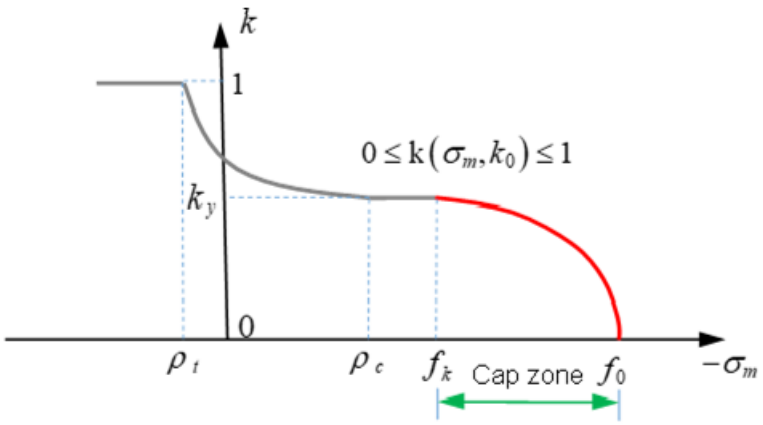


Figure 179: k Function in Drücker-Prager Criteria with a Cap

The material constant k_y should be $0 \le k_y \le 1$. A higher value of k_y results in a higher yield surface.

For example, if I_{cap} =2 (yield with cap), the difference of yield surface between $k_y = 0.8$ and $k_y = 0.6$ (Figure 180). The default value of k_y in LAW24 is 0.5.

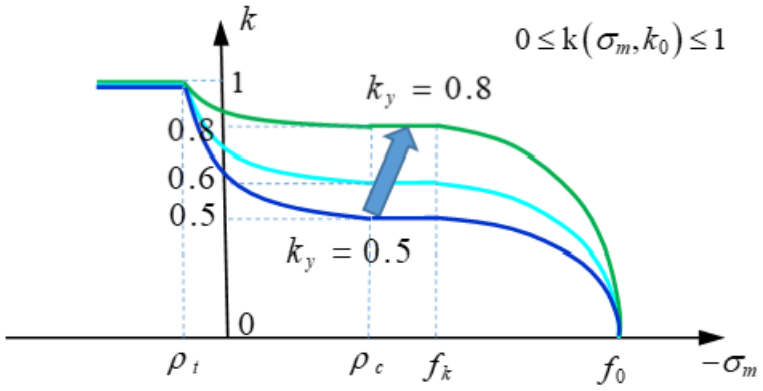


Figure 180: Affect of Different k Function Values

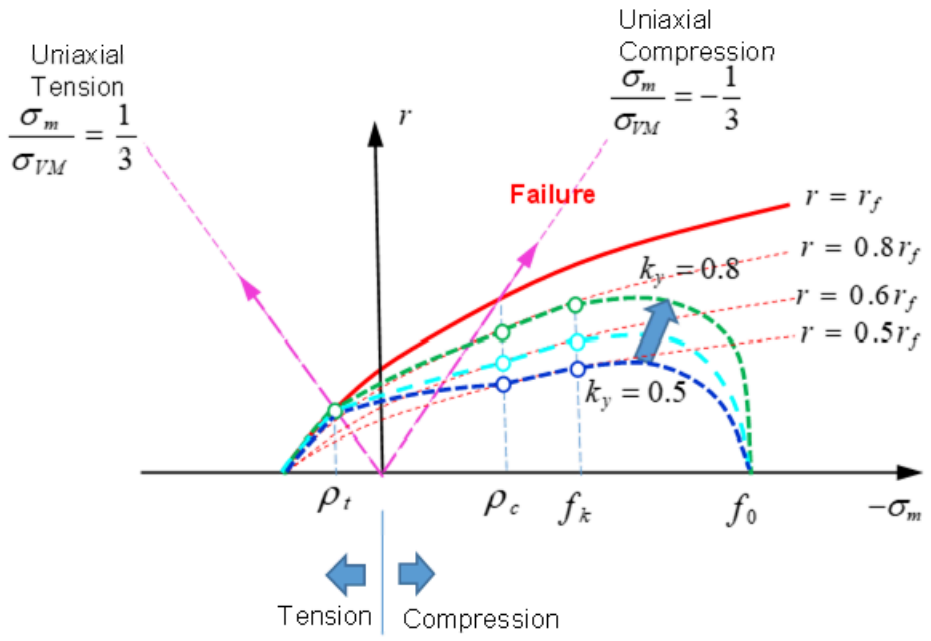


Figure 181: Drücker-Prager Criteria with Different k Function Values

Concrete Plastic Flow Rule in Compression

A non-associated plastic flow rule is used in LAW24. The plastic flow rule is:

$$g = \alpha I_1 + \sqrt{J_2} \tag{133}$$

Where,

α	Plastic dilatancy.
$\alpha = \frac{\partial g}{\partial I_1}$	Governs the volumetric plastic flow.
I_1	First stress invariant (hydrostatic pressure).

Experimentally, α is a linear function of k_0:

$$\alpha = \frac{(1-k_0)\alpha_y + (k_0 - K_y)\alpha_f}{1-K_y} \tag{134}$$

If $k_0 = K_y$	then, $\alpha = \alpha_y$ which means the material is in yield.
If $k_0 < K_y$	then, α becomes negative is the cap region.
If $k_0 = 1$	then, $\alpha = \alpha_f$ which means the material has failed.

The values of $\alpha_y,\ \alpha_f$ are used to describe the material beyond yield, but before failure. It is recommended to use -0.2 and -0.1 for $\alpha_y,\ \alpha_f$ in LAW24. If very small values of $\alpha_y,\ \alpha_f$ are used, there is no volumetric plasticity (no cap region).

Concrete Crushing Failure in Compression

Failure surface is given by:

$$f = r - r_f(\sigma_m, \theta) = 0 \tag{135}$$

Where, $r = \sqrt{2J_2}\,/\,f_c$, $\sigma_m = I_1/3f_c$ and θ is Lode angle, such as:

$$\cos 3\theta = \frac{J_3}{2}\left(\frac{3}{J_2}\right)^{3/2} \tag{136}$$

An Ottosen surface is built to design this surface using:

$$r_f(\sigma_m, \theta) = \frac{1}{a}\left(-b + \sqrt{b^2 - a(\sigma_m - c)}\right) \tag{137}$$

Where, a, b_c, b_t and c are 4 values which shape the surface and

$$b(b_c, b_t, \theta) = \frac{1}{2}\left[b_c(1 - \cos 3\theta) + b_t(1 + \cos 3\theta)\right] \tag{138}$$

For concrete, the compression failure curve r_f can be defined with a strength of:

f_t	Uniaxial tension (triaxiality is 1/3)
f_c	Uniaxial compression (triaxiality is -1/3)
f_b	Biaxial compression (triaxiality is -2/3)
f_2	Confined compression strength (tri-axial test)
s_0	Under confined pressure

The best way to fully determine the 3D failure envelope is to get experimental data for all of these values, f_c, f_t, f_b, f_2, s_0 which are schematically illustrated in Figure 182.

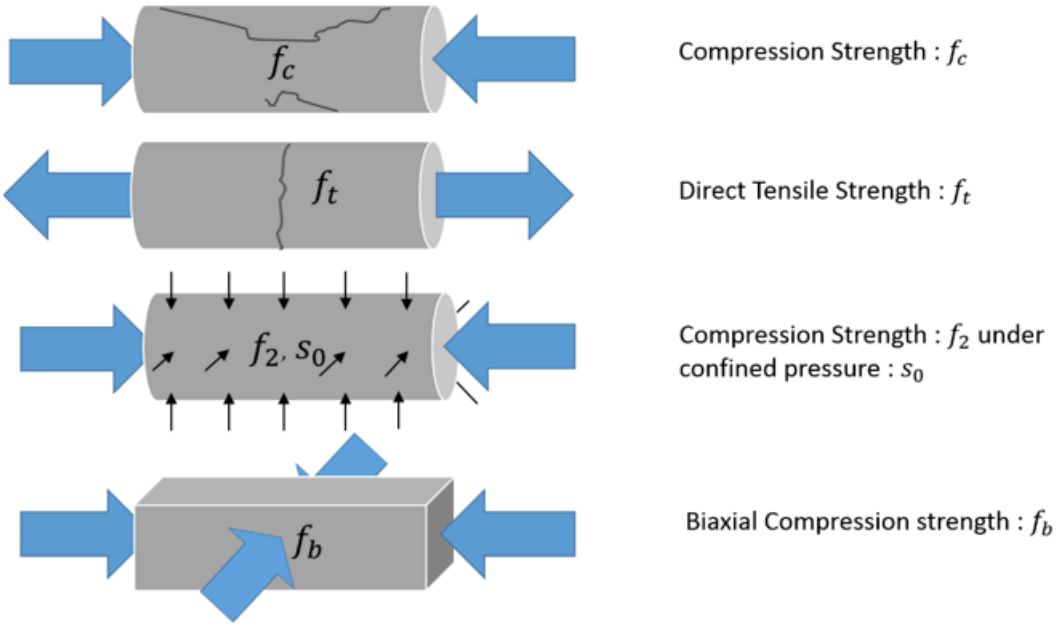


Figure 182: Failure Parameters that Fully Determine 3D Failure Envelope

Table 17: Input from the 4 Experimental Tests

Load Type	Surface Point	Default Input	Criteria r	Pressure σ_m	Lode Angle θ
Compression	$(f_c, 0, 0)$	Mandatory	$r=\sqrt{2/3}$	$\sigma_m=-1/3$	$\cos\theta=-1$
Direct Tensile	$(f_t, 0, 0)$	$f_t=0.1f_c$	$r=\sqrt{2/3}\left(\frac{f_t}{f_c}\right)$	$\sigma_m=1\Big/3\left(\frac{f_t}{f_c}\right)$	$\cos\theta=1$
Biaxial Compression	(f_2, s_0, s_0)	If I_{cap} = 1 $f_2=4.0f_c$ If I_{cap} = 2 $f_2=7.0f_c$ $s_0=1.25f_c$	$r=\sqrt{2/3}\left(\frac{f_t}{f_c}\right)\sigma_m$	$\sigma_m=2\Big/3\left(\frac{f_t}{f_c}\right)$	$\cos\theta=1$
Compression Strength under Confinement Pressure	$(f_b, f_b, 0)$	$f_b=1.2f_c$	$r=\sqrt{2/3}\frac{f_2-s_0}{f_c}$	$\sigma_m=\frac{f_2+2s_0}{3f_c}$	$\cos\theta=-1$

Figure 183 and Figure 184 show the points that determine the failure surface.

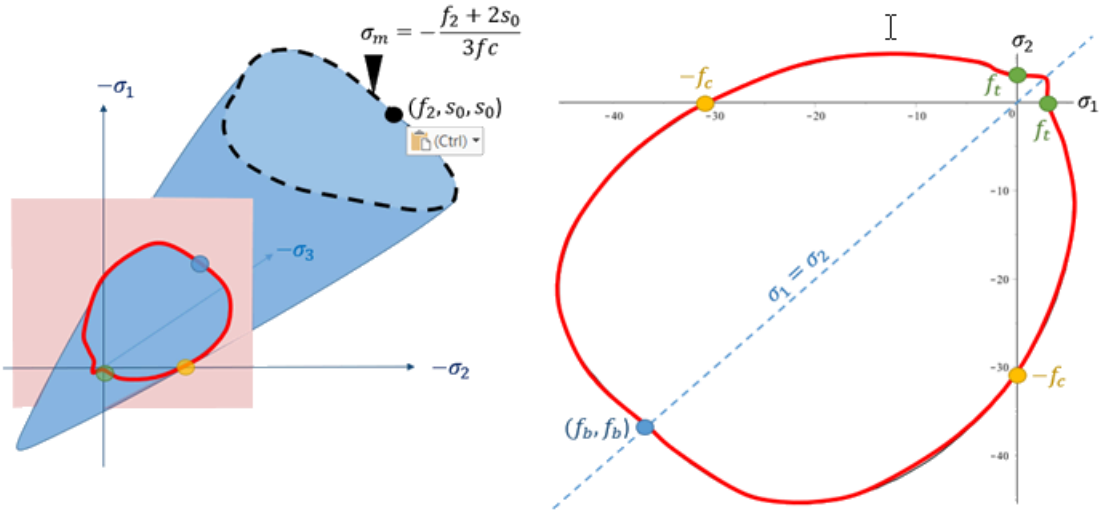


Figure 183: Trace of failure surface with planar stress plane

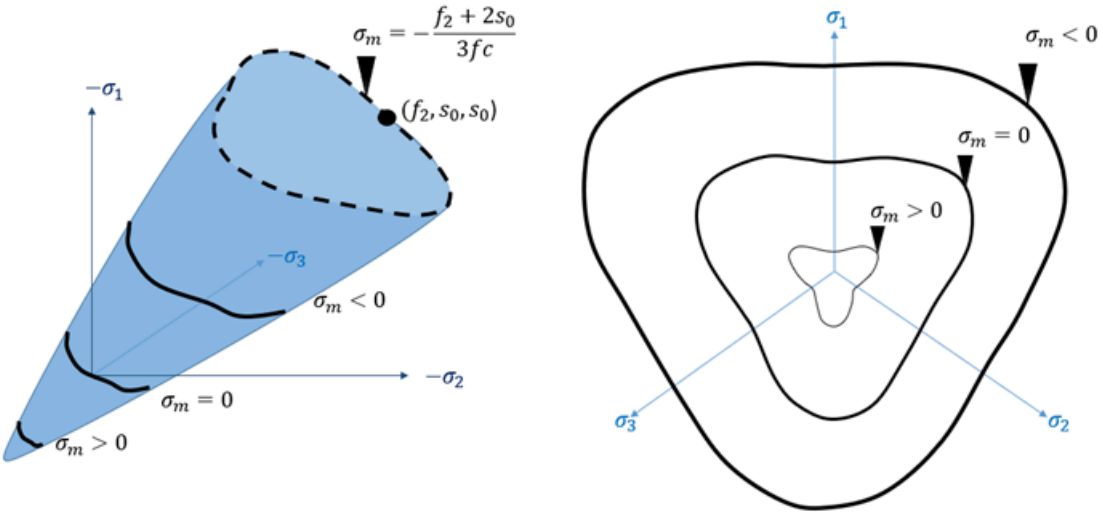


Figure 184: Failure trace with several cut plan which are normal to the hydrostatic axis

From these plots that the failure envelope is not a convex surface. Figure 185 shows this behavior.

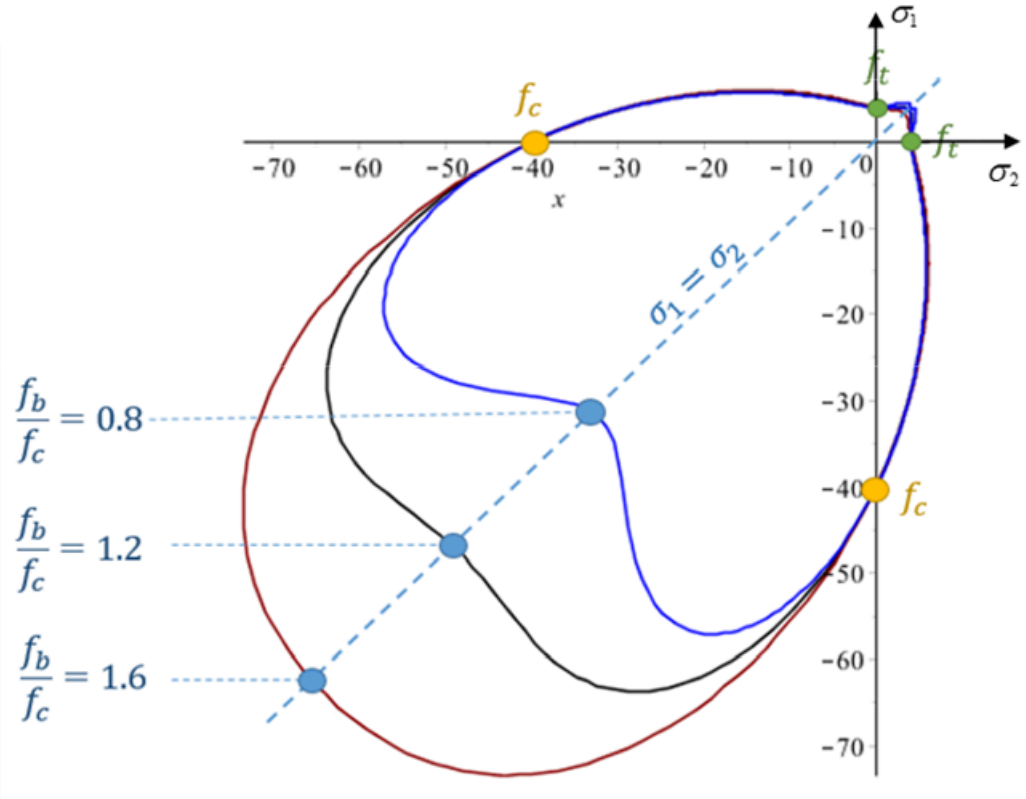


Figure 185: Influence of the biaxial compressive strength value with all other characteristic failure points fixed

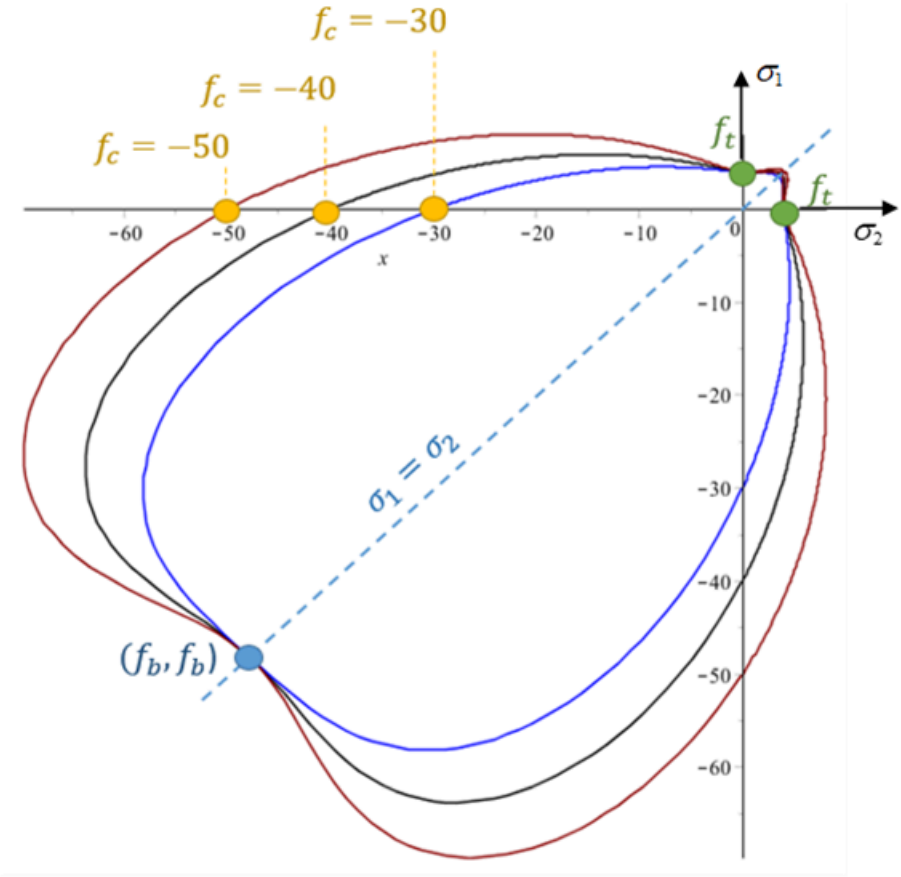


Figure 186: Influence of the compressive strength value with all other characteristic failure points fixed

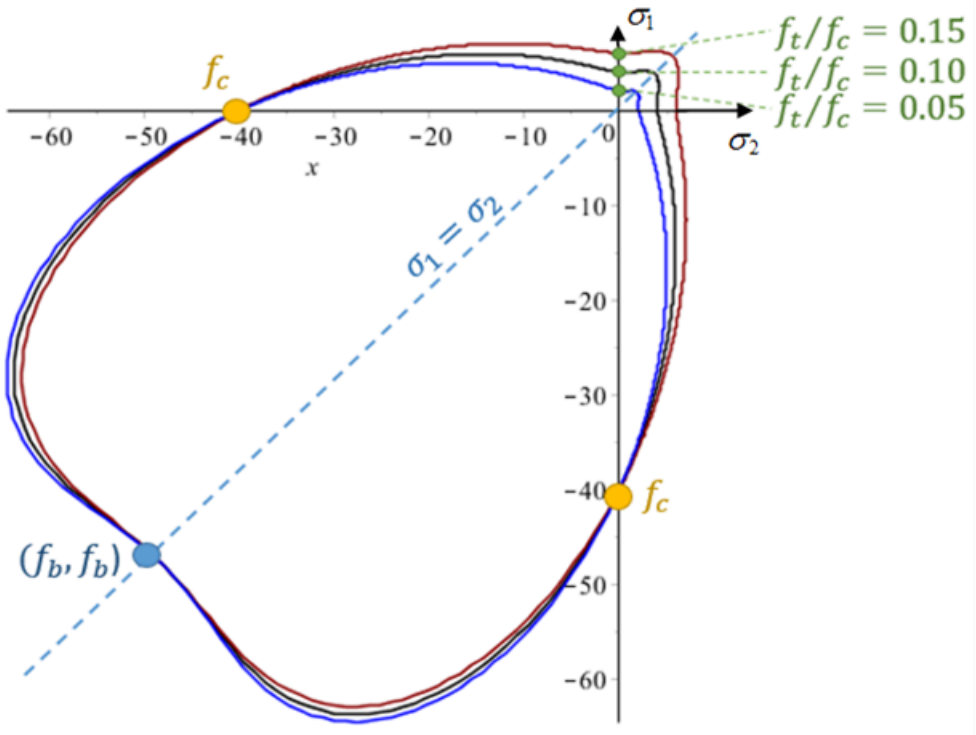


Figure 187: Influence of the tensile strength value with all other characteristic failure points fixed

In this particular case, the compressive strength is changing but all other ratios are fixed f_t/f_c, f_b/f_c, f_2/f_c, s_0/f_c. This leads to an envelope scaling, as shown in Figure 188.

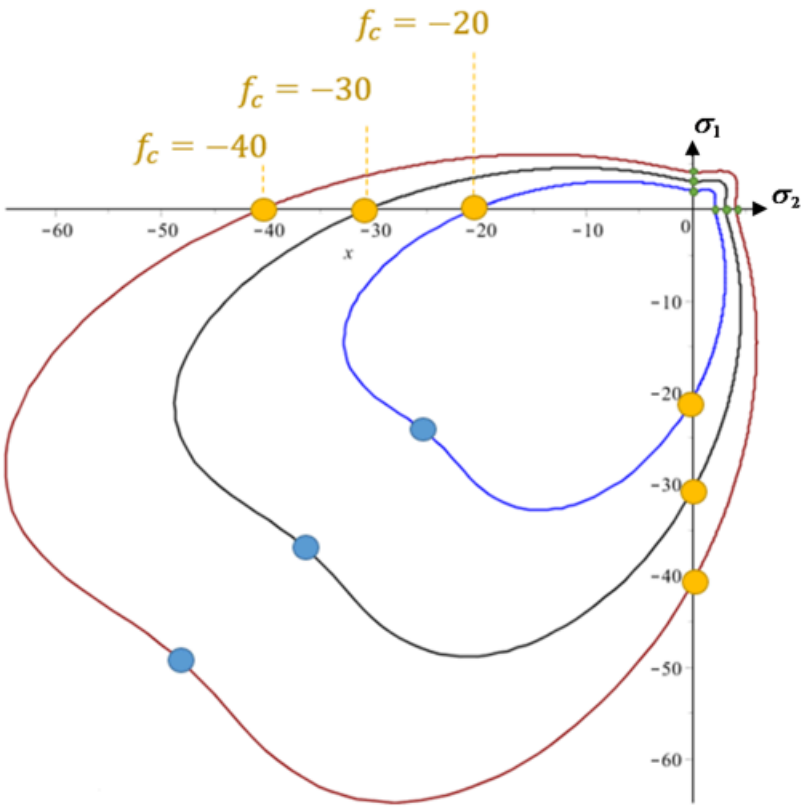


Figure 188: Influence of compressive strength value
All other ratios are fixed.

Here with same strength in LAW24, but different confined compression strength f_2.

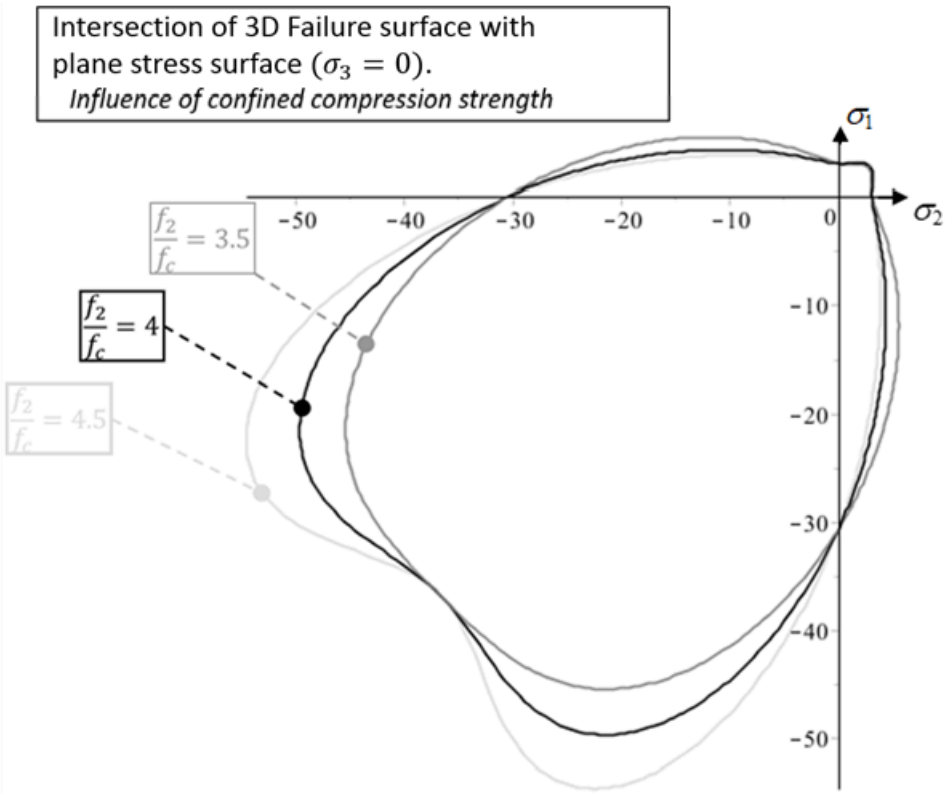


Figure 189:Failure envelope on the plane stress surface influenced by the triaxial failure point $(\sigma_1, \sigma_2, \sigma_3) = (f_2, s_0, s_0)$

f_c and the ratios f_t / f_c, and f_b / f_c in the $r - \sigma_m$ space (used to define the concrete failure) are:

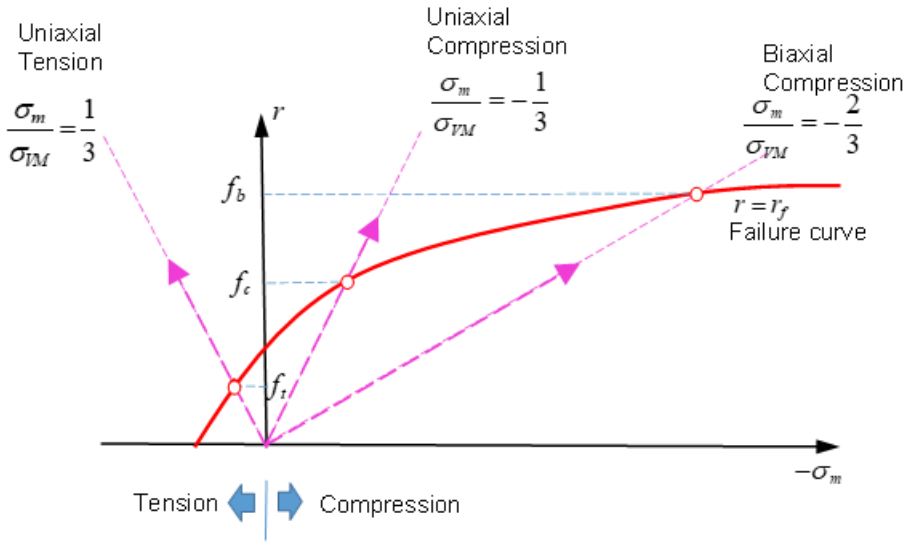


Figure 190: Different tests (uniaxial tension, uniaxial compression, and biaxial compression) to determine failure curve

Where the failure curve is defined using $r = \sqrt{2J_2} = \sqrt{\frac{2}{3}}\,\sigma_{VM}$ and $\sigma_m = \frac{I_1}{3}$ is the mean stress (pressure), then I_1 and J_2 are the first and second stress invariants.

The material fails once it reaches the failure curve r_f.

Concrete Reinforcement

In Radioss there are two different ways to simulate the reinforcement in concrete.

- One way is to use beam or truss elements and connect them to the concrete with kinematic conditions.
- Another way is to use the parameters in LAW24 along with the orthotropic solid property `/PROP/TYPE6` to define the reinforced direction. Parameters $\alpha_1,\ \alpha_2,\ \alpha_3$ in LAW24 are used to define the reinforcement cross-section area ratio to the whole concrete section area in direction 1, 2, 3.

$$\alpha_i = \frac{Area_{steel}}{Area_{concrete}} \tag{139}$$

Where, σ_y is the yield stress of the reinforcement. If steel is used as a reinforcement, then σ_y is the yield stress of steel and E_t is the modulus of steel in the plastic phase.

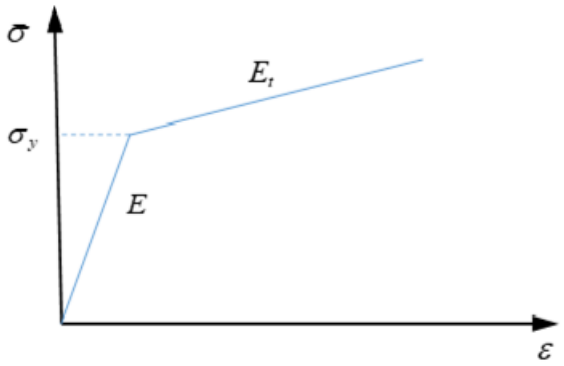


Figure 191: Stress-Strain Curve of Reinforcement (steel)

Concrete Material (/MAT/LAW81)

LAW81 can be used to model rock or concrete materials.

Drücker-Prager Yield Criteria

LAW81 uses a Drücker–Prager yield criterion where the yield surface and the failure surface are the same. The yield criteria is:

$$F = \underbrace{q}_{J_2\,\text{part}} - \underbrace{\mathrm{r}_c(p)\cdot(p\tan\phi + c)}_{I_1\,\text{part}} = 0 \qquad (140)$$

Where,

q	von Mises stress with $q = \sigma_{VM} = \sqrt{3J_2}$
p	Pressure is defined as $p = \frac{1}{3}I_1$

12. Han, D. J., and Wai-Fah Chen. "*A nonuniform hardening plasticity model for concrete materials.*" Mechanics of materials 4, no. 3-4 (1985): 283-302

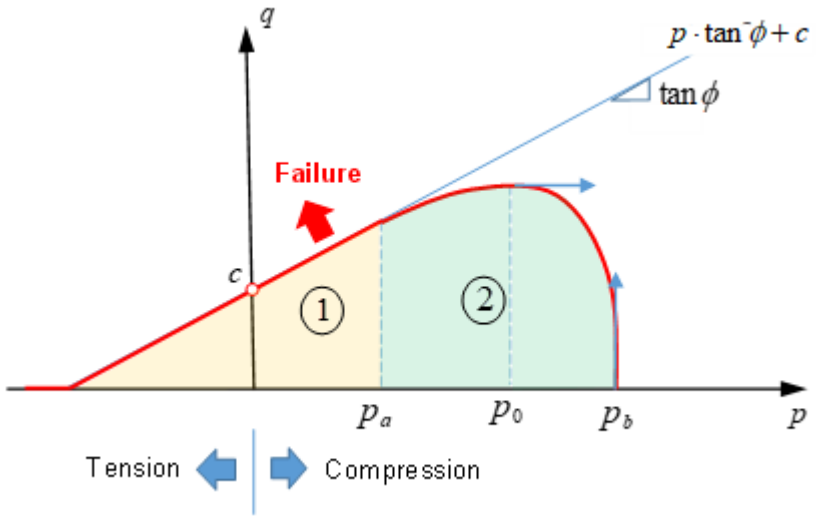


Figure 192: Yield Surface (LAW81)

The yield surface can be described in two parts:

- The linear part ① ($p \le p_a$), where the scale function is $r_c(p) = 1$ which leads to the von Mises stress being linearly proportional to pressure:

  $$q = p\tan\phi + c \qquad (141)$$

  Where,

c	Cohesive and is the intercept of yield envelope with the shear strength. If $c = 0$, the material has no strength under tension.
ϕ	Angle of internal friction, which defines the slope of the yield envelope.

  c and ϕ are also used to define the Mohr-Coulomb yield surface. The Drücker-Prage yield surface is a smooth version of the Mohr-Coulomb yield surface.

- The second part ② ($p_a < p < p_b$) of the yield surface simulates a cap limit. An increase of pressure in a rock or concrete material will increase the yield of the material; but, if pressure increases enough, then the rock or concrete material will be crushed. The Drücker-Prager model with the cap limit can be used to model this behavior. The cap limit defined in part ② and uses the scale function:

  $$r_c(p) = \sqrt{1 - \left(\frac{p - p_a}{p_b - p_a}\right)^2} \qquad (142)$$

  The von Mises stress is:

$$q = \sqrt{1 - \left(\frac{p - p_a}{p_b - p_a}\right)^2} \cdot (p\tan\phi + c) \quad (143)$$

Where,

p_b Curve is defined using the *fct_ID*$_{Pb}$ input

p_a Computed by Radioss using the input α ratio value.

$p_a = \alpha \cdot p_b$ with $0 < \alpha < 1$.

Where, p_0 is the maximum point of yield curve, where $\frac{\partial F}{\partial p}(p_0) = 0$

If $p = p_b$, then $r_c(p_b) = 0$ and the yield function is then,

$q = 0 \cdot (p\tan\phi + c) = 0$ which means the material is crushed.

The input parameters ϕ, c, p_b, α need to determine for the Drücker–Prager yield surface. At least four tests are needed to fit these parameters. In the simplest case, uniaxial tension and uniaxial compression can be used to determine the linear part, ϕ, and c. To determine p_b, and α biaxial compression tests and compression/compression tests are needed (refer to CC00 and CC01 in RD-E: 4701 Concrete Validation with Kupfer Tests).

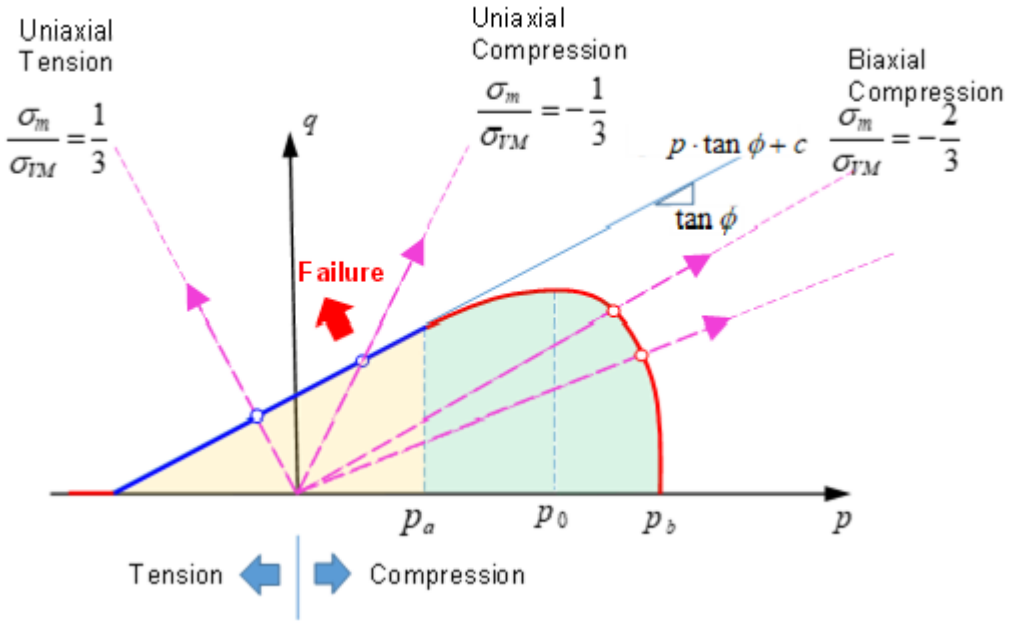


Figure 193: Yield Surface of LAW81 Showing Different Load Conditions

For most materials such as metal, the plastic strain increment could be considered normal to yield surface. However, if the plastic strain increment normal to yield surface is used for rock or concrete materials, the plastic volume expansion is overestimated. Therefore, a non-associated plastic flow rule is used in these materials. In LAW81 the plastic flow function G defined as:

- $G = q - p \cdot \tan\psi = 0$ if $p \le p_a$
- $G = q - \tan\psi\left(p - \frac{(p - p_a)^2}{2(p_0 - p_a)}\right) = 0$ if $p_a < p \le p_0$
- $G = F$ if $p > p_0$

Since the pressure is p_0, the yield function F and plastic flow function G are the same and the following condition is fulfilled:

$$G(p_0) = F(p_0) \tag{144}$$

$$\frac{\partial G}{\partial p}|_{p_0} = \frac{\partial F}{\partial p}|_{p_0} = 0 \tag{145}$$

The pressure p_0 can be calculated using the yield surface where $\frac{\partial F}{\partial p}|_{p_0} = 0$. With G defined as:

$$G = q - \tan\psi\left(p - \frac{(p - p_a)^2}{2(p_0 - p_a)}\right) = 0 \tag{146}$$

The parameter ψ can be determined using the von Mises stress at pressure, p_0 in the function.

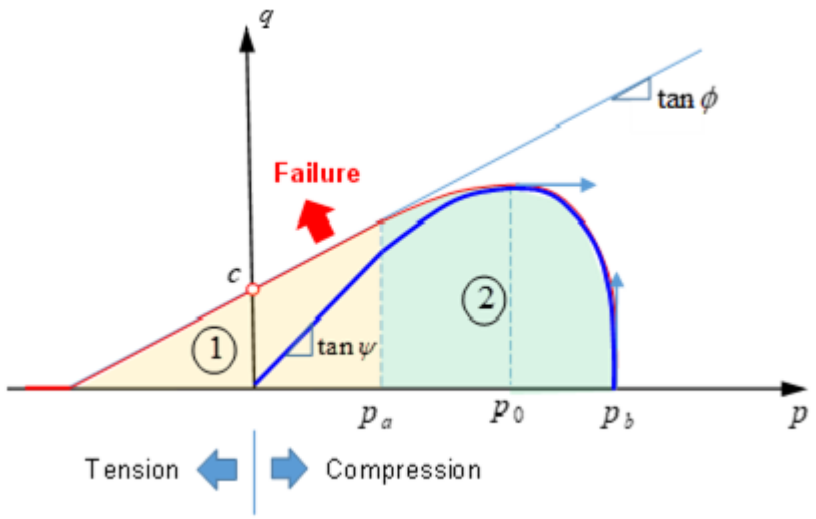


Figure 194: Yield Surface of LAW81 with Plastic Flow

Failure

Ductile

The /FAIL/BIQUAD, /FAIL/JOHNSON, and /FAIL/TAB1 failure models define material failure by relating the plastic strain at failure to the stress state in the material.

These failure models are often used to describe the ductile failure of materials. The state of stress in the material can be defined by using stress triaxiality.

Stress Triaxiality (Normalized Mean Stress)

For ductile materials, the state of stress (compression, shear, tension, etc.) of the material affects the plastic strain value at which the material will fail. An important and useful characteristic to describe the state of stress, stress triaxiality is defined as:

$$\sigma^* = \frac{\sigma_m}{\sigma_{VM}} \tag{147}$$

Where,

$\sigma_m = \frac{1}{3}(\sigma_1 + \sigma_2 + \sigma_3)$ Mean (hydrostatic) stress

$\sigma_{VM} = \sqrt{\frac{1}{2}\left[(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2\right]}$ von Mises stress

Triaxiality values for some commons stress states can be derived as:

- In pure tension:

  $\sigma_2 = \sigma_3 = 0$, then $\sigma^* = \frac{\sigma_m}{\sigma_{VM}} = \frac{1}{3}$
- In biaxial compression:

  $\sigma_1 = \sigma_2$, and $\sigma_3 = 0$, then $\sigma^* = \frac{\sigma_m}{\sigma_{VM}} = -\frac{2}{3}$

Stress triaxiality for various stress states:

Stress Triaxiality σ^*	**Stress State**
$-\frac{2}{3}$	Biaxial compression
$-\frac{1}{3}$	Uniaxial compression
0	Pure shear
$\frac{1}{3}$	Uniaxial tension
$\frac{1}{\sqrt{3}}$	Plain strain

Stress Triaxiality σ^*	**Stress State**
$\frac{2}{3}$	Biaxial tension

/FAIL/JOHNSON

The Johnson-Cook failure model is often used to describe the ductile failure of metals. It uses a Johnson-Cook equation to define failure strain as a function of stress triaxiality.

In the Johnson-Cook failure model, there are three parts to the failure model;

$$\varepsilon_f = \underbrace{\left[D_1 + D_2 \exp\left(D_3 \sigma^*\right)\right]}_{\substack{Influence\,of \\ triaxiality}} \underbrace{\left[1 + D_4 \ln\left(\dot{\varepsilon}^*\right)\right]}_{\substack{Influence\,of \\ strainrate}} \underbrace{\left[1 + D_5 T^*\right]}_{\substack{Influence\,of \\ temperature}} \quad (148)$$

Where,

ε_f	Plastic failure strain
$\dot{\varepsilon}^* = \frac{\dot{\varepsilon}}{\dot{\varepsilon}_0}$	Current strain rate divided by the input reference strain rate
T^*	Computed in the material law or `/HEAT/MAT`

Ignoring the influence of strain rate and temperature a plot of the Johnson-Cook failure is:

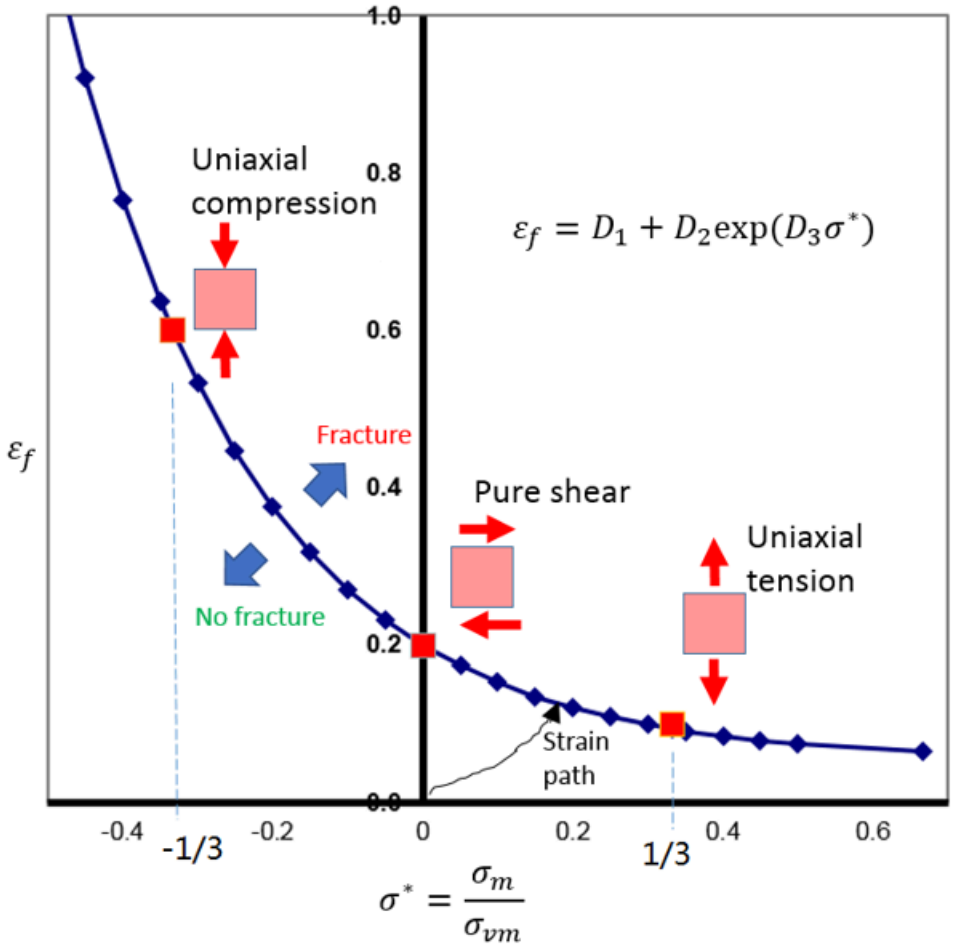


Figure 195: Example Plot of a Johnson-Cook Failure Model

Plastic strains above the curve represent material fracture and below the curve no material fracture.

In a simple case where only the triaxiality influence is considered, the failure strain is:

$$\varepsilon_f = D_1 + D_2 \cdot \exp(D_3 \cdot \sigma^*) \quad (149)$$

Using 3 failure data points from test:

$\varepsilon_f = 0.1585$ by uniaxial tension ($\sigma^* = 1/3$)

$\varepsilon_f = 0.19$ by pure shear ($\sigma^* = 0$)

$\varepsilon_f = 0.2419$ by uniaxial compression ($\sigma^* = -1/3$)

The parameters D_1, D_2 and D_3 could be calculated analytically by solving the following equations:

$$\begin{cases} 0.1585 = D_1 + D_2 \cdot \exp\left(D_3 \cdot \frac{1}{3}\right) \\ 0.19 = D_1 + D_2 \cdot \exp(D_3 \cdot 0) \\ 0.2419 = D_1 + D_2 \cdot \exp\left(D_3 \cdot -\frac{1}{3}\right) \end{cases} \quad (150)$$

Element Failure treatment

A cumulative damage method is used to sum the amount of plastic strain that has occurred in the element using:

$$D = \sum \frac{\Delta \varepsilon_p}{\varepsilon_f} \geq 1 \quad (151)$$

What happens when $D \geq 1$ depends on the values of element failure flags (I_{fail_sh} andI_{fail_so}) and XFEM formulation flag (I_{xfem}). When the XFEM formulation is not used (I_{xfem}=0), the following table summarizes the different element failure flag options:

Table 18: Element Failure Options

Element	Element Failure Flag	If $D \geq 1$	Failure Behavior
Shell	I_{fail_sh}=1 (Default)	In 1 IP or layer	Element deleted
Shell	I_{fail_sh}=2	In 1 IP or layer	Stress tensor set to zero in IP or layer
Shell	I_{fail_sh}=2	All IP or layer	Element deleted
Solid	I_{fail_sh}=1 (Default)	In 1 IP	Element deleted
Solid	I_{fail_sh}=2	In 1 IP	Stress tensor set to zero in IP
Solid	I_{fail_sh}=2	All IP	Stress tensor set to zero in element

Details on the XFEM formulation (I_{xfem}=1), can be found in `/FAIL/JOHNSON`.

The damage, D, can be plotted in animation files using `/ANIM/SHELL/DAMA` or `/ANIM/BRICK/DAMA`. This will show the risk of material damage.

/FAIL/BIQUAD

In Radioss, `/FAIL/BIQUAD` is the most user-friendly failure model for ductile materials. It uses a simplified, nonlinear strain-based failure criteria with linear damage accumulation.

The failure strain is described by two parabolic functions calculated using curve fitting from up to 5 user input failure strains.

By default, `/FAIL/BIQUAD` (`S-Flag`=1) uses two parabolic curves to describe the plastic failure strain ε_f, as a function of stress triaxiality σ^*. The two parabolic curves use:

$$f_1(x) = ax^2 + bx + c \quad (152)$$

$$f_2(x) = dx^2 + ex + f \quad (153)$$

Where,

a, b, c, d, e, **and** f	Parabolic coefficients
x	Stress triaxiality
$f_1(x)$ **and** $f_2(x)$	Plastic failure strain

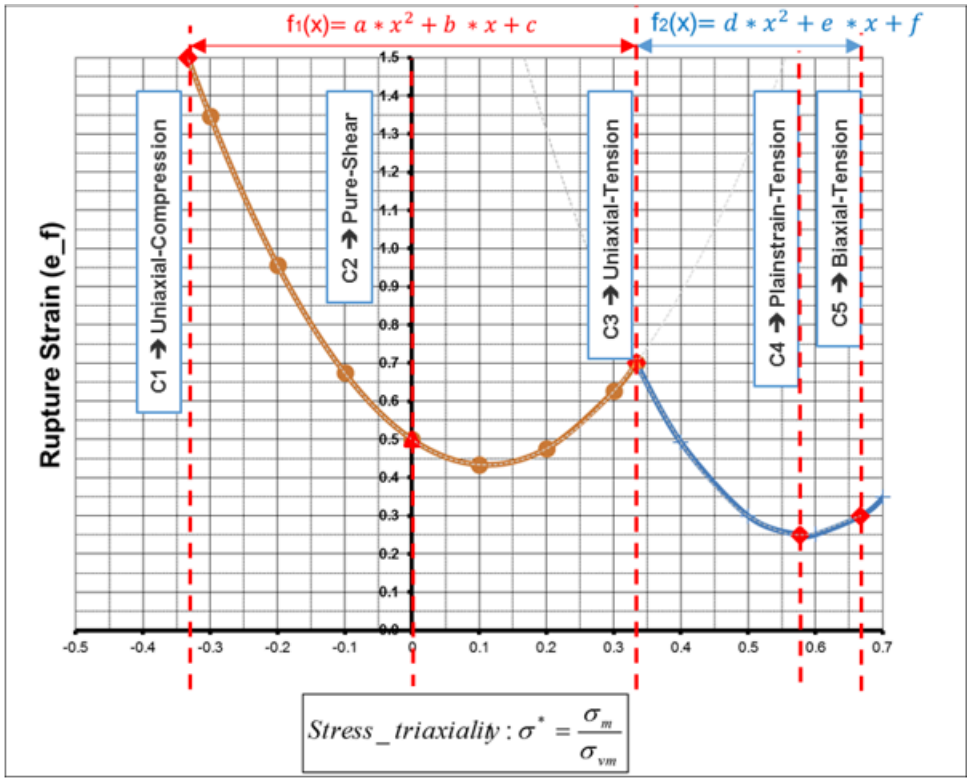


Figure 196: `/FAIL/BIQUAD` Failure Strain Curve Made of 2 Parabolic

The parabolic coefficients a, b, c, d, e, and f are computed by Radioss using a curve fit based on the plastic failure strain *c1-c5* input values. If the calculated parabolic failure strain curves have negative failure strain values, these negative values will be replaced by a failure strain of 1E-6 which results in a very high damage accumulation and brittle behavior. The results of the curve fit are in the Starter `*0000.out` file.

```
      Bi-Quadratic FAILURE
       --------------------
     c1. . . . . . . . . . . . . . . . . .=  0.2419E+00
     c2. . . . . . . . . . . . . . . . . .=  0.1900E+00
     c3. . . . . . . . . . . . . . . . . .=  0.1585E+00
     c4. . . . . . . . . . . . . . . . . .=  0.1437E+00
     c5. . . . . . . . . . . . . . . . . .=  0.1394E+00

      COEFFICIENTS OF FIRST PARABOLA
      ------------------------------
     a . . . . . . . . . . . . . . . . . .=  0.9180E-01
     b . . . . . . . . . . . . . . . . . .= -0.1251E+00
     c . . . . . . . . . . . . . . . . . .=  0.1900E+00
```

```
 COEFFICIENTS OF SECOND PARABOLA
 -------------------------------
d . . . . . . . . . . . . . . . . . . . .=  0.3753E-01
e . . . . . . . . . . . . . . . . . . . .= -0.9483E-01
f . . . . . . . . . . . . . . . . . . . .=  0.1859E+00
```

The *c1–c5* plastic failure strains definitions are:

c1	Plastic failure strain in uniaxial compression
c2	Plastic failure strain in shear
c3	Plastic failure strain in uniaxial tension
c4	Plastic failure strain in plane strain tension
c5	Plastic failure strain in biaxial tension

M-Flag Input Options

Depending on the *M-Flag* input option, there are three different ways to define the *c1-c5* values.

MI-*M-Flag*=0, User-defined Test Data

For this case, you must enter *c1-c5* which represents the plastic failure strain for the 5 different stress states. Ideally this data would be obtained from test or the material supplier.

MII-*M-Flag*=1-7, Predefined Material Data

If failure strain data is not available, you can pick from 7 predefined materials. Figure 197 shows the plastic strain at failure curves for the 7 materials.

> **Note:** The predefined values are supplied for early design exploration and it is your responsibility to verify that their material has the same properties.

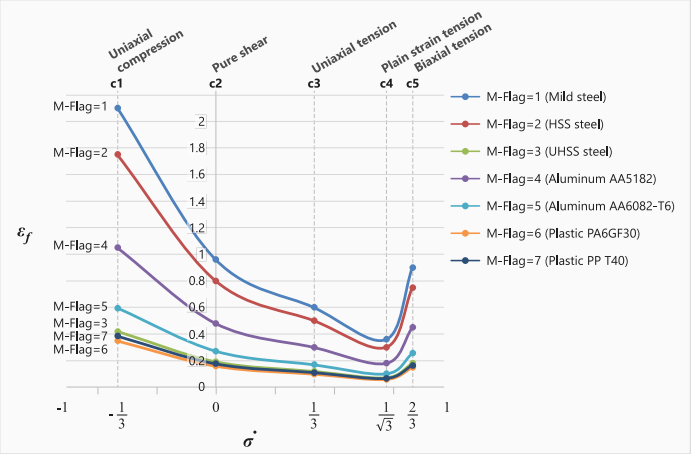


Figure 197: Predefined Material Failure Curves

- *M-Flag*=99, Plastic Failure Strain Ratio Input, *r1-r5*

  The last input method is to enter the plastic failure strain in uniaxial tension, *c3*, and plastic failure strain ratios for the other four stress states. These ratios are defined as:

r1	Failure plastic strain ratio, Uniaxial Compression (*c1*) to Uniaxial Tension (*c3*), so $c1 = r1 \cdot c3$
r2	Failure plastic strain ratio, Pure Shear (*c2*) to Uniaxial Tension (*c3*), so $c2 = r2 \cdot c3$
r4	Failure plastic strain ratio, Plane Strain Tension (*c4*) to Uniaxial Tension (*c3*), so $c4 = r4 \cdot c3$
r5	Failure plastic strain ratio, Biaxial Tension (*c5*) to Uniaxial Tension (*c3*), so $c5 = r5 \cdot c3$

  Using this method, it is easy to change the failure curve by adjusting the single plastic failure strain in uniaxial tension value, *c3*.

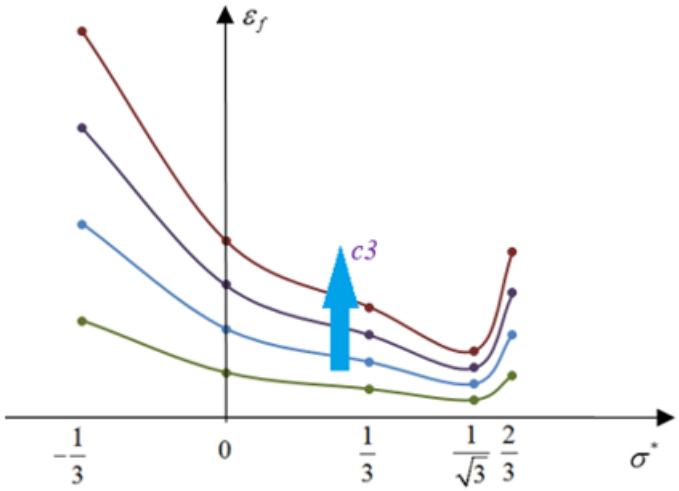


Figure 198: Changes in Plastic Failure Strain Curve
by increasing the uniaxial tension failure, c3, with the same failure plastic strain ratios

Default Behavior

By default, values different than 0 need to be entered for *c1* to *c5*. However, specific default behaviors exists, in case failure information are missing.

- In case the material failure behavior is unknown, *c1* to *c5* are set to 0.0 and the mild steel behavior (*M-Flag*=1) is used.
- If only the tensile failure value is known, *c3* is defined ($c1 = c2 = c4 = c5 = 0.0$). The mild steel behavior is used and scaled by the user- defined *c3* value.
- In case the material behavior is known, *M-Flag* is defined and *c3* can be used to adjust the failure model according the expected tensile failure. The selected material behavior is scaled by the user-defined *c3* value.
- For all other cases, all *c1* to *c5* are intended to be defined and default value of 0.0 is used.

Element Failure Treatment

A cumulative damage method is used to sum the amount of plastic strain that has occurred at each integration point in the element using:

$$D = \sum \frac{\Delta\varepsilon_p}{\varepsilon_f} \geq 1 \qquad (154)$$

Where,

D	Damage
$\Delta\varepsilon_p$	The change in plastic strain of the integration point
ε_f	Plastic failure strain for the current stress triaxiality

In shell elements after an integration point reaches $D = 1$, the integration point's stress tensor is set to zero. The element fails and is deleted when the ratio of through thickness failed integration points equals *P_thickfail*. In solid elements, the element is deleted when any integration point reaches $D = 1$.

Plane Strain as Global Minimum

The *S-Flag*=2 option can be used to force the global minimum of the plastic failure strain curve to occur at the plane strain stress triaxiality location, *c4*. This is accomplished by splitting the second equation into 2 separate quadratic sub-functions.

Modeling Material Instability (Localized Necking)

In materials such as sheet metal, thickness thinning and diffuse necking may appear during tensile loading of the material. This is called *localized necking* and normally occurs in the stress triaxiality range of $\frac{1}{3} \le \sigma^* \le \frac{2}{3}$:

Figure 199:

In `/FAIL/BIQUAD` it is possible to simulate this localized necking using the option, *S-Flag*=3 and *Inst_start*. This option uses the same plastic failure strain curve as *S-Flag*=2 and adds two additional quadratic functions that define a curve that represents the start of localized necking between stress triaxiality $\frac{1}{3}$ and $\frac{2}{3}$. The minimum value of this curve is a user-defined value in the *Inst_start* field and occurs at plane strain tension $\sigma^* = \frac{1}{\sqrt{3}}$. Using this localized necking curve, a second localized necking damage value is calculated and failure due to necking only occurs when all integration points reach $D = 1$. The localized necking criteria is based on the Marciniak-Kuczynski analysis. [13]

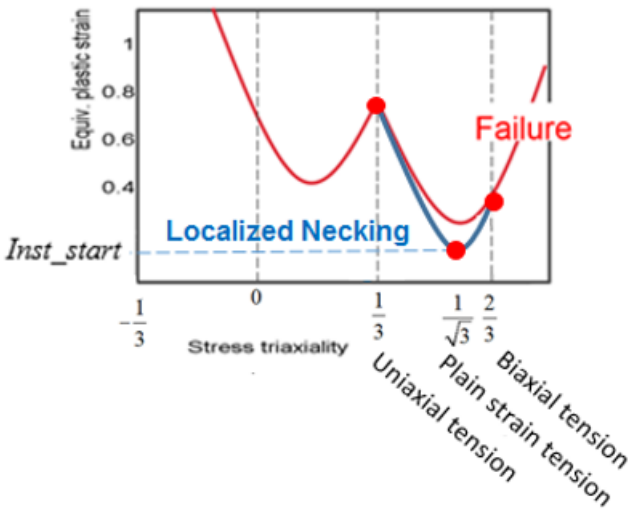


Figure 200: Default Failure Strain Curve
with additional localized necking curve (blue)

When using *S-Flag*=1 or 2, the damage accumulation begins once the plastic strain reaches in failure curve (red in Figure 201).

If *S-Flag*=3is used to describe the localized necking, damage accumulation begins once the plastic strain reaches in the localized necking curve (blue curve in Figure 201). For localized necking, the element is deleted when all integration points reach damage, $D = 1$, whereas element deletion not due to localized necking is defined by *P_thickfail*.

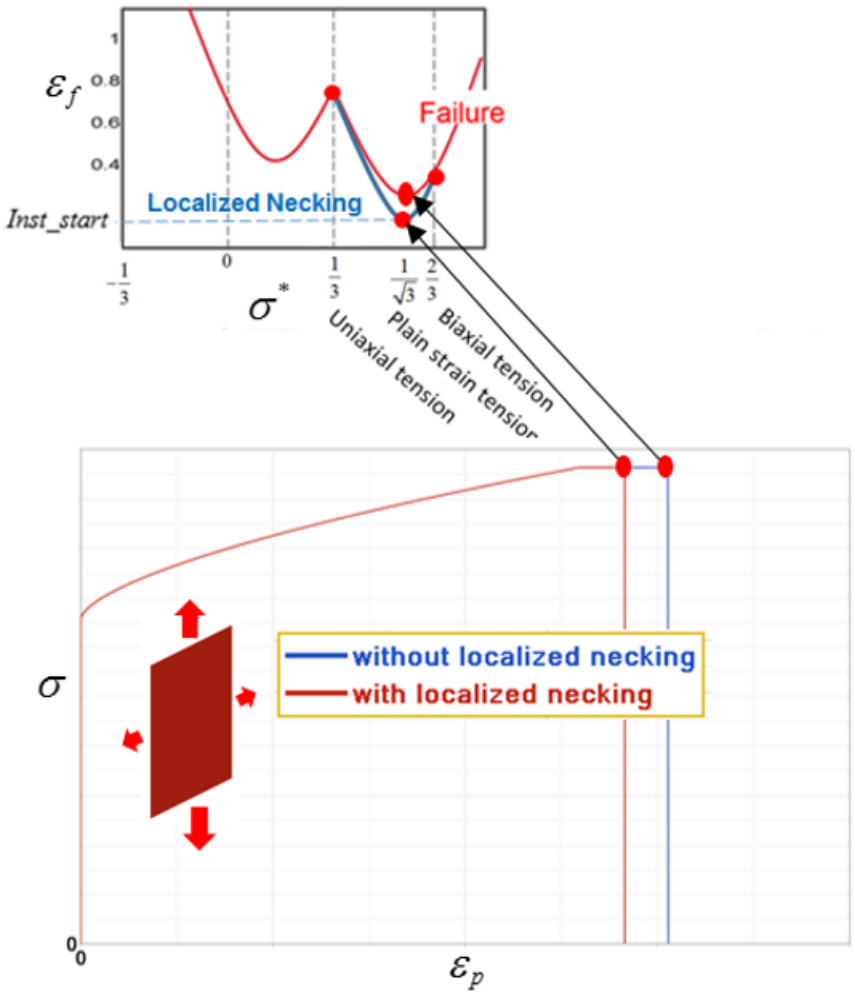


Figure 201:

Perturbation of the Failure Limit

Due to a materials imperfection or production process, a material's failure strain may not be exactly the same everywhere and thus very small perturbations of failure limit may exist. When using *M-Flag*>0 with `/PERTURB/FAIL/BIQUAD`, a statistical distribution of the failure limit is applied to each element assigned the failure model. This is accomplished by calculating the normal or random distribution of a failure scale factor which is applied to `/FAIL/BIQUAD`, *c3*. The two different distributions methods are shown in Figure 202 and Figure 203.

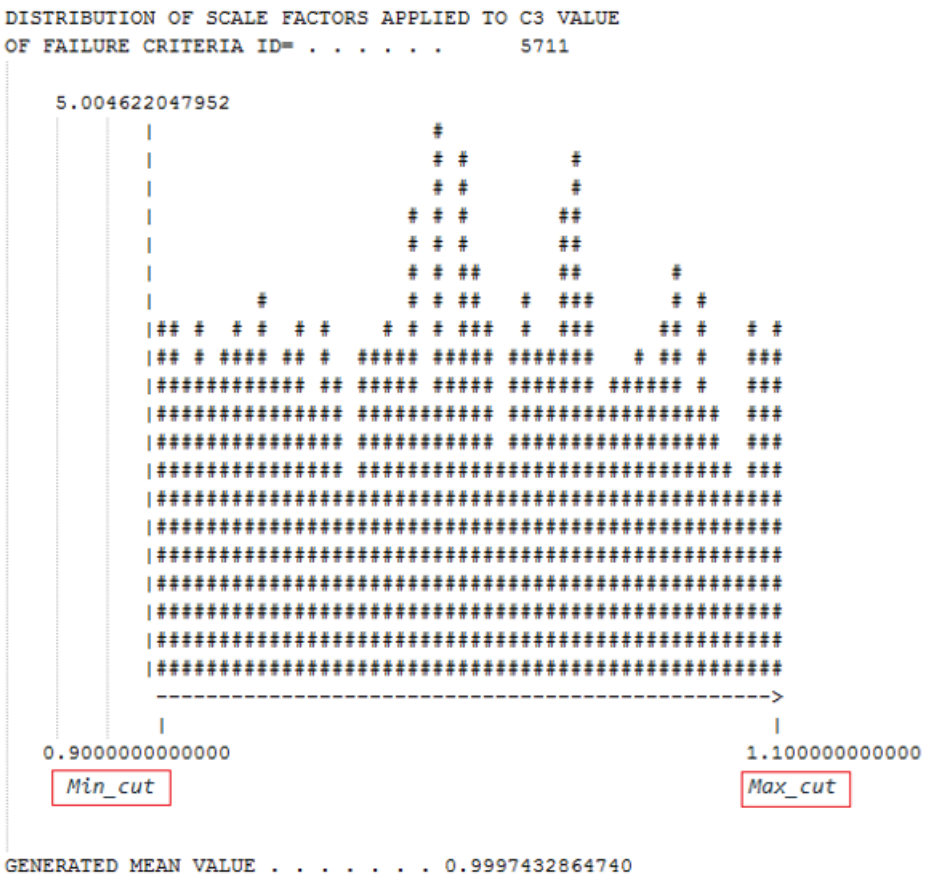


Figure 202: Random Distribution

Idistri=1, of the failure limit in the Starter `*0.out` file

Figure 203: Normal (Gaussian) Distribution
Idistri=2, of the failure limit in the Starter `*0.out` file

`/FAIL/BQUAD` uses small perturbations of *c3* generated by `/PERTURB/FAIL/BIQUAD` to scale the entire failure curve using the strain ratio values for the predefined materials or user-defined ratios, *r1*, *r2*, *r4* and *r5*, depending on the value of *M-Flag*.

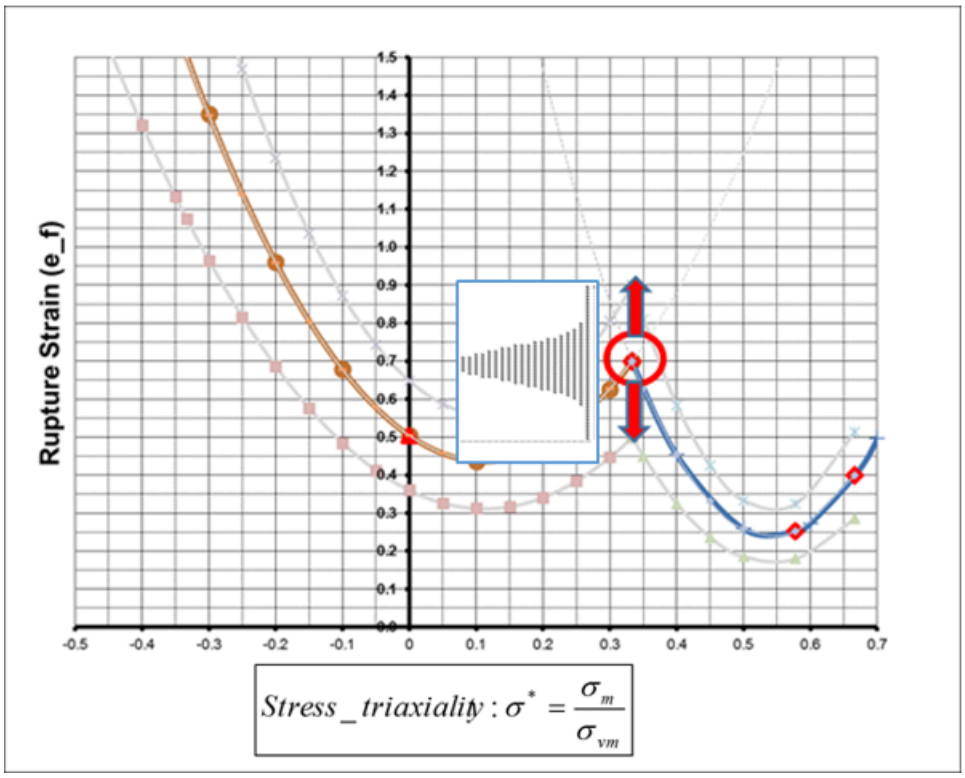


Figure 204:

Reference

Tabulated Failure Model /FAIL/TAB1

In Radioss, /FAIL/TAB1 is the most sophisticated failure model for ductile material. The plastic failure strain can be defined as a function of: stress triaxiality, strain rate, Lode angle, element size, temperature, and instability strain.

Damage is accumulated based on user-defined functions. The functionality of this failure model will be described starting with the most basic input to the most complex options.

Plastic Failure Strain

Similar to /FAIL/JOHNSON and /FAIL/BIQUAD, it is possible to define a curve that represents the plastic failure strain, ε_f, as a function of stress triaxiality, σ^*. Unlike /FAIL/JOHNSON and /FAIL/BIQUAD where the failure strain curve is defined using parameters in predefined equations, in /FAIL/TAB1 any number of discrete points can be entered to create an arbitrary function that represents the failure strain curve. This curve is defined using the /TABLE entity and is referenced in *table_ID*$_1$. This method can be used for shell and solid elements.

13. Pack, Keunhwan, and Dirk Mohr. "Combined necking & fracture model to predict ductile failure with shell finite elements." *Engineering Fracture Mechanics* 182 (2017): 32-51

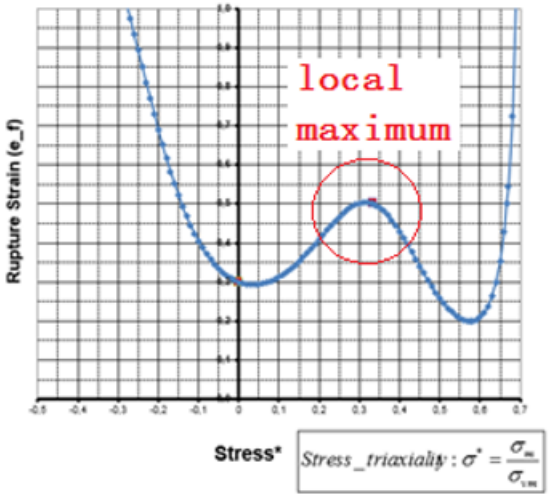


Figure 205: Material Failure Curve Defined using Discrete Points with a Local Maximum

Example `/TABLE`, *dimension*=1

Input for failure plastic-strain versus triaxiality using *table_ID*$_1$.

```
/TABLE/1/4711
failure plastic-strain vs triaxiality
#dimension
         1
#         Triaxiality      Failure_Strain
              -0.7000              0.3386
              -0.6000              0.3068
              -0.5000              0.2794
              -0.4000              0.2558
              -0.3333              0.2419
              -0.3000              0.2355
              -0.2000              0.2180
              -0.1000              0.2029
               0.0000              0.1900
               0.1000              0.1789
               0.2000              0.1693
               0.3000              0.1610
               0.3333              0.1585
               0.4000              0.1539
               0.5000              0.1478
               0.6000              0.1425
               0.7000              0.1380
```

Strain Rate Dependency

`/FAIL/TAB1` can also include the influence of strain rate on material failure. For this case the `/TABLE` must be defined such that, the first dimension is the function ID for the failure curve and the second dimension is the strain rate where that failure curve is applied.

Example `/TABLE`, *dimension*=2

```
/TABLE/1/4711
failure plastic-strain vs triaxiality and strain rate
#dimension
         2
#    FCT_ID                    strain_rate
       3000                          1E-4
       3001                           0.1
       3002                           1.0
/FUNCT/3000
```

```
failure plastic-strain vs triaxiality
#        Triaxiality      Failure_Strain
              -0.7000              0.3386
              -0.6000              0.3068
              -0.5000              0.2794
              -0.4000              0.2558
              -0.3333              0.2419
              -0.3000              0.2355
              -0.2000              0.2180
              -0.1000              0.2029
               0.0000              0.1900
               0.1000              0.1789
               0.2000              0.1693
               0.3000              0.1610
               0.3333              0.1585
               0.4000              0.1539
               0.5000              0.1478
               0.6000              0.1425
               0.7000              0.1380
/FUNCT/3001
failure plastic-strain vs triaxiality
#        Triaxiality      Failure_Strain
                 -0.7             0.27088
                 -0.6             0.24544
                 -0.5             0.22352
                 -0.4             0.20464
              -0.3333            0.19352
                 -0.3              0.1884
                 -0.2              0.1744
                 -0.1             0.16232
                    0                0.152
                  0.1             0.14312
                  0.2             0.13544
                  0.3              0.1288
               0.3333              0.1268
                  0.4             0.12312
                  0.5             0.11824
                  0.6               0.114
                  0.7              0.1104
```

Lode Angle with Solid Elements

For solid elements, the failure strain can also depend on the 3D stress state defined using the Lode angle.

This can be included by adding failure strain as a function of Lode angle parameter in the `/TABLE` entity referenced by *table_ID1*. For shells elements, it is only necessary to define the failure strain as a function of stress triaxiality. But for solid elements, it is more accurate to include the failure strain as a function of stress triaxiality and Lode angle.

In Radioss, the Lode angle is entered using a normalized and dimensionless Lode Angle parameter ξ and is defined here.

The stress state at a point, *P*, could be expressed with principle stresses ($\sigma_1, \sigma_2, \sigma_3$) or could also be expressed using stress invariants (I_1, J_2, J_3). The advantage of using stress invariants is that they are constant and do not depend on the orientation of the coordinate system. In Figure 206, to correctly describe stress state of point *P* $\sigma_1, \sigma_2, \sigma_3$ using stress invariants, the magnitude of $\mathbf{OO'}$ as:

$$\sqrt{3}\sigma_m = \frac{\sqrt{3}}{3} I_1 \qquad (155)$$

Where,

σ_m Mean stress

I_1	First stress invariant $I_1 = \sigma_1 + \sigma_2 + \sigma_3$

OO' is in the hydrostatic axis, which means the principle stress in this axis is the same ($\sigma_1 = \sigma_2 = \sigma_3$). **|OO'|** is the hydrostatic pressure.

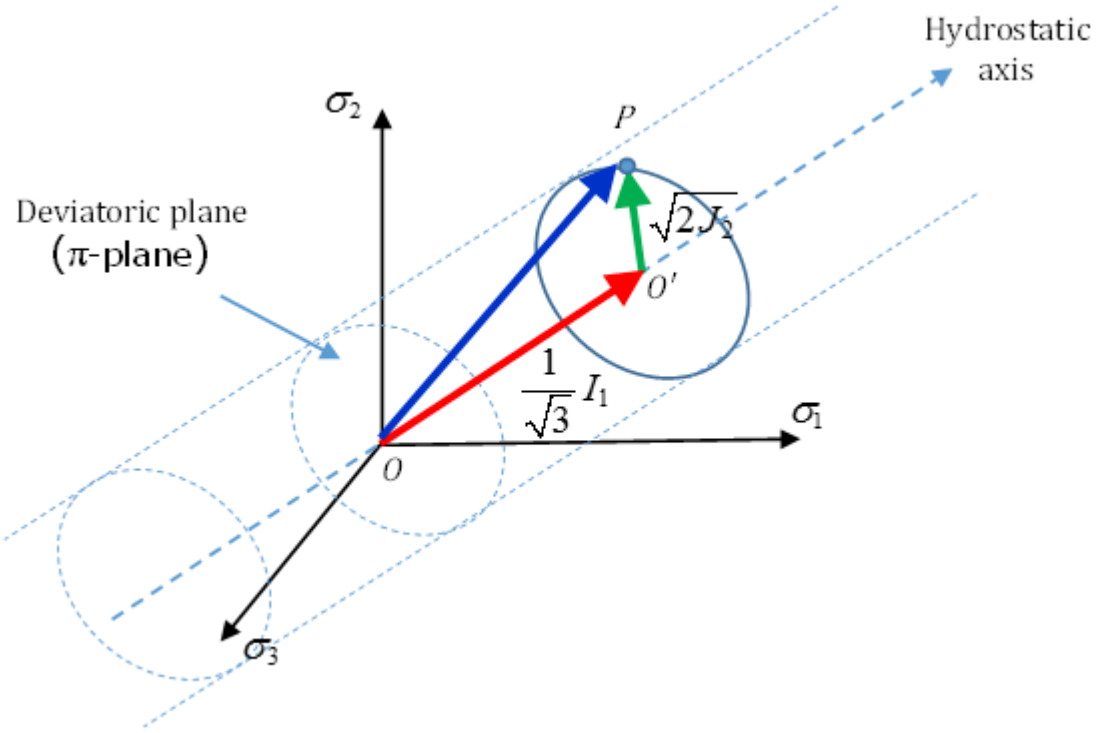


Figure 206: Stress State of P

The magnitude of **O'P** is:

$$\sqrt{2J_2} = \sqrt{\frac{2}{3}}\,\sigma_{VM} \tag{156}$$

Here, J_2 is the second invariant of deviatoric stress $\mathbf{s}(\mathbf{s} = \boldsymbol{\sigma} - \mathbf{p})$ with
$J_2 = \frac{1}{2}(S_1^2 + S_2^2 + S_3^2) = \frac{1}{2}[(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2]$.

To identify the point *P*, the angle in the circular plane must be calculated. This angle is called the Lode Angle θ:

$$\cos(3\theta) = \frac{27}{2}\frac{J_3}{\sigma_{VM}^3} = \frac{3\sqrt{3}}{2}\frac{J_3}{J_2^{3/2}} \tag{157}$$

with $0 \le \theta \le \frac{\pi}{3}$ and J_3 is the third invariant of deviatoric stress calculated as:

$$J_3 = S_1 S_2 S_3 \tag{158}$$

In `/FAIL/TAB1` a normalized and dimensionless Lode Angle parameter ξ with the range $(-1 \le \xi \le 1)$ is used and defined as:

$$\xi = \cos(3\theta) \tag{159}$$

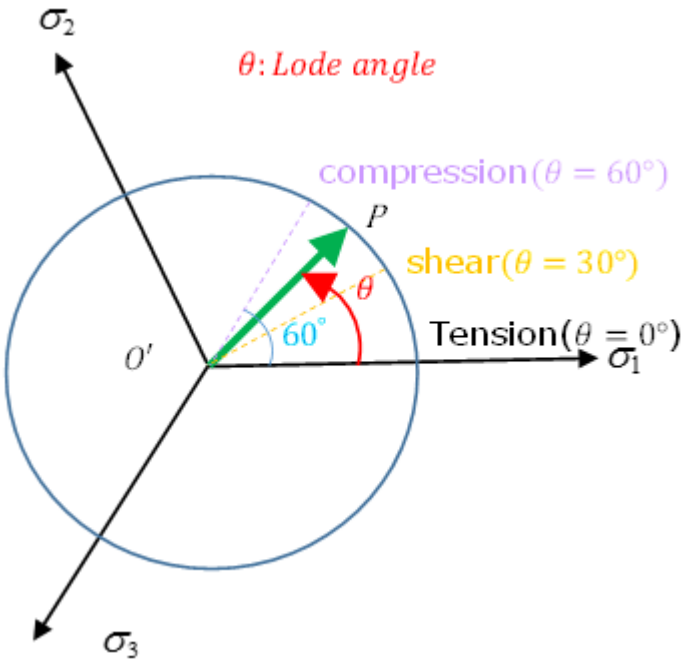


Figure 207: Stress State with Different Lode Angle

The special characters for Lode Angle θ and Load Angle Parameter ξ are:

Lode Angle Parameter ξ	**Lode Angle** θ	**Stress State**
1	0	Uniaxial tension + hydrostatic pressure (triaxial tension or axisymmetric tension)
0	30	Pure shear + hydrostatic pressure (plane strain)
-1	60	Uniaxial compression + hydrostatic pressure (axisymmetric compression)

A failure strain surface can be created from the stress triaxiality and Lode angle failure data.

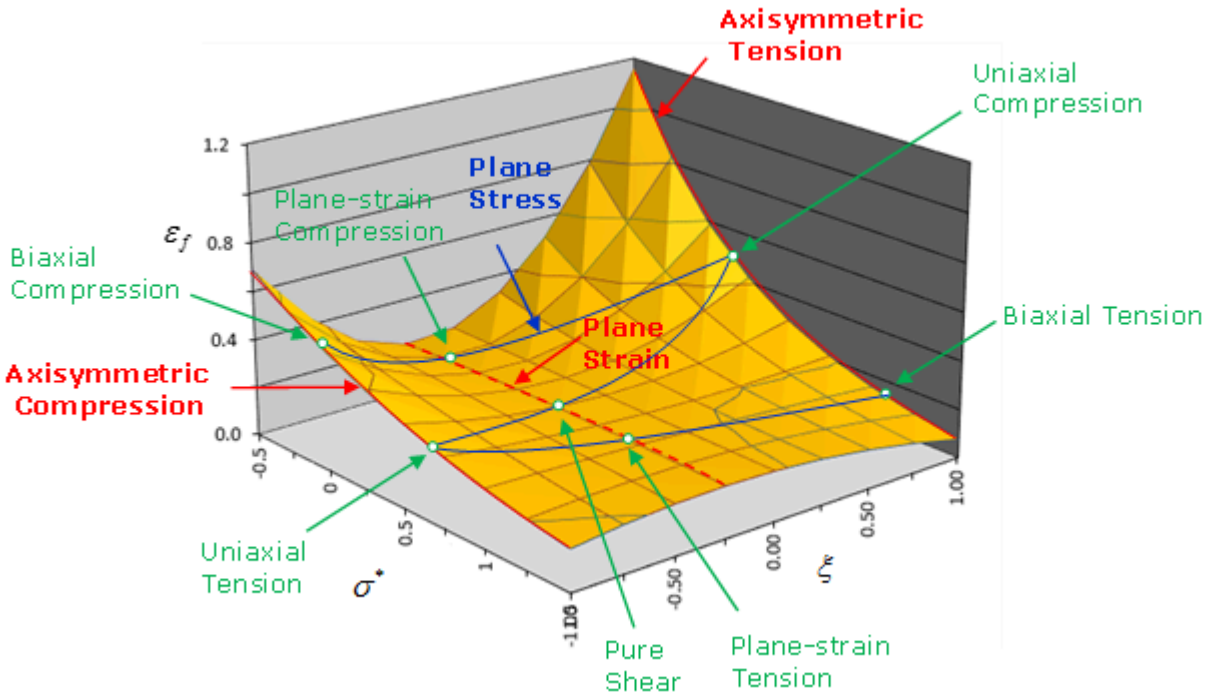


Figure 208: 3D Failure Surface

The material failure surface could be created using the following material tests.

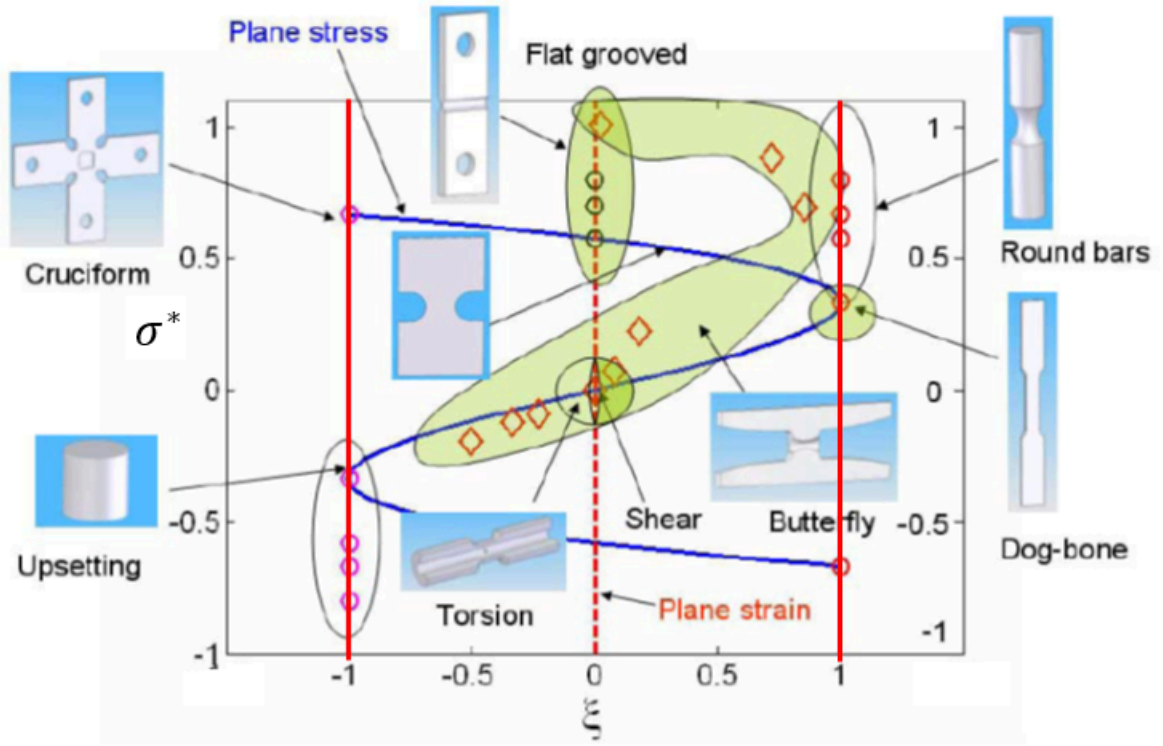


Figure 209: Stress State and Lode Angle for Various Tests

Example `/TABLE`, *dimension*=3

Input for failure plastic-strain versus triaxiality, strain rate, and Lode angle using *table_ID*$_1$

```
/TABLE/1/4711
failure plastic-strain vs triaxiality and strain rate
#dimension
         3
#---1----|----2----|----3----|----4----|----5----|----6----|----7----|----8----|----9----|---10----|
#    FCT_ID                  strain_rate          Lode_angle
      3000                        1E-4                  -1
      3001                         0.1                   0
      3002                         1.0                   1
....
```

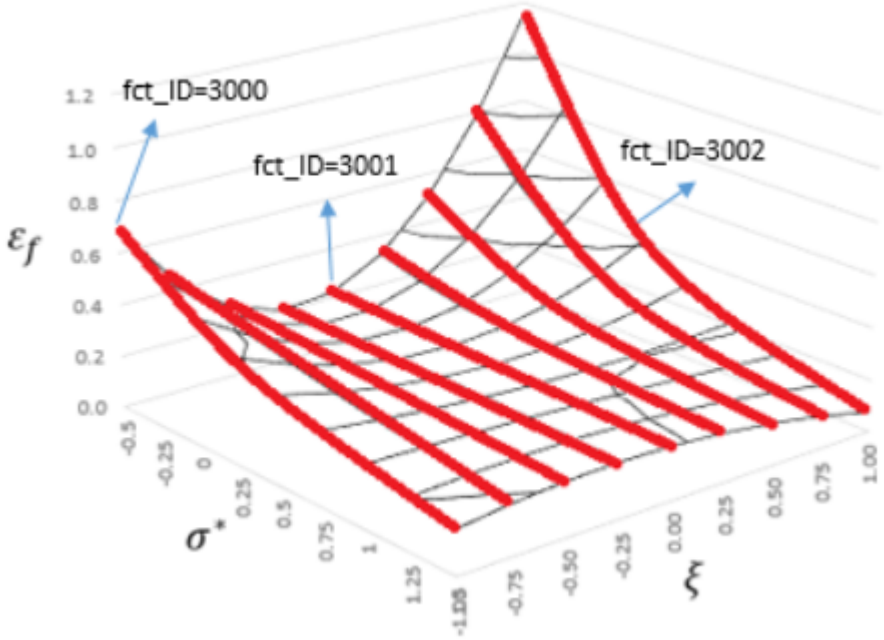


Figure 210: Failure Surface when table_ID$_1$ references a `/TABLE` *with dimension=3*

Scaling Failure Strain

Material failure based on temperature and element size can be considering in `/FAIL/TAB1` by including functions that scale the failure strain depending on the element size and/or temperature using:

$$\varepsilon_f = Xscale1 \cdot f(\sigma^*, \dot{\varepsilon}, \xi) \cdot factor_{el} \cdot factor_T \qquad (160)$$

Where,

$Xscale1$	General scaling factor
$factor_{el}$	Scale factor based on element size
$factor_T$	Scale factor based on temperature

Element Length Dependency

In numerical simulations, the element size will affect the material failure. Using the same failure parameters, a coarse mesh will fail earlier than a fine mesh.

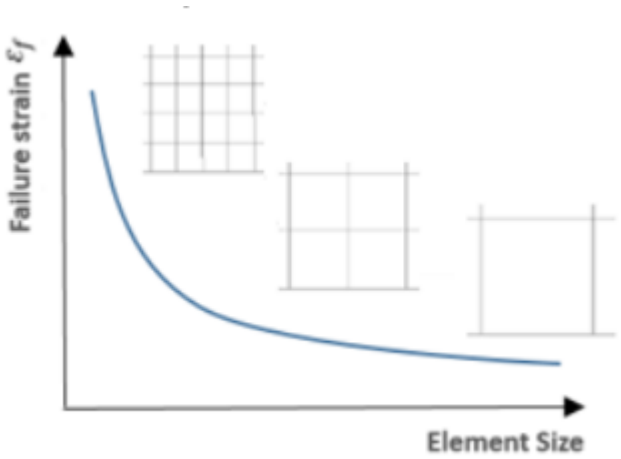


Figure 211: Influence of Element Mesh Size on Material Failure in Numerical Simulation

To account for the variation in results based on mesh size, the element size scale factor can be defined to scale the failure strain based on mesh element size. The scale factor defined in Equation 160 is:

$$factor_{el} = Fscale_{el} \cdot \mathrm{f}_{el}\left(\frac{Size_{el}}{El\_ref}\right) \quad (161)$$

Where,

$Fscale_{el}$	Element size function scale factor
$\mathrm{f}_{el}\left(\frac{Size_{el}}{El\_ref}\right)$	failure strain scale factor as a function of the normalized element size (defined via *fct_ID*$_{el}$), and El_ref is the reference element size used to normalize the element size

For example, a mesh size of 2mm was used in a material validation simulation. After the initial validation, the same simulation is reran using El_ref =2 with different element sizes to determine the correct scaling factor for each element size. Using the results of the second validation, a failure strain scale factor function is constructed as (Figure 212):

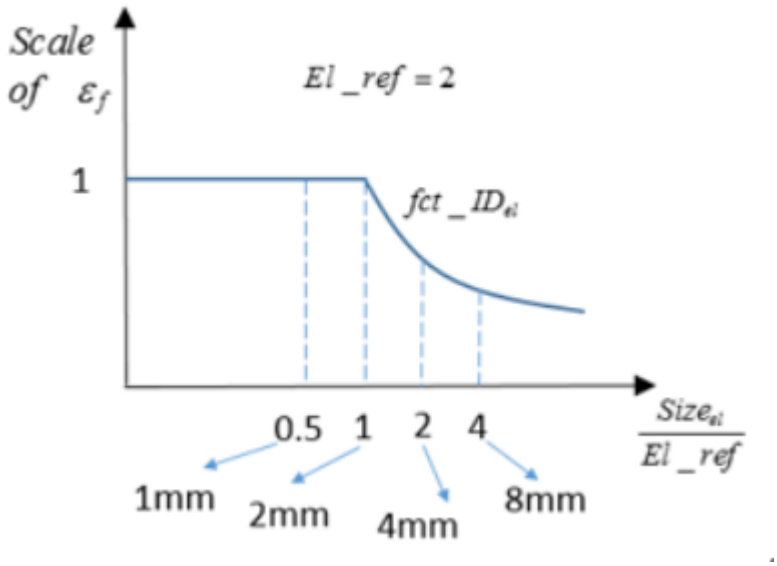


Figure 212: Example of Element Size Scale Factor Function fct_ID$_{el}$ in `/FAIL/TAB1`

Temperature Dependency

To account for how the temperature affects material failure, failure strain values defined in Equation 160 can be scaled using a temperature scale factor function:

$$factor_T = Fscale_T \cdot \mathrm{f}_T(T_{start}) \tag{162}$$

Where,

$Fscale_T$	Temperature function scale factor
$\mathrm{f}_T(T_{start})$	Failure strain scale factor as a function temperature defined via *fct_ID$_T$*.

Scaling the failure strain based on temperature works with any material that has `/HEAT/MAT` defined or materials that include thermos plasticity such as `/MAT/LAW2 (PLAS_JOHNS)`.

In *fct_ID$_T$*, the temperature is defined relative to the melting and initial temperature.

$$T^* = \frac{T - T_{ini}}{T_{melt} - T_{ini}} \tag{163}$$

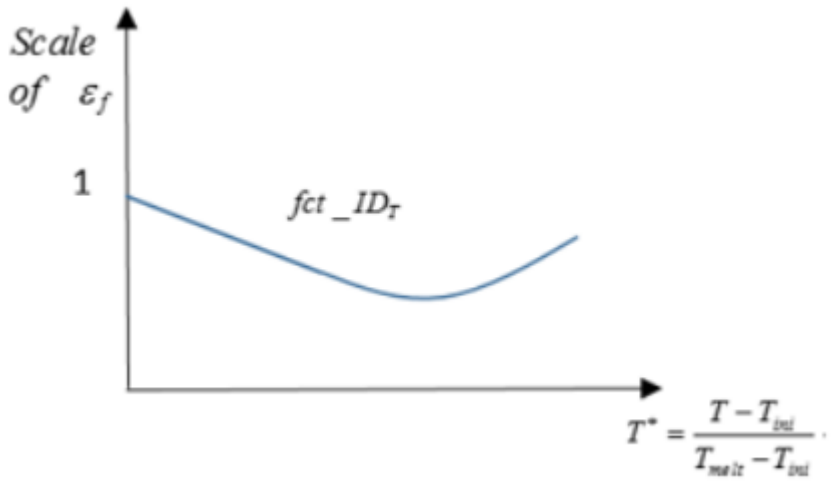


Figure 213: Influence of Temperature on Material Failure with fct_ID$_T$ in `/FAIL/TAB1`

Element Failure Treatment

In `/FAIL/TAB1` an accumulative damage model is used. The damage can be output for contour plotting using the Engine options, `/ANIM/SHELL/DAMA` or `/ANIM/BRICK/DAMA`.

The accumulated damage is calculated by first calculating a damage increment:

$$\Delta D = \frac{\Delta \varepsilon_p}{\varepsilon_f} \cdot n \cdot D_p^{\left(1-\frac{1}{n}\right)} \quad (164)$$

Where,

$\Delta \varepsilon_p$	The change in plastic strain of the integration point
ε_f	Plastic failure strain for the current stress triaxiality
D_p **and** n	Damage parameters

The accumulated damage is:

$$D = \frac{\sum \Delta D}{D_{crit}} \quad (165)$$

Where, D_{crit} is defined in `/FAIL/TAB1` with a recommended between 0 and 1. The elements fail when $D \geq 1$ which means $\sum \Delta D \geq D_{crit}$.

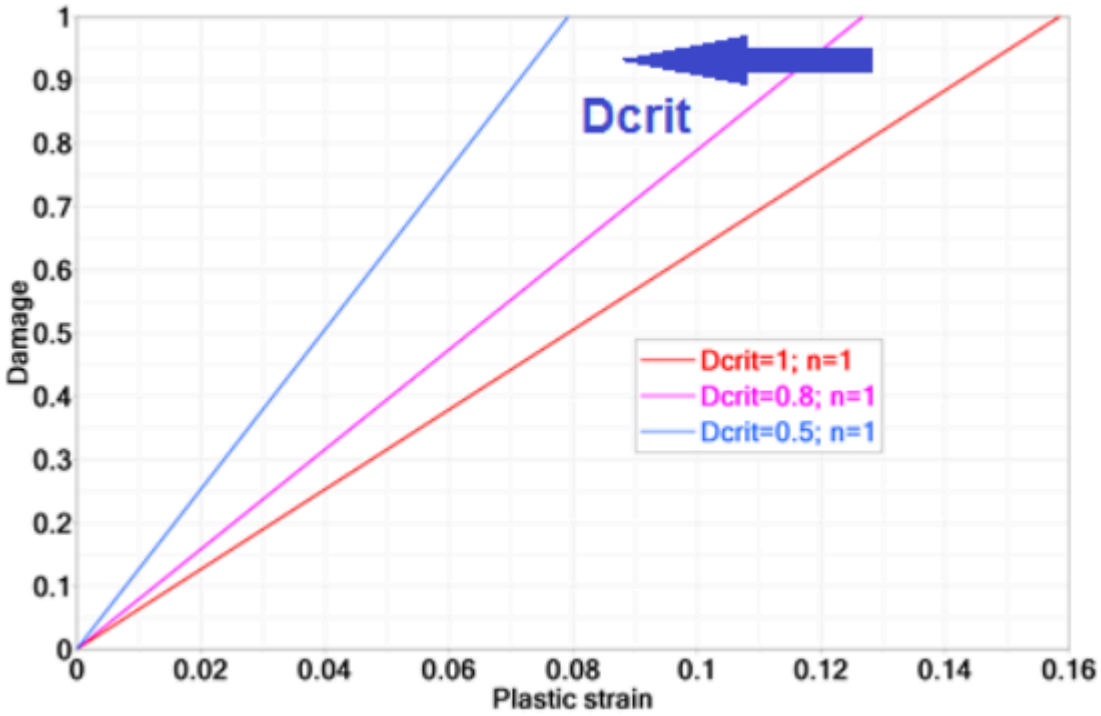


Figure 214: Influence of D_{crit} in `/FAIL/TAB1`

It is also interesting to understand the influence of the damage accumulation parameter, n in Equation 164.

When n=1 damage is linear, otherwise the damage curve is nonlinear.

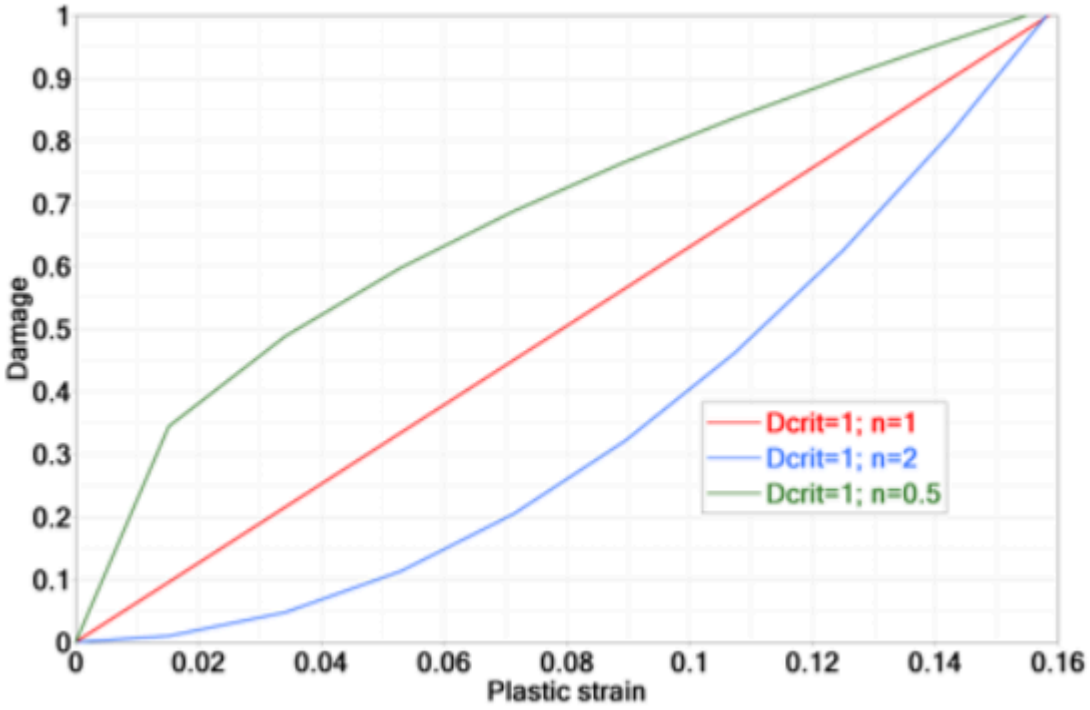


Figure 215:Influence of Damage Parameter n

Material Instability (Diffuse Necking)

In a tensile test, a material will reach a maximum engineering stress and then the stress will decrease or sometimes called *soften*. This maximum engineering stress point is called the *necking point*. After necking, the true stress actually increases, due to the decrease in the cross sectional area. This is called *diffuse necking*. In sheet metal, thickness thinning or localized necking in diffuse area may appear if the material continues to be loaded in the tensile direction. The diffuse necking normally appears in stress triaxiality ranges of $0 < \sigma^* \leq \frac{2}{3}$ and localized necking in the range of $\frac{1}{3} \leq \sigma^* \leq \frac{2}{3}$.

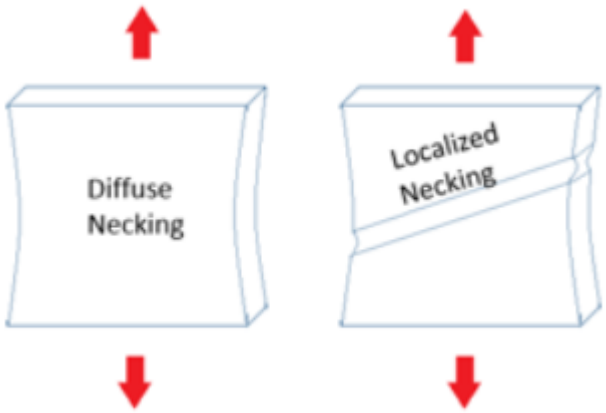


Figure 216: Sketch of Diffuse Necking and Localized

In `/FAIL/TAB1` it is possible to account for the material instability (diffuse necking) with options *table_ID*$_2$, or *Inst_start* and *Fad_exp*.

The material will start to weaken due to diffuse necking when the instability strain is reached. The reduced stress in the material is:

$$\sigma_{reduced} = \sigma \cdot \left(1 - \left(\frac{D_{instability} - inst\_start}{1 - inst\_start}\right)^{Fad\_\exp}\right) \quad (166)$$

Where,

$$D_{instability} = \sum \frac{\Delta \varepsilon_p}{\varepsilon_f} \quad (167)$$

with ε_f being the diffuse necking strain based on the current stress triaxiality.

The strain at which instability starts could be either input with a curve (blue curve in Figure 217) using *table_ID*$_2$ or input as a constrain strain using the option *Inst_start*.

In a uniaxial tensile test $\sigma^* = \frac{1}{3}$.

- If material instability is not included then damage is calculated using the red failure curve in Figure 217
- If material instability is included and modeled using the curve input (*table_ID*$_2$) then:
  - Damage due to diffuse necking starts when the plastic strain defined by the blue curve in Figure 217 is reached.
  - Damage due to diffuse necking is linear if *Fad_exp*=1 is used. Increasing the *Fad_exp* leads to more energy dissipated during damage. Figure 217 shows the influence of *Fad_exp* from 1 to 10 in stress-strain curve. It is recommend to use a *Fad_exp* value of 5 to 10.
  - Once the strain reaches the red curve the element fails.

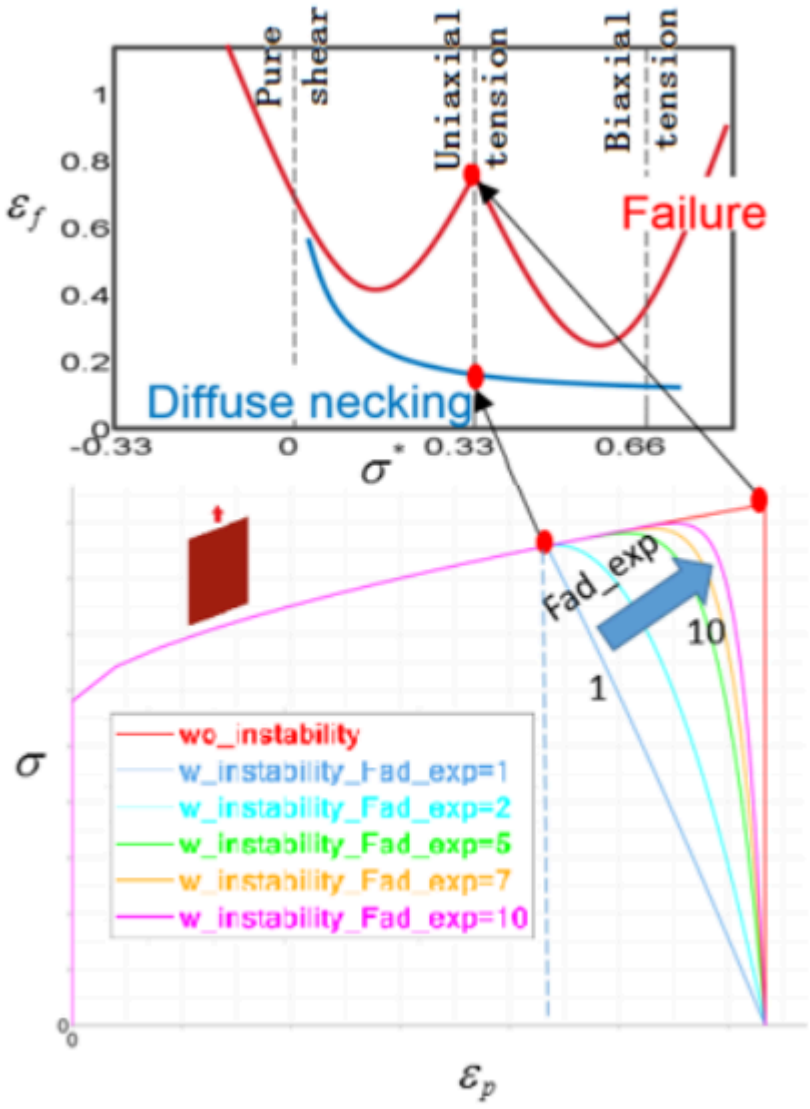


Figure 217: Influence of Parameter Fad_exp and Material Instability Region

- If only *Inst_start* is used without curve input in *table_ID*$_2$, then the diffuse necking plastic strain is the constant value, *Inst_start*, for all stress triaxiality.

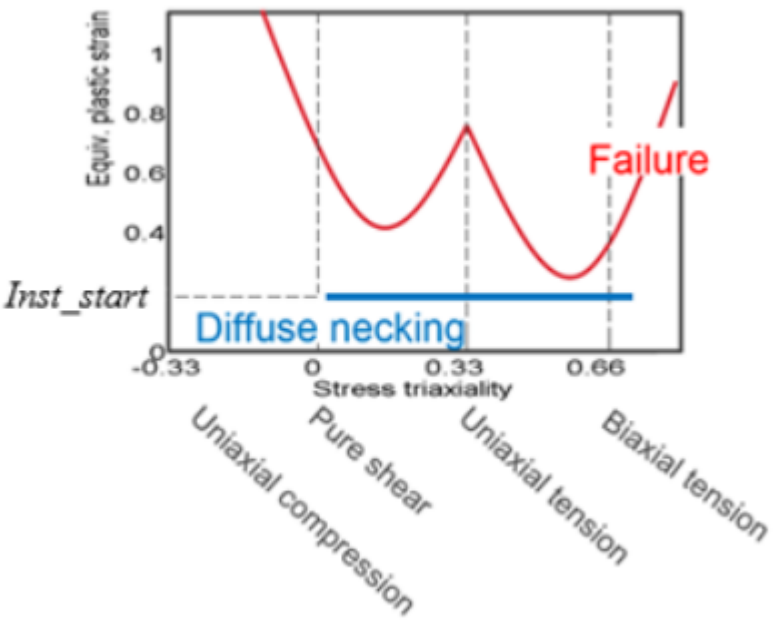


Figure 218: Constant Inst_start Describing Diffuse Necking

Currently diffuse necking (material instability) in `/FAIL/TAB1` can only be used with material law numbers > 28.

14. Wierzbicki, Tomasz, "*Addendum to the Research Proposal on Fracture of Advanced High Strength Steels*", page 19, January 2007.
15. Wierzbicki, Tomasz. "*Fracture of AHSS Sheets–Addendum to the Research Proposal on Fracture of Advanced High Strength Steels*." Impact and Crashworthiness Laboratory, Massachusetts Institute of Technology (2007).

Composites

Composite materials consist of two or more materials combined each other. Most composites consist of two materials, binder (matrix) and reinforcement. Reinforcements come in three forms, particulate, discontinuous fiber, and continuous fiber.

Composite Material

In Radioss the following material laws are used to describe composite material:

- `LAW12` and `LAW14`
- `LAW15` (recommended to use `LAW25`+`/FAIL/CHANG`)
- `LAW25`
- `LAW19` (For Fabric and with only `/PROP/TYPE9`)
- `LAW58` (For Fabric only with `/PROP/TYPE16`)

Table 19: Composite Material, Property, Failure Model, and Element Type Compatibility

	Shell Element	Brick Element	Failure Model
LAW12		/PROP/TYPE6 (SOL_ORTH) /PROP/TYPE21 (TSH_ORTH) /PROP/TYPE22 (TSH_COMP)	/FAIL/HASHIN /FAIL/PUCK /FAIL/LAD_DAMA
LAW14		/PROP/TYPE6 (SOL_ORTH) /PROP/TYPE21 (TSH_ORTH) /PROP/TYPE22 TSH_COMP)	/FAIL/HASHIN /FAIL/PUCK /FAIL/LAD_DAMA
LAW15	/PROP/TYPE9 (SH_ORTH) /PROP/TYPE10 (SH_ORTH) /PROP/TYPE11 (SH_SANDW) /PROP/TYPE17 (STACK)		/FAIL/CHANG

	Shell Element	Brick Element	Failure Model
	/PROP/TYPE19 (PLY)		
LAW25	/PROP/TYPE10 (SH_ORTH) /PROP/TYPE11 (SH_SANDW) /PROP/TYPE17 (STACK) /PROP/TYPE19 (PLY) /PROP/TYPE51 /PROP/PCOMPP	/PROP/TYPE6 (SOL_ORTH) /PROP/TYPE14 (SOLID) /PROP/TYPE20 (TSHELL) /PROP/TYPE21 (TSH_ORTH) /PROP/TYPE22 TSH_COMP)	/FAIL/CHANG /FAIL/HASHIN /FAIL/PUCK /FAIL/LAD_DAMA (for solid only)
LAW19	/PROP/TYPE9 (SH_ORTH)		
LAW58	/PROP/TYPE16 (SH_FABR)		

LAW12 and LAW14

Describes orthotropic solid material which use the Tsai-Wu formulation. The materials are 3D orthotropic-elastic, before the Tsai-Wu criterion is reached. LAW12 is a generalization and improvement of LAW14.

Elastic Phase

Both material laws require Young's modulus, shear modulus and Poisson ratio (9 parameters) to describe the material orthotropic in elastic phase.

E_{11}	E_{22}	E_{33}
ν_{12}	ν_{23}	ν_{31}
G_{12}	G_{23}	G_{31}

Figure 219:

$$\begin{bmatrix}\varepsilon_{11}\\ \varepsilon_{22}\\ \varepsilon_{33}\\ \gamma_{12}\\ \gamma_{23}\\ \gamma_{31}\end{bmatrix} = \begin{bmatrix}\frac{1}{E_{11}} & -\frac{\nu_{12}}{E_{11}} & -\frac{\nu_{31}}{E_{33}} & 0 & 0 & 0\\ & \frac{1}{E_{22}} & -\frac{\nu_{23}}{E_{22}} & 0 & 0 & 0\\ & & \frac{1}{E_{33}} & 0 & 0 & 0\\ & & & \frac{1}{2G_{12}} & 0 & 0\\ & symm. & & & \frac{1}{2G_{23}} & 0\\ & & & & & \frac{1}{2G_{31}}\end{bmatrix} \cdot \begin{bmatrix}\sigma_{11}\\ \sigma_{22}\\ \sigma_{33}\\ \sigma_{12}\\ \sigma_{23}\\ \sigma_{31}\end{bmatrix} \quad (168)$$

Stress Damage

σ_{t1}	σ_{t2}	σ_{t3}	δ

Figure 220:

Stress limits σ_{t1}, σ_{t2} and σ_{t3} (in tensile/compression) are requested for damage. These stress limits could be observed from a tensile test in 3 related directions.

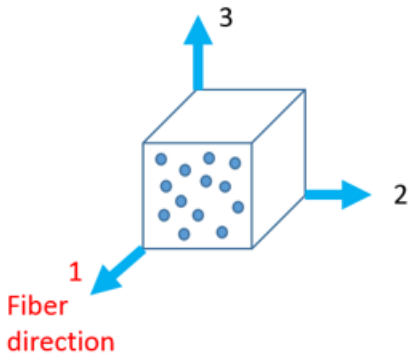


Figure 221:

Once stress limit is reached, then damage to material begins (stress reduced with damage parameter δ). If Damage ($D_i = D_i + \delta$) reaches D=1, then stress is reduced to 0.

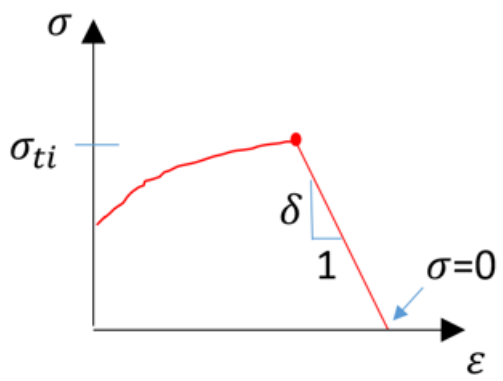


Figure 222:

Tsai-Wu Yield Criteria

In `LAW12 (3D_COMP)`, the Tsai-Wu yield criteria is:

$$F(\sigma)=F_1\sigma_1+F_2\sigma_2+F_3\sigma_3+F_{11}\sigma_1^2+F_{22}\sigma_2^2+F_{33}\sigma_3^2+F_{44}\sigma_{12}^2+F_{55}\sigma_{23}^2 \\ +F_{66}\sigma_{31}^2+2F_{12}\sigma_1\sigma_2+2F_{23}\sigma_2\sigma_3+2F_{13}\sigma_1\sigma_3 \quad (169)$$

The 12 coefficients of the Tsai-Wu criterion could be determined using the yield stress from the following tests:

Tensile/Compression Tests

- Longitude tensile/compression (in direction 1):

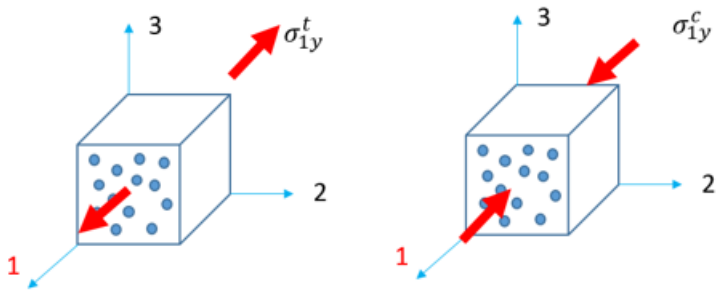


Figure 223:

$$F_1=-\frac{1}{\sigma_{1y}^c}+\frac{1}{\sigma_{1y}^t} \quad (170)$$

$$F_{11}=\frac{1}{\sigma_{1y}^c\sigma_{1y}^t} \quad (171)$$

- Transverse tensile/compression (in direction 2):

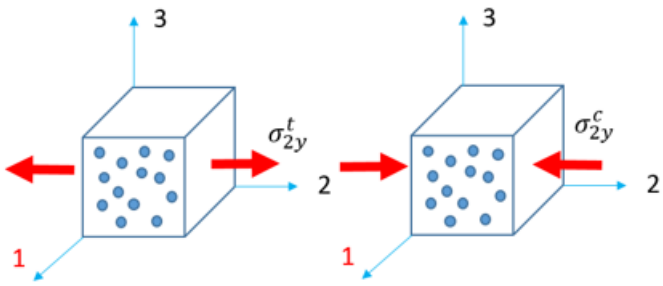


Figure 224:

$$F_2 = -\frac{1}{\sigma_{2y}^{c}} + \frac{1}{\sigma_{2y}^{t}} \tag{172}$$

$$F_{22} = \frac{1}{\sigma_{2y}^{c}\sigma_{2y}^{t}} \tag{173}$$

- Transverse tensile/compression (in direction 3):

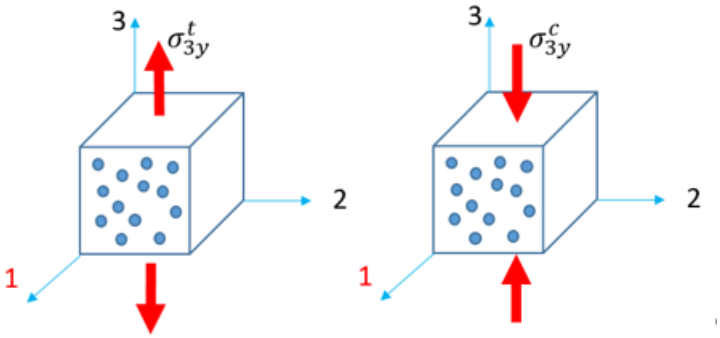


Figure 225:

$$F_3 = -\frac{1}{\sigma_{3y}^{c}} + \frac{1}{\sigma_{3y}^{t}} \tag{174}$$

$$F_{33} = \frac{1}{\sigma_{3y}^{c}\sigma_{3y}^{t}} \tag{175}$$

Then the interaction coefficients can be calculated as:

$$F_{12} = -\frac{1}{2}\sqrt{(F_{11}F_{22})} \tag{176}$$

$$F_{23} = -\frac{1}{2}\sqrt{(F_{22}F_{33})} \tag{177}$$

$$F_{13} = -\frac{1}{2}\sqrt{(F_{11}F_{33})} \tag{178}$$

Shear Tests

- Shear in plane 1-2 test:

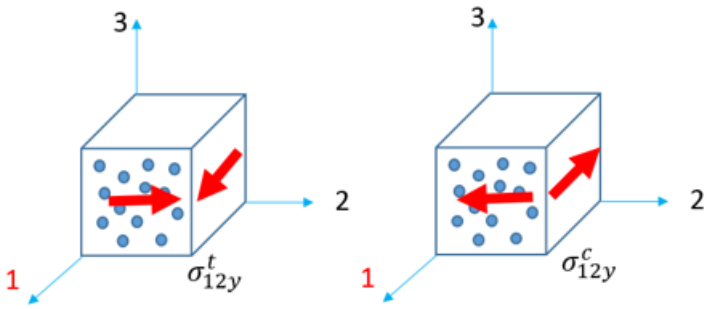


Figure 226:

σ^t_{12y} and σ^c_{12y} can result from the sample tests below:

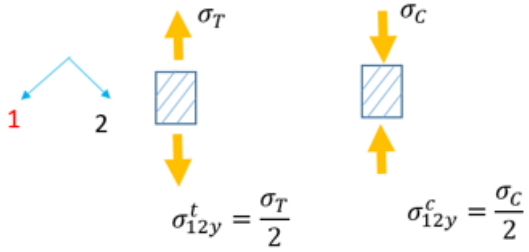


Figure 227:

$$F_{44} = \frac{1}{\sigma^c_{12y}\sigma^t_{12y}} \quad (179)$$

- Shear in plane 1-3

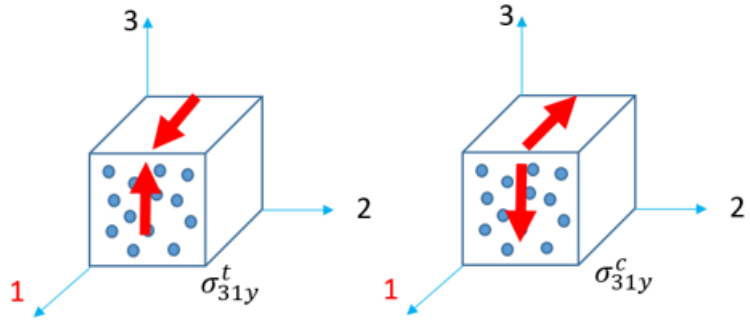


Figure 228:

σ^t_{31y} and σ^c_{31y} can result from the sample tests below:

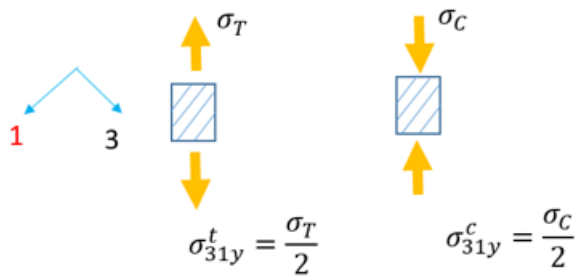


Figure 229:

$$F_{66} = \frac{1}{\sigma_{31y}^c \sigma_{31y}^t} \tag{180}$$

- Shear in plane 2-3:

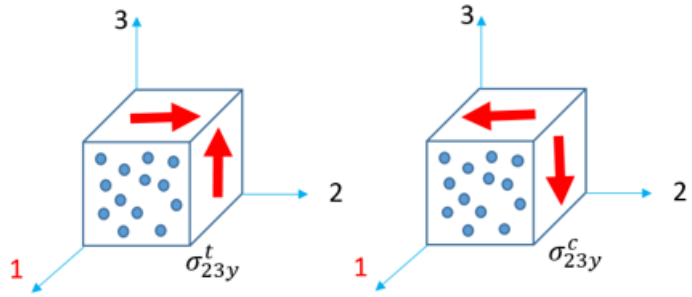


Figure 230:

$$F_{55} = \frac{1}{\sigma_{23y}^c \sigma_{23y}^t} \tag{181}$$

The parameters shown below in `LAW12` and `LAW14` are requested to calculate the Tsai-Wu criteria:

σ_{1y}^t	σ_{2y}^t	σ_{1y}^c	σ_{2y}^c
σ_{12y}^t	σ_{12y}^c	σ_{23y}^t	σ_{23y}^c
σ_{3y}^t	σ_{3y}^c	σ_{13y}^t	σ_{13y}^c

→ Additional parameters in LAW12

Figure 231:

The yield surface for Tsai-Wu is $F(\sigma) = 1$. As long as $(F(\sigma) \leq 1)$, the material is in the elastic phase. Once $(F(\sigma) > 1)$, the yield surface is exceeded and the material is in nonlinear phase.

In these two material laws, the following factors could also be considered for the yield surface.

- Plastic work W_p with parameter *B* and *n*
- Strain rate $\dot{\varepsilon}$ with parameter $\dot{\varepsilon}_0$ and *c*.

$$F(W_p, \dot{\varepsilon}) = (1 + BW_p^n)\left(1 + c\ln\frac{\dot{\varepsilon}}{\dot{\varepsilon}_o}\right) \quad (182)$$

Then the yield surface will be $F(\sigma) = F(W_p, \dot{\varepsilon})$.

- Material will be in elastic phase, if $F(\sigma) \le F(W_p, \dot{\varepsilon})$
- Material will be in nonlinear phase, if $F(\sigma) > F(W_p, \dot{\varepsilon})$

This yield surface $F(W_p, \dot{\varepsilon})$ will be limited with $f_{\max}$ ($F(W_p, \dot{\varepsilon}) \le f_{\max}$), where $f_{\max}$ is the maximum value of the Tsai-Wu criterion limit.

$$f_{\max} = \left(\frac{\sigma_{\max}}{\sigma_y}\right)^2$$

Depending on parameter *B*, *n*, *c* and $\dot{\varepsilon}_0$, the yield surface is between 1 and $f_{\max}$.

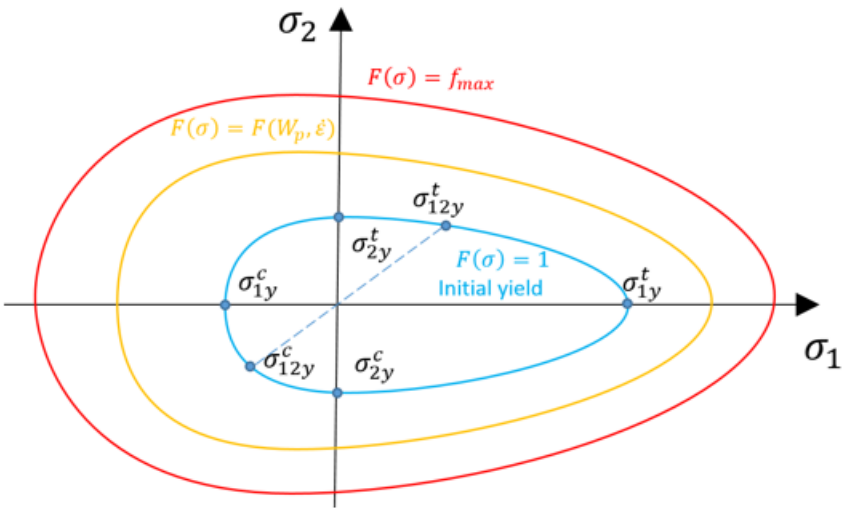


Figure 232: Tsai-Wu Yield Criteria in 1-2 Plane

LAW25 (Tsai-WU and CRASURV)

`LAW25` is the most commonly used composite material in Radioss. It can be used with shell and solid elements. The two formulations available in `LAW25` are the Tsai-Wu and CRASURV formulations.

Elastic Phase

In the elastic phase, Young's modulus (3 parameters), shear modulus (3 parameters) and one parameter for Poisson ratio are required to describe the orthotropic material.

$$\begin{bmatrix}\varepsilon_{11}\\ \varepsilon_{22}\\ \varepsilon_{33}\\ \gamma_{12}\\ \gamma_{23}\\ \gamma_{31}\end{bmatrix}=\begin{bmatrix}\frac{1}{E_{11}} & -\frac{\nu_{12}}{E_{11}} & -\frac{\nu_{12}}{E_{33}} & 0 & 0 & 0\\ & \frac{1}{E_{22}} & -\frac{\nu_{12}}{E_{22}} & 0 & 0 & 0\\ & & \frac{1}{E_{33}} & 0 & 0 & 0\\ & & & \frac{1}{2G_{12}} & 0 & 0\\ & symm. & & & \frac{1}{2G_{23}} & 0\\ & & & & & \frac{1}{2G_{31}}\end{bmatrix}\cdot\begin{bmatrix}\sigma_{11}\\ \sigma_{22}\\ \sigma_{33}\\ \sigma_{12}\\ \sigma_{23}\\ \sigma_{31}\end{bmatrix} \tag{183}$$

Tsai-Wu Yield Criteria for I_{form}=0 and =1

The Tsai-Wu yield surface in `LAW25` is defined with 6 coefficient:

I_{form}=0: Tsai-Wu $\left(F(\sigma)\le F(W_p^*,\dot{\varepsilon})\right)$

I_{form}=1: CRASURV $\left(F(W_p^*,\dot{\varepsilon},\sigma)\le 1\right)$

$$F(\sigma)=F_1\sigma_1+F_2\sigma_2+F_{11}\sigma_1^2+F_{22}\sigma_2^2+F_{44}\sigma_{12}^2+2F_{12}\sigma_1\sigma_2$$

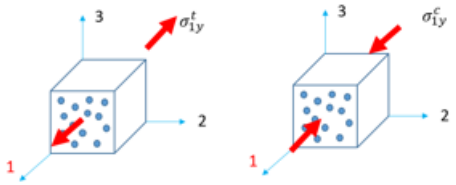
$$F(W_p^*,\dot{\varepsilon},\sigma)=F_1(W_p^*,\dot{\varepsilon})\sigma_1+F_2(W_p^*,\dot{\varepsilon})\sigma_2+F_{11}(W_p^*,\dot{\varepsilon})\sigma_1^2+F_{22}(W_p^*,\dot{\varepsilon})\sigma_2^2+F_{44}(W_p^*,\dot{\varepsilon})\sigma_{12}^2+2F_{12}(W_p^*,\dot{\varepsilon})\sigma_1\sigma_2$$

To check if material in yield, in Tsai-Wu (I_{form}=0) $F(\sigma)$ will be compared with $F(W_p^*,\dot{\varepsilon})$ at each stress state and in CRASURV (I_{form}=1) $F(W_p^*,\dot{\varepsilon})$ will be simply compared with 1 at each stress state.

These 6 coefficients could be determined with yield stress from these tests:

- Tensile/compression tests

  Longitudinal tensile/compression tests (in direction 1 which is fiber direction):



I_{form} = 0: Tsai-Wu $\left(F(\sigma)\le F(W_p^*,\dot{\varepsilon})\right)$

$$F_1=-\frac{1}{\sigma_{1y}^c}+\frac{1}{\sigma_{1y}^t}$$

$$F_{11}=\frac{1}{\sigma_{1y}^c\sigma_{1y}^t}$$

$$F(W_p^*,\dot{\varepsilon})=\left(1+b(W_p^*)^n\right)\cdot\left(1+c\ln\left(\frac{\dot{\varepsilon}}{\dot{\varepsilon}_0}\right)\right)$$

I_{form} = 1: CRASURV $\left(F(W_p^*,\dot{\varepsilon},\sigma)\le 1\right)$

$$F_1(W_p^*\cdot\dot{\varepsilon})=-\frac{1}{\sigma_{1y}^c(W_p^*\cdot\dot{\varepsilon})}+\frac{1}{\sigma_{1y}^t(W_p^*\cdot\dot{\varepsilon})}$$

$$F_{11}(W_p^*\cdot\dot{\varepsilon})=\frac{1}{\sigma_{1y}^c(W_p^*\cdot\dot{\varepsilon})\cdot\sigma_{1y}^t(W_p^*\cdot\dot{\varepsilon})}$$

In tension:

$$\sigma_{1y}^t(W_p^*\cdot\dot{\varepsilon})=\sigma_{1y}^t\left(1+b_1^t(W_p^*)^{n_1^t}\right)\left(1+c_1^t\ln\left(\frac{\dot{\varepsilon}}{\dot{\varepsilon}_0}\right)\right)$$

***I**form* **= 0: Tsai-Wu**
$\left(F(\sigma) \le F(W_p^*, \dot{\varepsilon})\right)$

Here, $W_p^* = \dfrac{W_p}{W_p^{ref}}$

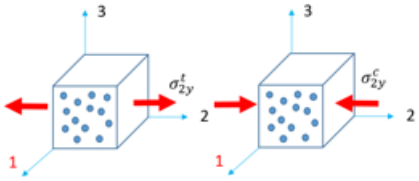
***I**form* **= 1: CRASURV** $\left(F(W_p^*, \dot{\varepsilon}, \sigma) \le 1\right)$

In compression:

$$\sigma_{1y}^c\left(W_p^* \cdot \dot{\varepsilon}\right) = \sigma_{1y}^c\left(1 + b_1^c (W_p^*)^{n_1^c}\right)\left(1 + c_1^c \ln\left(\frac{\dot{\varepsilon}}{\dot{\varepsilon}_0}\right)\right)$$

Here $W_p^* = \dfrac{W_p}{W_p^{ref}}$

Transverse tensile/compression tests (in direction 2)



***I**form* **= 0: Tsai-Wu**
$\left(F(\sigma) \le F(W_p^*, \dot{\varepsilon})\right)$

$$F_2 = -\frac{1}{\sigma_{2y}^c} + \frac{1}{\sigma_{2y}^t}$$

$$F_{22} = \frac{1}{\sigma_{2y}^c \sigma_{2y}^t}$$

Here, $W_p^* = \dfrac{W_p}{W_p^{ref}}$

***I**form* **= 1: CRASURV** $\left(F(W_p^*, \dot{\varepsilon}, \sigma) \le 1\right)$

$$F_2\left(W_p^* \cdot \dot{\varepsilon}\right) = -\frac{1}{\sigma_{2y}^c\left(W_p^* \cdot \dot{\varepsilon}\right)} + \frac{1}{\sigma_{2y}^t\left(W_p^* \cdot \dot{\varepsilon}\right)}$$

$$F_{22}\left(W_p^* \cdot \dot{\varepsilon}\right) = \frac{1}{\sigma_{2y}^c\left(W_p^* \cdot \dot{\varepsilon}\right) \cdot \sigma_{2y}^t\left(W_p^* \cdot \dot{\varepsilon}\right)}$$

In tension:

$$\sigma_{2y}^t\left(W_p^* \cdot \dot{\varepsilon}\right) = \sigma_{2y}^t\left(1 + b_2^t (W_p^*)^{n_2^t}\right)\left(1 + c_2^t \ln\left(\frac{\dot{\varepsilon}}{\dot{\varepsilon}_0}\right)\right)$$

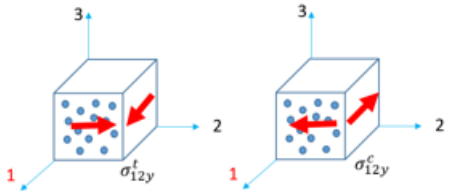
In compression:

$$\sigma_{2y}^c\left(W_p^* \cdot \dot{\varepsilon}\right) = \sigma_{2y}^c\left(1 + b_2^c (W_p^*)^{n_2^c}\right)\left(1 + c_2^c \ln\left(\frac{\dot{\varepsilon}}{\dot{\varepsilon}_0}\right)\right)$$

Here $W_p^* = \dfrac{W_p}{W_p^{ref}}$

- Shear tests

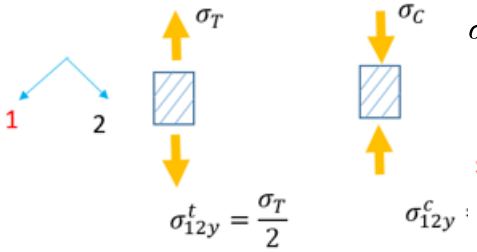
  Shear in plane 1-2



***I**form* **= 0: Tsai-Wu**
$\left(F(\sigma) \le F(W_p^*, \dot{\varepsilon})\right)$

$$F_{44} = \frac{1}{\sigma_{12y}^c \sigma_{12y}^t}$$

σ_{12y}^t and σ_{12y}^c can result from the sample tests:



$$F\left(W_p^*, \dot{\varepsilon}\right) = \left(1 + b\left(W_p^*\right)^n\right) \cdot \left(1 + c\ln\left(\frac{\dot{\varepsilon}}{\dot{\varepsilon}_0}\right)\right)$$

Here, $W_p^* = \frac{W_p}{W_p^{ref}}$

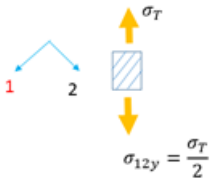
***I**form* **= 1: CRASURV** $\left(F(W_p^*, \dot{\varepsilon}, \sigma) \le 1\right)$

$$F_{44}\left(W_p^*, \dot{\varepsilon}\right) = \frac{1}{\sigma_{12y}\left(W_p^*, \dot{\varepsilon}\right) \cdot \sigma_{12y}\left(W_p^*, \dot{\varepsilon}\right)}$$

In shear:

$$\sigma_{12y}\left(W_p^* \cdot \dot{\varepsilon}\right) = \sigma_{12y}\left(1 + b_{12}\left(W_p^*\right)^{n_{12}}\right)\left(1 + c_{12}\ln\left(\frac{\dot{\varepsilon}}{\dot{\varepsilon}_0}\right)\right)$$

σ_{12y} can result from the sample test:



- Interaction coefficients

***I**form* **= 0: Tsai-Wu** $\left(F(\sigma) \le F(W_p^*, \dot{\varepsilon})\right)$

I_{form} = 1: CRASURV $\left(F(W_p^*, \dot{\varepsilon}, \sigma) \le 1\right)$

$$F_{12} = -\frac{\alpha}{2}\sqrt{F_{11}F_{22}}$$

The default reduction factor, $\alpha = 1$, is typically used.

$$F\left(W_p^*, \dot{\varepsilon}\right) = \left(1 + b\left(W_p^*\right)^n\right) \cdot \left(1 + c\ln\left(\frac{\dot{\varepsilon}}{\dot{\varepsilon}_0}\right)\right)$$

Here, $W_p^* = \frac{W_p}{W_p^{ref}}$

$$F_{12}\left(W_p^*, \dot{\varepsilon}\right) = -\frac{\alpha}{2}\sqrt{F_{11}\left(W_p^*, \dot{\varepsilon}\right)F_{22}\left(W_p^*, \dot{\varepsilon}\right)}$$

The default reduction factor, $\alpha = 1$, is typically used.

Note that the relative plastic work W_p^* is used in Tsai-Wu to calculate the yield surface; whereas in CRASURV, the relative plastic work is used to calculate the yield stress.

***I**form* **= 0: Tsai-Wu** $\left(F(\sigma) \le F(W_p^*, \dot{\varepsilon})\right)$

$$F(\sigma) \le F(W_p^*, \dot{\varepsilon})$$

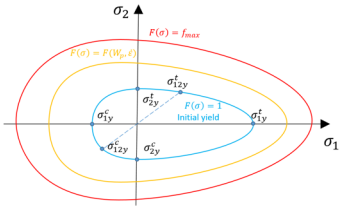
With

$$F\left(W_p^*, \dot{\varepsilon}\right) = \left(1 + b\left(W_p^*\right)^n\right) * \left(1 + c\ln\left(\frac{\dot{\varepsilon}}{\dot{\varepsilon}_0}\right)\right)$$

Here $W_p^* = \frac{W_p}{W_p^{ref}}$

Material in elastic phase, if $F(\sigma) \le F(W_p^*, \dot{\varepsilon})$

Material in nonlinear phase, if $F(\sigma) > F(W_p^*, \dot{\varepsilon})$

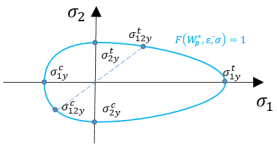


***I**form* **= 1: CRASURV** $\left(F(W_p^*, \dot{\varepsilon}, \sigma) \le 1\right)$

$$F(W_p^*, \dot{\varepsilon}, \sigma) \le 1$$

Material in elastic phase, if $F(W_p^*, \dot{\varepsilon}, \sigma) \le 1$

Material in nonlinear phase, if $F(W_p^*, \dot{\varepsilon}, \sigma) > 1$



The yield stress limit $F(W_p^*, \dot{\varepsilon})$ **is in range of 1 and** $f_{\max}$

In `LAW25` (Tsai-Wu and CRASURV) damage is a function of the total strain and the maximum damage factor.

If the total strain $\varepsilon > \varepsilon_t$ or out of plane strain $\gamma_{ini} < \gamma < \gamma_{\max}$, then the material is softened using the following method:

$$\sigma^{reduce} = \sigma \cdot (1 - d_i) \quad (184)$$

with *i*=1,2,3

Where, d_i is the damage factor and is defined as:

$$d_i = \min\left(\frac{\varepsilon_i - \varepsilon_{ti}}{\varepsilon_i} \cdot \frac{\varepsilon_{mi}}{\varepsilon_{mi} - \varepsilon_{ti}},\ d_{\max}\right) \quad (185)$$

with *i*=1,2

$$d_3 = \min\left(\frac{\gamma - \gamma_{ini}}{\gamma_{\max} - \gamma_{ini}} \cdot \frac{\gamma_{\max}}{\gamma}, d_{3\max}\right) \quad (186)$$

in direction 3 (delamination)

- If the total strain is between $\varepsilon_t < \varepsilon < \varepsilon_f$, the material begins to soften, but this damage is reversible. Once $\varepsilon > \varepsilon_f$, then the damage is irreversible and if $\varepsilon \geq \varepsilon_m$, then stress in material is reduced to 0.
- Damage could be in elastic phase or in plastic phase. It depends on which phase ε_t and ε_f are defined in.
- Element deletion is controlled by I_{off}. Select a different I_{off} option to control the criteria of element deletion. For additional information, refer to I_{off} in `LAW25` in the *Reference Guide*.

LAW19 and LAW58 for Fabric

Radioss has two material laws for modeling fabrics `LAW19` and `LAW58`. `LAW19` is an elastic orthotropic material and must be used with `/PROP/TYPE9`. `LAW58` is hyperelastic anisotropic fabric material and must be used with `/PROP/TYPE16`.

Coupling between warp and weft directions could be defined in this material law to reproduce physical interaction between fibers. Both material laws are often used for airbag modeling.

In `LAW58`, two methods are provided to define the stress-strain behavior.

- Nonlinear function (*fct_ID*$_i$) curve to define the warp, weft and shear behavior
- Young's modulus, soften coefficient *B*, straightening strain S_i and fiber bending modulus reduction factor *Flex*$_i$

  In warp and weft direction:

  $$\sigma_{ii} = E_i\varepsilon_{ii} - \frac{(B_i\varepsilon_{ii}2)}{2} \quad (i = 1, 2) \text{ when } \frac{d\sigma}{d\varepsilon} > 0$$

  $$\sigma_{ii} = \max_{\varepsilon_{ii}}\left(E_i\varepsilon_{ii} - \frac{(B_i\varepsilon_{ii}2)}{2}\right) \quad (i = 1, 2) \text{ when } \frac{d\sigma}{d\varepsilon} \leq 0$$

  For in-plane shear in initial state, use G_0. Once α (angle between wrap and weft) reaches α_T (shear lock angle), then use G_T to describe the strengthening.

  $\tau = G_0\tan(\alpha) - \tau_0$ if $\alpha \leq \alpha_T$

  $$\tau = \frac{G_T}{1+\tan^2(\alpha_T)}\tan(\alpha) + \left(G_0 - \frac{G_T}{1+\tan^2(\alpha_T)}\right)\tan(\alpha_T) - \tau_0 \text{ if } \alpha > \alpha_T$$

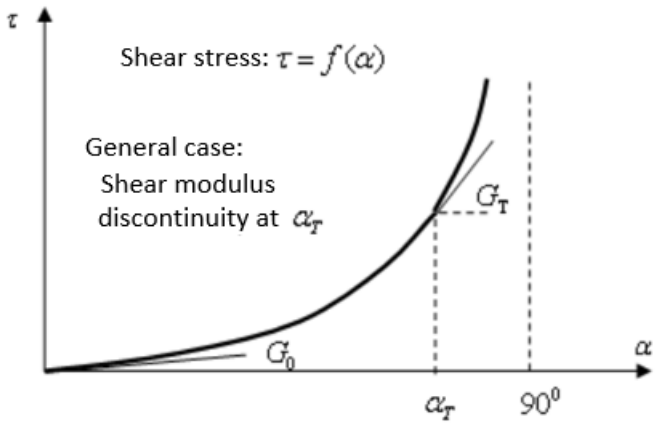


Figure 233:

For out-of-plane shear stress-strain is described with G_{sh}.

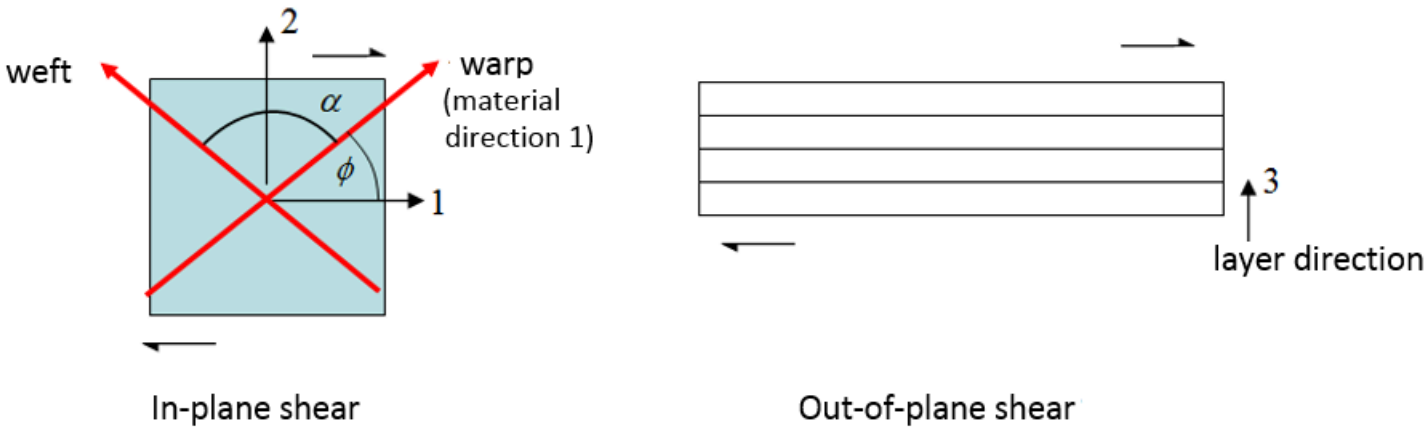


Figure 234:

For fabric, there is straight process at the beginning of tension. At this phase, fiber shows very soft, due to not being tightened yet.

Figure 235:

In `LAW58`, use *Flex*$_i$ to describe this behavior:

$$E_{fi} = Flex_i * E_i \quad (187)$$

After the fabric is tight (strengthening strain S_i is reached), then normal fiber elasticity E_i could be used.

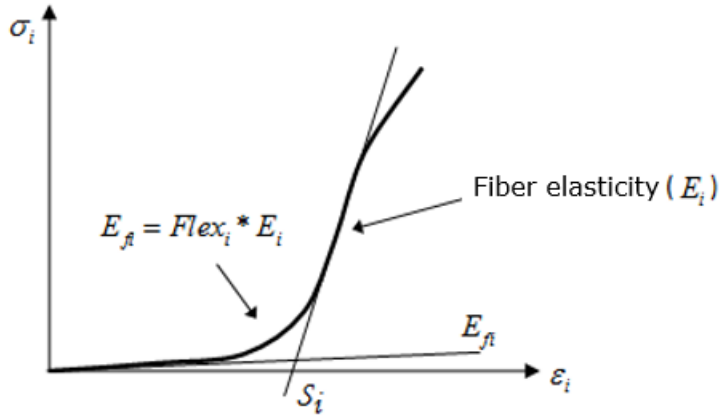


Figure 236:

Composite Modeling

Composite modeling techniques used in Radioss.

Each Layer with at Least One Solid

The model is large which leads to high accuracy results but need more computation resource.

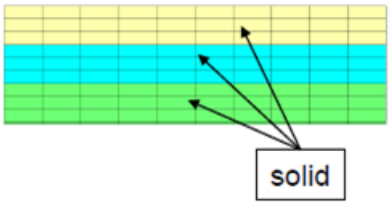


Figure 237:

Mixed Approach (Middle Layer Thick)

Defines shells for top and bottom layer and solid for middle layer.

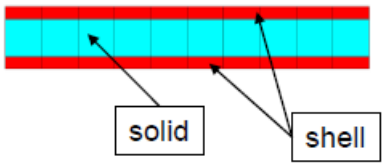


Figure 238:

Sandwich Shell Approach

Use only one shell element through the thickness. Use different materials to define multiple layers.

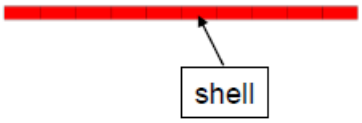


Figure 239:

In composite shell modeling there are two types of problems.

- The first is composite shell with isotropic layers (orthotropic in z direction). For example, by Windshield modeling. Where, material `LAW27(PLAS_BRIT)` or `LAW36 (PLAS_TAB)` can be used.
- The second is composite shell with orthotropic layers (orthotropic in x, y and z). For example, Sandwich shell with external fiber glass layers and internal foam layer. Where, material `LAW25 (COMPSH)` and all shell formulation can be used, except for `QEPH`.

Composite Properties

Composite could be modeled with solid or shell element. Depending on the element type, the following properties can be used in Radioss to model a composite.

Shell Elements

Modeling composite in Radioss could be defined by Layer based or Ply based with different property.

- Layer based modeling with `/PROP/TYPE10 (SH_COMP)`, `/PROP/TYPE11 (SH_SANDW)`
- Ply based modeling with `/PROP/TYPE17 (STACK)`, `/PROP/TYPE51`, `/PROP/PCOMPP+/STACK`, `/PROP/TYPE19 (PLY)` and `/PLY`

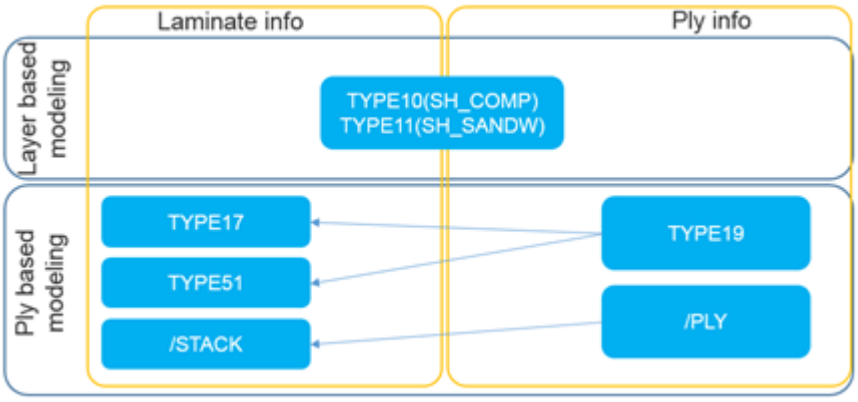


Figure 240:

For ply-based modeling, info (like material, thickness, anisotropic angle, anisotropic axis angle and number of integration points) for each ply defined in `/PROP/TYPE19` (or `/PLY`) and assembled in `/PROP/TYPE17` or `/PROP/TYPE51` (or `/STACK`) with option *Pply_ID$_i$*.

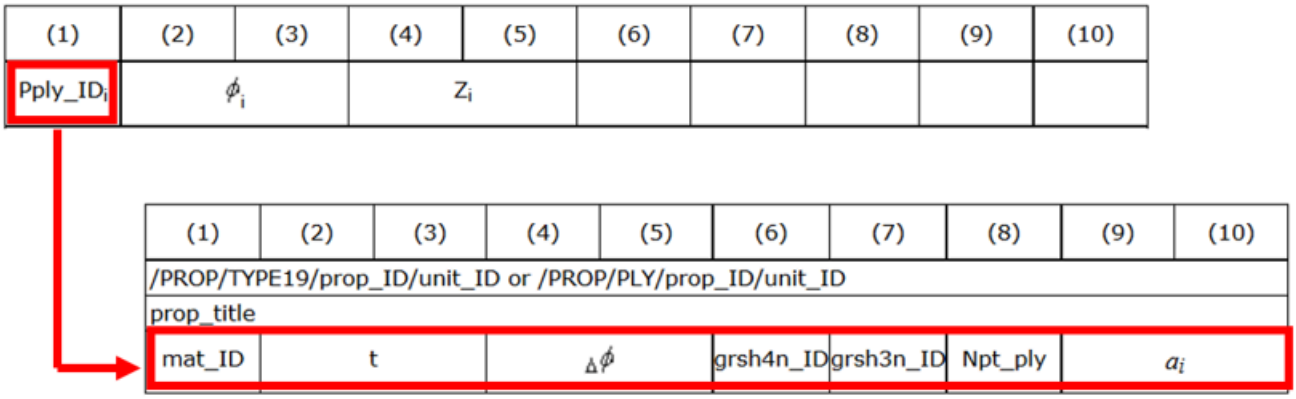


Figure 241:

For ply-based modeling, the way of assemble plies could be "by ply" or "by substack". "By ply" is simply pile up plies one by one from bottom to top. The way "by substack" need first pile up plies one by one to create each substack, and then each substack could either be combined or could also be stacked with substack connection "*INT*".

Figure 242:

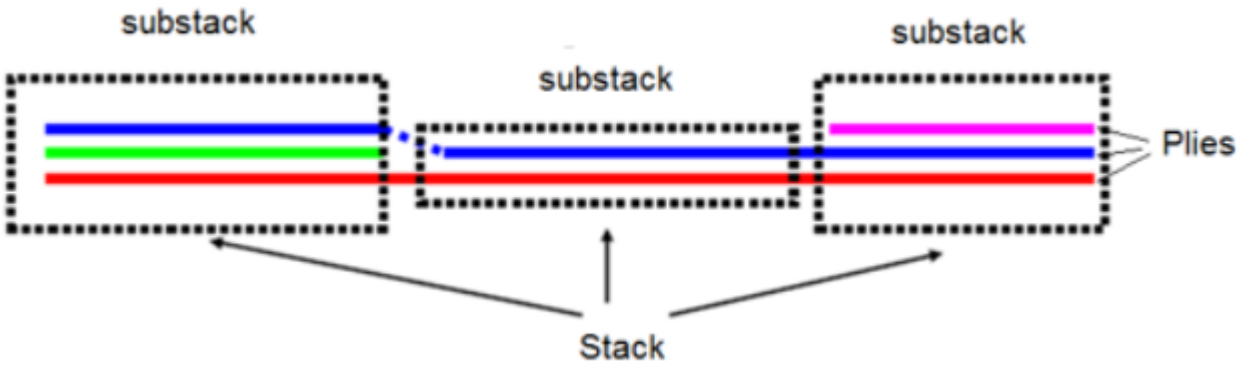


Figure 243: Stack in `/PROP/TYPE51`

For composite property following topics are:

- Layer (ply) number and integration points each layer (ply)
- Anisotropy in layer (ply)
- Layer (ply) thickness and position
- Composite material used for Layer (ply)

	Layer-based Properties	
	/PROP/TYPE10 (SH_COMP)	**/PROP/TYPE11 (SH_SANDW)**
N Layer Numbers Or *Pply_ID*$_i$ Ply Numbers	N=0~100	N=1~100
IP of each layer/ply	1 per Layer	1 per Layer
Iint Integration formulation		
$\phi_i + \mathbf{V}$ Anisotropic direction	√	√
ϕ_i+ skew Anisotropic direction		√
θ_{drape}		

	Layer-based Properties	
	/PROP/TYPE10 (SH_COMP)	**/PROP/TYPE11 (SH_SANDW)**
Ply orientation change		
α_i Angle between anisotropic axis		
t_i Layer/Ply thickness		√
$I_{pos} + Z_i$ Layer/Ply position		√
I_{pos}=2,3,4 Layer/Ply offset		
mat_ID$_i$ Material for each layer/ply	Use material defined in `/PART`	√ Must use same material type for all layers.
Commonly used Composite Material Law	15, 25 and user material	15, 25 and user material
XFEM compatibility (crack propagation)		√ With `/FAIL/JOHNSON`, `/FAIL/TAB1` and `/FAIL/TBUTCHER`
Plyxfem Delamination between layer/ply		
Minterply Material between layer/ply		

	Ply-based Properties		
	/PROP/TYPE17 + /PROP/TYPE19	**/PROP/TYPE51 + /PROP/TYPE19**	**/PROP/PCOMPP+/STACK+/PLY**
N Layer Numbers Or *Pply_ID*$_i$ Ply Numbers	*Pply_ID*$_i$=1~200	*Pply_ID*$_i$=1~200	*Pply_ID*i$_i$1~n
IP of each layer/ply	1 per Ply *Npt_ply*=1 in `/PROP/TYPE19`	1~9 per Ply *Npt_ply*=1~9 in `/PROP/TYPE19`	1~9 per Ply *Npt_ply*=1~9 in `/PLY`
Iint Integration formulation		√ Uniformed or Gauss	√ Uniformed or Gauss
$\phi_i + \mathbf{V}$ Anisotropic direction	√	√	√
ϕ_i+ skew Anisotropic direction	√	√	√
θ_{drape} Ply orientation change	√ Defined in `/PROP/TYPE19`	√ Defined in `/PROP/TYPE19`	√ Defined in `/PLY`
α_i Angle between anisotropic axis	√ in `/PROP/TYPE19`	√ in `/PROP/TYPE19`	√ in `/PLY`
t_i Layer/Ply thickness	√ Different in `/PROP/TYPE19`	√ Different in `/PROP/TYPE19`	√ Different in `/PLY`
I_{pos} + Z_i Layer/Ply position	√	√	√
I_{pos}=2,3,4	√	√	√

	Ply-based Properties		
	/PROP/TYPE17 + /PROP/TYPE19	**/PROP/TYPE51 + /PROP/TYPE19**	**/PROP/PCOMPP+/STACK+/PLY**
Layer/Ply offset			
mat_ID$_i$ Material for each layer/ply	√ Must use same material type for all plies.	√ Different material type allows for each ply.	√ Different material type allows for each ply.
Commonly used Composite Material Law	25, 27, 36, 60, 72, 93 and user material	25, ≥28 and user material	25 and user material
XFEM compatibility (crack propagation)		√ With `/FAIL/JOHNSON` and `/FAIL/TBUTCHER`	
Plyxfem Delamination between layer/ply	√		
Minterply Material between layer/ply	√ Only with LAW1+`/FAIL/LAD_DAM`		

Layer (Ply) Number N (*Nply_ID$_i$*) and Integration Points each Layer (Ply)

For layer-based modeling which use `/PROP/TYPE10`, `/TYPE11`. N is the number of layers through the shell thickness. For these properties, there is one integration point (IP) each layer.

For ply-based modeling, which use `/PROP/TYPE17`, `/TYPE51` and `/PCOMPP`. *Pply_ID$_i$* is the number of plies through the shell thickness. Plies could be combined until n plies for these properties.

For TYPE17 only one integration point is allowed while for TYPE51 and `/STACKF` until 9 integration points are allowed. Number of integration point defined with option "*Npt_ply*" in property TYPE19 or `/PLY`.

Example (Ply) (`/PROP/TYPE51`)

In this example, *Npt_ply*=3 defined in `/PROP/TYPE19`, means 3 integration points defined per ply and with option I_{int}=0 defined in `/PROP/TYPE51`, then these 3 integration points are uniformly distributed through each ply thickness.

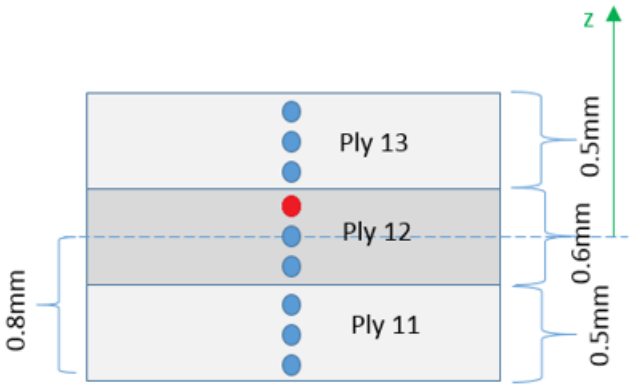


Figure 244:

If I_{int}=1 in `/PROP/TYPE51`, then integration points are distributed follow Forces and Moments Calculation *Gauss Integration Scheme* through each ply thickness.

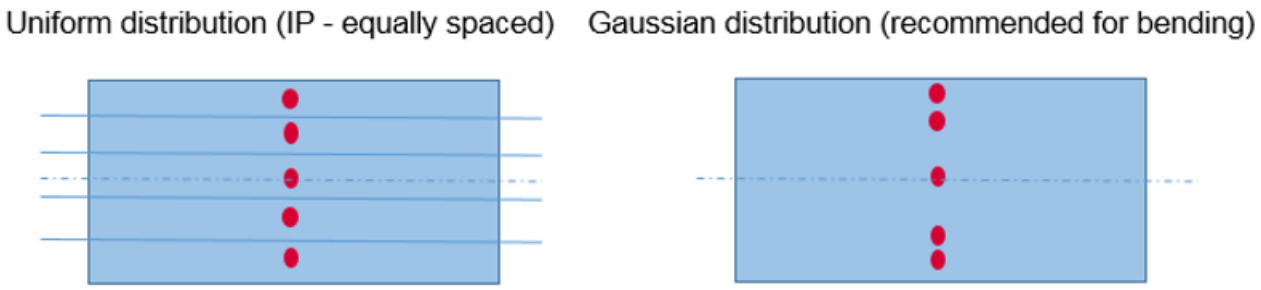


Figure 245:

It is possible to print animation results (plastic strain, damage, stress and strain tensor) in each specific integration point with `/ANIM/SHELL/IDPLY/Keyword4/I/J` (or `/ANIM/SHELL/Keyword3/N/NIP`).

For instance, use /ANIM/SHELL/IDPLY/EPSP`/2/3` (or /ANIM/SHELL/EPSP`/2/3`) to print plastic strain in third integration point (red highlighted integration point in Figure 245) of second ply (ply name Ply12). For additional print info about Integration points through shell thickness for composite properties, refer to Shell stress tensor output in animation. in the FAQs.

Anisotropy in Layer (Ply)

- The first anisotropic direction of material could be defined with angle ϕ and gloval vector $\mathbf{V}(V_X,V_Y,V_Z)$. It is also possible to use angle ϕ and skew. In this case, x-axis of skew replaced the global vector $\mathbf{V}$.

  Project the global vector $\mathbf{V}$ in shell element and then rotate ϕ degree is the first anisotropic direction (also called Material direction 1). The positive direction of ϕ is coding to shell normal $\mathbf{n}$. For example, in Figure 246 rotate counterclockwise ϕ degree is material direction 1.

The material direction 1 of the local element reference is normally the fiber direction. Then the material character (E-Module, yield stress, and so on) of direction 1 which defined in material law could be then applied in the correct direction on local element reference.

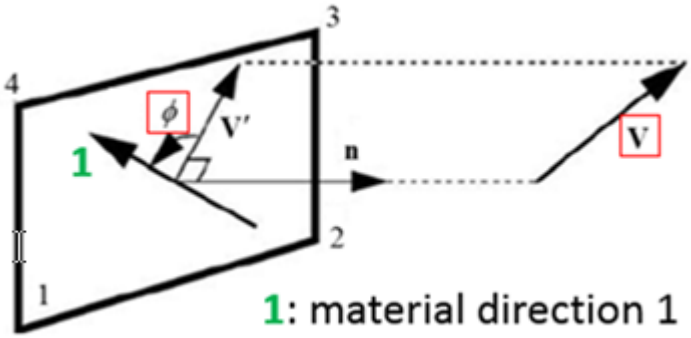


Figure 246:

- Composite material could be orthotropic or anisotropic. In Radioss it is possible to describe this character with anisotropic axis angle α_i in ply-based properties. In case of $\alpha_i = 90^\circ$, then it describes orthotropic material. For layer based properties (TYPE10 and TYPE11) which without this option α_i, so that only orthotropic material could be defined.

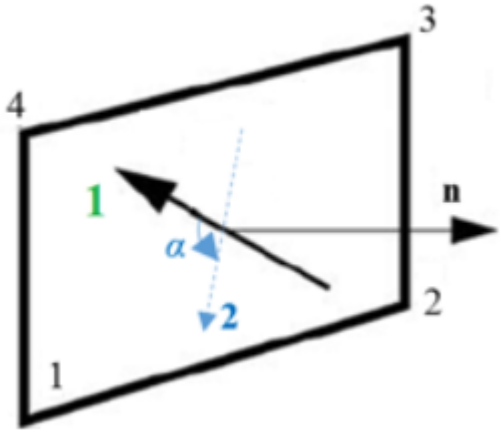


Figure 247:

Figure 248 shows an Example in `/PROP/TYPE11` which use skew to define global vector $\mathbf{V}$.

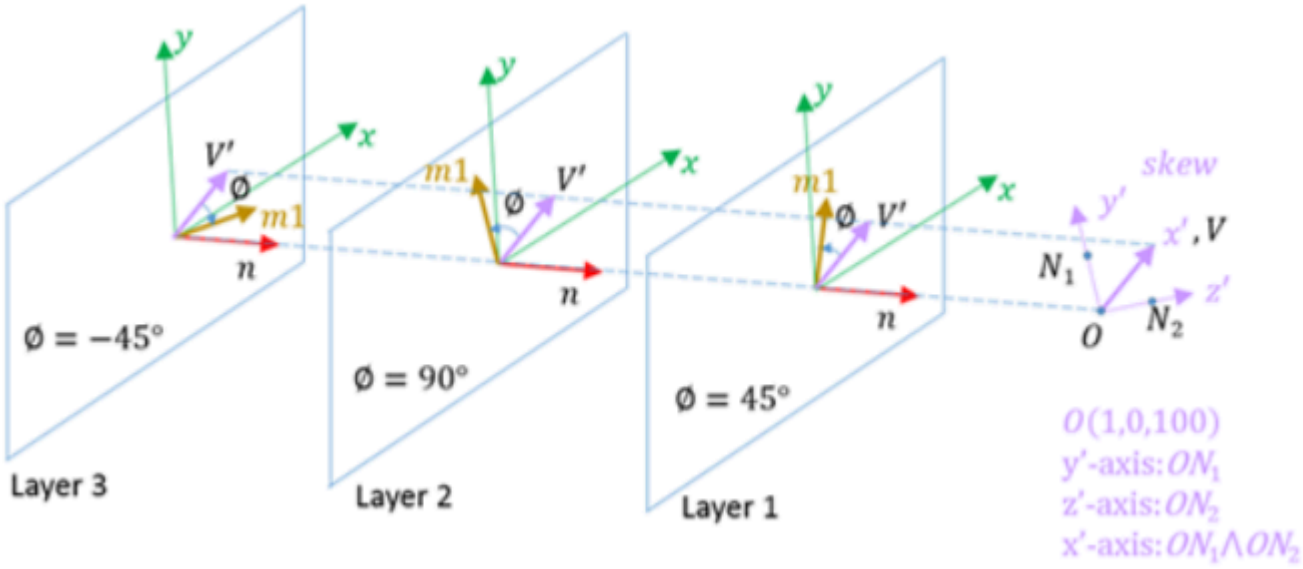


Figure 248:

For property TYPE11 and TYPE51, the anisotropic direction for some shell could also be initialized with ϕ in keyword `/INISHE/ORTH_LOC` or `/INISH3/ORTH_LOC`. For property TYPE51, anisotropic axis angle could also be initialized with these keywords.

- The orientation of anisotropy for specific shell elements or shell element groups could be change again with option *drape_ID* and def_orth in `/PROP/TYPE19` and `/PLY`. *drape_ID* defined in `/DRAPE`. With this feature, angle of anisotropic direction could be changed with θ_{drape}.
  - If use *def_orth*=1:

    Angle ϕ_i, skew or global vector $\mathbf{V}$ will be ignored. Take shell local x-axis as vector $\mathbf{V'}$ and then rotate ϕ degree is the first anisotropic direction. ϕ_s (defined in `/SHELL` or `/SH3N`) is taken into account by compute angle ϕ.

    $$\phi = \phi_s + \Delta\phi + \theta_{drape} \quad (188)$$

  - If use *def_orth*=2 (Default)

    Project the global vector $\mathbf{V}$ in shell element to vector $\mathbf{V'}$ and then rotate ϕ degree is the first anisotropic direction. The angle ϕ is computed as:

    $$\phi = \phi_i + \Delta\phi + \theta_{drape} \quad (189)$$

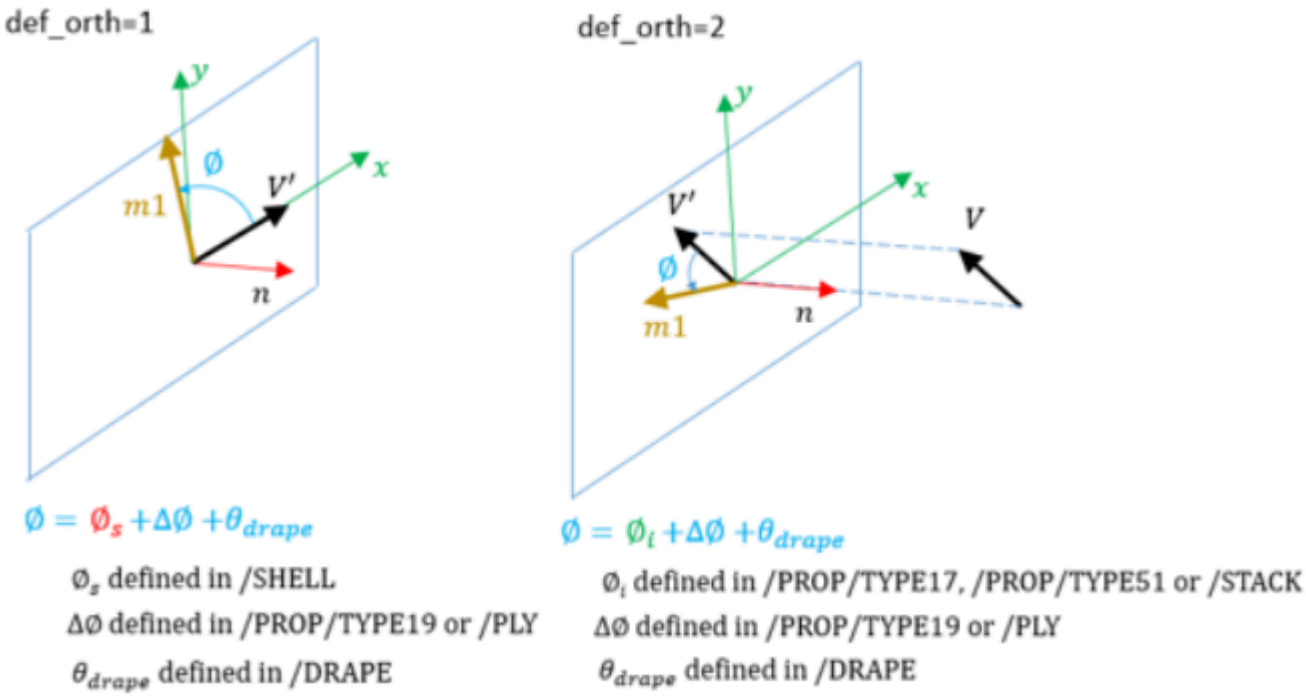


Figure 249:

Layer (Ply) Thickness and Position

- For `/PROP/TYPE10`, layer thickness is simply averaged by layer number

  $$t_i = Thick / N \qquad (190)$$

  and layers are automatically overlying one by one from bottom to top.

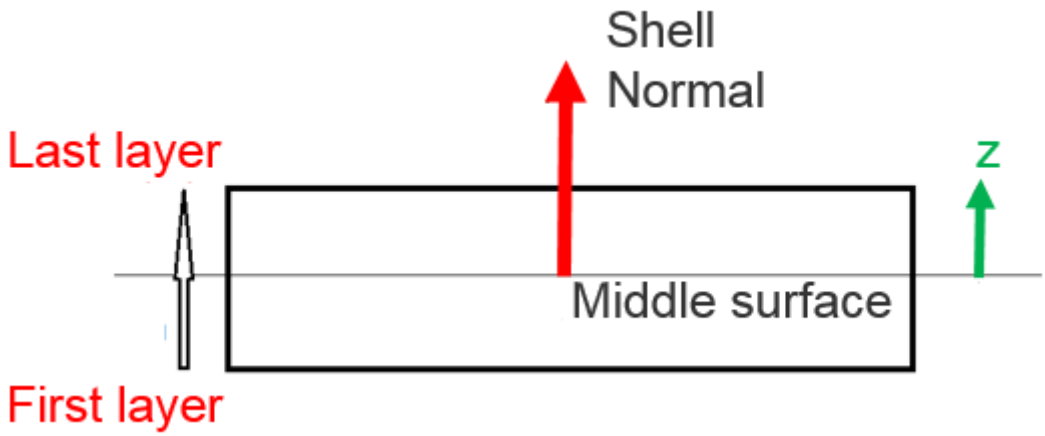


Figure 250:

- For property TYPE11, TYPE17, TYPE51 and `/STACK`, layer (ply) position and thickness depend on option I_{pos}

- If I_{pos}=0

  User layer (ply) thickness input t_i will be taken, and layer (ply) position will be automatically overlying one by one from bottom till top; but if,

  $$\sum_i t_i \neq Thick \quad (191)$$

  Then, layer (ply) thickness will be adjusted to t_i^{new}, so that

  $$\sum_i t_i^{new} = Thick \quad (192)$$

  Layer (ply) position will be then adjusted, as well.
- If I_{pos}=1

  User layer(ply) input of thickness t_i and position Z_i will be taken. Sum of layer thickness will not be checked with $Thick$.

  For additional information, refer to Layer thickness and position calculation. in the FAQs.

- For property TYPE17, TYPE51 and `/STACK`, it is also possible to offset the plies with I_{pos}=2, 3, 4
  - I_{pos}=2: the shell element mid-surface is at Z_0 from the bottom of the ply layout

Figure 251:

  - I_{pos}=3: the top of the ply layout is coincident with the element mid-surface

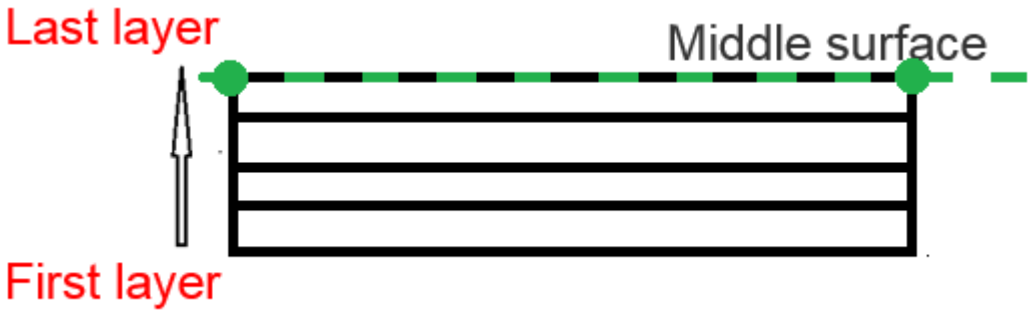


Figure 252:

- I_{pos}=4: the bottom of the ply layout is coincident with the element mid-surface

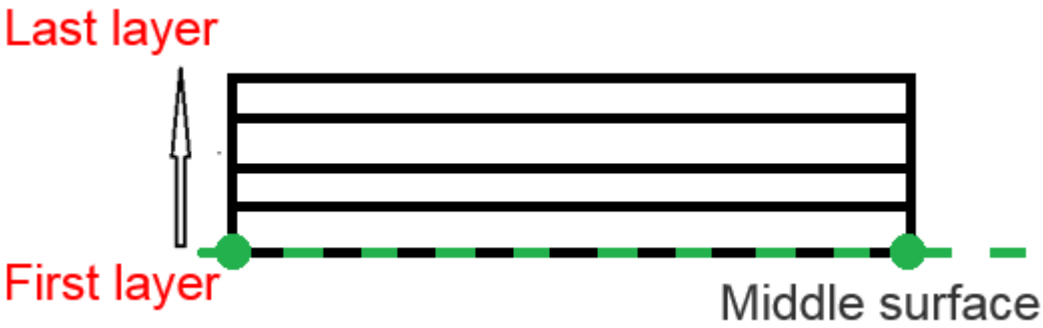


Figure 253:

- For `/PROP/TYPE17`, `/PROP/TYPE51` and `/STACK`, ply thickness could be changed by option *Thinning* in `/DRAPE` (which is used in `/PROP/TYPE19` or `/PLY`). Then updated ply thickness is:

$$t_i^{new} = t_i \cdot Thinning \quad (193)$$

Composite Material Used for Layer (Ply)

Composite material LAW15 and LAW25 could be used for shell element. Failure model `/FAIL/HASHIN`, `/FAIL/PUCK` and `/FAIL/LAD_DAMA` with LAW25 and `/FAIL/CHANG` with LAW15 could be used to describe composite behavior for shell element. For additional information, refer to Composite Material.

- Material for layer (ply)
  - For property TYPE10, composite used material defined in `/PART`
  - For property TYPE11 and TYPE17, composite used material defined in option *mat_ID*$_i$. Different material ID could be defined for each layer (ply). But they must use same material type. If using LAW25, then several different LAW25 cards can be used for different layer (ply).
  - For property TYPE51 and `/STACK`, composite also used material defined in option *mat_ID*$_i$ and different material type or ID could be used for each ply.
- Material between ply

  For property TYPE17, it is possible to define delamination between plies or stacks (with *Plyxfem*=2). This is very useful for delamination is the main driven of composite failure. The

material between plies defined with *Minterply*. For the moment, LAW1+`/FAIL/LAD_DAM` could be used to describe three different type of ply delamination.

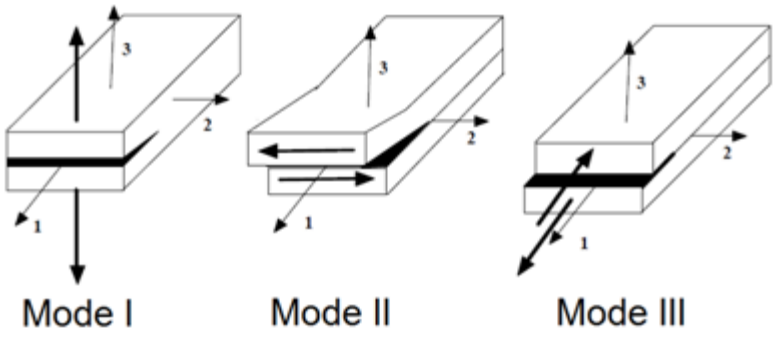


Figure 254:

And then in this case additional variable $\delta_1, \delta_2, \delta_3$ are added on each node of ply by computation to simulate the delamination failure between plies.

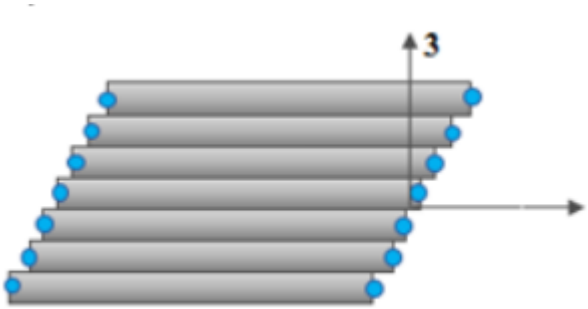


Figure 255:

Solid Element

The new composite technology allows you to make parts production more and thicker, as modeling those parts with shell elements is not enough. Thick shell can solve this problem. Compare with shell element, thick shell could direct connect with other solid part.

For Solid element, for the moment only layer-based modeling with property `/PROP/TYPE22 (TSH_COMP)` is available. This solid property is similar to shell property `/PROP/TYPE11` by define composite.

	Layer-based Properties	
	`/PROP/TYPE22 (TSH_COMP)`	
Layer Numbers	I_{solid}=14: I_{int}=9~200	I_{solid}=15:

	Layer-based Properties	
	/PROP/TYPE22 (TSH_COMP)	
IP of each layer/ply	I_{npts}=ijk=2~9	I_{npts}=j=1~200
Integration formulation		
ϕ_i + $\mathbf{V}$, Anisotropic direction	√	
ϕ_i + skew, Anisotropic direction	√	
θ_{drape}, Ply orientation change		
α_i, Angle between anisotropic axis		
t_i, Layer/Ply thickness	√ defined with factor t_i/t	
I_{pos} + Z_i, Layer/Ply position	√	
I_{pos}=2,3,4, Layer/Ply offset		
mat_ID$_i$, Material for each layer/ply	√ Different material type allows for each layer.	
Commonly used Composite Material LAW	LAW12, LAW14, LAW25 and user material	
Plyxfem, Delamination between layer/ply		
Minterply, Material between layer/ply		

- Layer Number and Integration Points Each Layer

  The layer number defined using the option I_{int}. I_{int} is only used for I_{solid}=14 when the number of layers > 9.

  In this case, the thickness direction integration point defined by I_{npts} should be zero.

  Example, I_{cstr} = 010; I_{npts} = 202; I_{int} = 100 for a number of 100 layers in "s" direction
- Anisotrophy in Layer (Ply)

Similar to shell property /PROP/TYPE11, reference vector $\mathbf{V}$ and angle ϕ are used to define the material direction 1. The reference vector $\mathbf{V}$ project to the middle surface of solid element and turn ϕ degree is the material direction 1.

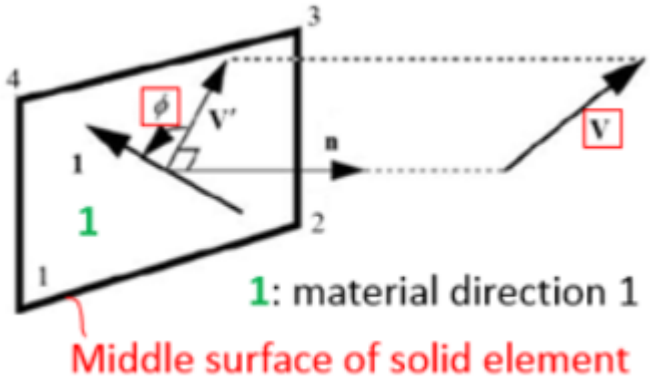


Figure 256:

- Layer Thickness and Position

  For solid element thickness and position defined by element mesh.
- Composite Material Used for Layer
  - With option *mat_ID$_i$*, it is possible to use different material type for each layer
  - Composite material LAW12, LAW14 and LAW25 could be used with this property
  - Failure model /FAIL/HASHIN, /FAIL/PUCK and /FAIL/LAD_DAMA with these composite material laws are also accounted for
  - Material referred to in the corresponding /PART card is only used for time step and interface stiffness calculation
  - For LAW25, it is assumed that (for solids and thick shells) the material is elastic in transverse direction (material direction 2 and 3) and the *E33* value must be specified in such cases

  For additional information, refer to “Composite material and Composite failure”.

See Also
/DRAPE (Starter)
Property and Elements (FAQ)

Composite Failure Model

In Radioss the following composite failure models may be used to describe composite material failure.

- /FAIL/HASHIN
- /FAIL/PUCK
- /FAIL/LAD_DAMA

16. L. Gornet, “Finite Element Damage Prediction of Composite Structures"

- /FAIL/CHANG

A composite material consists of two different materials (matrix and reinforcement fiber). Each material has a different failure behavior. In Radioss it is possible to use different failure models for matrix and fiber in one composite element (for elements with property TYPE11, TYPE16, TYPE17, TYPE51, PCOMPP or TYPE22). For example, you may use /FAIL/HASHIN for fiber failure, /FAIL/PUCK for matrix failure and /FAIL/LAD_DAMA for delamination between layers or plies (if there is more than one layer or plies defined for the composite).

Besides the above typical composite failure models, /FAIL/FLD (used for isotropic brittle composite materials in layers(plies) as in, glass), /FAIL/ENERGY, /FAIL/TBUTCHER and /FAIL/TENSSTRAIN may also be used to describe failure for composite layers(plies).

/FAIL/HASHIN

In HASHIN failure, two primary failure modes are considered.

- Fiber mode: composite fails, due to fiber rupture in tension or fiber buckling in compression. So, in /FAIL/HASHIN, tensile/shear fiber mode, compression fiber mode and crush mode are the fiber modes. If direction 1 is the fiber direction, then plane 23 is the predominant failure plane for fiber mode.
- Matrix mode: composite fails, due to matrix cracking from the fiber. Failure matrix mode (or shear failure matrix mode) and delamination mode are both matrix modes. The failure plane for matrix mode is parallel to the fiber, and stress σ_{11} will not be considered in this mode.

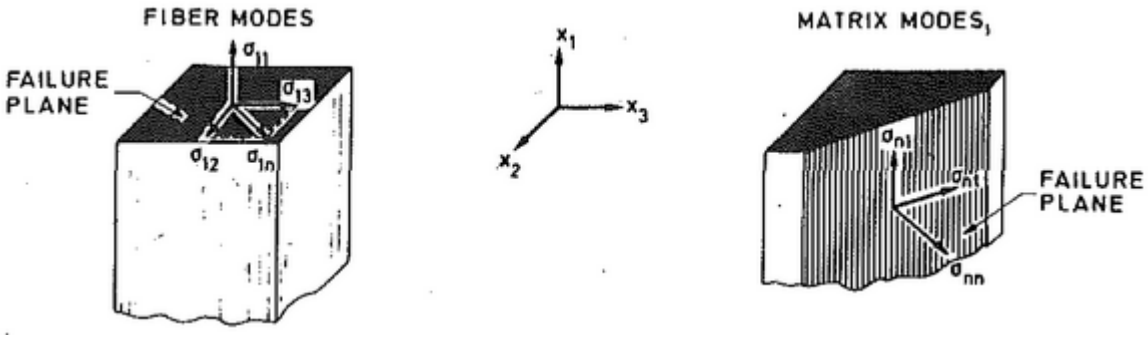


Figure 257: Fiber modes and matrix modes for uni-directional lamina model

Fibers in uni-directional lamina model [17] are only in one direction and in fabric lamina model are in two directions.

	Uni-directional Lamina Model	**Fabric Lamina Model**
Damage criteria	If D=1, then failure. If $0 \le D < 1$, D then no failure. With $D = Max(F_1, F_2, F_3, F_4)$	If D=1, then failure. If $0 \le D < 1$, D then no failure. With $D = Max(F_1, F_2, F_3, F_4)$

	Uni-directional Lamina Model	**Fabric Lamina Model**
Tensile/shear fiber mode	$F_1=\left(\frac{\langle\sigma_{11}\rangle}{\sigma_1^t}\right)^2+\left(\frac{\sigma_{12}^2+\sigma_{13}^2}{\sigma_{12}^{f2}}\right)$	$F_1=\left(\frac{\langle\sigma_{11}\rangle}{\sigma_1^t}\right)^2+\left(\frac{\sigma_{12}^2+\sigma_{13}^2}{\sigma_a^{f2}}\right)$ $F_2=\left(\frac{\langle\sigma_{22}\rangle}{\sigma_2^t}\right)^2+\left(\frac{\sigma_{12}^2+\sigma_{23}^2}{\sigma_b^{f2}}\right)$ With, $\sigma_a^f=\sigma_{12}^f, \quad \sigma_b^f=\sigma_{12}^f\frac{\sigma_2^t}{\sigma_1^t}$
Compression fiber mode	$F_2=\left(\frac{\langle\sigma_a\rangle}{\sigma_1^c}\right)^2$ With $\sigma_a=-\sigma_{11}+\left\langle-\frac{\sigma_{22}+\sigma_{33}}{2}\right\rangle$	$F_3=\left(\frac{\langle\sigma_a\rangle}{\sigma_1^c}\right)^2$ With $\sigma_a=-\sigma_{11}+\langle-\sigma_{33}\rangle$ $F_4=\left(\frac{\langle\sigma_b\rangle}{\sigma_2^c}\right)^2$ With $\sigma_b=-\sigma_{22}+\langle-\sigma_{33}\rangle$
Crush mode	$F_3=\left(\frac{\langle p\rangle}{\sigma_c}\right)^2$ With $p=-\frac{\sigma_{11}+\sigma_{22}+\sigma_{33}}{3}$	$F_5=\left(\frac{\langle p\rangle}{\sigma_c}\right)^2$ With $p=-\frac{\sigma_{11}+\sigma_{22}+\sigma_{33}}{3}$
Shear failure matrix mode		$F_6=\left(\frac{\sigma_{12}}{\sigma_{12}^m}\right)^2$
Failure matrix mode	$F_4=\left(\frac{\langle\sigma_{22}\rangle}{\sigma_2^t}\right)^2+\left(\frac{\sigma_{23}}{S_{23}}\right)^2+\left(\frac{\sigma_{12}}{S_{12}}\right)^2$ Where, $S_{12}=\sigma_{12}^m+\langle-\sigma_{22}\rangle\tan\phi$ $S_{23}=\sigma_{23}^m+\langle-\sigma_{22}\rangle\tan\phi$	
Delamination mode	$F_5=S_{del}^2\left[\left(\frac{\langle\sigma_{33}\rangle}{\sigma_3^t}\right)^2+\left(\frac{\sigma_{23}}{\tilde{S}_{23}}\right)^2+\left(\frac{\sigma_{13}}{S_{13}}\right)^2\right]$ Where,	$F_7=S_{del}^2\left[\left(\frac{\langle\sigma_{33}\rangle}{\sigma_3^t}\right)^2+\left(\frac{\sigma_{23}}{S_{23}}\right)^2+\left(\frac{\sigma_{13}}{S_{13}}\right)^2\right]$ Where,

	Uni-directional Lamina Model	**Fabric Lamina Model**
	$S_{13}=\sigma_{13}^{m}+\langle-\sigma_{33}\rangle\tan\phi$ $\tilde{S}_{23}=\sigma_{23}^{m}+\langle-\sigma_{33}\rangle\tan\phi$	$S_{13}=\sigma_{13}^{m}+\langle-\sigma_{33}\rangle\tan\phi$ $\tilde{S}_{23}=\sigma_{23}^{m}+\langle-\sigma_{33}\rangle\tan\phi$

Note: $\langle a\rangle=\begin{cases}a\ if\ a>0\\0\ if\ a<0\end{cases}$

In `/FAIL/HASHIN`, material strength σ_1^t, σ_2^t, σ_3^t, σ_1^c, σ_2^c are derived from tension/compression test for composite.

Crush strength σ_c and fiber shear strength σ_{12}^f may be obtained from a quasi-static punch shear test (QS-PST). [22] Crush strength σ_c from the span to punch ratio (SPR) =0 and fiber shear strength σ_{12}^f from SPR=1.1.

ϕ is the Coulomb friction angle. It is observed that composite shear strength is higher if the composite is also under compression (rather than under tension). This is due to the friction between matrix and fiber.

The shear strength is assumed to be proportional to compression stress and is computed as:

$$S_{12}=\sigma_{12}^{m}+\langle-\sigma_{22}\rangle\tan\phi \qquad (194)$$

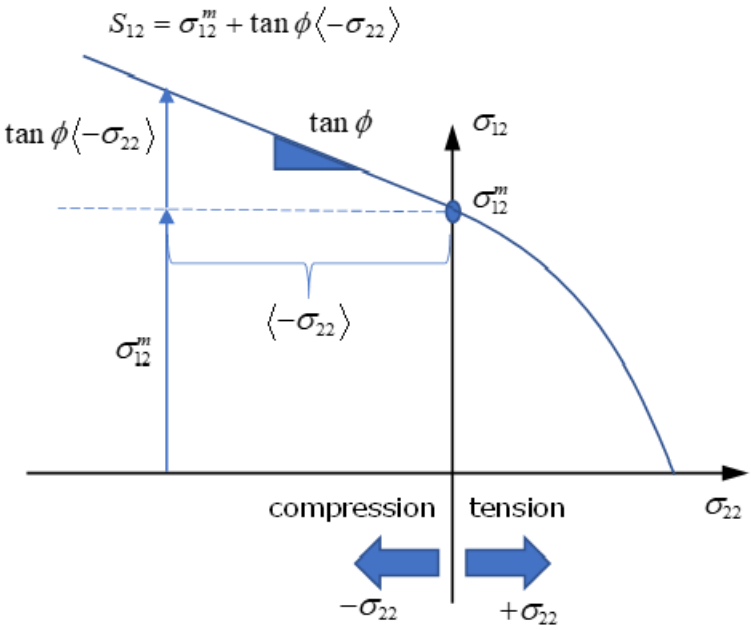


Figure 258:

Friction angle ϕ may be fitted with an Off-Axis compression test with different angles θ (for example, $30^{\circ}, 45^{\circ}, 60^{\circ}, \ldots$).

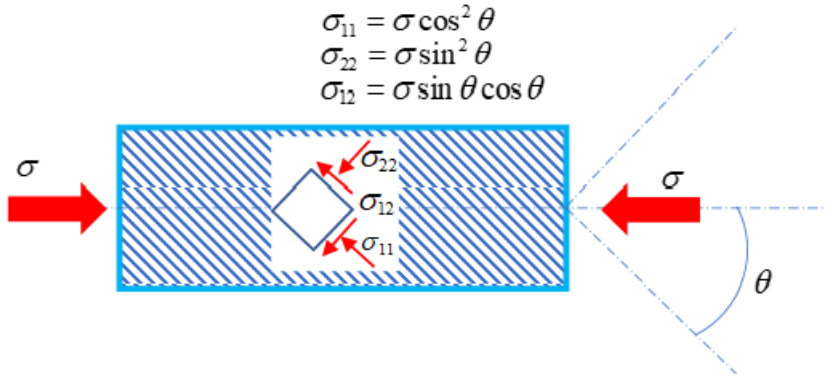


Figure 259:

$\sigma_{12}^{m}, \sigma_{13}^{m}, \sigma_{23}^{m}$ may be derived from a matrix shear test in three directions.

S_{del} is the scale factor for delamination criteria. It may be fitted with composite delamination experimental datain order for delamination failure to correlate with the damage area in experiment.

/FAIL/PUCK

In Puck failure, two types of failure are considered.

- Fiber fracture: composite fails, due to the fiber reaching the tensile or compression strength limit.
- Inter fiber failure (IFF): composite fails, due to the fiber matrix cracking.

Damage criteria	If D=1, then failure. If $0 \le D < 1$ D, then no failure. With $D = Max\left(e_f(tensile), e_f(compression), e_f(ModeA), e_f(ModeB), e_f(ModeC)\right)$
Fiber fraction failure	Tensile fiber failure mode: $\sigma_{11} > 0$ $e_f(tensile) = \frac{\sigma_{11}}{\sigma_1^t}$
	Compressive fiber failure mode: $\sigma_{11} < 0$ $e_f(compression) = \frac{\lvert\sigma_{11}\rvert}{\sigma_1^c}$
Inter fiber failure (IFF)[18]	Mode A, if $\sigma_{22} > 0$:

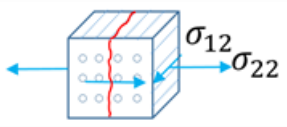


Figure 260:

$$e_f(ModeA) = \frac{1}{\bar{\sigma}_{12}}\left[\sqrt{\left(\frac{\bar{\sigma}_{12}}{\sigma_2^t} - p_{12}^+\right)^2 \sigma_{22}{}^2 + \sigma_{12}{}^2} + p_{12}^+ \sigma_{22}\right]$$

Mode C, if $\sigma_{22} < 0$:

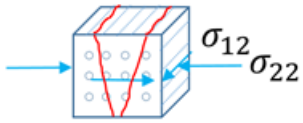


Figure 261:

$$e_f(ModeC) = \left[\left(\frac{\sigma_{12}}{2(1+p_{22}^-)\bar{\sigma}_{12}}\right)^2 + \left(\frac{\sigma_{22}}{\sigma_2^c}\right)^2\right]\left(\frac{\sigma_2^c}{-\sigma_{22}}\right)$$

Mode B:

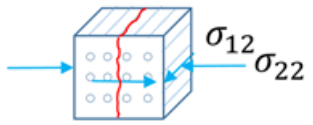


Figure 262:

$$e_f(ModeB) = \frac{1}{\bar{\sigma}_{12}}\left(\sqrt{\sigma_{12}^2 + \left(p_{12}^- \sigma_{22}\right)^2} + p_{12}^- \sigma_{22}\right)$$

In inter fiber failure, Mode A shows failure under tension in transverse fiber direction (90 degrees to fiber direction), and in this case, shear loading could reduce the failure limit.

If under compression in transverse fiber direction, at first increasing compression will increase composite shear loading (Mode B). If compression continues to increase, then shear loading will decrease (Mode C).

Input Parameters

For fiber fracture failure, you could obtain fiber strengths σ_1^t, σ_1^c from tension and compression composite tests in the fiber direction.

For inter fiber failure, you could obtain strengths σ_2^t, σ_2^c from tension and compression composite tests in the transverse fiber direction.

You could obtain shear strength $\bar{\sigma}_{12}$ with a pure shear test ($\sigma_2 = \sigma_1 = 0$).

With σ_2^t, σ_2^c, $\bar{\sigma}_{12}$, then p_{22}^- and p_{12}^- for Mode C and Mode B may be determined.

With σ_2^t, $\bar{\sigma}_{12}$ and additional tension-shear tests in the transverse fiber direction, p_{12}^+ may be determined. The additional tension-shear test in transverse fiber direction could take equal tension-shear (by $\sigma_{22} = \sigma_{12}$) loading.

Then you may derive the fracture curve in $\sigma_{22} - \sigma_{12}$ plane, as shown below.

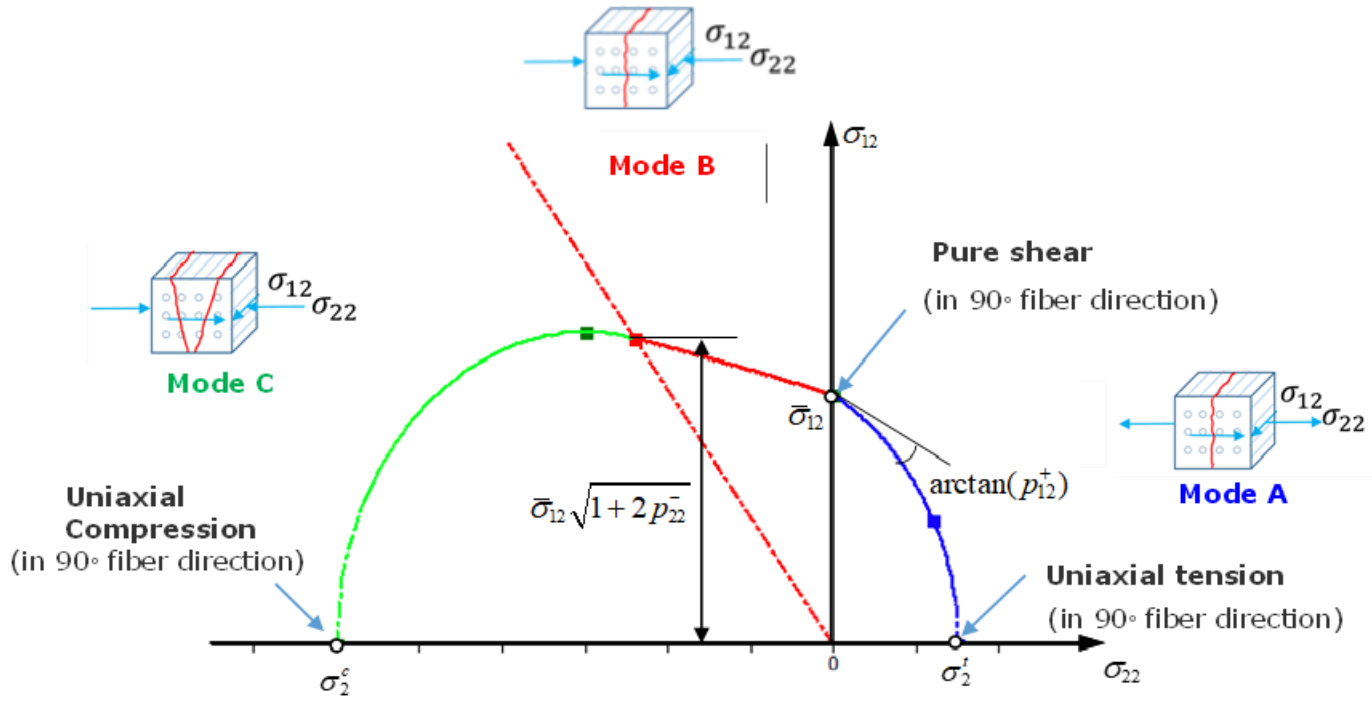


Figure 263: IFF fracture curve in $\sigma_{22} - \sigma_{12}$ plane

For p_{12}^+, p_{12}^-, p_{22}^- parameters [19]. For carbon fiber composite, use $p_{12}^+ = 0.35$, $p_{12}^- = 0.3$, $p_{22}^- = 0.2$ and for glass fiber composite, use $p_{12}^+ = 0.3$, $p_{12}^- = 0.25$, $p_{22}^- = 0.2$.

/FAIL/LAD_DAMA

`/FAIL/LAD_DAMA` is used to describe delamination between composite layers (damage propagation in the matrix). Assume that layers are connected through a virtual interface (contact).

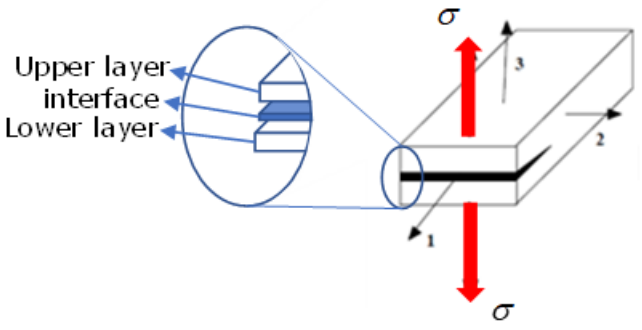


Figure 264:

For example, if composite is under loading, shown below, the traction σ and displacement in direction 3 δ is as shown in curve.

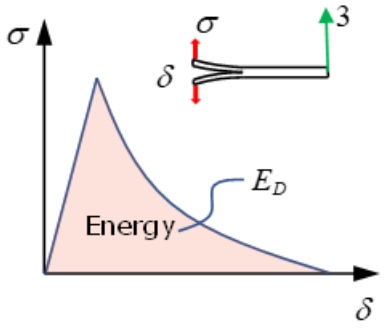


Figure 265:

The area under the traction versus displacement curve is the absorbed energy by delamination, which is also termed the *strain energy* of the damage interface. The failure is governed by this strain energy is described below, considering 3 Modes of delamination:

$$E_D = \frac{1}{2}\left[\frac{\langle \sigma_{33} \rangle^2}{K_3(1-d_3)} + \frac{\langle -\sigma_{33} \rangle^2}{K_3} + \frac{\sigma_{32}{}^2}{K_2(1-d_2)} + \frac{\sigma_{31}{}^2}{K_1(1-d_1)}\right] \quad (195)$$

Where, $\sigma_{33}, \sigma_{32}, \sigma_{31}$ are the stresses shown below in three modes of delamination behavior.

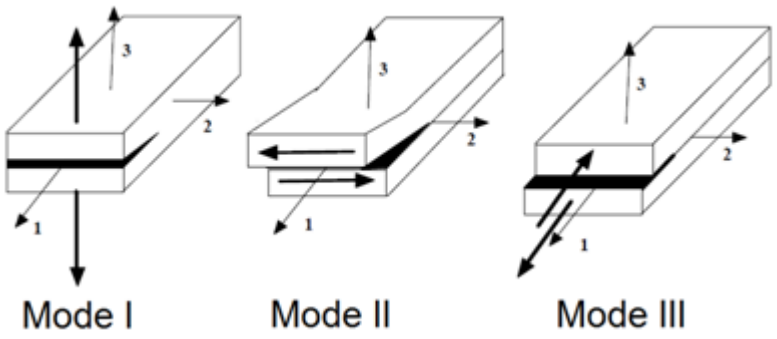


Figure 266:

With strain energy of delamination E_D, the thermodynamic force (contact force of virtual interface), also termed *damage energy* release rates may be calculated for these three modes:

- Model I (DCB specimen [21])

$$Y_{d_3} = \frac{\partial E_D}{\partial d_3}\bigg|_{\sigma=cst} = \frac{1}{2}\frac{\langle\sigma_{33}\rangle^2}{K_3(1-d_3)^2}$$

- Model II (ENF specimen [21])

$$Y_{d_2} = \frac{\partial E_D}{\partial d_2}\bigg|_{\sigma=cst} = \frac{1}{2}\frac{\sigma_{32}{}^2}{K_2(1-d_2)^2}$$

- Model III

$$Y_{d_1} = \frac{\partial E_D}{\partial d_1}\bigg|_{\sigma=cst} = \frac{1}{2}\frac{\sigma_{31}{}^2}{K_1(1-d_1)^2}$$

Where, K_3, K_2, K_1 are the stiffnesses of the virtual interface, also called *interlaminar stiffness*. These values may be computed as:

(196)

$$K_3 = \frac{2E_{33}}{t}$$

$$K_2 = \frac{2G_{23}}{t}$$

$$K_1 = \frac{2G_{13}}{t}$$

Where,

t	Thickness of the virtual interface. It may be assumed to be 1/5 layer thickness.
G_{13}, G_{23}, E_{33}	From upper or lower layer.

d_i	(with i=1,2,3), the damage variable. It has a range from 0~1. It starts to accumulate once the composite reaches Y_0.

An example of Mode I, during traction in direction 3, at the beginning d_3 always remains 0 until thermodynamic force Y_{d_3} reaches Y_0 (left figure).

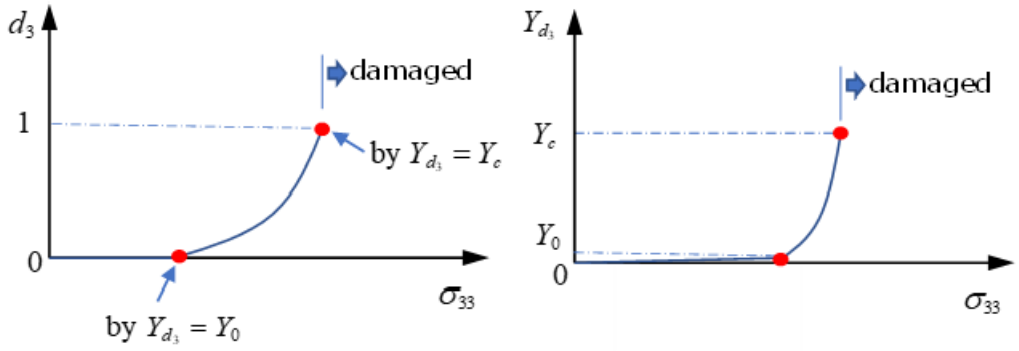


Figure 267:

After Y_0 is reached, the damage variable starts to increase and when it reaches 1, $d_3 = 1$ (thermodynamic force Y_{d_3} at this point then becomes the critical damage Y_c). The composite could be considered as fully delaminated and may be deleted immediately or the stress may be reduced. In Radioss, the option $\tau_{\max}$ is used to simulate exponential function stress reduction nd the stress at Y_c is $\sigma_d(t_r)$ (Stress Decrease in Damage).

The relation with thermodynamic force Y_{d_i} and d_i is:

- If $d \geq 1$, then take $d = 1$
- If $d < 1$, then d is function of Y (damage evaluation law):

$$d = w(Y) = \frac{\langle \sqrt{Y} - \sqrt{Y_0} \rangle}{\sqrt{Y_c} - \sqrt{Y_0}} \qquad (197)$$

$Y = Y_{d_3} + \gamma_1 Y_{d_1} + \gamma_2 Y_{d_2}$ with $Y_{d_i}|_t = \sup Y_{d_i}|_{\tau \leq t}$

Here, γ_1, γ_2 are scale factors to consider two other delamination modes. This may be validated with experiments (DCB and ENF specimen test [21]).

In the example of Mode I, γ_1, γ_2 may be 0 as this is pure delamination in direction 3 and then $Y = Y_{d_3}$, the relation of Y_{d_3} and d_3 is:

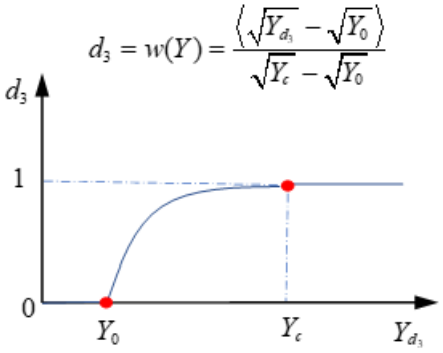


Figure 268:

How quickly will the damage variable increase? The damage velocity $\dot{d}$ (also called *damage evaluation law*) is computed as:

- If $d = 1$, then $\dot{d} = const.$
- If $d < 1$, then $\dot{d} = \frac{k}{a}[1 - \exp(-a\langle w(Y) - d \rangle)]$

$\frac{k}{a}$ is the maximum damage rate, which means minimum duration of the failure phenomenon. Its reciprocal, $\frac{a}{k}$ is called the *characteristic time*, which may be obtained with a one-dimension tensile test. [23]

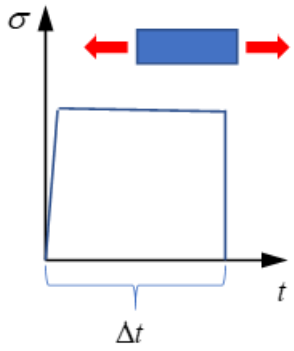


Figure 269:

Through tensile sample with different stress to find the minimum time of composite damage Δt, the $\sigma - \Delta t$ curve is vertical asymptote corresponding to the characteristic time $\frac{a}{k}$.

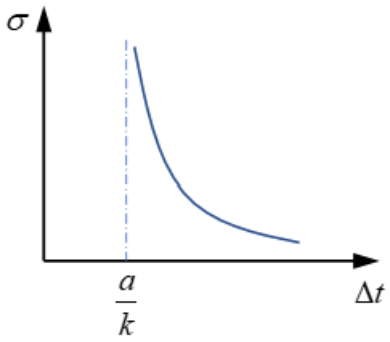


Figure 270:

Parameters a and k govern the damage evaluation law. For example, with constant parameter a (here $a = 1$), decreasing values of k leads to more brittle failure of the composite.

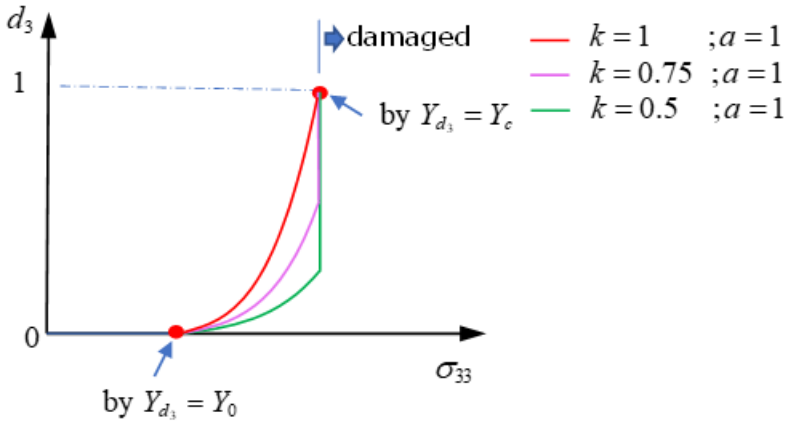


Figure 271:

With constant parameter k (here $k = 1$), increasing values of a leads to more brittle failure of the composite.

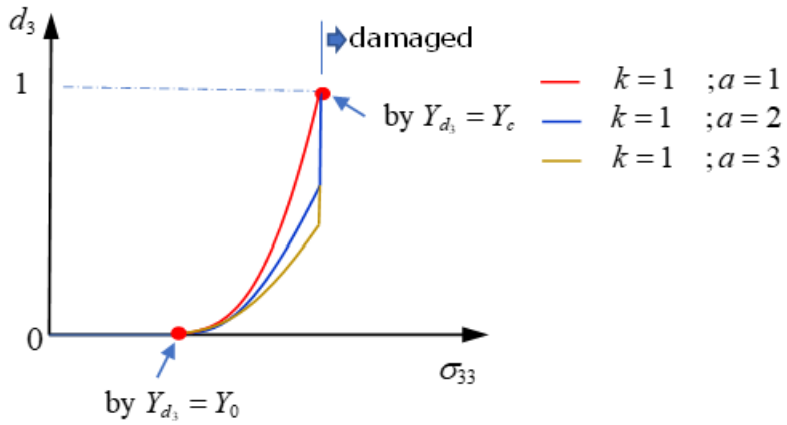


Figure 272:

/FAIL/CHANG

In Chang-Chang failure, two primary failure modes are considered.

- Fiber mode: composite fails, due to fiber rupture in tension or fiber buckling in compression.
- Matrix mode: composite fails, due to matrix failure under tension or compression.

This failure criteria is used only for shell elements.

Damage criteria		If $D=1$, then failure. If $0 \le D < 1$ D, then no failure. With $D = Max(e_{f2}, e_{c2}, e_{m2}, e_{d2})$.
Fiber breakage	Tensile fiber mode $\sigma_{11} > 0$	$e_{f2} = \left(\frac{\sigma_{11}}{\sigma_1^t}\right)^2 + \beta\left(\frac{\sigma_{12}}{\bar{\sigma}_{12}}\right)^2$
	Compression fiber mode $\sigma_{11} < 0$	$e_{c2} = \left(\frac{\sigma_{11}}{\sigma_1^c}\right)^2$
Matrix cracking	Tensile matrix mode $\sigma_{22} > 0$	$e_{m2} = \left(\frac{\sigma_{22}}{\sigma_2^t}\right)^2 + \beta\left(\frac{\sigma_{12}}{\bar{\sigma}_{12}}\right)^2$
	Compressive matrix mode $\sigma_{22} < 0$	$e_{d2} = \left(\frac{\sigma_{22}}{2\bar{\sigma}_{12}}\right)^2 + \left[\left(\frac{\sigma_2^c}{2\bar{\sigma}_{12}}\right)^2 - 1\right]\frac{\sigma_{22}}{\sigma_2^c} + \left(\frac{\sigma_{12}}{\bar{\sigma}_{12}}\right)^2$

Where,

direction 1 Fiber direction.

σ_1^t, σ_1^c	Fiber tensile/compressive strength.
σ_2^t, σ_2^c	Matrix strength. Tensile or compressive loading in direction 2 (transverse to direction 1).
$\bar{\sigma}_{12}$	Shear strength in composite ply plane.
β	Shear scale factor, which can be determined experimentally.

Stress Decrease in Damage

After reaching the damage criteria:

- HASHIN:

  $D = Max(F_1, F_2, F_3, F_4) \geq 1$
- PUCK:

  $D = Max(e_f(tensile), e_f(compression), e_f(ModeA), e_f(ModeB), e_f(ModeC)) \geq 1$
- LAD_DAMA:

  $d \geq 1$
- CHANG:

  $D = Max(e_{f2}, e_{c2}, e_{m2}, e_{d2}) \geq 1$

Stresses start decreasing and decrease gradually by using an exponential function to avoid numerical instabilities.

$$\begin{aligned}\boldsymbol{\sigma}(t) &= \boldsymbol{\sigma}_d(t_r) \cdot \mathrm{f}(t) \\ &= \boldsymbol{\sigma}_d(t_r) \cdot \exp\left(-\frac{t-t_r}{\tau_{\max}}\right)\end{aligned} \qquad (198)$$

with, $t \geq t_r$

The option $\tau_{\max}$ controls how gradually the stress is decreased in damage.

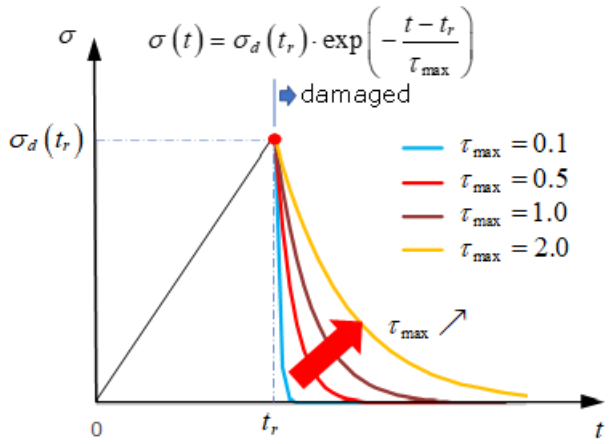


Figure 273:

Where,

$\boldsymbol{\sigma}_d(t_r)$	Stress components when damage is reached $D \geq 1$.
t_r	Time of $\boldsymbol{\sigma}_d(t_r)$.
$\tau_{\max}$	Time of dynamic relaxation. The higher the value of $\tau_{\max}$, the slower stress decreases during damage. Normally, it takes 10~20 time step.

17. Hashin, Z., "Failure Criteria for Unidirectional Fiber Composites," Journal of Applied Mechanics, Vol. 47, 1980, pp. 329-334.
18. A. Puck, J. Kopp, and M. Knops., "Failure analysis of FRP laminates by means of physically based phenomenological models". Composites Science Technology, 62. pp. 1633-1662. 2002.
19. A. Puck, J. Kopp, and M. Knops. "Guidelines for the determination of the parameters in Puck's action plane strength criterion". Composites Science Technology 62. pp. 371-378. 2002.
20. L. Gornet, "Finite Element Damage Prediction of Composite Structures".
21. Ladevèze, P., Allix, O., Douchin, B., Lévêque, D., "A Computational Method for damage Intensity Prediction in a Laminated Composite Structure", Computational mechanics—New Trends and Applications In: Idelsohn, S., Oñate E., and Dvorkin E., (eds.) CIMNE, Barcelona, Spain (1998).
22. Gama B.A., Gillespie J.W., Punch Shear Behavior of Composites at Low and High Rates[M]// Fracture of Nano and Engineering Materials and Structures. Springer Netherlands, 2006.
23. Allix, O. & Deü, Jean-François. (1997). Delay-damage modeling for fracture prediction of laminated composites under dynamic loading. Engineering Transactions. 45. 29-46.

Connections

Spotweld (Bolt or Adhesive Connection)

There are three different ways of modeling spotwelds:

- Nodal connection
- Spring (`/PROP/TYPE13`) connection
- Solid connection

Spring (`/PROP/TYPE13`) connection and solid connection may also be used to model a bolted or adhesive connection (glue).

Nodal Connection

A single interface TYPE2 with the first surface as the main side and some nodes from the second surface as secondary nodes: With this solution the mesh of the main surface can be independent of the spotweld location. Hourglass problems disappear on the main surface. On the second shell, the surface mesh must respect the spotweld location and the hourglass problem will remain. The main problem with this modeling approach is the undeformability of the connection and its infinite strength.

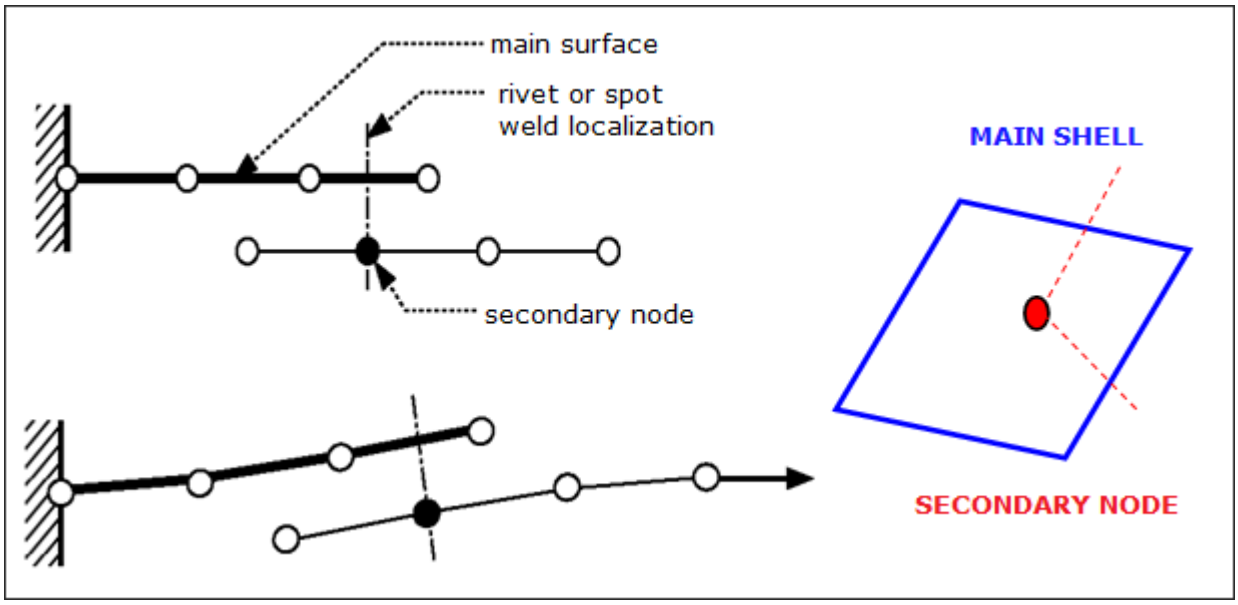


Figure 274: Example of Connection between 2 Shell Surfaces

Spring (`/PROP/TYPE13`) Connection

Two tied interfaces and a spring: The use of two tied interfaces will provide a full symmetrical solution allowing a free mesh on the two surfaces and avoiding hourglass. The spotweld is modeled with a beam type spring element. The spring element uses independent nodes not connected directed to the shell elements. One of the two nodes is located on the first surface (or near to it, there is no need to be located exactly on the shell surface) and the second node is located on the second surface. One tied interface connects one spring node with the first surface and a second tied interface does the same for the second node on the second surface.

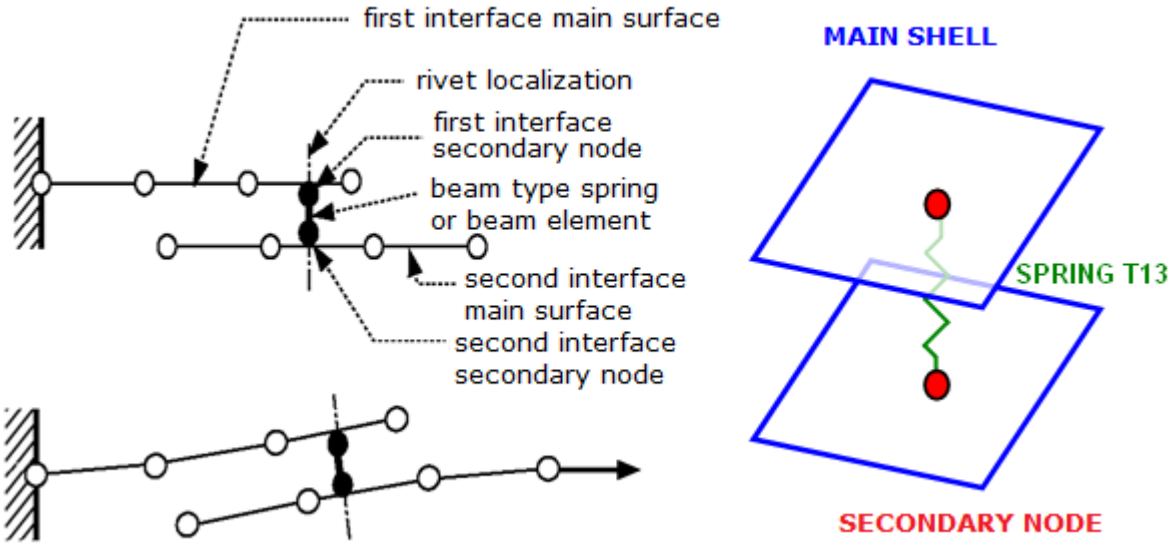


Figure 275: Spotweld Modeling

To create a spotweld using this method is a good alternative solution with this approach the connection location is independent from the shell mesh. It is more accurate than the above node connection modeling, since the spotweld properties are input directly for the spring TYPE13.

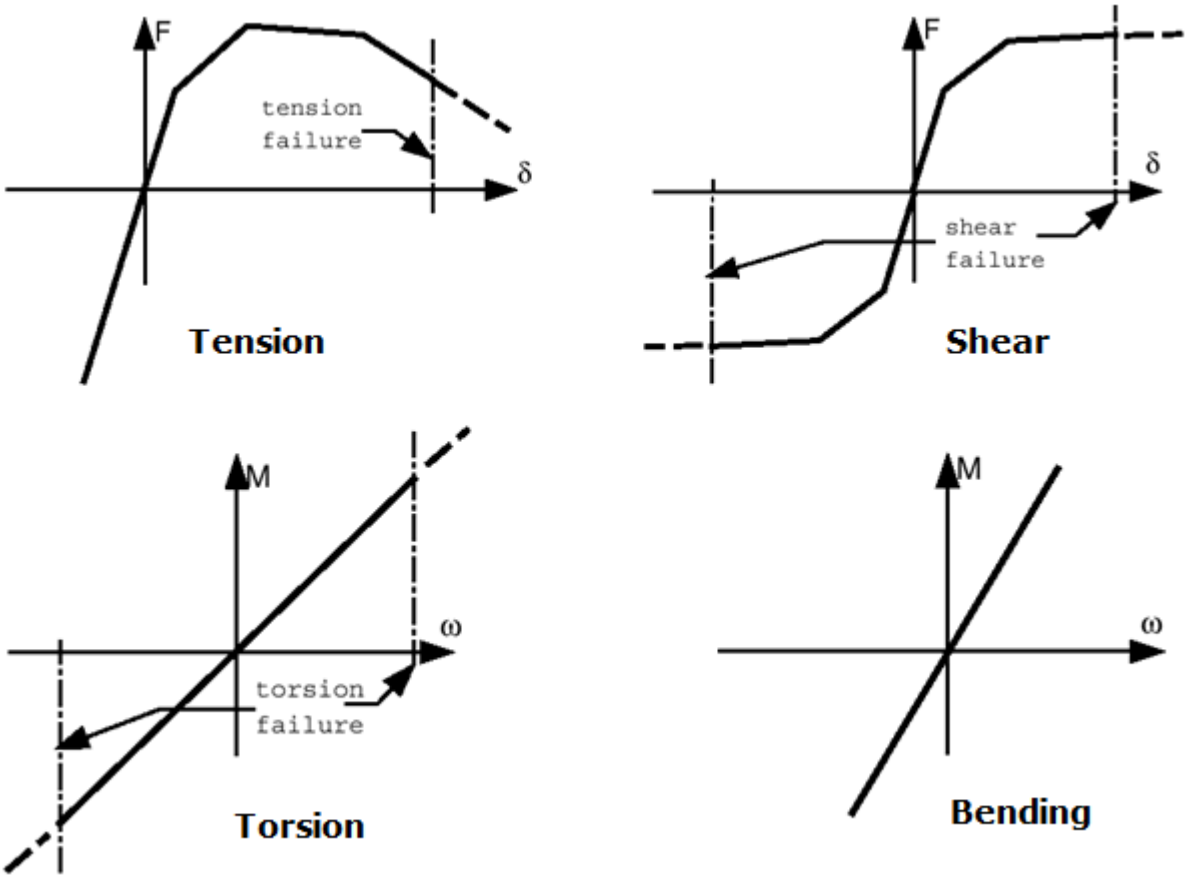


Figure 276: Spring TYPE13 - Typical Input for Spotweld

Moreover, there are two different ways to model rupture of the spotweld:

1. Use failure criteria which are available for a spring TYPE13. For more details, see the comments for failure criteria in /PROP/TYPE13 (SPR_BEAM).
2. Use $Spot_{flag}$= 20, 21, or 22 in the Tied Contact (Tied Contact (/INTER/TYPE2)).

> **Note:** The spring TYPE13 modeling technique for spotwelds can also be used for other kinds of connections such as welding lines, hemming, glue and bolts. For bolt modeling, the use of a tied interface is not necessary, as the shell nodes can be put directly in the rigid bodies.

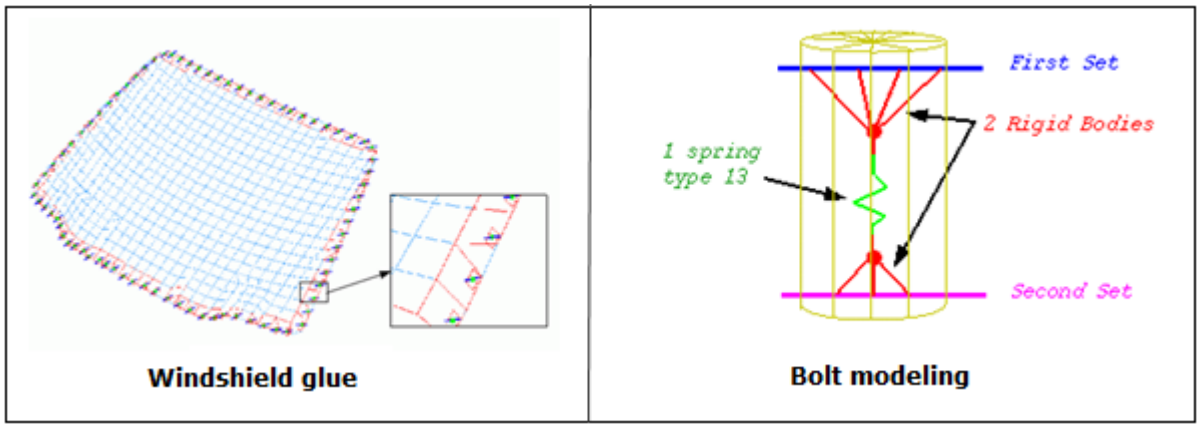


Figure 277: Glue and Bolt Modeling Examples

Note: With a tied interface, the secondary node mass is transferred to the main nodes, if $Spot_{flag}$ is set to 1. The secondary node inertia is equally distributed over the main nodes by adding mass, so that the induced inertia (at the center of the main surface) is equal to the inertia of the secondary node. If the main surface is a perfect square, the added mass is computed as:

$$l_s = 4\Delta m \cdot L^2$$

Δm: added mass
L: distance between the main node and the center
l_s: inertia of the secondary node

As long as the secondary node inertia is realistic, the added mass will be very small. A large added mass is observed if the secondary node is too great a distance from the main surface. The ideal is for the secondary node to lie on the main surface plane, right at its center. If this is not the case, the secondary node has inertia at the center of the shell surface:

$$l_s = m_s \cdot L_s 2$$

m_s	Secondary node mass
L_s	Distance between the secondary node and the center
l_s	Inertia of the secondary node

Consequently, a new added mass is set to the main nodes, so that the inertia (due to this new added mass) is equal to the inertia, due to the off-centering of the secondary node.

$$4\Delta m \cdot L^2 = m_s \cdot L_s 2 \qquad (199)$$

If $Spot_{flag}$=0, there is no added mass, since the secondary node inertia instead is transferred as inertia to the main node. An added inertia that is too large will seriously decrease the accuracy.

See Also
RD-E: 4802 Solid and Spring Spotweld in Crash Box

Solid Spotweld

Uses 8-node brick element (with `/PROP/TYPE43`) and `/MAT/LAW59+/FAIL/CONNECT` (or `/MAT/LAW83+/FAIL/SNCONNECT`) to model solid spotwelds, which may provide more accurate results.

Solid Element and Property

The brick element uses `/PROP/TYPE43` and it has 4 integration points on the shear plane, which is between plane (1, 2, 3, 4) and plane (5, 6, 7, 8). There is one integration point in normal direction *t*. This element type does not have a time step itself and its stability is ensured by its nodal connections. This means that the thickness of spotweld can be very small. This characteristic is very useful for modeling glue.

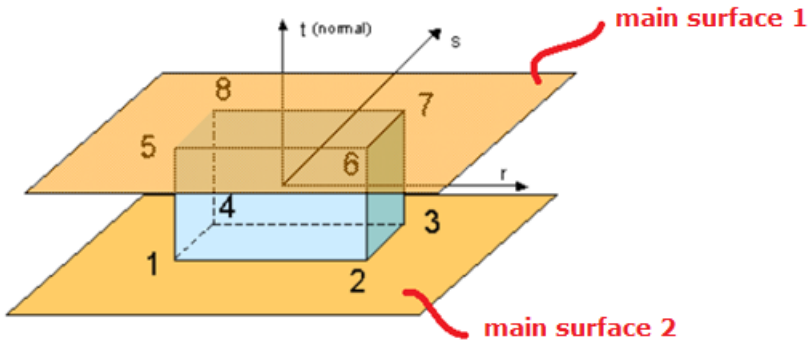


Figure 278:

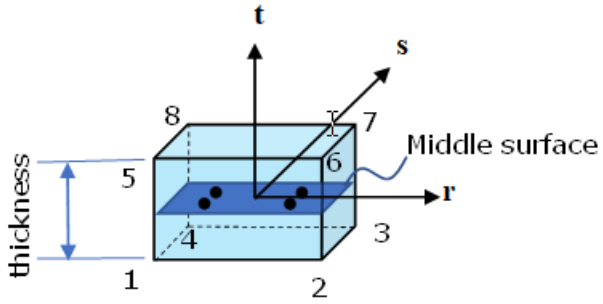


Figure 279:

Connected to Shell Sheet

`/INTER/TYPE2` may be used to connect solid spotwelds with two (upper and lower) main surfaces. Nodes of plane (1,2,3,4) tied on one shell, and nodes of plane (5,6,7,8) tied on another shell. It is not allowed to have any other plane (for example, plane (1,4,8,5)) tied on a shell.

Material and Failure Model

Solid spotwelds in Radioss may be modeled with `/MAT/LAW59+/FAIL/CONNECT` (or `/MAT/LAW83+/FAIL/SNCONNECT`). This material model should be validated with four load cases of spotweld tests.

- Shear test (angle of loading and spotweld upper surface is 0 degrees below named 0 degree test)
- Normal tensile test (90 degree test),
- Shear and normal combined test (for example, 30 degree test, 45 degree test or 90 degree test)
- Moment test (peel test)

E-Modulus

The stiffness of the spotweld in different tests is different. In the normal test, it is lower than in the shear test, due to deformation of the upper and lower sheets. Therefore, normally the stiffness measured is taken from true stress versus displacement curve of the shear test.

/MAT/LAW59+/FAIL/CONNECT

- Material yield curve:

  In LAW59, spotweld material yield curves in normal direction and in shear direction are requested. The yield curve ($Y\_fct\_ID_N$) in normal direction may be determined from the normal tensile test (90 degree test) and the yield curve ($Y\_fct\_ID_T$) in shear direction may be determined from the shear test (0 degree test).

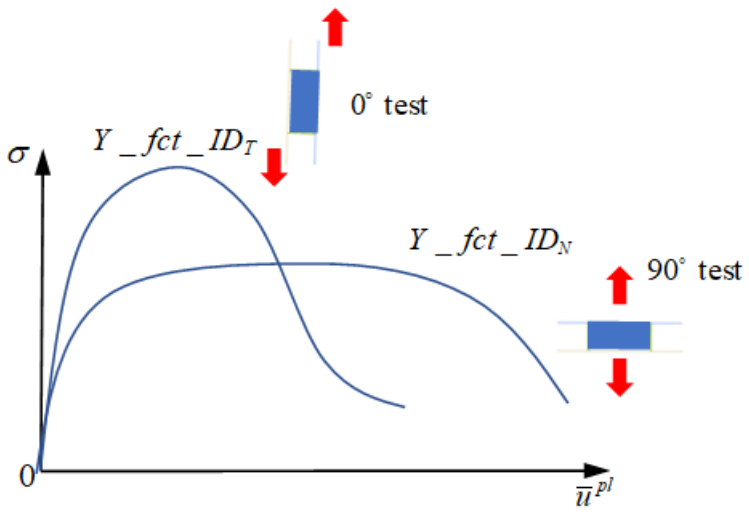


Figure 280:

  In this case, the maximum stress is also described inside the curves. Given the reference displacement rate SR_{ref} of the input yield curve, Radioss will consider the displacement rate effect with respect to this reference displacement rate.

- Spotweld failure:

  Solid spotweld damage and failure may be considered with `/FAIL/CONNECT`. Displacement criteria and/or energy criteria may be used to describe the failure of the spotweld.

- For displacement criteria, failure occurs when the normal displacement or shear displacement is reached according to 2 alternative behavior types:
  - Uncoupled failure (I_{fail}=0: uni-directional failure)

$$\bar{u}_i \cdot \mathrm{f}(\dot{\bar{u}}) > \bar{u}_{maxi} \qquad (200)$$

    with i = 33 for normal direction and i = 13 or 23 for tangent directions.

    In the normal tensile test (90 degree test), element fails once user-defined maximum displacement $\bar{u}_{\max N}$ is reached.

    In the shear tensile test (0 degree test), element fails once user-defined maximum displacement $\bar{u}_{\max T}$ is reached.

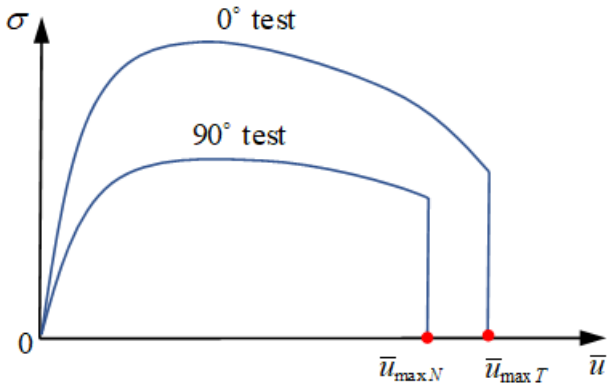


Figure 281:

    In a combined mode test (for example, 30 degree test or 60 degree test), failure in the solid spotweld does not consider shear and normal combined stress effect. Failure in each direction is considered separately. The element fails as soon as either of these two stresses reaches its corresponding maximum displacement. To consider combined stresses, instead set I_{fail}=1 and combined stress effect will be then considered.

  - Coupled failure (I_{fail}=1: multi-directional failure)

$$\left|\frac{\bar{u}_N}{\bar{u}_{\max N}} \cdot \alpha_N \cdot \mathrm{f}_N(\dot{\bar{u}}_N)\right|^{\exp_N} + \left|\frac{\bar{u}_T}{\bar{u}_{\max T}} \cdot \alpha_T \cdot \mathrm{f}_T(\dot{\bar{u}}_T)\right|^{\exp_T} > 1 \qquad (201)$$

    With I_{fail}=1, in combined mode test, the element fails before reaching the maximum stress $\bar{u}_{\max N}$ or $\bar{u}_{\max T}$ which is closer to reality. To describe the curve failure surface you need at least 4 different combined tests to fit the parameters α_N, α_T, $\exp_N$, $\exp_T$.

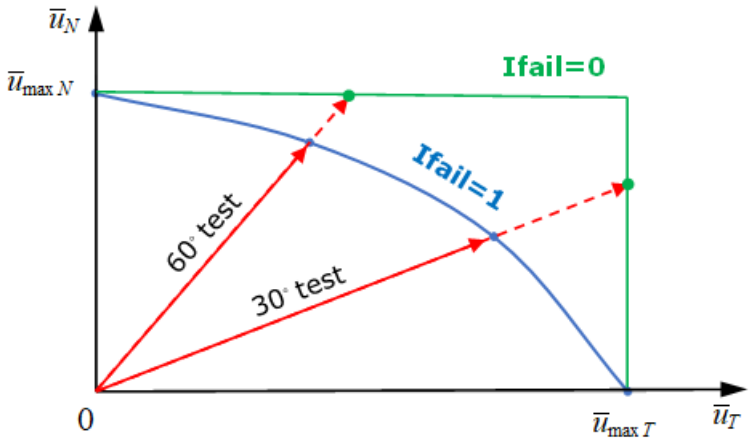


Figure 282: Failure surface

- For energy criteria, failure occurs when the internal energy in normal direction or internal energy in shear direction is reached, corresponding to maximum internal energy $EN_{\max}$, $ET_{\max}$.

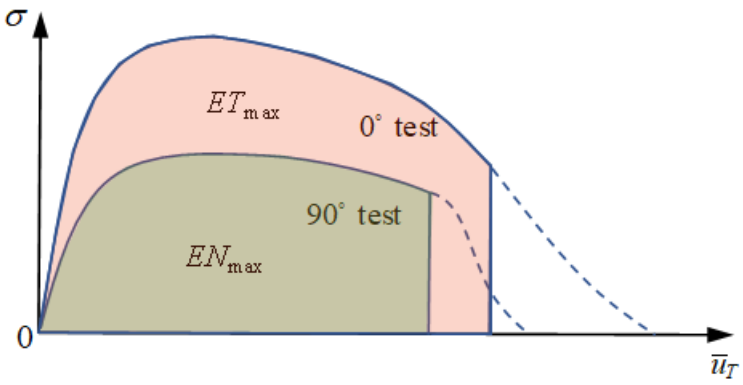


Figure 283:

In a combined mode test, element failure is also considered with respect to the multi-direction effect on internal energy. If internal energy in normal direction and in shear direction are input, the element fails, if satisfied via:

$$\left(\frac{En}{EN_{\max}}\right)^{N_n}+\left(\frac{Et}{ET_{\max}}\right)^{N_t}\geq 1 \qquad (202)$$

To input only total internal energy $EI_{\max}$, the element fails, if satisfied via:

$$\frac{E(t)}{EI_{\max}}\geq 1 \qquad (203)$$

If both $EI_{\max}$ and $EN_{\max}$, $ET_{\max}$ are input, the element fails, due to whichever of these two criteria is reached first.

Both displacement criteria and energy criteria may be defined. The element fails, due to whichever criteria is reached first. The element deletion occurs when one integration point reaches the failure criteria, if I_{solid}=1 or when all integration points reach the failure criteria, if I_{solid}=2.

- Spotweld softening:

  After reaching the failure criteria (either displacement criteria or energy criteria) stress is reduced to 0 directly or may be gradually controlled with parameters $T_{\max}$ and N_{soft} with:

$$\sigma = \sigma_f \left(1 - \frac{D}{T_{\max}}\right)^{N_{soft}} \quad (204)$$

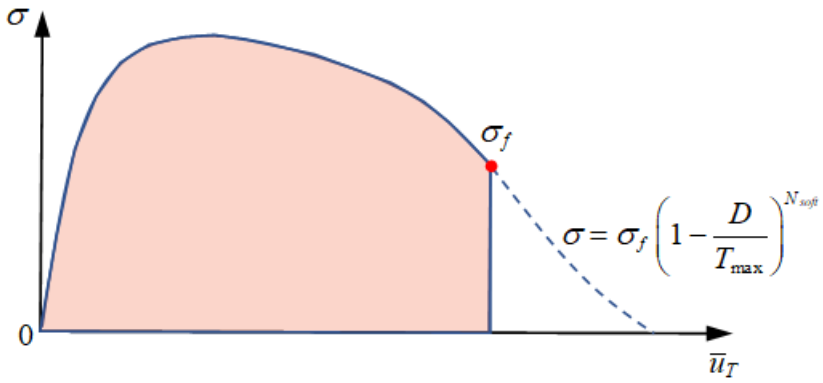


Figure 284:

Figure 285 shows the effect of different $T_{\max}$ and N_{soft} on stress reduction behavior.

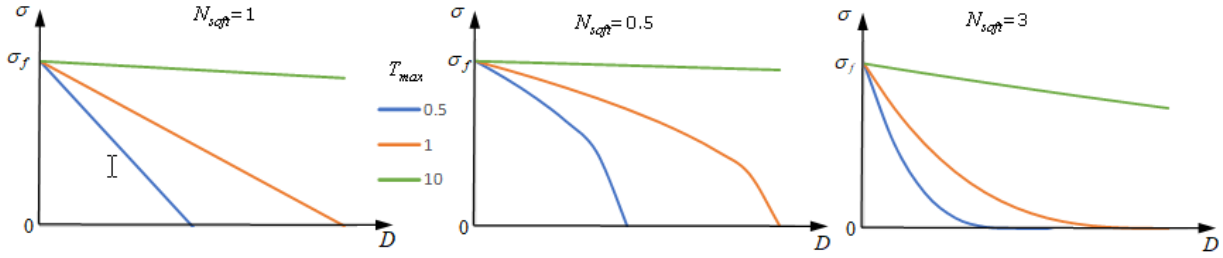


Figure 285:

/MAT/LAW83+/FAIL/SNCONNECT

- Material yield curve:

  In LAW83, the spotweld material curve may be input with *fct_ID*$_1$. Where in LAW59 input, two yield curves for normal direction and shear direction are required, LAW83 uses just one curve. This curve should take the yield curve from shear test. Furthermore, the yield curve *fct_ID*$_1$ for LAW83 is not defined as true stress versus plastic displacement (as in LAW59), but should be a normalized stress

versus plastic displacement curve. Yield stress is normalized by maximum stress which are input as parameters R_N, R_S in LAW83.

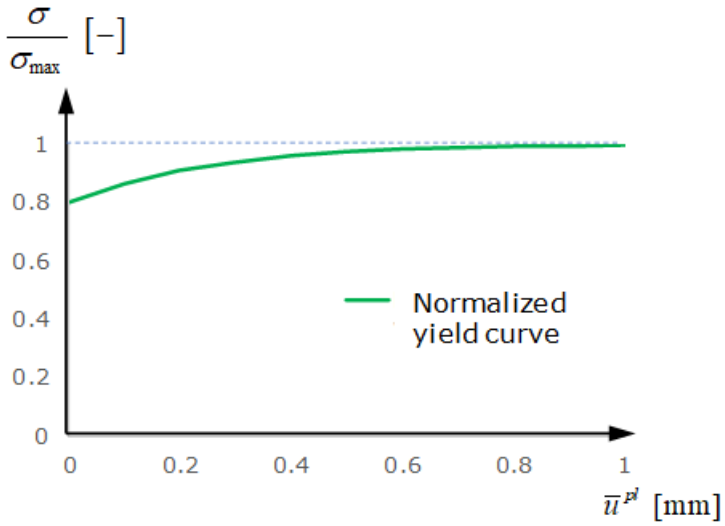


Figure 286:

The yield curve is different due to different combinations of normal stress and shear stress in the spotweld. This may be described with parameter β in LAW83 (it is not considered in LAW59). The normalized yield stress in LAW83 is:

$$\sigma_y = \left[\left(\frac{\sigma_n}{R_N \cdot \mathrm{f}_N(1-\alpha \cdot \mathrm{sym})}\right)^{\beta} + \left(\frac{\sigma_s}{R_S \cdot \mathrm{f}_S}\right)^{\beta}\right]^{\frac{1}{\beta}} \quad (205)$$

In cases where the moment effect is not considered, the normalized yield stress in LAW83 is:

$$\sigma_y = \left[\left(\frac{\sigma_n}{R_N \cdot \mathrm{f}_N}\right)^{\beta} + \left(\frac{\sigma_s}{R_S \cdot \mathrm{f}_S}\right)^{\beta}\right]^{\frac{1}{\beta}} \quad (206)$$

Figure 287 shows the difference of normalized maximum stress in combined tests between LAW83 and LAW59.

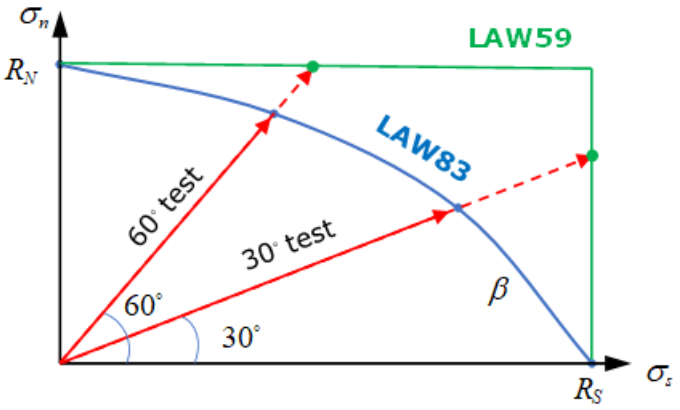


Figure 287:

Figure 288 shows the effect of varying β on normalized maximum stress in combined tests using LAW83.

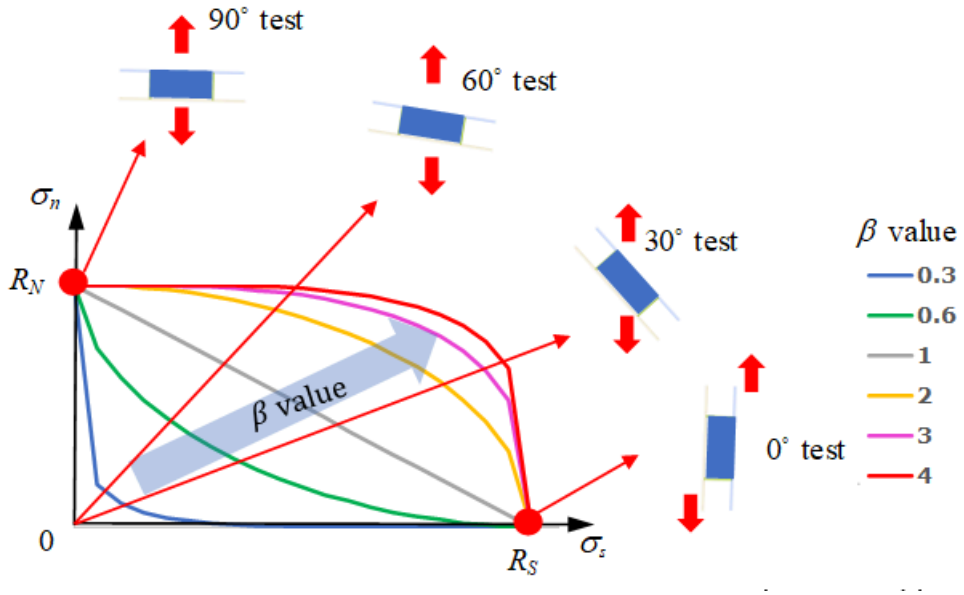


Figure 288:

Parameter α is used to describe the moment effect in the spotweld.

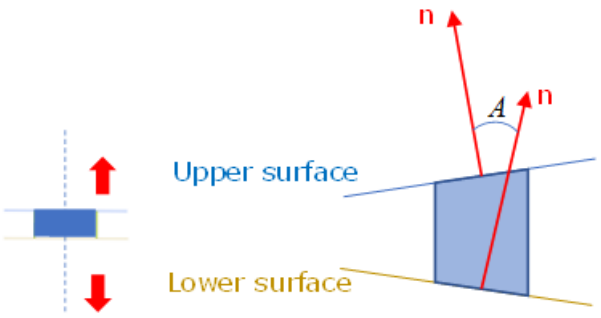


Figure 289: Non-central tensile test (peel test)

Use $\alpha \cdot \mathrm{sym}$ to reduce the maximum stress of peel test. sym is the sin of the angle between the spotweld upper surface and lower surface. It is changed during spotweld deformation and is in range of [-1,1]. The α parameter may be fitted with a simple FEM model to match the real experiment data. [24]

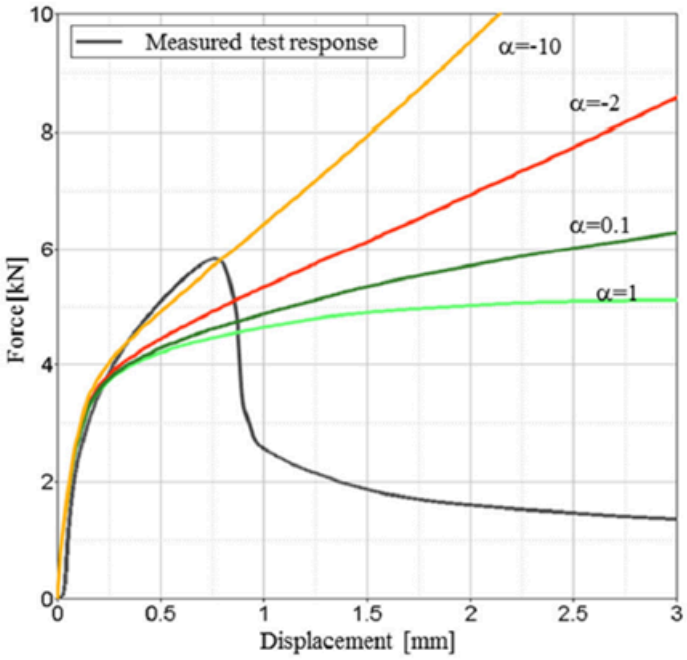


Figure 290: Different α effects on peel test on force versus displacement

The displacement rate effect on the material yield curve may also be considered with curve inputs *fct_ID$_N$* and *fct_ID$_S$*.

- Material damage and failure:

  For spotweld failure, `/FAIL/SNCONNECT` may be used. In this failure model, the plastic displacement (in both normal and shear directions) of damage beginning and failure are needed.

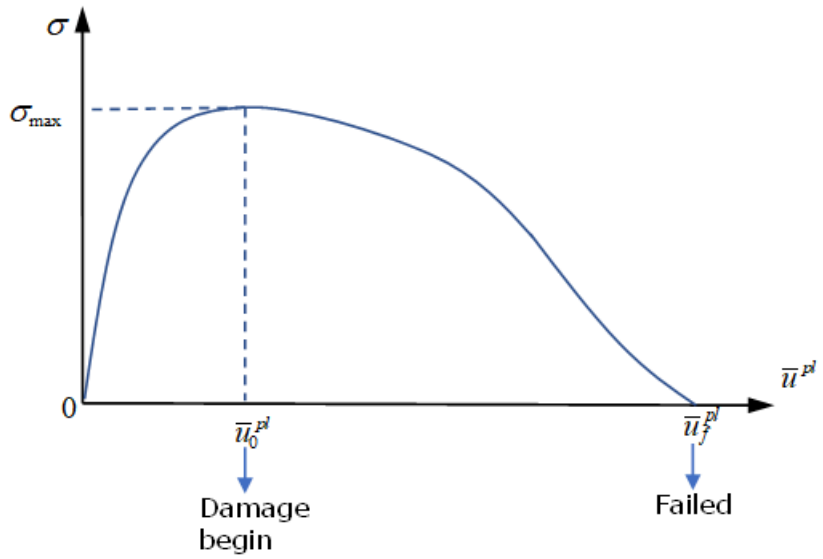


Figure 291:

For a combined mode test, similar to maximum stress in LAW83, β_0 is needed to describe plastic displacement at damage beginning and β_f to describe plastic displacement at failure.

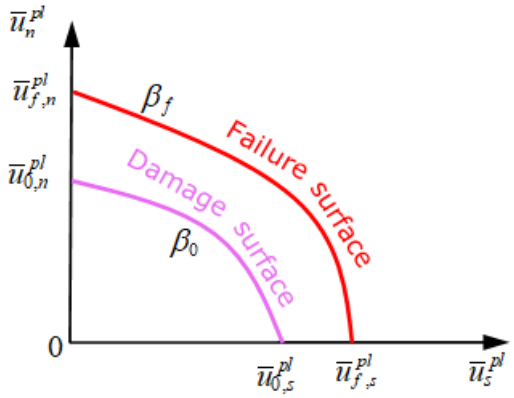


Figure 292:

For spotwelds with moment (peel test), similar to maximum stress in LAW83, α_0 is needed to describe plastic displacement at damage beginning during the peel test and α_f to describe plastic displacement at failure of peel test.

Table 20: General Capability of the Two Spotweld Modeling Approaches

		/MAT/LAW59+/FAIL/CONNECT		/MAT/LAW83+/FAIL/SNCONNECT
Yield curve		Two yield curves (in normal and shear directions)		One normalized yield curve with maximum stress R_N, R_S
Maximum stress in combined mode test		Normal and shear effect in combined test not considered.		Use α to consider normal and shear effect in combined mode tests.
Failure	Failure criteria	Displacement criteria	Uni-direction failure Multi-direction failure	Displacement criteria, Multi-direction failure
		Energy criteria	Uni-direction failure Multi-direction failure	
	Failure in combined mode test	Proportionally controlled with α_T, α_N, $\exp_T$, $\exp_N$ in displacement criteria and with N_n, N_t in energy criteria.		Controlled with β_0, β_f
	Moment effect (peel test)	No control in input		Controlled with α_0, α_f
	Softening	With $T_{\max}$, N_{soft} (stress curve decreases)		With reference to damage displacement and failure displacement (stress linear decrease)

24. Pasligh, N., Schilling, R., and Bulla, M., "Modeling of Rivets Using a Cohesive Approach for Crash Simulation of Vehicles in Radioss," SAE Int. J. Trans. Safety 5(2):2017, doi:10.4271/2017-01-1472

Bolt Modeling for Crash

There are different ways for modeling bolt connection, and more detailed model accuracy of the results:

- Rigid bolt
- Rigid bolt with spring (and pretension)
- Deformable bolt with spring and pretension
- Deformable bolt with initial stress

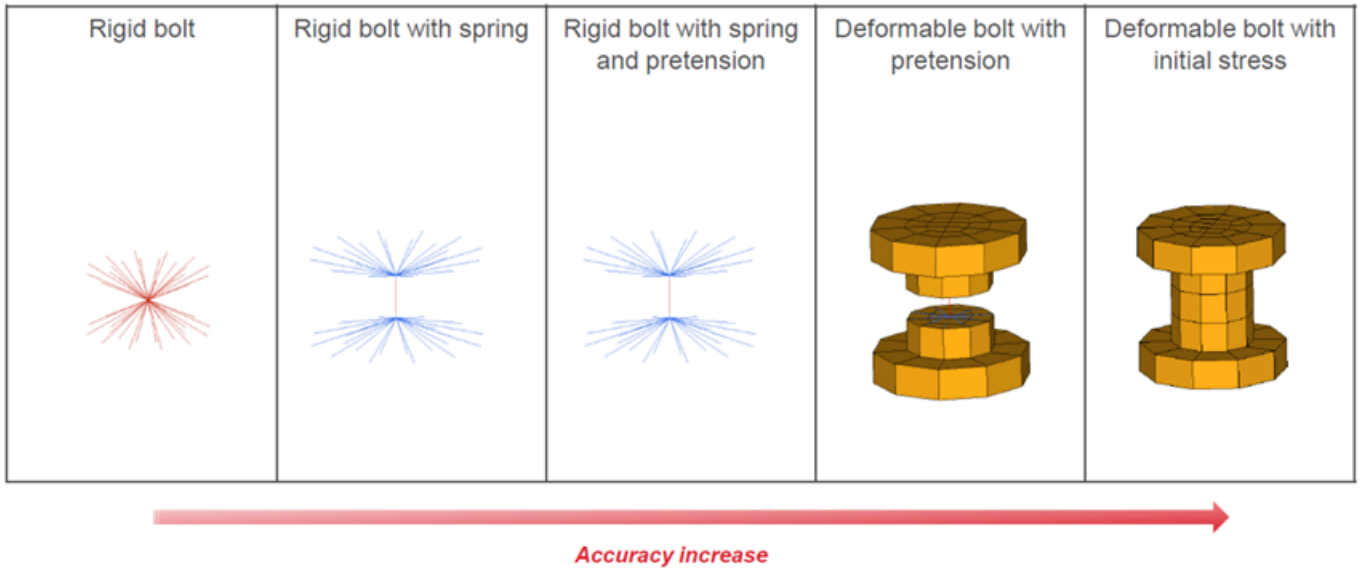


Figure 293: Bolt Modeling Methods

Rigid Bolt

Use one rigid body (`/RBODY`) connected to the parts where the bolt is linked is the simplest method. Although this method is very stable and easy to modeling, you cannot consider pretension, elasto-plastic and rupture behavior for bolt, and there is no force output and model the bolt head passing through the hole is not possible.

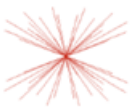


Figure 294: Rigid Bolt Modeling Method

Note: The main node should be a free node.

Rigid Bolt with Spring (and Pretension)

Use two rigid bodies (`/RBODY`) connected to each set of parts which are linked with the bolt and then set one spring element (`/PROP/SPR_BEAM`) between both rigid bodies. With this method it is possible to describe the elasto-plastic behavior rupture behavior in any direction, and the force passing through

the bolt (normal force, shear force, moments) could be output, using /TH/SPRING. This method is often used in the automotive crash analysis, but it has limited capability to model the pretension.

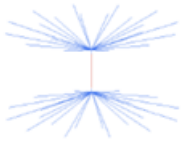


Figure 295: Rigid Bolt with Spring

Note:

1. The main node should be a free node.
2. The spring element is attached to the rigid bodies as secondary node.

To describe pretension with this method another two spring elements (/PROP/SPR_PRE and /PROP/SPR_BEAM) need to be added. /IMPDISP or /CLOAD could be used to simulate preloading. Auto-balance state is calculated by Radioss.

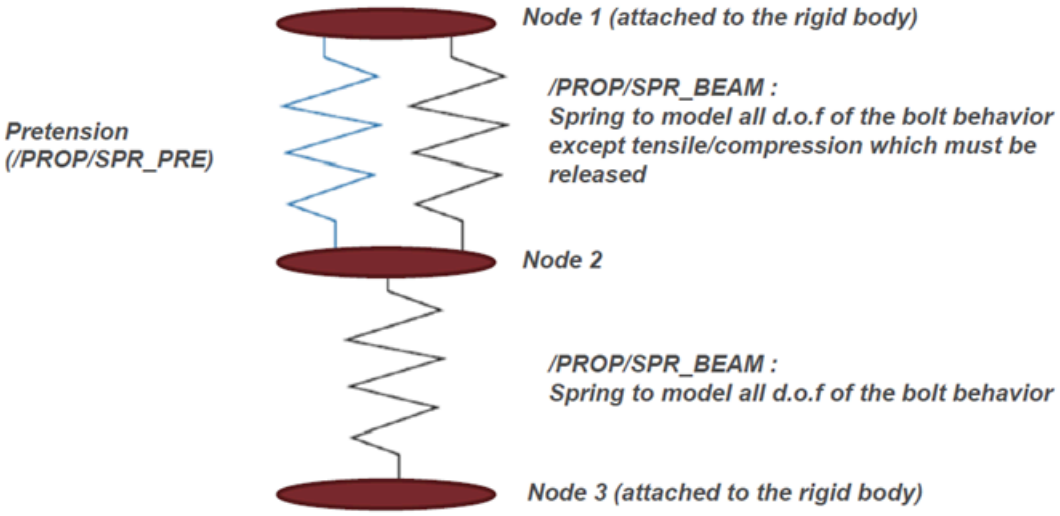


Figure 296: Modeling Pretension with Springs

Deformable Bolt with Spring and Pretension

Meshed bolt with solid elements (/BRICK or any other solid element type) and cut in the middle.
Two rigid bodies connected to each side of meshed bolt and spring series (two springs with /PROP/SPR_BEAM and one spring with /PROP/SPR_PRE) like above between rigid bodies are used to describe the pretension. With this more detailed model, the bolt head passing through the hole could be considered in simulate.

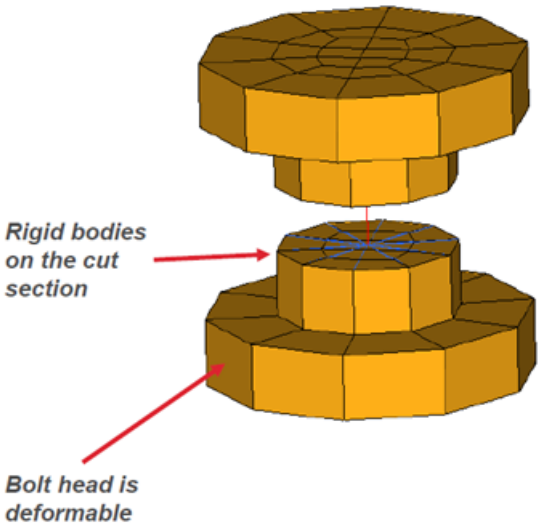


Figure 297: Deformable Bolt with Spring and Pretension

Deformable Bolt with Initial Stress

With this method, bolt is fully-meshed with solid elements (`/BRICK`), which is more realistic, and the material and failure character could be easily defined. Use `/SECT` to get section force/moment (`/TH/SECT`) in bolt. To define the pretension, initial stress (`/INIBRI/STRS_F`) is needed to get an initial analysis (tensile test on the bolt).

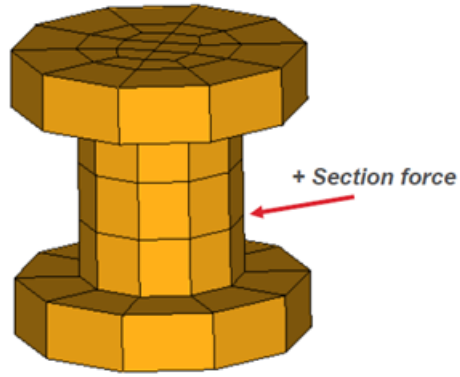


Figure 298: Fully-meshed Deformable Bolt

Note:

1. The rupture and stress accuracy depend of the mesh size.
2. Section force in the bolt is necessary to measure the forces passing through the hole.

Kinematic Constraints

In Radioss, a kinematic condition is a nodal constraint applied to a set of nodes.

There are several different types of nodal constraints:

- Rigid Wall
- Rigid Body
- Boundary Conditions
- Tied Interface
- Cylindrical Joint
- Rigid Link
- Gear Type Joint

Numerical methods available in Radioss that apply to the kinematic conditions are:

Penalty method	Rigid walls
Main secondary kinematic condition	Interface TYPE2, rigid body
Local kinematic condition	Rigid link, cylindrical joint
Lagrange multiplier	Old and new interfaces, rigid wall, rigid body, etc.

Rigid Body (/RBODY)

A rigid body is defined by a set of secondary nodes and a main node. It can be compared to a part with an infinite stiffness. No relative displacement is allowed between secondary nodes, and the general motion of the rigid body manages the main node.

As a kinematic condition is applied on each secondary nodes and for all directions, no other nodal constraint is allowed. However, in the case of the Lagrange Multiplier method, the solution can be found if no incompatible kinematic conditions are applied.

Four options are available to position the center of gravity of the rigid body:

Case 1:	COG is computed with secondary and main nodes mass and coordinates Rigid body mass is the sum of the added mass and secondary node mass
Case 2:	COG is computed with secondary nodes mass and coordinates only Rigid body mass is the sum of the added mass and secondary node mass
Case 3:	COG is set at main node's coordinates

	Rigid body mass is the sum of the added mass and secondary node mass
Case 4:	COG is set at main node's coordinates
	Rigid body mass is equal to the added mass

It is strongly recommended to use an artificial node (not part of an element) as a main node, since Radioss Starter is likely to move the main node. The main node is moved by Radioss Starter to the center of gravity; unless the *ICoG* flag is set to 3 or 4. It is advised to set *ICoG* to 2 to get the most realistic behavior; the center of gravity is then computed taking into account only the secondary nodes. If *ICoG* is set to 1, the main node, with its own mass, is included to compute the center of gravity.

If the main node is initially set to the center of gravity, the behavior using *ICoG* = 1 or 2 is similar.

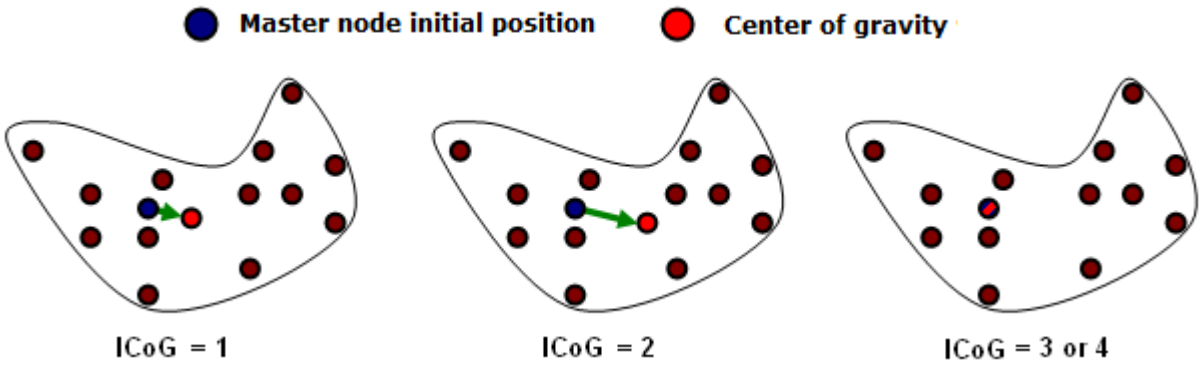


Figure 299: Center of Gravity Computation

In a car crash simulation, rigid bodies are largely used and three typical uses of rigid bodies can be distinguished:

- Rigid body covering a part of a finite element model, including shells, solids or other elements: In this case, the mass of secondary nodes gives the total rigid body mass and no added mass is needed. The main node can be located anywhere, and it will be moved to the center of mass. This kind of rigid body saves a large amount of CPU time.
- Rigid body representing a non-modeled component connected on some structural nodes: In this case, only a few number of secondary nodes are used to connect the rigid body to the finite element model. The mass and inertia are added and the main node is located at the component center of mass. The main node will only move a little, taking into account the mass of the secondary nodes. In some cases, a dummy mesh is used to visualize the rigid body or to simulate the contacts, but if the dummy elements have a small mass, the previous remarks are still true.
- Rigid body used to connect two or more parts together: For these rigid bodies no added mass is needed and the main node can be located anywhere. A spherical inertia must be used for these rigid bodies; as these rigid bodies are usually very small (4 to 8 nodes), the inertia is often very small in one direction and very large for one specific direction. This may lead to instability; therefore, through the use of spherical inertia, inertia will be identical for any direction.

Activation/Deactivation

Rigid bodies can be activated or deactivated with `/SENSOR` or using the `/RBODY/ON` or `/RBODY/OFF` Engine options.

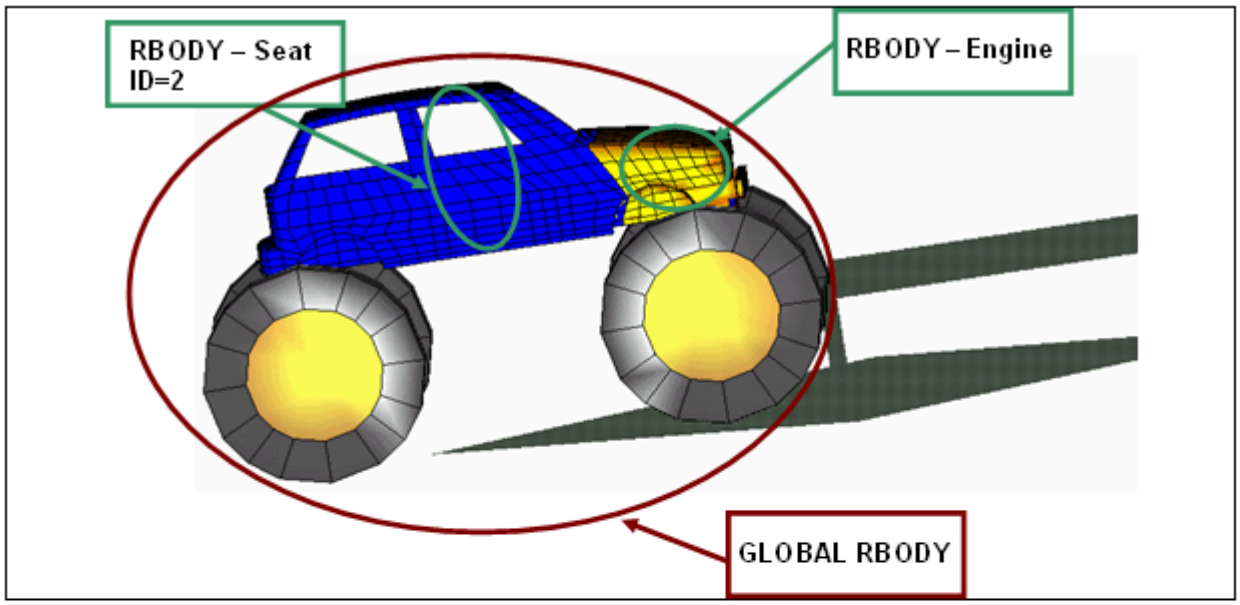
To activate two rigid bodies with the main node 1 and 2, add the following option in `Rootname_0001.rad`:

/RBODY/ON	1 2 All elements included in both rigid body are deactivated. To deactivate one rigid body whose main node ID is 3, add:
/RBODY/OFF	3 All elements are reactivated, strains and stresses are those from previous deactivation time.

Rigid Body Motions

One of the main uses of rigid body activation and deactivation concerns the roll-over motions, as in roll-over simulations. During the free flight of the car, the elements deformation can be neglected. A large part of CPU time can be saved if the whole structure is replaced with a temporary rigid body during the fly. Before the impact on the ground, this rigid body is deactivated and eventually activated again after rebound.

Engine



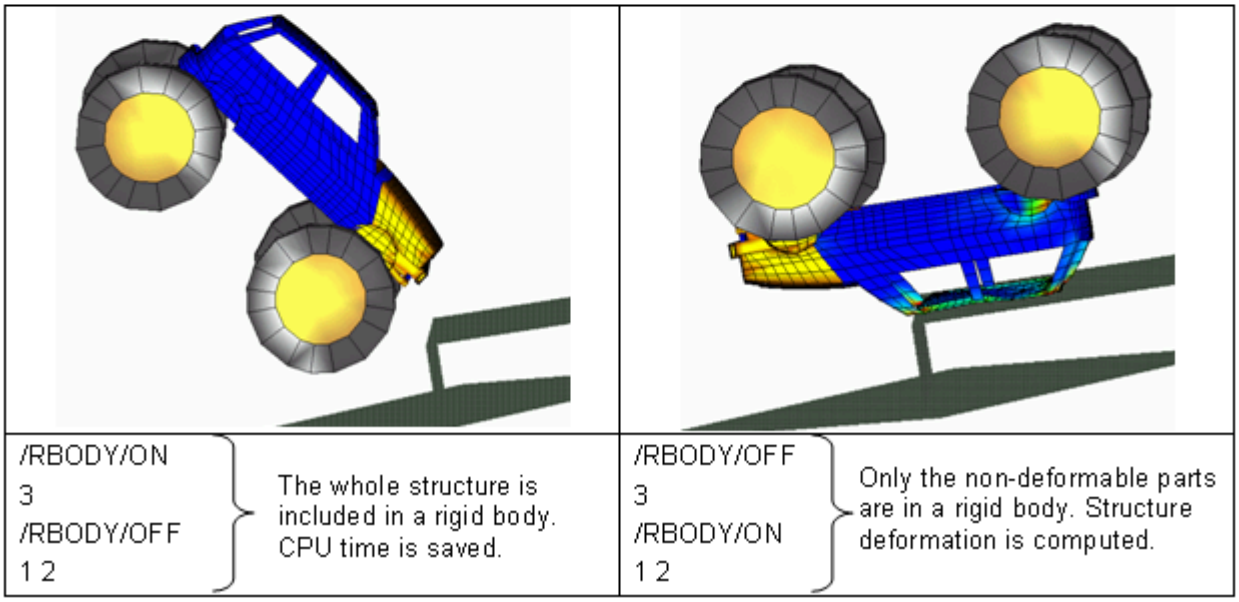
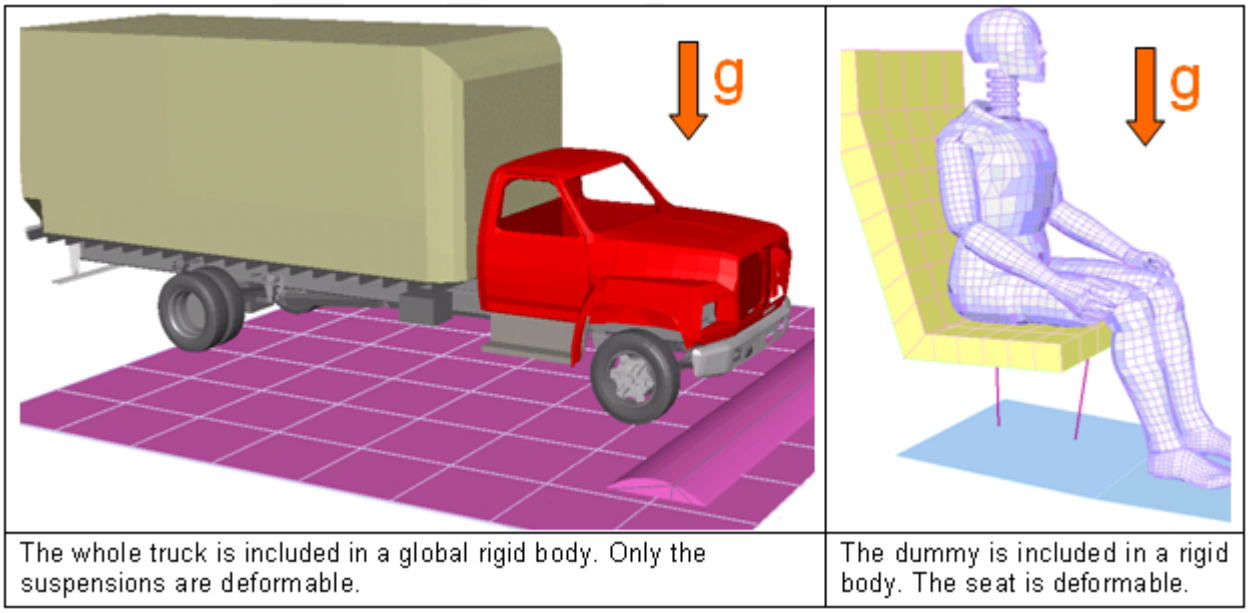


Figure 300: Activation - Deactivation of Rigid Bodies in a Roll-Over Example

Initial Static Equilibrium

Another use of rigid body activation and deactivation is for initial static equilibrium, where only gravity is applied. In a crash analysis, it may be of interest to set the whole car in equilibrium over the suspension, and the dummy on the seat. For an explicit analysis, it takes a long time to reach a static equilibrium; consequently using a large rigid body allows a solution to be obtained faster. However, in this case the deformation in a car or in a dummy under gravity is neglected and validity of this assumption needs to be carefully watched.



The whole truck is included in a global rigid body. Only the suspensions are deformable.	The dummy is included in a rigid body. The seat is deformable.

Figure 301: Static Equilibrium Examples

Merge Rigid Bodies

The `/MERGE/RBODY` option can be used to merge rigid bodies together or add secondary nodes to an existing rigid body. The secondary entities defined become secondary to a main rigid body. The secondary entities can be rigid bodies, single nodes, or node sets. Multiple secondary entities can be merged into one main rigid body by defining multiple lines in `/MERGE/RBODY`.

Some use cases include merging two different assemblies defined in two separate include files by defining the merge in the main input file. This can also be useful when separate parts defined as rigid need to be merged into one rigid body to model a complex component like an engine.

The main rigid body defined in `/MERGE/RBODY` can be defined as a secondary rigid body in another `/MERGE/RBODY`. However, complex hierarchies should be avoided as they can become difficult to debug. The secondary entities can only be defined in one `/MERGE/RBODY` and not in any `/RBODY`; otherwise, incompatible kinematic conditions occur.

Before merging, the inertia, mass, and center of gravity of each the secondary and main rigid body is calculated based on their `/RBODY` properties. Next, the secondary entities are merged to the main rigid body and new rigid body properties are calculated based on the `/MERGE/RBODY` *Iflag* option.

Rigid Wall (/RWALL)

A rigid wall is a nodal constraint applied to a set of secondary nodes in order to avoid the node penetration to the wall. If contact is detected, then the secondary node acceleration and velocity are modified.

There is no gap to determine whether or not a secondary node is in contact. Contact occurs only when the secondary node impacts the rigid wall surface. The secondary node's tangential velocity can also be modified depending on the flag *Slide*. The default value (=0) enables the model pure sliding during

contact. If set to 1, no sliding is allowed, the secondary nodes are "tied" in the tangential direction. If set to 2, friction based on Coulomb model is enabled.

Four types of rigid walls are available in Radioss:

- Infinite rigid wall
- Infinite cylindrical wall
- Spherical wall
- Finite planar wall

The rigid walls can be *fixed* or *moving*. A fixed wall is a pure kinematic condition on all impacted nodes; whereas a moving wall is similar to a main secondary option. A main node defines the wall position at each time step and imposes velocity on impacted secondary nodes. Impacted secondary node forces are applied to main node. The secondary node forces are computed with momentum conservation. The mass of a secondary node is not transmitted to a main node, assuming a large rigid wall mass compared to impacted secondary nodes mass.

Infinite Rigid Wall

An infinite rigid wall is a planar surface which extends to infinity. It is defined by two points representing the rigid wall normal (Figure 302).

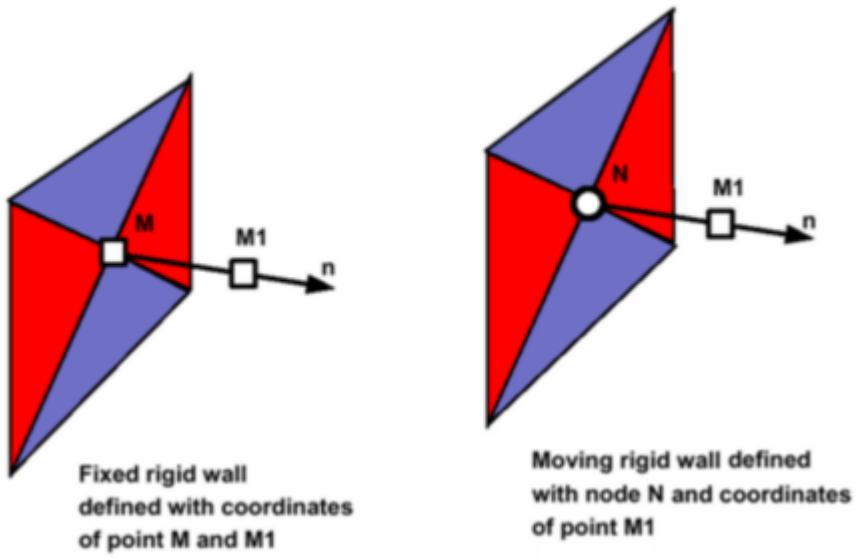


Figure 302: Infinite Rigid Wall

Infinite Cylindrical Wall

An infinite cylindrical wall is a cylinder which extends to infinity. It is defined with two points (or one point and one node) and a diameter.

> **Note:** contact is only possible from outside the cylindrical wall.

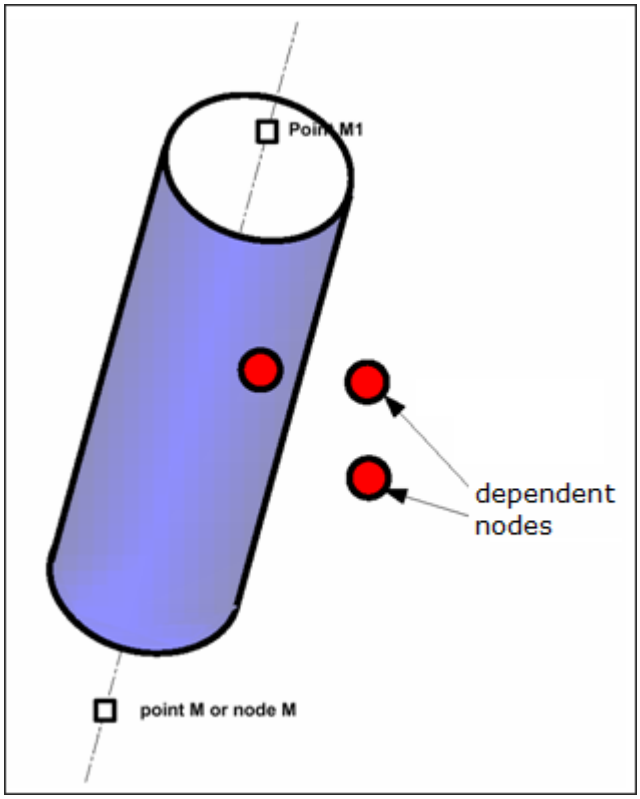


Figure 303: Infinite Cylindrical Wall

Spherical Wall

A spherical wall is a sphere defined by a point *M* (or a node *N*) and a diameter.

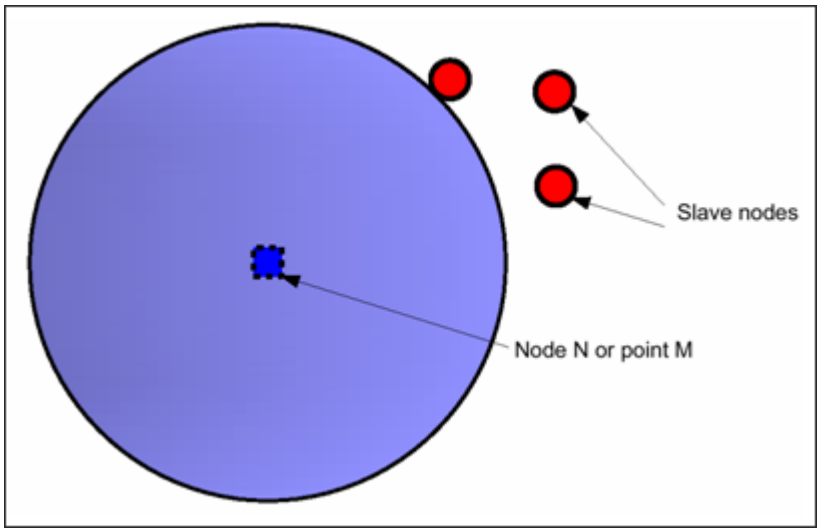


Figure 304: Spherical Wall

Finite Planar Wall

A finite planar wall is a parallelogram defined by three points M, M1 and M2. In case of a moving wall, M will be considered as the main node.

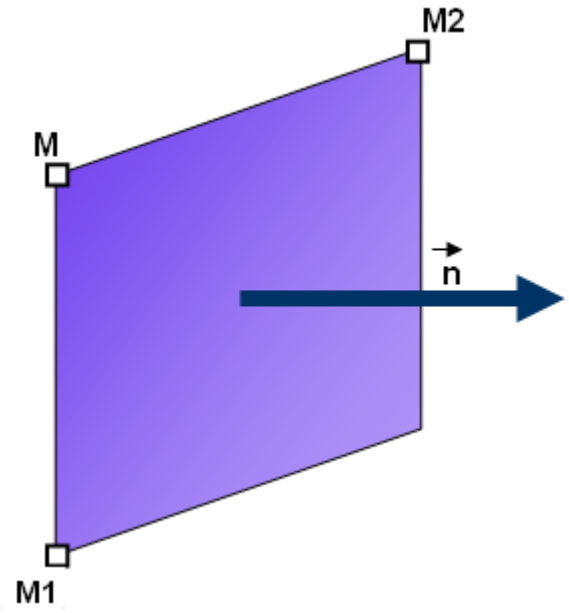


Figure 305: Finite Planar Rigid Wall

Comments

1. During the simulation of a moving wall is following the main node *N* but the wall orientation remains constant and parallel to initial normal. A moving rigid wall does not respect the moment equilibrium, only force equilibrium is applied. An external moment is; therefore, applied from laboratory to the wall.
2. If the secondary nodes are defined with a distance from the wall only, the nodes with positive or zero distance are considered (that is the nodes behind the infinite wall or in the cylinder wall are not considered as secondary).
3. A node initially impacted into a wall cannot rebound; except if the rebound velocity is high enough to go out from the wall in only one cycle.
4. If a random noise is added to the node coordinates, the initial position of secondary nodes are modified and for a moving wall, the wall location is changed. Therefore, it is possible that some secondary nodes with zero or near zero distance from the wall move inside the wall. If secondary nodes are defined with a distance, these nodes are not secondary nodes. If these nodes are explicit secondary nodes, they will stay inside the wall without possible rebound.
5. With random noise (`/RANDOM`) the orientation of a moving wall is also affected. Location of main node *N* is moved with random value and the normal defined with node *N* and point M1 is changed. This is especially critical if point M1 is close to point *N*.

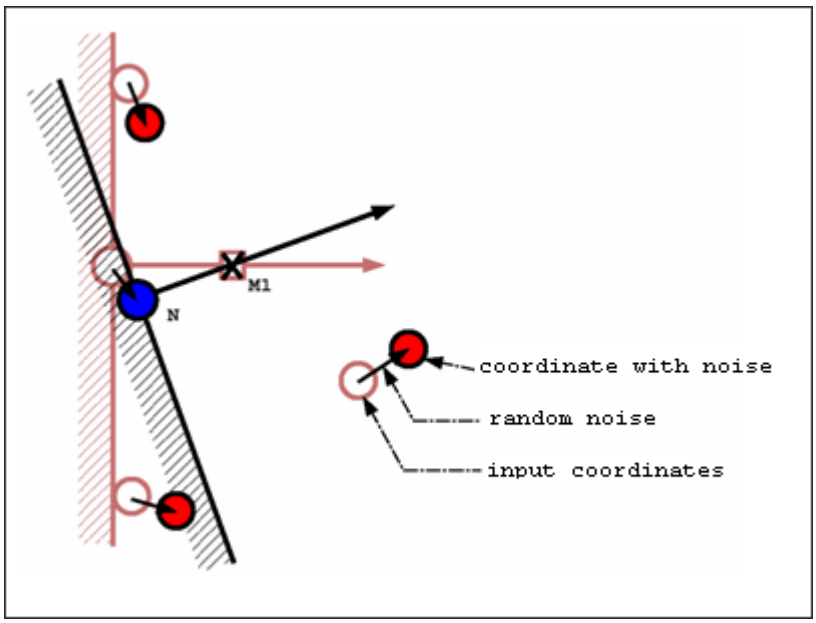


Figure 306: Change in Wall Orientation, Due to Random Noise

Tied Contact (/INTER/TYPE2)

Interface TYPE2, also called *tied interface* is a nodal constraint to rigidly connect a set of secondary nodes to a main surface. The secondary nodes forces and moments are transferred to the main nodes, and then secondary nodes are positioned kinematically according to the motion of the main nodes.

This interface ensures a full force and moment equilibrium.

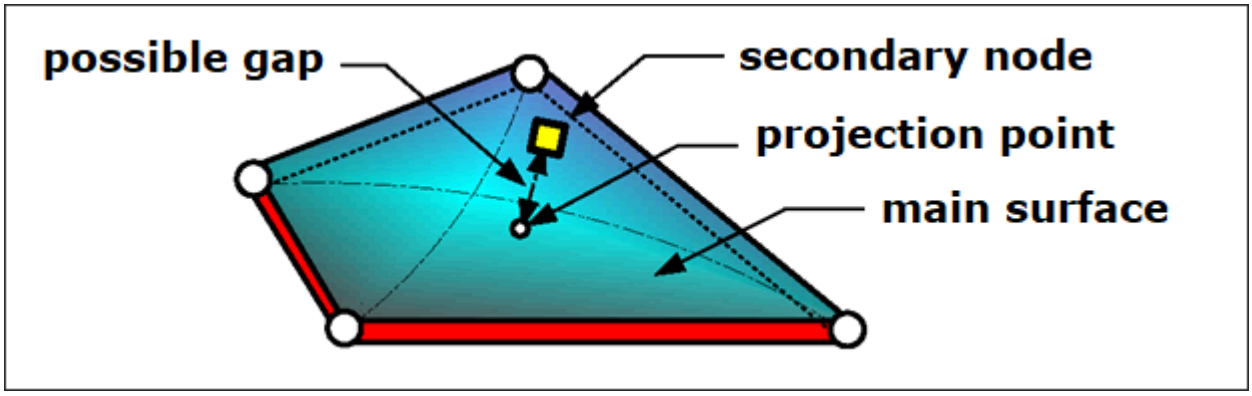


Figure 307: Interface TYPE2 - Tied

There are four formulations are available to describe this connection.

- Default spotweld formulation
- Optimized spotweld formulation
- Formulation with failure
- Penalty formulation

Default Spotweld Formulation

$Spot_{flag}$=0

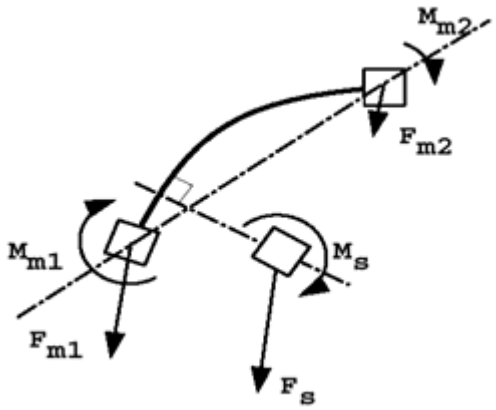


Figure 308:

When the flag is set to 0, the spotflag formulation is a default formulation:

- Based on element shape functions
- Generate hourglass with under-integrated elements
- Give a connection stiffness function of secondary node localization
- Recommended with fully-integrated shells (main)
- Recommended for connecting brick secondary nodes to brick main segments (mesh transition without rotational freedom)

Optimized Spotweld Formulation

$Spot_{flag}$=1

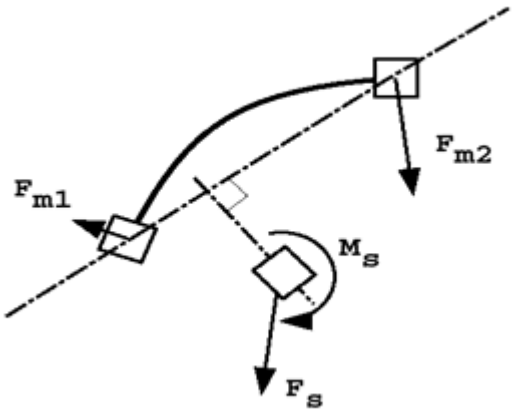


Figure 309:

When the flag is set to 1, the spotflag formulation is an optimized formulation:

- Based on element mean rigid motion
- No hourglass problem
- Constant connection stiffness
- Recommended with under-integrated shells (main)
- Recommended for connecting beam, spring, and shell secondary nodes to brick main segments

Formulation with Failure

Spot$_{flag}$=20, 21 and 22

Using these options, these two failure criteria can be defined:

Rupt= 0 (independent rupture parameters):
Failure when *Max_N_Dist* or *Max_T_Dist* are reached (default)

Rupt= 1 (coupled rupture parameters):
Failure when $\sqrt{\left(\frac{N\_Dist}{Max\_N\_Dist}\right)^2+\left(\frac{T\_Dist}{Max\_T\_Dist}\right)^2} > 1$

During the computation, a normal stress, shear stress, normal displacement and tangential displacement are computed and compared to the maximum values defined in the interface. As soon as the maximum criteria have been reached, the normal stress and shear stress will be set to 0.

Penalty Formulation

Spot$_{flag}$=25

The main goal for interface TYPE2 using penalty method is to tie secondary node to main segment without any kinematic constraints. Using the penalty method may avoid

```
"INCOMPATIBLE KINEMATIC CONDITIONS"
```

.

Cylindrical Joint (/CYL_JOINT)

A cylindrical joint is like a rigid body, except that one specific direction is defined with the first two secondary nodes. All nodes are free to move along this direction and to rotate around it.

A kinetic condition is applied on all secondary nodes, including the first two defining the privileged direction No main node is used.

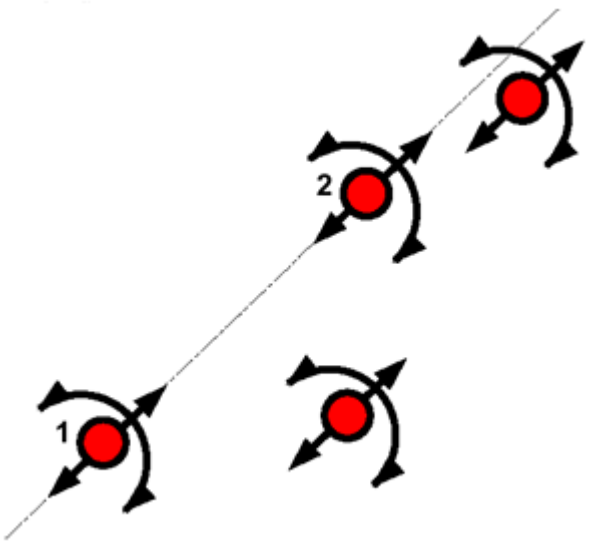


Figure 310: Cylindrical Joint

If all the secondary nodes are initially aligned, they will always remain aligned. As visualized in Figure 310, the rotation freedom is a local rotation for each node and not a global rotation around axis 1-2. Therefore, it is recommended to use a cylindrical joint with aligned nodes.

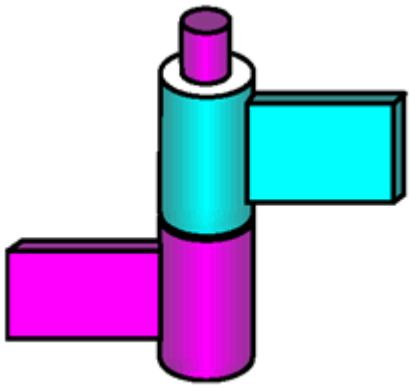


Figure 311: Door Hinge Example

Figure 311 shows how a hinge could be modeled using a cylindrical joint. The cylindrical joint is made of nodes 2, 5, 3 and 4. Note that in hinge modeling nodes must be aligned to get a realistic rotation, then beams or any other finite elements are used to connect nodes 1-2, 2-5, 4-3 and nodes 3-6. Finally, it is possible to link nodes 2-3 with a nonlinear elastic spring to improve the connection.

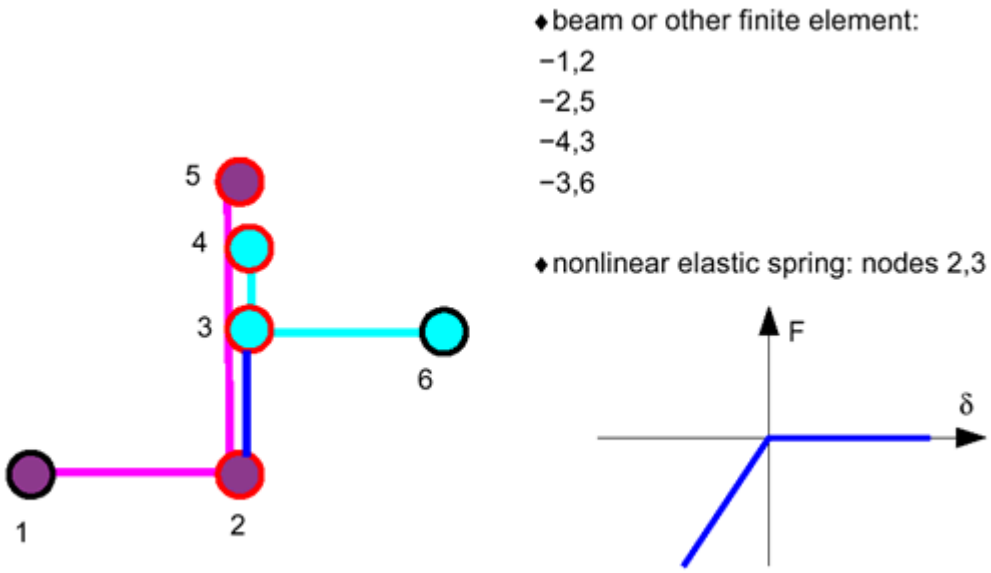


Figure 312: Hinge Modelization

Rigid Link (/RLINK)

The rigid link option imposes the same velocity on all secondary nodes for one or more directions. Directions are defined to a skew or a global frame, velocity is computed with momentum conservation.

No global moment equilibrium is respected. A rigid link is equivalent to an infinite stiff spring TYPE8.

> **Note:** A rigid link is a Radioss Engine input. A rigid link is alive only during the current restart run. For permanent rigid bodies, the input has to be repeated in each Engine input deck.

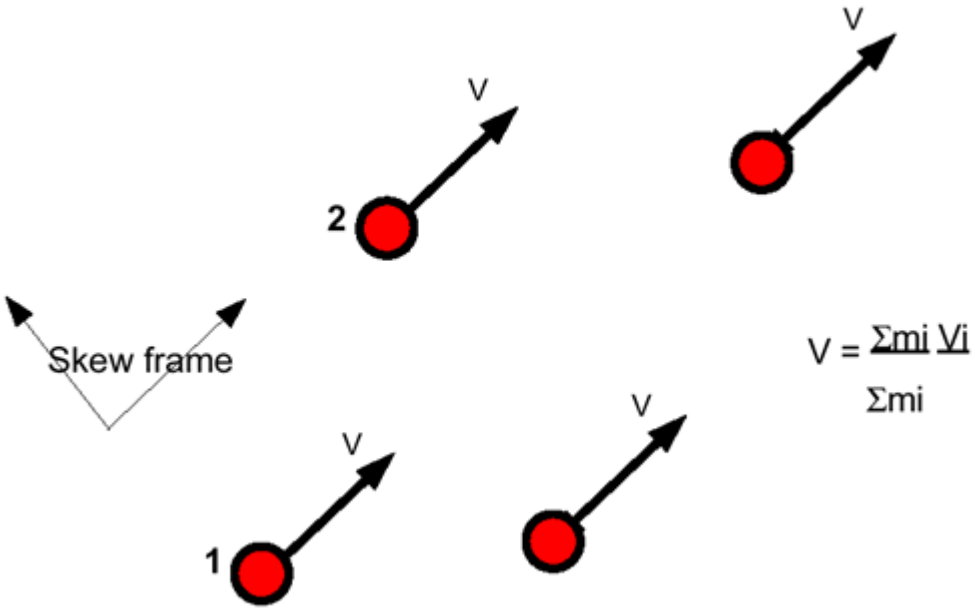


Figure 313: Rigid Link Model

Multi-Point Constraints (/MPC)

Gear type joints are more complex than other kinematic joints. They use the Lagrange Multiplier method and are compatible with all other Lagrange Multiplier kinematic conditions and incompatible with all classical kinematic conditions.

Three examples of these joints are explained:

- Rotational gear type joint
- Rack and pinion joint
- Differential gear joint

Mass and inertia may be added to all nodes. MPC joints impose relations between nodes velocities. The MPC cannot be applied to the translational degrees of freedom of a node without mass or the rotational degrees of freedom of a node without inertia.

Rotational Gear Type Joint

This joint is used to impose a rotational velocity relation between input and output node as:

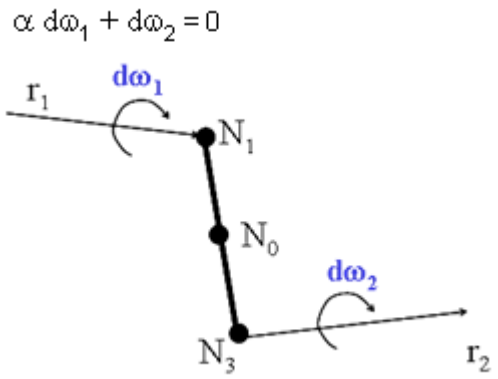


Figure 314: Rotational Type Joint

Translational velocities of gear joint nodes are constrained by a rigid link relation. For the rotational degrees of freedom, a scale factor is imposed between velocities of nodes N_1 and N_2, measured in their local coordinates. The corresponding constraint equations are:

$$\alpha(\Delta\omega_1 \cdot r_1) + (\Delta\omega_2 \cdot r_2) = 0$$

$$\Delta\omega_1 \cdot s_1 = 0, \quad \Delta\omega_2 \cdot s_2 = 0 \Delta\omega_1 \cdot t_1 = 0, \quad \Delta\omega_2 \cdot t_2 = 0$$

Where, $\Delta\omega_1 = \omega_1 - \omega_0$, $\Delta\omega_2 = \omega_2 - \omega_0$ are relative rotational velocities of nodes N_1 and N_2 in respect of the rigid body rotational velocity.

Rack and Pinion Joint

This joint allows the rotational velocity of node to be transformed to a translational velocity as:

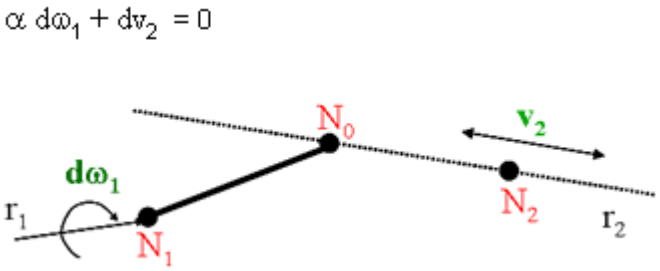


Figure 315: Rack and Pinion Type Joint

The constraint equations for these velocities are:

$$\Delta\omega_1 = \omega_1 - \omega_0, \quad \Delta\omega_2 = \omega_2 - \omega_0 \tag{207}$$

$$\alpha(\Delta\omega_1 \cdot s_1) + (\Delta V_2 \cdot s_2) = 0 \tag{208}$$

$$\alpha(\Delta\omega_1 \cdot t_1) + (\Delta V_2 \cdot t_2) = 0 \tag{209}$$

Differential Gear Joint

This joint is used to impose rotational velocity relations between an input node and two output nodes as:

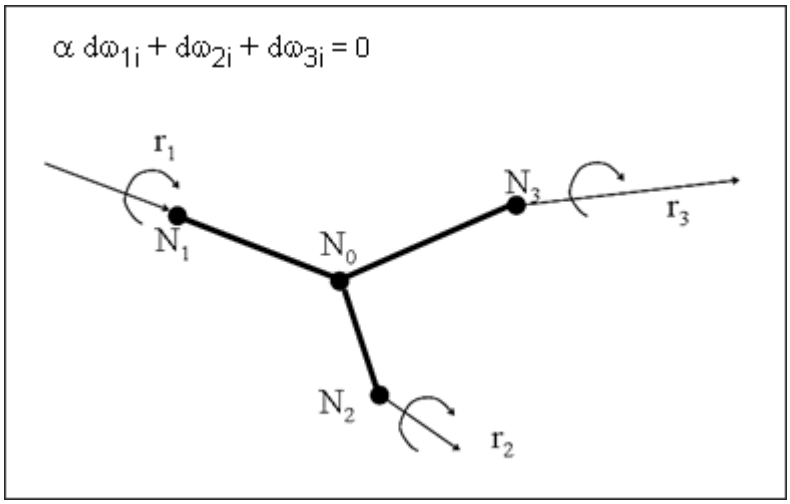


Figure 316: Differential Joint Type

The rotational velocities of a differential gear joint are constrained by the relations:

$$\alpha(\Delta\omega_1 \cdot r_1) + (\Delta\omega_2 \cdot r_2) + (\Delta\omega_3 \cdot r_3) = 0 \quad (210)$$

$$\alpha(\Delta\omega_1 \cdot s_1) + (\Delta\omega_2 \cdot s_2) + (\Delta\omega_3 \cdot s_3) = 0 \quad (211)$$

$$\alpha(\Delta\omega_1 \cdot t_1) + (\Delta\omega_2 \cdot t_2) + (\Delta\omega_3 \cdot t_3) = 0 \quad (212)$$

Methods to Apply Kinematic Conditions

Numerical methods available in Radioss to apply the kinematic conditions are:

Penalty Method (Rigid Walls)
: The node displacements are constrained by the application of penalty forces.

Main Secondary Kinematic Condition (Interface TYPE2 and Rigid Body)
: All secondary nodes lose their independence. The degrees of freedom of the secondary nodes are expressed as a function of a main node or element.

Local Kinematic Condition (Rigid Link and Cylindrical Joint)
: Constraint equations are applied to the nodes to satisfy local kinematic conditions. No main node is used.

Lagrange Multiplier (Radioss V44)
: This method consists in the nonlinear resolution of the equilibrium equations under constraints. This method is optional for some classical kinematic conditions as: interfaces, rigid wall, rigid body and boundary conditions. It is a unique method that is available for some new kinematic conditions, such as: gear joints and interface TYPE16 and TYPE17.

This method is generally used when other methods are not applicable. For example, when the kinematic condition must be satisfied exactly while a node contacts a surface. On the other hand, as the method is based in the nonlinear resolution of equations by Newton-Raphson method, its efficiency may be poor for a general case.

Incompatible Kinematic Conditions

As nodal constraints are based on kinematic conditions applied on nodal DOF, therefore it is not allowed to apply two nodal constraints to the same set of nodes, unless the induced kinematic conditions are perfectly orthogonal (for example: boundary condition in the X-direction and rigid link in the Y-direction).

Radioss Starter will issue the following warning each time two nodal constraints are applied to the same set of nodes.

```
WARNING ID: 147
*** WARNING: INCOMPATIBLE KINEMATIC CONDITIONS
2 KINEMATIC CONDITIONS ON NODE xxxxxx,
IN DIRECTION TRANSLATION X, FOR:
- Nodal constraint 1 (e.g. BOUNDARY CONDITION)
- Nodal constraint 2 (e.g. RIGID WALL)
```

It is of very important to take all warnings about incompatible kinematic conditions into account. True incompatible kinematic conditions (that is, nodes belonging to several rigid bodies) can generate energy and local instability. In such a case, accuracy of the results will be seriously decreased.

Radioss Starter does not check if the kinematic conditions are really incompatible. If they are strictly orthogonal, or if they are not applied simultaneously, just ignore the warning. Figure 317 illustrates two cases: in the first case, a node is secondary on a rigid wall and has a boundary condition in a non-orthogonal direction. If the rigid wall is fixed, there are no possible incompatible conditions (the node cannot impact on the wall). If the wall is moving, it is impossible after impact to respect both conditions. Therefore, the boundary condition is not applied and the reaction forces on the wall are incorrect. In the second case, a node is defined as a secondary for two parallel walls. If the two rigid walls are fixed, there are no possible incompatible conditions, as the node cannot impact the two walls at the same time. If one wall is moving, that results no problem, as long the moving wall is not crossing the fixed wall.

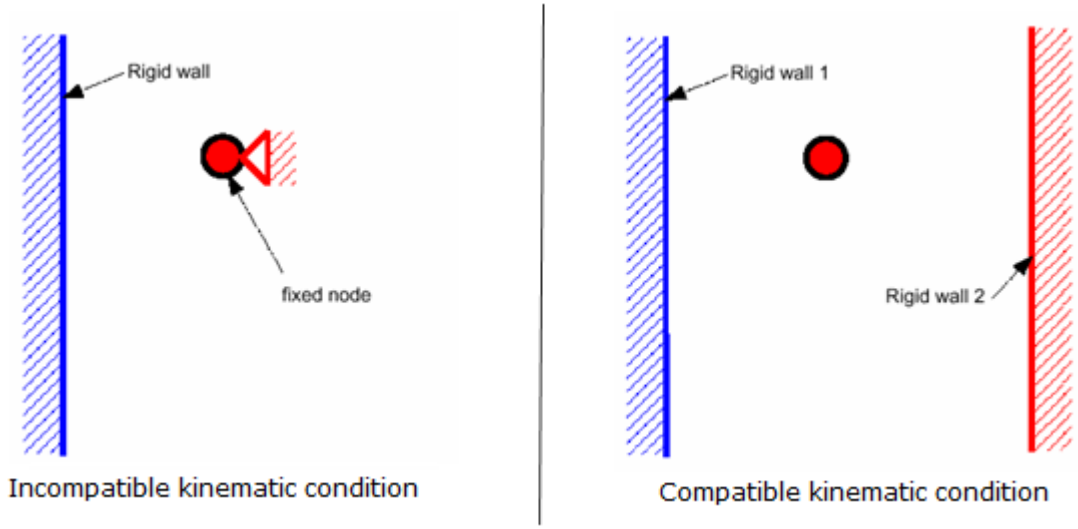


Figure 317: Radioss Warning for Kinematic Conditions

On a multi-processor Radioss version, a run executed twice can give different results if some kinematic conditions are incompatible. This is still true if option `/PARITH/ON` is being used. For example, if a secondary node impacting on two non-orthogonal rigid walls, as shown in Figure 318, the results obtained on a multi-processor can be arbitrary. If rigid wall 1 is computed before rigid wall 2, velocity V0 is replaced with velocity V12. If rigid wall 2 is computed before rigid wall 1, velocity V0 becomes V21. On multi-processor computers, the order in which rigid walls and other kinematic conditions are applied is arbitrary and can be changed from one cycle to the next and from one run to another.

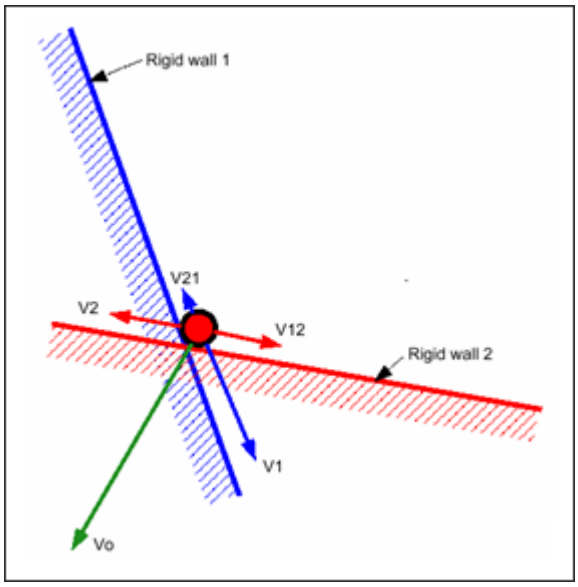


Figure 318: Arbitrary Results with Incompatible Kinematic Conditions

The Langrange Multiplier method allows applying several nodal constraints to the same set of nodes as it resolves a global system of equations with all Langrange Multiplier constrains. However, it is not allowed to mix both methods for the same set of nodes. Nevertheless, both methods can be used successfully in a model, as long as they are applied to different nodes.

Interfaces

Several interfaces are available in Radioss, this section deals with contact interfaces only. Each interface is distinguished with a type number.

Interface TYPE2 is a kinematic condition used to connect two Lagrange meshes and is not detailed in this section (refer to Kinematic Constraints). A brief overview of the contact interfaces are shown in Table 21.

Each of these interfaces was developed for a specific application field, but this field is not the only choice criteria. Some limitations of the different algorithms used in each interface can also determine your choice.

The algorithm limitations mainly concern the search of the impacted segment. This search can be done directly (interface TYPE7, TYPE10 and TYPE11) or via the search of the nearest node (other interface types). The nearest node search is done with a cheap, but limited algorithm (interface TYPE3, TYPE5 and TYPE6).

Interface TYPE3, TYPE5 and TYPE6 also have some limitations concerning the choice between the segments connected to the nearest node. Normal orientation is one limitation for these interfaces.

Interface TYPE23 defines a contact interface for airbag fabrics. This is a soft penalty contact which can deal with penetrations and intersections often coming in the folded airbag mesh.

Interface TYPE24 use constant penalty stiffness and therefore the time step is not affected.

Heat friction could be computed using interface TYPE7 and TYPE21.

Table 21: Interfaces in Radioss

Type	Description	Application	Contact Treatment
1 / 9	ALE/LAG with sliding	Fluid-structure interaction	Main-Secondary
2	Tied interface	Change of mesh density (solid)	Main-Secondary or LM
3 & 5	Contact impact between two parts	Use of TYPE7 is recommended	Penalty
6	Contact impact between two rigid bodies	User-defined contacts	Penalty
7	General purpose contact impact between two parts	Solid contact impact at all speeds	Penalty or LM
8	Drawbead contact	Stamping applications	Penalty
10	Like TYPE7, but tied contact	Special purpose interface	Penalty
11	Impact between two lines	For beams, bars or springs	Penalty
12	Fluid / fluid contact	Fluid to fluid contact	Penalty

Type	Description	Application	Contact Treatment
16 & 17	Contact between nodes to quadratic shape solids and solid-shells or between quadratic shapes	Meshes with 8-node or 16-node thick-shell or 20 bricks	LM
18	CEL Lagrange / Euler interface	Fluid-structure interactions	Penalty
19	Combination of interface TYPE7 and TYPE11	Solid contact impact at all speeds and For beams, bars or springs	Penalty
21	Specific interface between a non-deformable main surface and a secondary surface	For stamping	Main-Secondary
23	Soft penalty contact	For airbag	Soft penalty
24	General contact interface, optional single surface or surface to surface or nodes to surface contacts	This contact interface can replace interface TYPE3, 5, 7	Penalty stiffness is constant and; therefore, the time step is not affected

Contact Treatment

There are two approaches that deal with contact:

- Penalty Method is the most commonly used in explicit codes and can be found in most of the Radioss interfaces
- Lagrange Multiplier Method (`/LAGMUL` and `/INTER/LAGMUL`) is used in special case studies

Penalty Method

Interfaces using the penalty method are based on mainr/secondary treatment. Contact can only occur between a set of secondary nodes and a set of main segments. Main segments are defined depending on the type of element they lie on. If it is a 3-node or a 4-node shell, the segment is the surface of the element. If it is a solid element, the segment is defined as a face. Finally, if it is a 2D solid element (quad), the segment is a side.

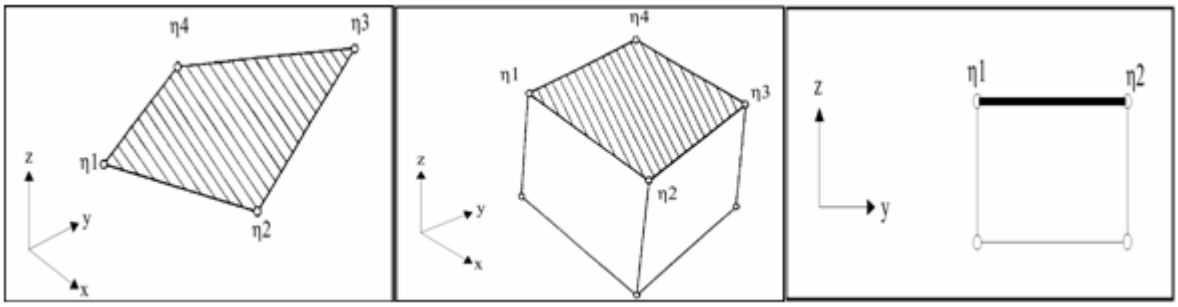


Figure 319: Segment Definition

A gap distance is defined to determine whether or not a node is in contact with a segment.

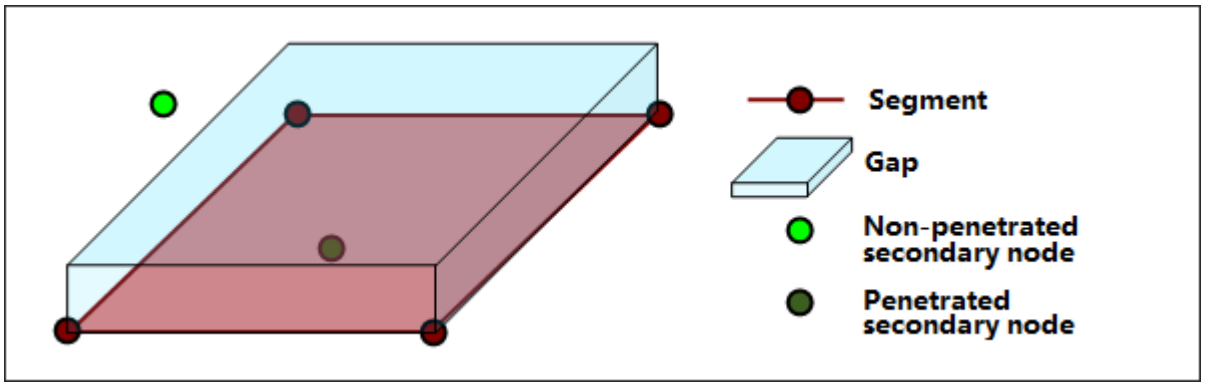


Figure 320: Gap and Penetration

As soon as a node penetrates into the gap, an elastic spring is added between the secondary node and the main segment. Therefore, a resistant force will tend to reject the secondary node.

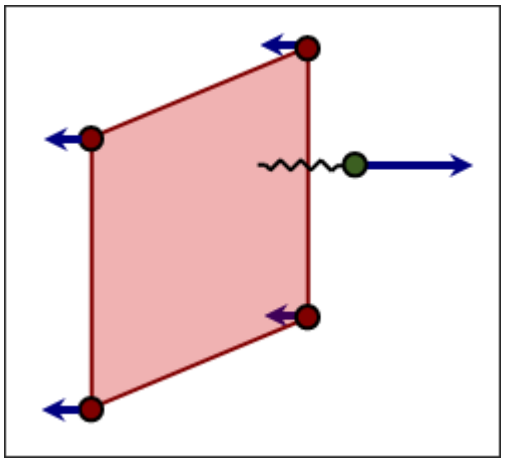


Figure 321: Reaction Force in Interface

The time step may be affected by the interface stiffness. During the penetration, as a spring is connected to the secondary node, the spring stiffness must be added to the overall stiffness acting on the node (stiffness from all elements connected to this node). The nodal time step must be reduced to account for the spring stiffness.

$$dt = \sqrt{\frac{2M_{node}}{K_{node} + K_{spring}}} \quad (213)$$

Contact ends when the penetrated node is completely pushed out of the gap. Therefore, the elastic spring and the reaction force are removed.

It is worthwhile to note that the contact interfaces with penalty method are fully compatible will all kinematic conditions (for example, rigid body, imposed velocity, etc.).

Lagrange Multiplier Method (/LAGMUL and /INTER/LAGMUL)

Contrary to the penalty method, the Lagrange Multiplier method is purely mathematical and does not require physical elements (springs) to model contact. A nonlinear system of equations is solved to account for contact conditions. Therefore, there is no time step collapse due to high interface stiffness, but it takes more CPU time to do one cycle as new equations need to be solved by a nonlinear solver. The method has the advantage to stop secondary nodes at the contact surface (the contact condition is precisely satisfied); however the friction cannot be computed.

Symmetric Interface (/INTER/TYPE3)

This interface is used to simulate symmetric impacts between two surfaces.

Both surfaces are defined through the use of oriented segments; therefore, contacts can only occur on one side. Each node on both surfaces is considered as a secondary node, and each surface is considered as a main segment.

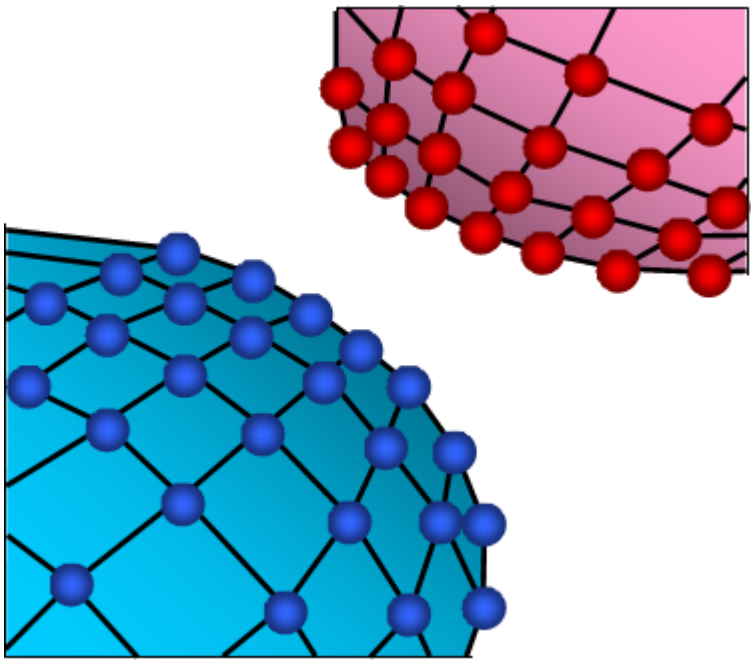


Figure 322: Interface TYPE3

Contrary to interface TYPE5, interface TYPE3 has two main surfaces; therefore, the contact algorithm is performed twice. The first pass solves the penetration of nodes from the first surface with respect of the

second surface. The second pass solves the penetration of nodes from the second surface with respect of the first surface. This leads to higher accuracy compared to interface TYPE5, but more CPU time is needed.

When contact is detected, an elastic spring is added and the spring stiffness is calculated using both surfaces stiffness. Based on material and geometric properties, stiffness is attributed to each surface, then the overall interface stiffness is computed:

$$K = s\frac{K_1 K_2}{K_1 + K_2} \tag{214}$$

The default value for the stiffness scale factor(s) is 0.2, for stability reason this value should not be modified. However, if the ratio of K_1 over K_2 is greater than 100 (or lower than 0.01), it is recommended to increase the stiffness scale factor to avoid too many penetrations. The spring stiffness is constant, allowing secondary nodes to pass through the mid-plane of the main segment.

This interface does not enable auto-contact simulation, as a node cannot belong to both surfaces.

Note: This interface is mainly used for shell plates initially in contact. If the two plates are modeled with a gap between the two meshes, it is better to use interface TYPE7; unless, if the two plates are modeled without a gap, interface TYPE3 is a good choice.

Non-Symmetric Interface (/INTER/TYPE5)

This interface is used to simulate impacts between a main surface and a list of secondary nodes.

The interface is oriented; therefore, contact can only occur on one side of the main segments. Thus, main segments normal must be oriented towards the secondary nodes.

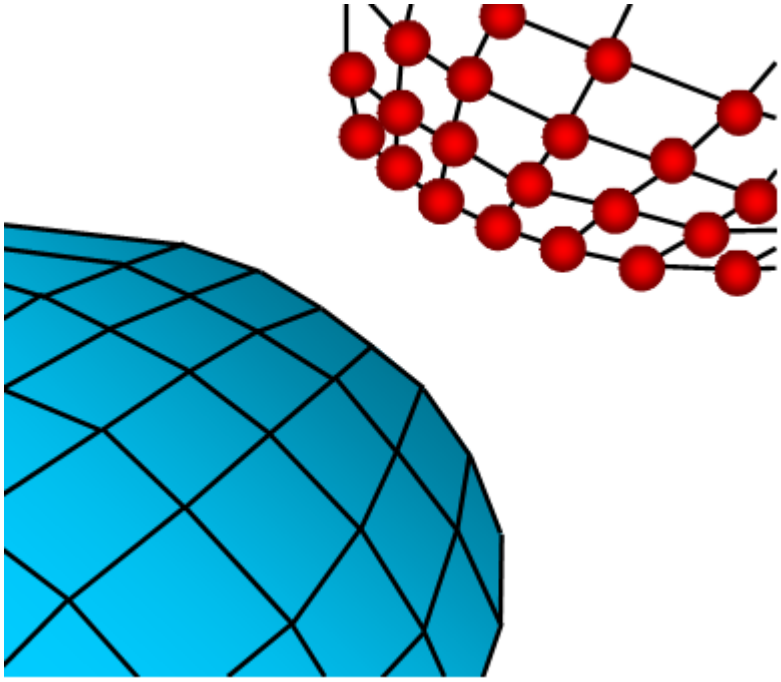


Figure 323: Interface TYPE5

The main surface mesh has to be regular with a good aspect ratio. It is not allowed to put the secondary node on the main surface. It works only with main segments connected to solid or shell elements. One use of this interface is to replace rigid wall. Replacing a rigid wall with interface TYPE5 will allow you to simulate rigid body impacts.

With interface TYPE5, a gap is used to determine contact between a node and a surface. This gap is user-defined and is located on the side normal to the surface. It is recommended to use a small or zero gap.

When contact is detected, an elastic spring is added and the spring stiffness is calculated using material and geometric properties of the main segment only. For stability reason, a stiffness scale factor of 0.2 is applied to the main side stiffness. It is strongly advised not to modify this factor unless the main side is very soft compared to the secondary side. If that is the case, it is recommended to use the ratio of the greater elastic modulus over the lower one as stiffness scale factor.

As mentioned previously, the stiffness depends on the geometric and material properties and also on the type of element. Figure 324 describes the way stiffness is computed according to the type of element the segment is attached to. In case a segment is common to a brick and a shell (for example: 3D part covered by a skin), the stiffness used is the one associated with the shell element.

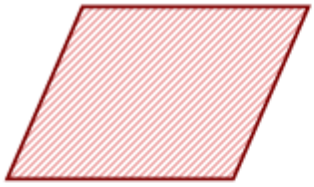
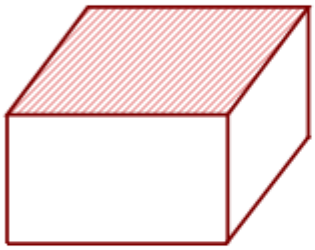
ELEMENT	STIFFNESS
	$K = s\frac{Et}{2}$ { E: Young's modulus t: shell thickness
	$K = s\frac{BA^2}{2V}$ { B: Bulk modulus A: Segment area V: Element volume
	$K = sE\sqrt{S}$ { E: Young's modulus S: Section area
	$K = \max(K_t)$ K_t : translational stiffness

Figure 324: Stiffness in Interface TYPE5

The main drawback of interface TYPE5 is that contact cannot occur on both sides of the main segment. For a problem with large rotations (usually the case in a crash analysis), contact is likely to occur on the wrong side of the surface; therefore, penetration will not be detected. Consequently, for a complex contact problem, a good understanding of the impact is necessary prior to the simulation to properly define normal surfaces.

Another important drawback is that a node is not allowed to be a secondary node and part of a main segment. Therefore, auto-contact cannot be simulated through the use of interface TYPE5.

Common Problems in Interfaces TYPE3 and TYPE5

Interfaces TYPE3 and TYPE5 have some common problems.

- Energy jumps
- Bad contact treatment
- Limited search algorithm

Energy Jumps

The gap used in interface TYPE5 (and TYPE3) is one sided and does not take into account the edges. This can lead to energy jumps in cases of a large gap (Figure 325).

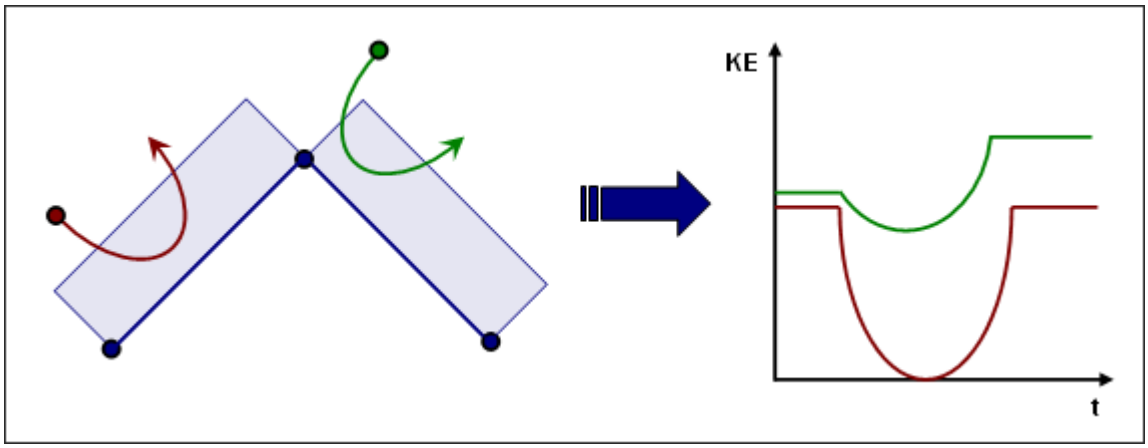


Figure 325: Jump in Energy

Bad Contact Treatment

Moreover, as the interface stiffness is constant, node penetration is allowed. This point can lead to a huge error, especially if sliding occurs during the penetration. Figure 326 illustrates how secondary nodes are allowed to go through the shell mid-surface because of bad contact treatment.

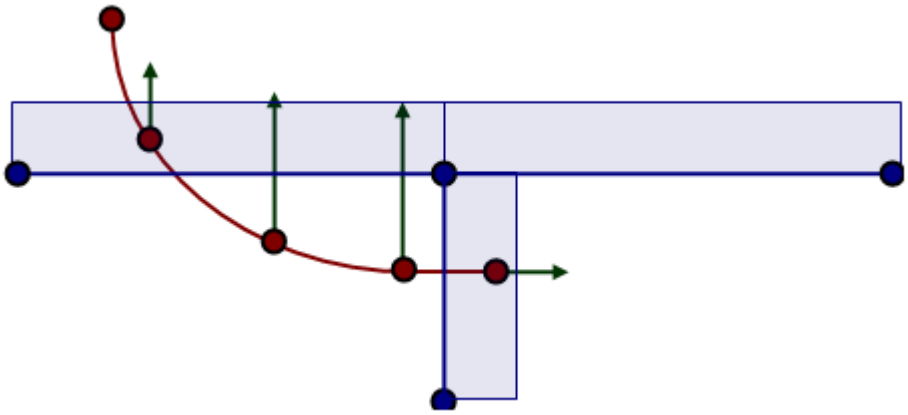


Figure 326: Bad Contact Treatment

Limited Search Algorithm

With interface TYPE5 (and TYPE3), the detection of the closest main node is limited to the segments topologically close to the previous one (topologically limited search algorithm). The first search is done by Radioss Starter to determine the closest initial main node, then the Engine determines the closest main node, taking into account only the segments topologically close to the previous one. This method is rather slow in CPU time, and it does not work very well, especially if high curvatures are involved (Figure 327).

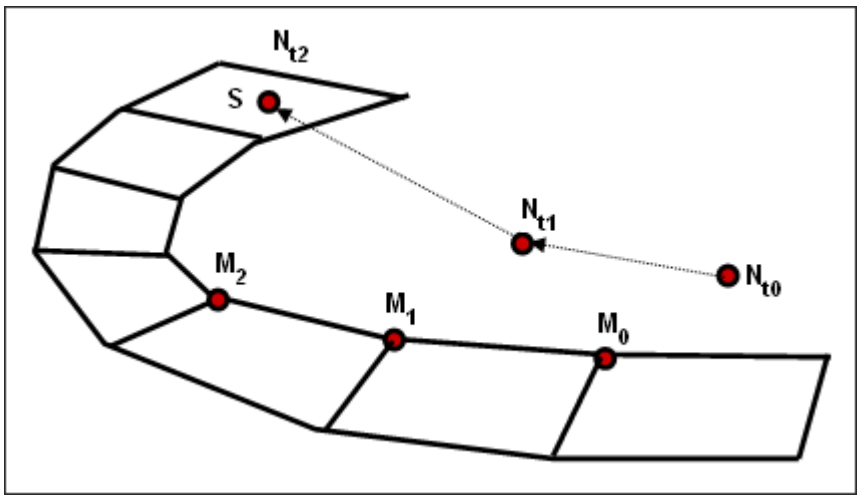


Figure 327: Bad Detection of the Closest Main Segment

Interface TYPE6 (/INTER/TYPE6)

Interface TYPE6 is used to simulate contact between two rigid bodies.

This interface is similar to interface TYPE3, except for the stiffness. The relationship between the contact force and the penetration is provided by a user-defined function. This interface is used particularly in vehicle occupant simulations, for example: knee bolsters. The main limitation of this interface is that surface 1 must be part of one rigid body, and also for surface 2. Moreover, both surfaces must be oriented so that normals are facing each others.

The stiffness used corresponds to the curve Force versus Penetration input by you. The instantaneous interface stiffness is the slope of the input curve at the given penetration; therefore, the time step can be affected since the interface stiffness is used to compute a stable time step:

$$\Delta t = 0.1\sqrt{\frac{M}{K}} \tag{215}$$

Where,

M	The min (Mass of the first rigid body and Mass of the second rigid body)
K	Slope of Force versus Penetration curve

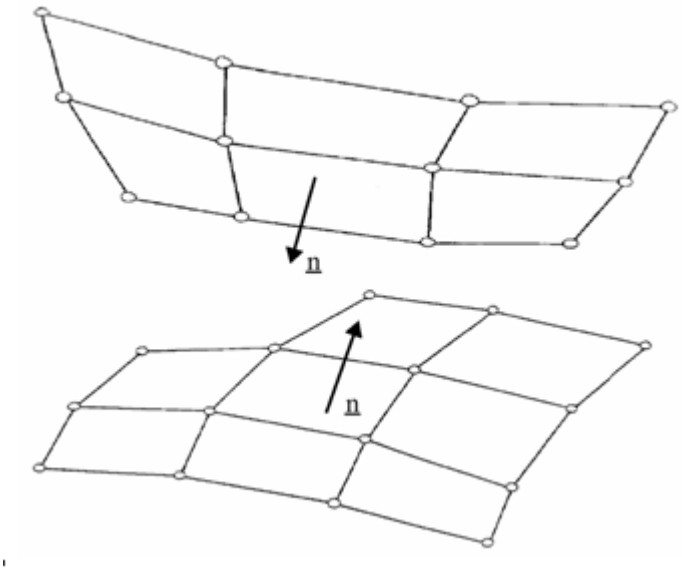


Figure 328: Interface TYPE6 - Normal Orientation

General Purpose Interface (/INTER/TYPE7)

Interface TYPE7 is a general purpose interface and can simulate all types of impact between a set of nodes and a main surface. Contrary to interface TYPE3 and TYPE5, interface TYPE7 is non-oriented and secondary nodes can belong to the main surface. Therefore, this interface can simulate self-impact, especially buckling during a high speed crash.

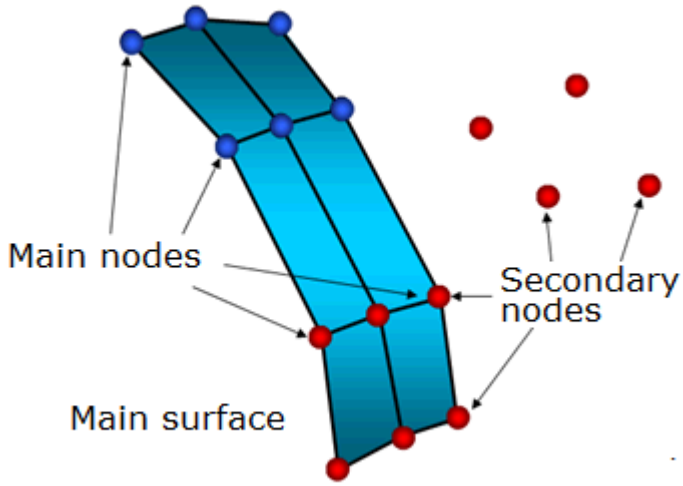


Figure 329: Interface TYPE7

Interface TYPE7 solves all problems and limitations encountered with interface TYPE3 and TYPE5. The search for the closest segment is done via a direct search algorithm; therefore, there are no search

limitations and all possible contacts are found. The energy jumps induced by a node impacting from the shell edges are removed by the use of a cylindrical gap around the edges.

Finally, the main advantage of interface TYPE7 is that the stiffness is not constant and increases with penetration preventing the node from going through the shell mid-surface. This solves many bad contact treatments (common when using either interface TYPE3 or TYPE5).

The gap used in interface TYPE7 is rather different than those of the previous interfaces. The gap used is on both sides of the shell mid-surface, and a cylindrical gap is added around the edges (Figure 330). The gap used is on both sides of the shell mid-surface, and a cylindrical gap is added around the edges.

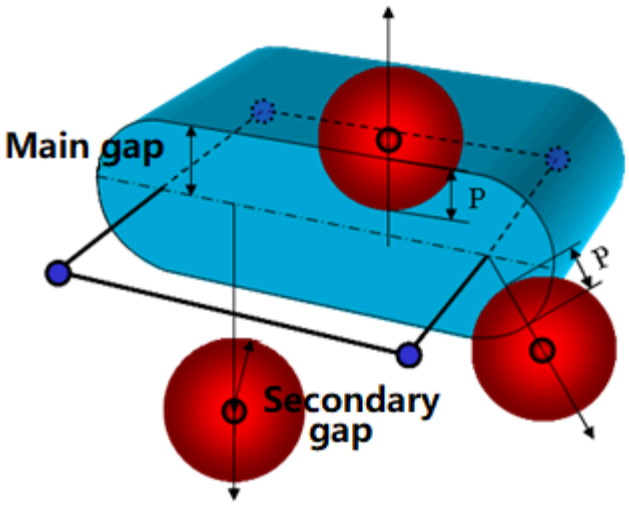


Figure 330: Gap in Interface TYPE7

The cylindrical gap allows getting rid of energy jumps, node impacting from the edges follow the same path during the penetration and the depenetration. Moreover, such a gap keeps the reaction force smooth during sliding between segments.

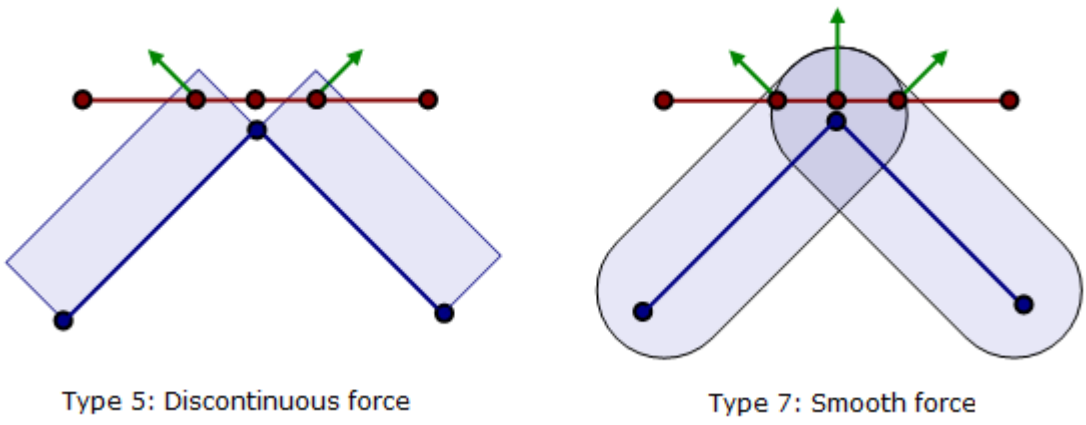


Figure 331: Sliding Between Segments

Contrary to interface TYPE3 and TYPE5, a variable gap in space is available. Depending on option I_{gap}, variable gap is computed for each impact as the sum of the main element gap (g_m) and the secondary node gap (g_s).

If I_{gap}=1, variable gap is computed as:

$$\max\left[Gap_{\min}, \left(g_s + g_m\right)\right] \tag{216}$$

If I_{gap}=2, variable gap is computed as:

$$\max\left\{Gap_{\min}, \min\left[Fscale_{gap} \cdot \left(g_s + g_m\right), Gap_{\max}\right]\right\} \tag{217}$$

If I_{gap}=3, variable gap is computed as:

$$\max\left\{Gap_{\min}, \min\left[Fscale_{gap} \cdot \left(g_s + g_m\right), \%mesh\_size \cdot \left(g_{s\_l} + g_{m\_l}\right), Gap_{\max}\right]\right\} \tag{218}$$

Table 22: Variable Gap Computation

Element	Main Element Gap (g_m)	Secondary Node Gap (g_s)
SHELL	$g_m = \frac{t}{2}$ t: thickness of the main segment	$g_s = \frac{t}{2}$ t: largest thickness of shell elements connected to the secondary node
BRICK	$g_m = 0$	$g_s = 0$
TRUSS and BEAM	Non-applicable	$g_s = \frac{1}{2}\sqrt{S}$ S: cross section

If a minimum gap for impact activation (Gap_{min}) is also used, the computed variable gap cannot be smaller than the minimum value. It is also possible to apply a scale factor to the gap and define a maximum gap value.

Initial Penetrations

In an FE model, initial penetrations are very common, are unavoidable and result from the discretization during the meshing process (Figure 332).

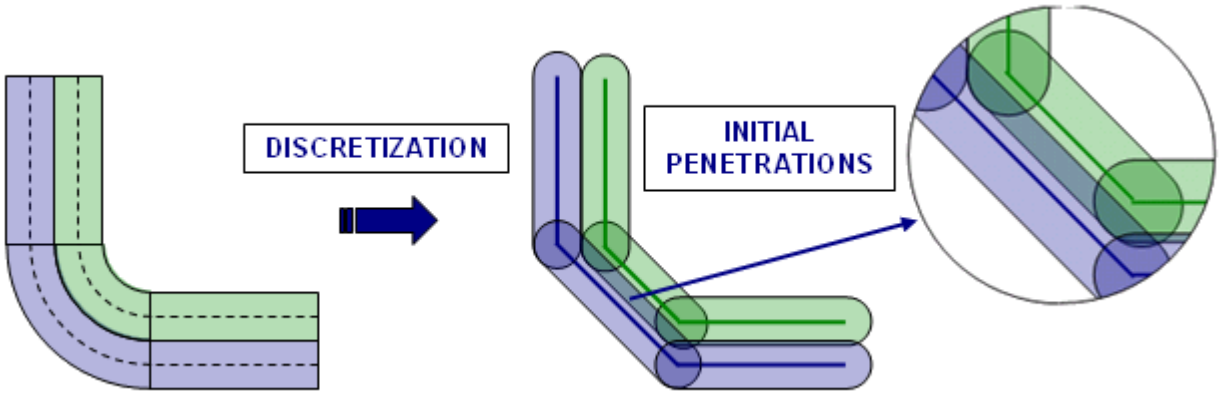


Figure 332: Initial Penetrations Due to Discretization

Inacti

Special treatment for initial penetrations can be accomplished through the use of the *Inacti* flag. It is possible to remove penetrated nodes from the interface or to remove the main segments relating to the penetrated nodes. Both treatments allow getting rid of initial penetrations very easily, but they may lead to poor results if the number of penetrated nodes is large.

Setting *Inacti* to 3 allows Radioss Starter to automatically modify the coordinates of penetrated nodes to avoid initial penetrations. Special care must be taken when doing so, since this operation can lead to initially constrained springs.

It is also possible to obtain a variable gap in time by setting *Inacti* to 5. The illustration Figure 333 explains how the effective gap is updated taking into account the previous penetrations.

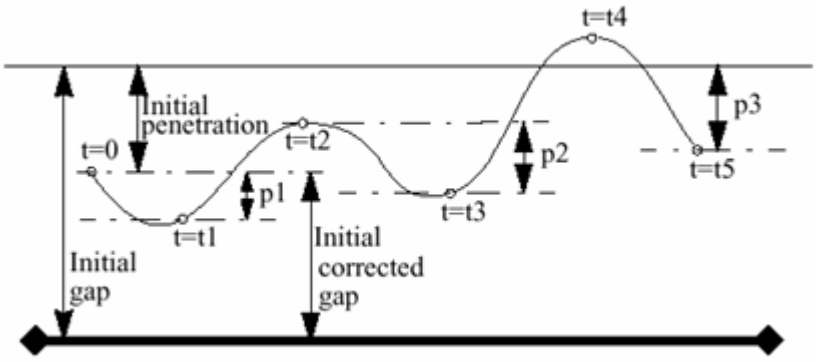


Figure 333: Variable Gap in Time

At t=0, if a node is initially penetrated, its gap is automatically corrected. Then this "*initial corrected gap*" will be increased every time the node is moving away from the main segment. This option is mainly used for unfolding the airbag, it allows a decent time step at the beginning of the unfolding, whereas nodes are all highly penetrated.

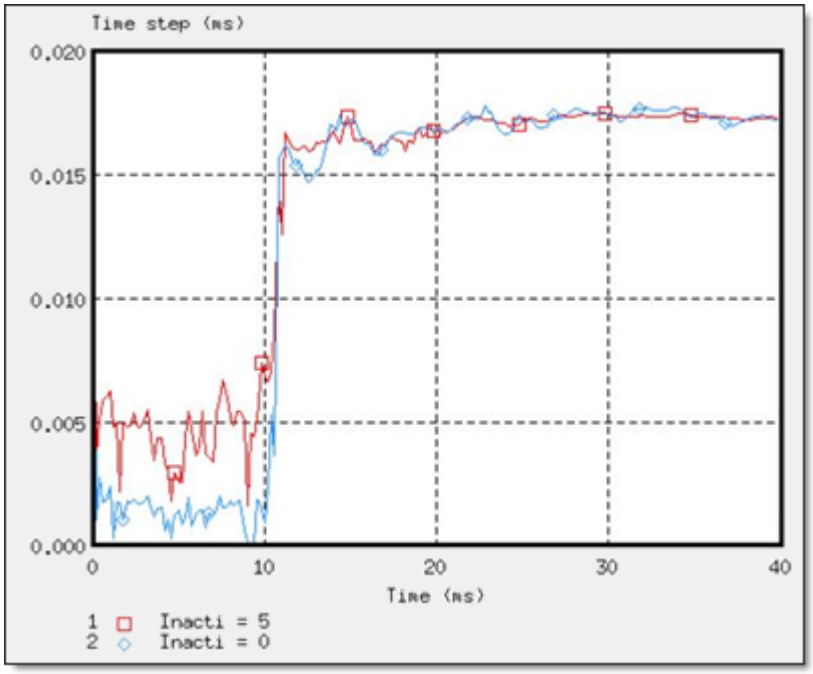


Figure 334: Time Step Using Inacti=5

To avoid high frequency effects *Inacti* = 6 is recommended instead of *Inacti* =5.

Fpenmax

Fpenmax (maximum fraction of initial penetration), is used to deal with big initial penetration. Node stiffness will be deactivated, if $Penetration \geq Fpenmax \cdot Gap$, whatever the value of *Inacti*.

Igap3 + %mesh_size

With I_{gap}= 3 and **`%mesh_size`**, the size of the mesh can be taken into account to avoid initial penetrations. In this case, the variable gap is computed as:

$$\max\left\{Gap_{\min}, \min\left[Fscale_{gap} \cdot \left(g_s + g_m\right), \%mesh\_size \cdot \left(g_{s\_l} + g_{m\_l}\right), Gap_{\max}\right]\right\} \quad (219)$$

Where,

g_{m_1}	Length of the smaller edge of element
g_{s_1}	Length of the smaller edge of elements connected to the secondary node

Irem_gap

The option *Irem_gap* is used to deactivate secondary nodes which close (Curvilinear $Distance < \sqrt{2} \cdot Gap$) to elements. This option is useful for self-impact contact when mesh size is very small.

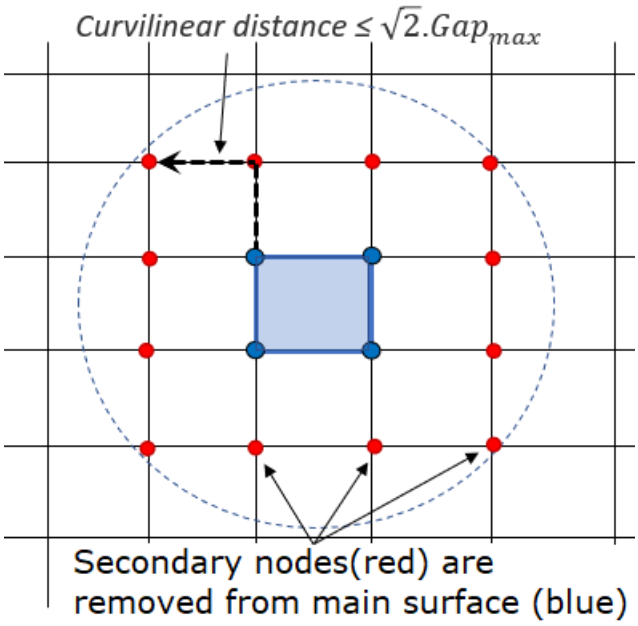


Figure 335: -1: Irem_gap Definition

Note: When dealing with initial penetrations, it is strongly advised to remove initial penetrations during the creation of the FE model, using pre-processing tools like HyperMesh and HyperCrash depenetrators.

Interface Stiffness

Like the other interface types, when using the penalty method, the interface has spring stiffness as a secondary node penetrates the gap; however, the reaction force is computed with much better approximation. The force variation versus penetration of a node is nonlinear, due to the increasing stiffness.

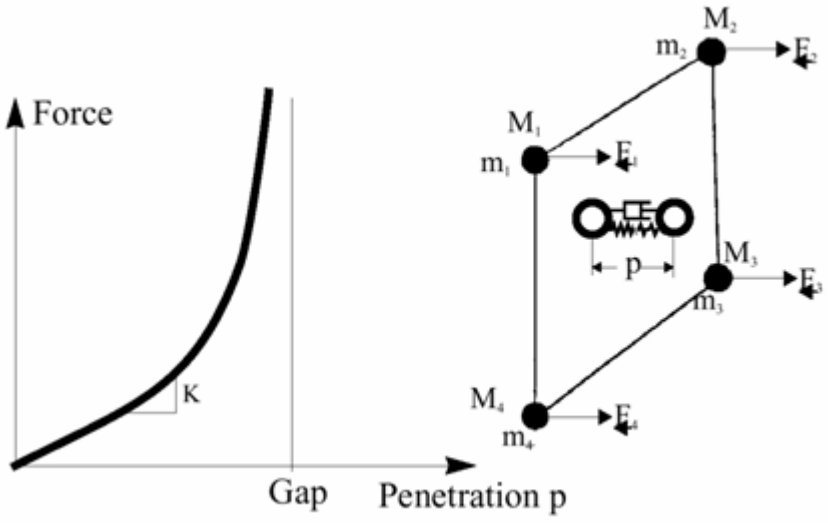


Figure 336: Interface Force Variation in Interface TYPE7

The interface stiffness (*K*) is not constant, it increases with the penetration. Moreover, there is a viscous damping acting on the rate of penetration. The contact force is then computed as:

$$F_n = K_S P + C\frac{dP}{dt}\begin{cases} K_S = K_0\frac{Gap}{Gap-P} \\ C = VIS_S\sqrt{2K_S M} \end{cases} \quad (220)$$

The instantaneous stiffness is then computed as:

$$K_t = \frac{\partial F}{\partial P} = K_0\frac{Gap^2}{(Gap-P)^2} \quad (221)$$

Nodal time step can be seriously affected if penetration is large. The stiffness, used to compute the nodal time step takes into account the interface stiffness.

There are two ways to decrease the interface stiffness:

- Increasing the gap
- Increasing the initial stiffness (through the use of the flag *Stfac*)

Both methods allow absorbing more energy by contact and smoothing the impact. Increasing the gap will allow nodes to slow down over a larger distance, therefore the penetration is reduced.

Comments

1. Even if an elementary time step is chosen for the simulation, a nodal time step is automatically computed if there is an interface TYPE7. The lowest time step is applied for the simulation.
2. Contrary to interface TYPE5, a *Stfac* smaller than 1.0 produces a large penetration at the first touch and results in high interface stiffness and reaction force. To avoid high penetration, a *Stfac* greater than or equal to 1.0 is recommended.

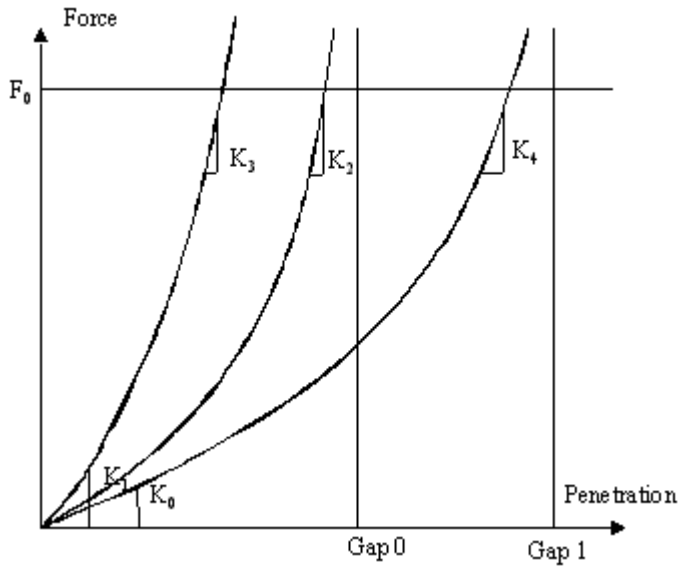


Figure 337: Force versus Penetration Curves

Although, increasing the initial stiffness leads to a smaller time step at the beginning of the penetration, it will increase the time step if penetration is large.

Friction

Several friction formulations are available within Radioss. The simplest one, which is also the most used, is the Coulomb friction law. This formulation provides accurate results in crash analysis and requires just one parameter (Coulomb friction coefficient, μ).

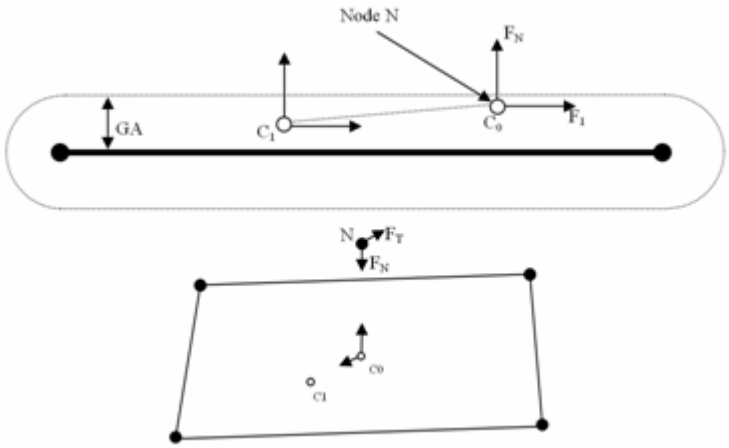


Figure 338: Normal and Tangential Forces Applied to a Node

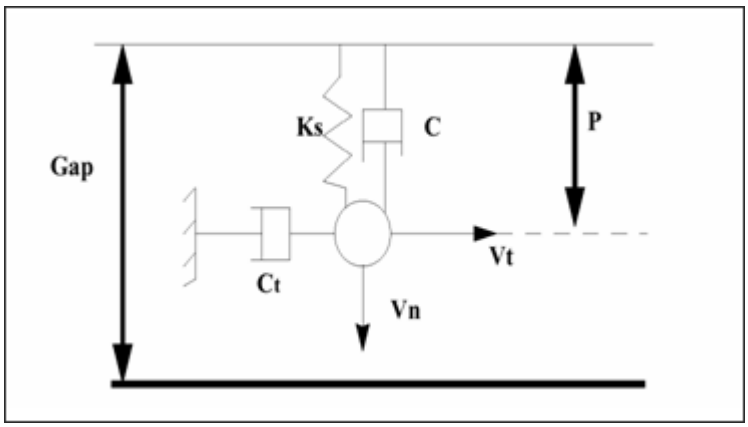


Figure 339: Adhesion Force Computation

The default value for μ is 0 (no friction between surfaces). To compute the friction force, the default friction penalty formulation is a viscous one, based on the tangential velocity. During sliding penetrate the node goes from position C_0 (contact point at time t) to C_1 (contact position at time $t+\Delta t$). As the contact is viscous, a viscous coefficient C is introduced to compute the adhesion force:

$$F_{adh} = C \cdot V_t \tag{222}$$

Where,

$$C = VIS_F \cdot \sqrt{2KM}$$

K	Instantaneous interface stiffness
VIS$_F$	Critical damping coefficient on interface friction
M	Main node mass

Once the adhesion force (F_h) is computed, if it is less than μF_n, the friction force is unchanged equaling F_h and sticking will occur. If the adhesion force is greater than μF_n, then the friction force is reduced and equals μF_n.

$$F_t = \min(\mu F_n, F_{adh}) \tag{223}$$

$$F_t = \min(\mu F_n, F_{ad})$$

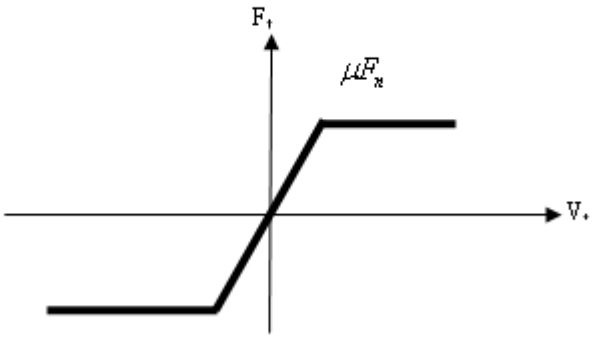


Figure 340: Friction Force Computation

If sliding occurs at a very low speed (for example: quasi-static simulation), the viscous formulation will not work, as the friction force is computed upon the tangential speed. To overcome this limitation, a new friction penalty formulation is available based on tangential displacement (stiffness incremental formulation). This method introduces an artificial stiffness, *K* to calculate the variation of the friction force:

$$\Delta F_t = K \cdot V_t \cdot \delta_t \qquad (224)$$

Where,

δ_t	Tangent displacement

Therefore, contrary to the previous formulation, the stiffness formulation is able to compute the proper friction force even at a low speed. Figure 341 illustrates this point. If an imposed displacement is applied to a part (a 3D cube) at a low speed (0.01 m/s), the viscous formulation will not work; whereas the stiffness formulation based on the tangential displacement will.

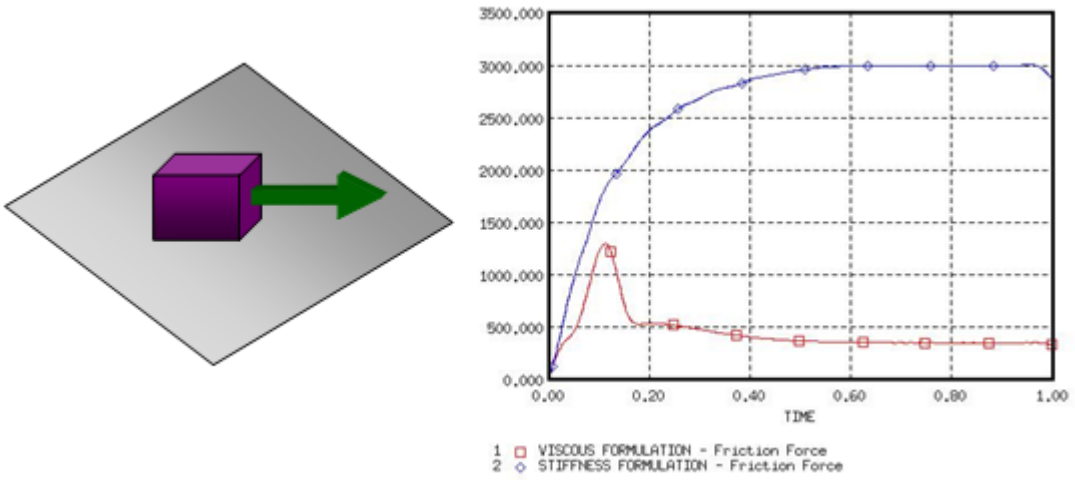


Figure 341: Viscous Formulation versus Stiffness Formulation

Other friction formulations are available, their principle is similar to the Coulomb friction law. Radioss first computes an adhesion force, which is then compared to μF_n. Their differences lie in the friction coefficient (μ) which is not constant anymore, but function on the pressure of the normal force on the main segment and on the tangential velocity of the secondary node. Depending on the flag I_{fric}, three new friction formulations are available:

Generalized Viscous Friction Law

$$\mu = Fric + C_1 p + C_2 \cdot V + C_3 p \cdot V + C_4 \cdot p^2 + C_5 \cdot V^2 \quad (225)$$

Modified Darmstad Friction Law

$$\mu = Fric + C_1 \cdot e^{(C_2 V)} \cdot p^2 + C_3 \cdot e^{(C_4 V)} \cdot p + C_5 \cdot e^{(C_6 V)} \quad (226)$$

Renard Friction Law

$$\mu = C_1 + (C_3 - C_1) \cdot \frac{V}{C_5} \cdot \left(2 - \frac{V}{C_5}\right) \quad (227)$$

if

$$V \in [C_5, C_6] \quad (228)$$

$$\mu = C_3 - \left((C_3 - C_4) \cdot \left(\frac{V - C_5}{C_6 - C_5}\right)^2 \cdot \left(3 - 2 \cdot \frac{V - C_5}{C_6 - C_5}\right)\right) \quad (229)$$

if

$$V \in [C_5, C_6] \quad (230)$$

$$\mu = C_2 - \frac{1}{\frac{1}{C_2 - C_4} + (V - C_6)^2} \quad (231)$$

if

$$V \geq C_6 \quad (232)$$

$$P = C_1 \cdot \mu + C_4 \cdot \rho \cdot C_v \cdot T = C_1 \cdot \mu + \alpha_v \cdot T \quad (233)$$

Note: Friction filtering is available for all friction formulations and allows you to smooth the friction force. Refer to Radioss Starter Input for more details.

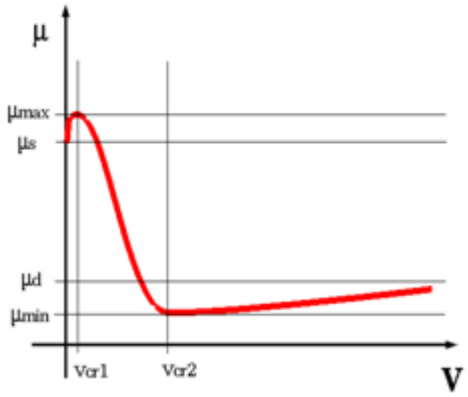


Figure 342: Graphical Representation of Renard Friction Model

Heat Exchange

In interface TYPE7 there are three heat exchanges: heat transfer, radiation and heat friction are allowed by I_{the}=1.

For heat transfer between secondary and main with *Iform_the*, two different heat transfer can be defined. One is set constant temperature in interface where, heat exchange is only between this interface and shell secondary side. The other is heat exchange between all pieces in contact.

If *Frad*≠ 0, then radiation is computed inside of distance *Drad* (max. distance for radiation computation). It is recommended to not set too high of a value for *Drad* , otherwise Radioss Engine performance may be reduced.

Using $Fheat_s$ and $Fheat_m$ frictional sliding energy will be converted into heat. Since friction heat is divided between secondary and main side, generally $Fheat_s + Fheat_m < 1.0$. The frictional heat Q_{Fric} is defined:

If I_{form}=2 (a stiffness formulation):

Secondary side:

$$Q_{Fric} = Fheat_s \cdot \frac{(F_{adh} - F_t)}{K} \cdot F_t \quad (234)$$

Main side:

$$Q_{Fric} = Fheat_m \cdot \frac{(F_{adh} - F_t)}{K} \cdot F_t \quad (235)$$

(*Ithe_form*=1)

Here, K is contact stiffness F_{adh} is adhesive force.

If I_{form}=1 (a penalty formulation):

Secondary side:

$$Q_{Fric} = Fheat_s \cdot C \cdot V_t 2 \cdot dt \quad (236)$$

Main side:

$$Q_{Fric} = Fheat_m \cdot C \cdot V_{t2} \cdot dt \tag{237}$$

Interface Time Step Control

The previous section explains that the time step can be seriously reduced during contact since stiffness is added to all penetrated nodes. Moreover, to prevent any node from going through the main segment during one cycle, a kinematic time step is also computed. If the node impact velocity is high enough to pass through the segment on one cycle, Radioss reduces the time step in order to apply the penalty force when the node is in the gap distance. If p is the penetration distance, then dp/dt is the penetration speed and the kinematic time step is necessary time for the node to go over half of the distance between the node and the segment. The nodal time step is also computed to ensure the numerical stability. The smallest time step is then used for the simulation.

Nodal time step during contact:

$$dt_{nodal} = \sqrt{\frac{2M_{nodal}}{\sum\left(K_{interface} + K_{element}\right)}} \tag{238}$$

Kinematic time step:

$$dt_{kin} = \frac{1}{2}\left[\frac{Gap - p}{dp/dt}\right] \tag{239}$$

Note: Regarding the impact velocities in automobile crash simulations, kinematic time step cannot be activated. Its activation may be due to an incoherency in the model.

If for some reason a node is highly penetrated, either the nodal time step or the kinematic time step may be very low. Then, it is possible to release this node from the interface using the option `/DT/INTER/DEL` in the Engine file. All nodes reaching $dt_{\min}$ will be removed from the interface.

Note: This option may be useful in order to keep a decent time step during contact, but if the number of released nodes is too large, poor results can be expected. Take note of the following message displayed in the Engine output file:

```
**WARNING MINIMUM TIME STEP XXX IN INTERFACE 1
REMOVE SECONDARY NODE XXX FROM INTERFACE
```

Increase in Mass

The use of mass scaling (`/DT/NODA/CST`) may lead to a mass instability. As a node penetrates, its global stiffness increases (the instantaneous interface stiffness, K_t is added); therefore, its nodal time step decreases. In order to match the minimum time step, Radioss adds the needed mass to the node. Unfortunately, this added mass increases the kinetic energy and the penetration gets bigger.

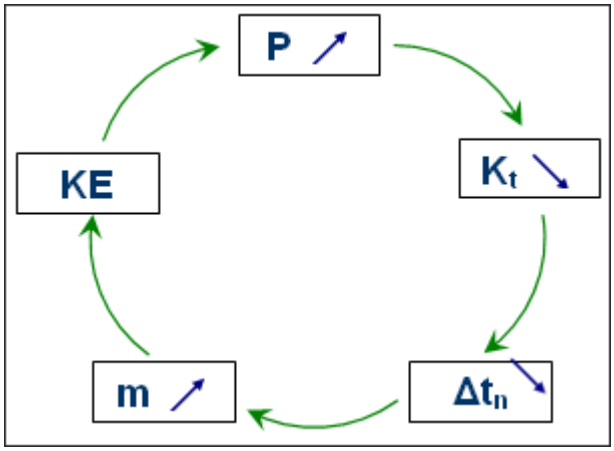


Figure 343: Effect of Mass Scaling on Interface TYPE7

Unless the interface is able to stop the penetration, the added mass (due to mass scaling) will keep on getting bigger and bigger. Therefore, the computation is likely to stop, since the mass variation may get huge very quickly (few cycles). If this is the case, the interface should be modified:

- Gap should be increased
- Initial stiffness can be increased
- Mesh should be modified to be refined and uniform in the contact zone

Soft Part against Hard Part

The default stiffness value computed by Radioss is often suitable to avoid very high penetration, leading to the time step dropping. When contact occurs between similar materials, there is no problem using the default stiffness; except when the materials are different. For instance, when mild steel impacts soft foam, the default stiffness may be too low to avoid a large penetration. When such contacts are willing to occur, it is advised to first compute the ratio of the secondary material stiffness over the main material stiffness. If this ratio is greater than 100, a scale factor (*Stfac*) equal to this ratio should be used to increase the interface stiffness.

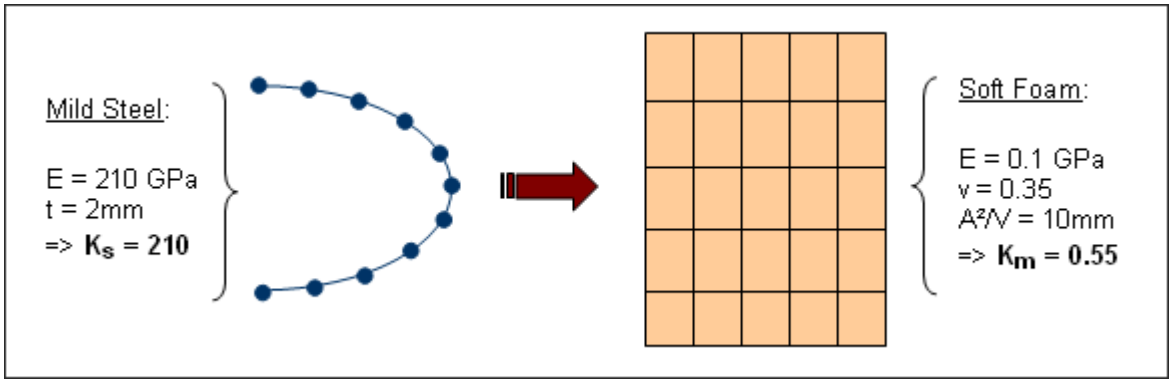


Figure 344: Impact Between Steel and Foam

Figure 344 shows contact between mild steel and soft foam. The ratio of stiffness is greater than 380, in such a case where the main side is the soft side, the flag *Stfac* may be set to 380 to avoid very high penetrations.

Edge to Edge Impact Locking

Interface TYPE7 does not deal with edge to edge impact. The limitation of this interface during the edge to edge contact is illustrated in Figure 345.

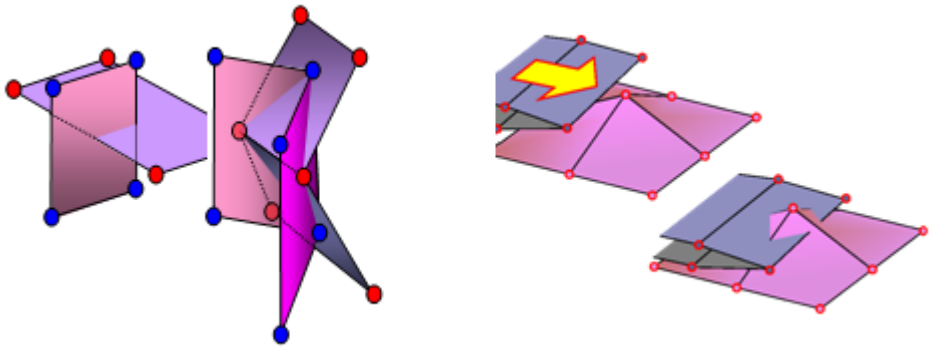


Figure 345: Edge to Edge Contact

When the mesh is fine enough, edge to edge penetration is often followed by a node to shell contact. The main problem with edge to edge impact is the locking situations. If there is a change of load after edge penetration, locking is unavoidable since a node to surface contact is detected (Figure 346). This usually leads to high penetration; therefore, the analysis comes to a standstill as time step is reduced. If locking occurs, the use of interface TYPE11 in this area is necessary to solve the problem.

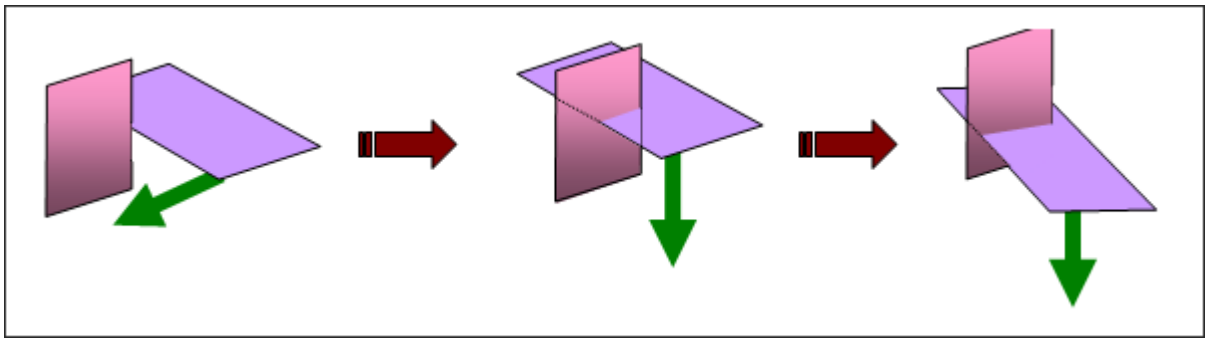


Figure 346: Locking After Edge Penetration

Generation of Tangential Force

A tangential force can be generated when a penetrated node is sliding without friction. This behavior is due to the gaps overlapping around the edges. Figure 347 illustrates that by the cylindrical gap around the edges, the force is no longer normal to the shell mid-surface.

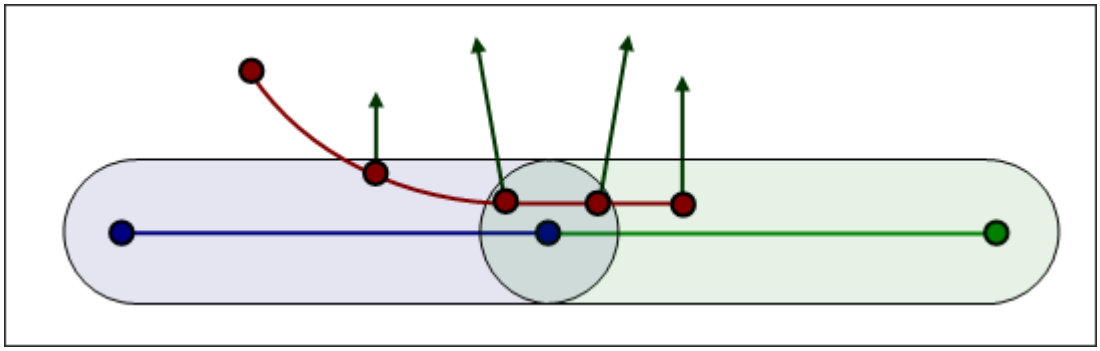


Figure 347: Generation of Tangential Force

Note: No problems were observed in crash simulations, but in metal forming where friction is very important this can seriously decrease the accuracy.

Comments

1. It is always advised to post-process the contact forces. If they are too large, taking into account the physical understanding, the model must be checked.
2. To post-process the forces for symmetric contact, the interface can be split into four interfaces. For example, for two parts A and B, you can create:
   - Interface 1 with A Secondary and B Main
   - +Interface 2 with A Main and B Secondary
   - +Interface 3 with A Main and Secondary
   - +Interface 4 with B Main and Secondary

Auto-Contact Gap Warning

This warning should only be taken into account of a self-impacting interface.

When simulating auto-contact, it is strongly advised to use a minimum gap of at least one half of the smallest segment edge. The message means that at least one element on the main side has a side length less than twice the `GAP`, and there is an over-stiffening risk.

```
WARNING ID: 94
** WARNING IN INTERFACE GAP
INPUT GAP 1.7
HOWEVER GAP IS RECOMMENDED TO BE LESS THAN 1
```

Figure 348 illustrates a self-impacting interface for which the `GAP` equals half the side length of an element. If this element is compressed more than 50%, the red node enters into the gap of the neighboring element; a self-contact is then detected which should not be the case. This leads to over-stiffening of the structure.

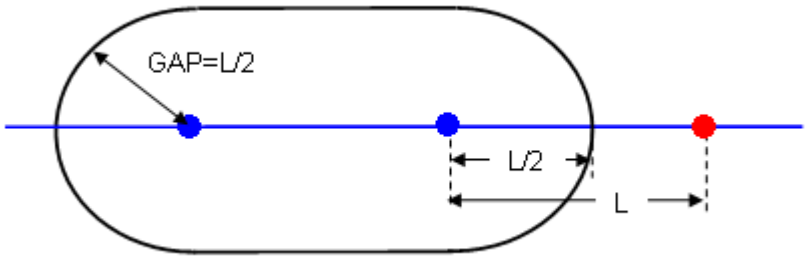


Figure 348: Recommended Gap When Modeling Auto-Contact

If the side length, *L* of an element is lower than the `GAP`, a self-contact will even be computed from the beginning of the computation.

Such a situation can be accepted if it is an exception, but not for frequent cases over the self-impacting interface. It is possible to localize the source of this message by using a pre-processor and selecting the elements through size criteria.

Rupture

When rupture is modeled and elements are expected to fail, it is important to deal with failed elements defined either as a main segment, or as secondary nodes.

When an element is deleted, its corresponding segment is still taken into account and its nodes are still considered for contact. This behavior can lead to a huge error, as deleted elements have no stiffness; therefore, it is recommended to set the flag I_{del} to 2 when failure is expected. This flag will remove any failed element from the main surface and corresponding nodes will be removed from the secondary nodes list.

Note: The I_{del} flag exists as of Radioss V5.1. For older versions, refer to `/DELINT/ON`.

Drawbead Interface (/INTER/TYPE8)

Interface TYPE8 is used to simulate drawbeads. This interface is mainly used in the process industry to model metal forming. Drawbeads are used to prevent the metal strip from sliding during the stamping process.

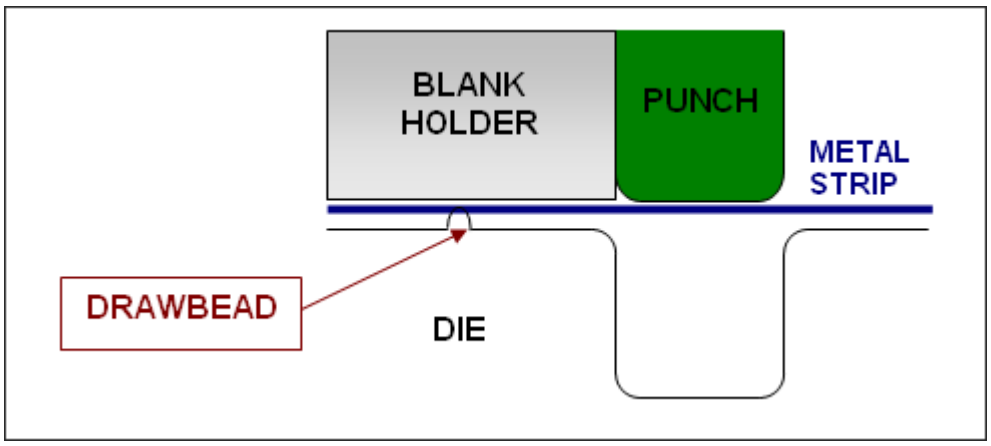


Figure 349: Stamping Process

The modeling of the drawbeads using interface TYPE8 consists in defining a line of secondary nodes and a main surface. The set of secondary nodes must be sorted since the input force is a force per unit length, and the length is computed as the distance between two successive secondary nodes. As soon as a secondary node projects onto a main surface, a tangential force is applied to all main nodes to counter their tangential velocity (Figure 350). Consequently the metal strip does not slide.

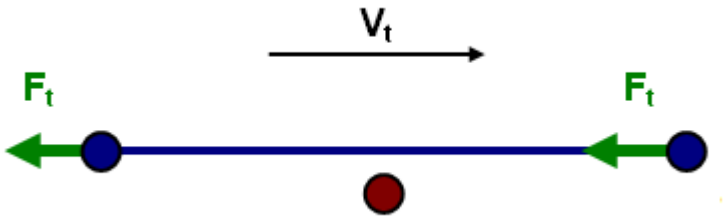


Figure 350: Drawbead Force

Note: The global force acting on the main surface cannot be greater than the drawbead force per unit length (user-defined) multiply by the distance of two successive secondary nodes. In that case, sliding of the metal strip will occur.

Common Problem

In case the drawbead is not continuous, it is necessary to create several TYPE8 interfaces, one interface for each continuous drawbead. Otherwise, the drawbead force can be far too high between two successive nodes that are not part of the same continuous drawbead.

Figure 351 illustrates this point. The grey part could be a car hood, and three drawbeads are needed during the stamping process. If only one interface TYPE8 is created, as the drawbeads are not continuous, the distance between node N17 and N18 is much larger than the average distance between two successive nodes. Therefore, the drawbead force applied to nodes N17 and N18 could be too high, leading to unrealistic deformation in these areas.

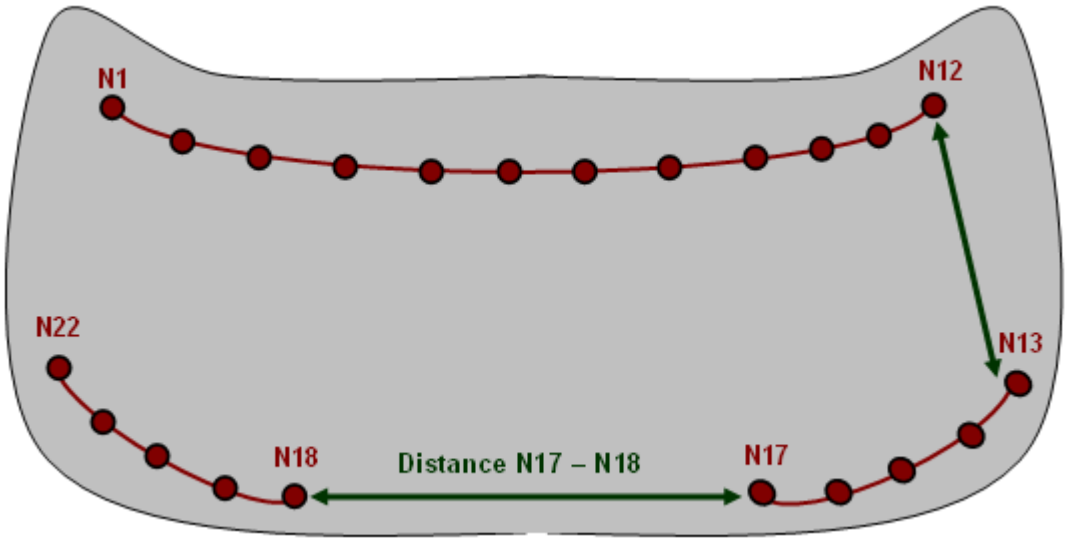


Figure 351: Example of Hood Stamping

Edge-to-Edge Interface (/INTER/TYPE11)

Interface TYPE11 is used to simulate impact between lines for 1D mesh, or edges in shell and solid meshes.

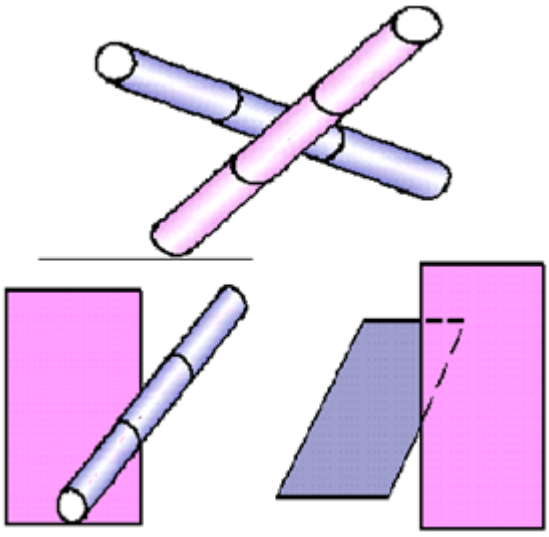


Figure 352: Edge-to-Edge Contact

In association with interface TYPE7, interface TYPE11 solves the locking cases and uses the same penalty formulation and search algorithm as interface TYPE7.

Interface TYPE 11 uses complicated mathematical algorithms; therefore, it is strongly advised to use it only when necessary. A huge interface TYPE11 can slow down the computation.

Quadratic Surface Contact (/INTER/LAGMUL/TYPE16 & /INTER/LAGMUL/TYPE17)

Interface TYPE16 allows defining contact conditions between a group of nodes (secondary) and a curve surface of quadratic elements (main part), as shown in Figure 353 for a symmetric contact. The main part may be made of 16-node thick shells or 20-node bricks. The Lagrange Multiplier method is used to apply the contact conditions; therefore, no gap is necessary to be applied. Some applications of this interface are sliding contacts without gaps as in gear box modeling.

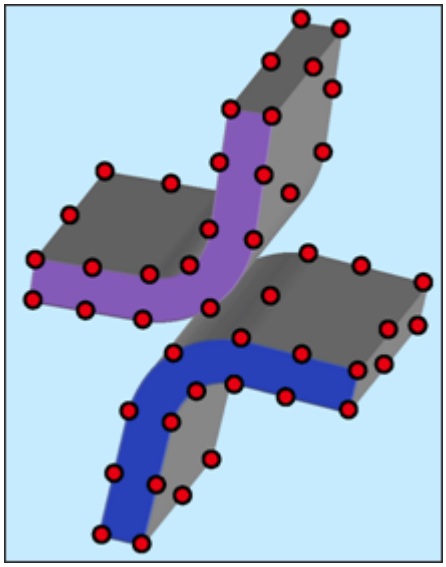


Figure 353: Interface TYPE16 - Node to Quadratic Surface Contact

Interface TYPE17 allows modeling contact between two quadratic surfaces using the Lagrange Multiplier method. It is a generalized form of interface TYPE16 in which the contact on the two quadratic surfaces are directly resolved. The contact is supposed to be sliding or tied.

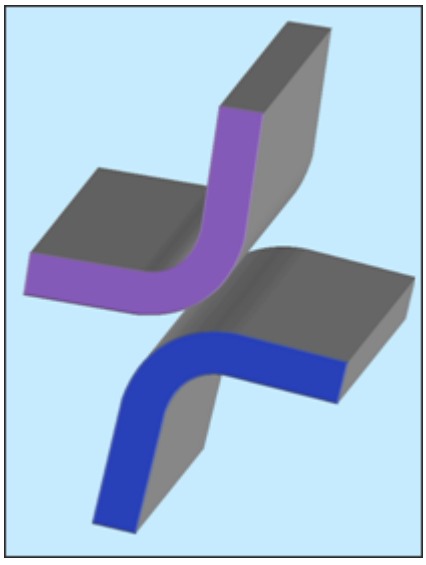


Figure 354: Interface TYPE17 - Quadratic Surface to Quadratic Surface Contact

As defined above, both interfaces do not model friction effects. Contact is either considered as tied or without friction (pure sliding).

> **Note:** When using interfaces with the Lagrange Multiplier method, it is not recommended to symmetrize the contact.

Tied Interface (/INTER/TYPE10)

This interface is similar to interface TYPE7, except for the stiffness which remains constant during the contact.

Moreover, after impact a secondary node becomes tied to the main surface with or without rebound (flag I_{tied}).

Stiffness

During impact, a mass-less spring with constant stiffness is added in order to push the secondary node back. Moreover, as the secondary node is tied, a second massless spring is used to maintain the secondary node close to its impact point. Depending on whether or not rebound is allowed, these two springs are maintained when the secondary node is no longer penetrated.

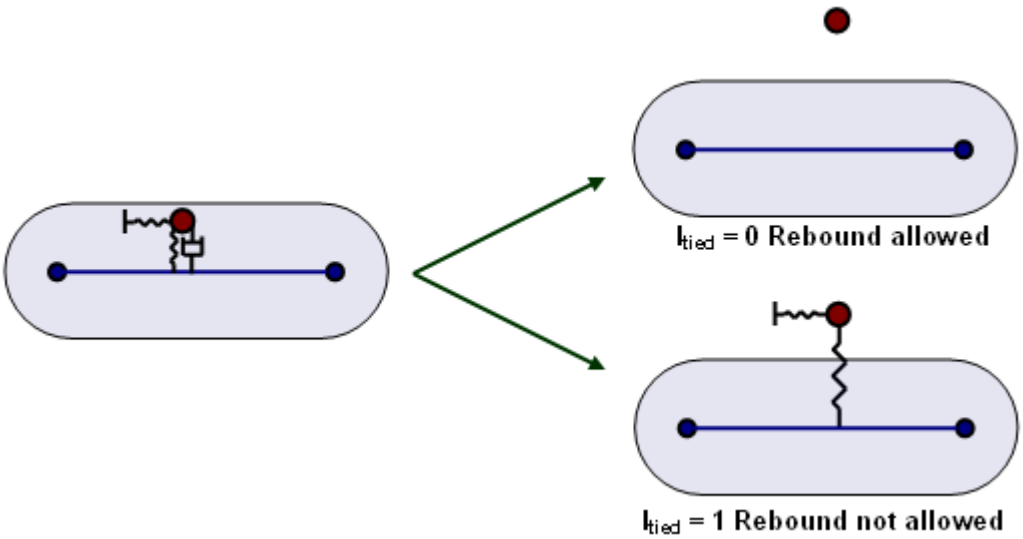


Figure 355: Interface TYPE10

See Also
/INTER/TYPE10 (Starter)

Loads

Preload

The ability to preload a section of elements is available using the `/PRELOAD` option. This can be used to model the pretension force in bolts used in assemblies.

Bolt Tensioning

Details of the physical bolt tensioning problem.

A typical sequence.

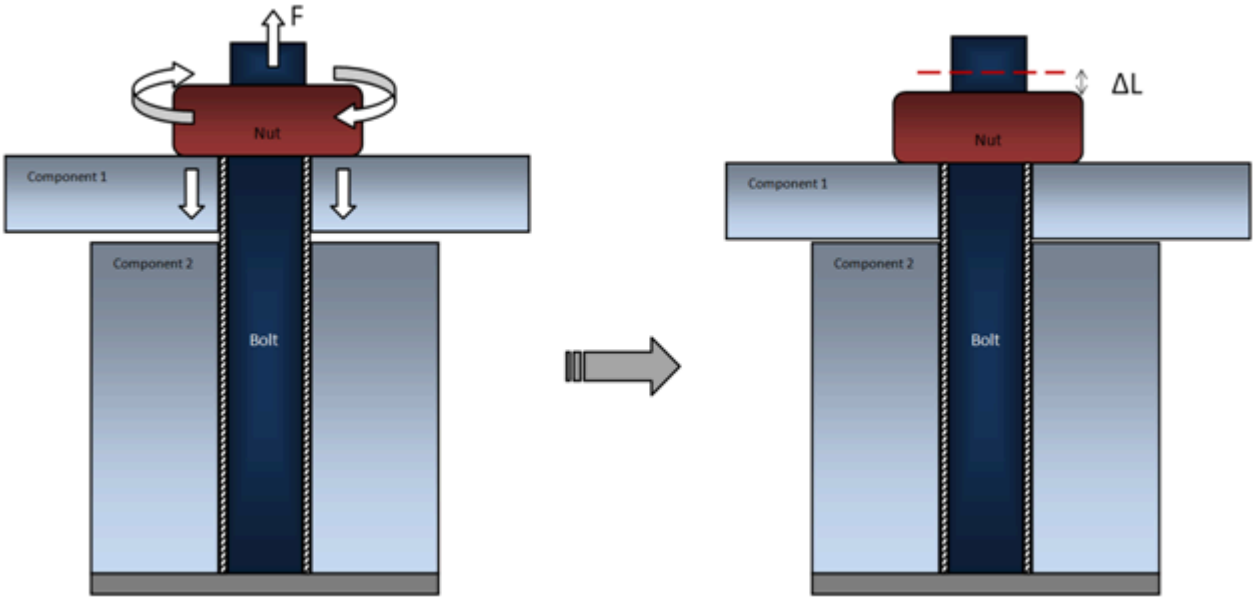


Figure 356: Pretensioned Assembly - Application of Pretensioning Loads (Step 1)

In Step 1, upon preliminary assembly of the structure, the nuts on respective bolts are tightened, usually by applying specified torque (which translates into specified tension force according to the pitch of the thread).

As the result, the working part of the bolt becomes shorter by a distance ΔL. This distance depends upon the applied force, the compliance of the bolt and of the assembly being pretensioned.

From the perspective of FEA analysis, it is important to recognize that:

- Pretensioning shortens the working part of the bolt by removing a certain length of the bolt from the active structure (in reality this segment slides through the nut, yet the net effect is the shortening of the working length of the bolt). Since the bolt stretches, there is a smaller effective length of the bolt material to span the distance from the bolt mount to the nut.
- Calculation of each bolt's shortening ΔL, due to applied forces $\mathbf{f}$, requires FEA solution of the entire model with the pretensioning forces applied. This is because the amount of nut movement due

to given force depends on the compliance of the bolts, of the assembly being bolted and is also affected by cross-interaction between multiple bolts being pretensioned.

At the end of Step 1, the amount of shortening ΔL for each bolt is established and "locked", simply by leaving the nuts at the position that they reached during the pretensioning step.

In Step 2, with the shortening ΔL of all the bolts "locked", other loads are applied to the assembly (Figure 357). At this stage the stresses and strains in the bolts will usually change, while the length of material removed ΔL remains constant for each bolt.

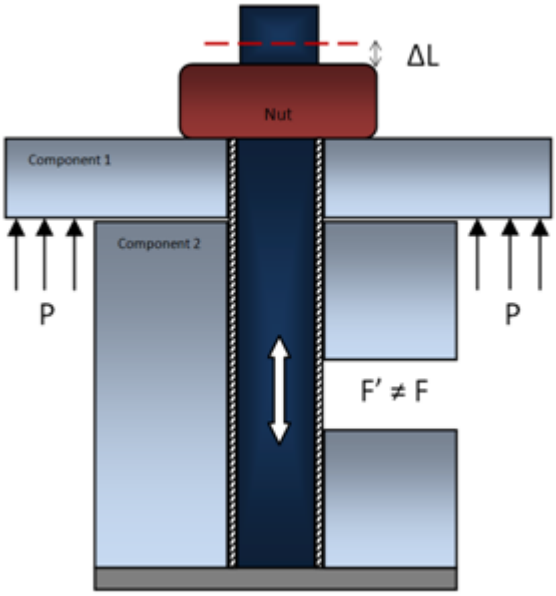


Figure 357: Pretensioned Assembly - Application of Working Loads with "locked" Bolt Shortening (Step 2)

Define a Preload

To create a preload in a bolt, the section option `/SECT` is first used to define a set of elements where the preload will be applied.

The section ID is then referenced in the `/PRELOAD` input. To create the preload, an initial tensile stress applied to the set of elements referenced in the section causes the length of the bolt to shorten in the local z axis of the defined `/SECT`.

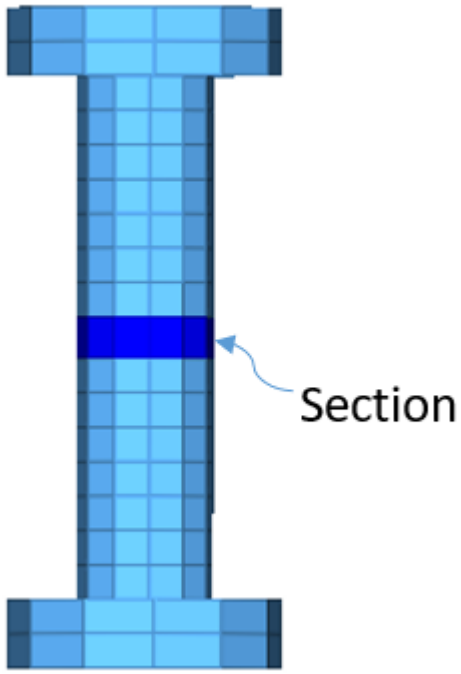


Figure 358: Section which Defines a Group of Elements for Preload

The preload amount can be entered as a force (default) or a stress depending on the `/PRELOAD`, I_{typ} input option. If a force is entered, the initial cross-section area of the `/SECT` is used to calculate a stress that is applied to cause the preload. At the defined `/PRELOAD` T_{start} time, the material stiffness of the elements defined in the section is reduced and the tensile stress is applied to the defined section elements which causes the elements to reduce in length resulting in the required preload. Since the material stiffness is reduced, the amount of strain created in the element is also reduced.

The preloading time between T_{start} and T_{stop} is when the material stiffness goes from a reduced stiffness back to its originaal material stiffness. As shown in Figure 359 , there is a constant reduced stiffness, then a linear increase in stiffness to the original material stiffness, followed by a constant stiffiness.

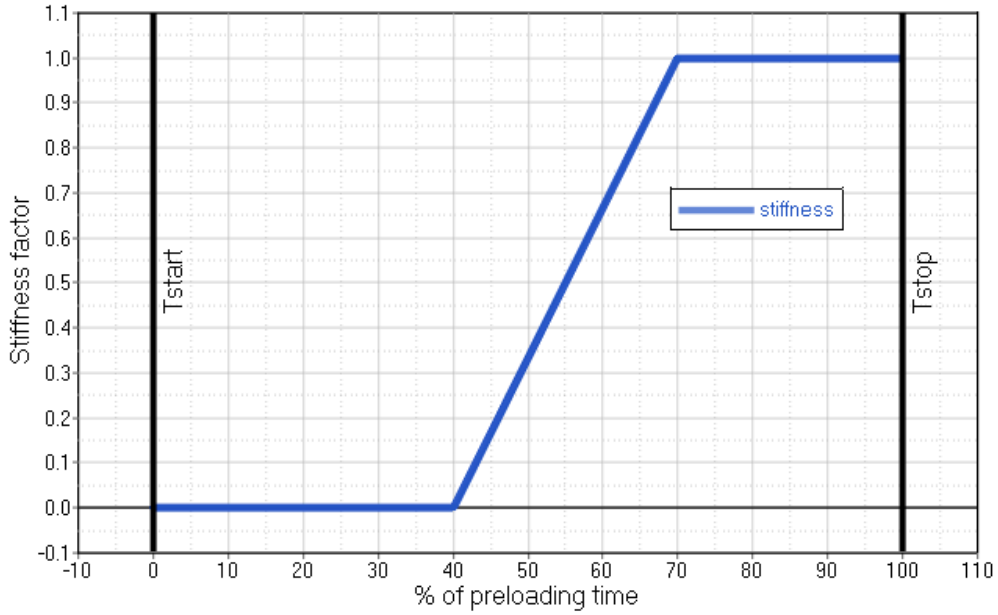


Figure 359: Increase in Material Stiffness

The preload can also be activated using a `/SENSOR` and in this case the T_{start} and T_{stop} times are shifted based on the sensor activation time.

A `/TH/SECTIO` should be created to output the pretension section force. Depending on the assembly being bolted, contact gap between the bolt parts and bolt material, the bolt pretension input may not exactly match the output section force.

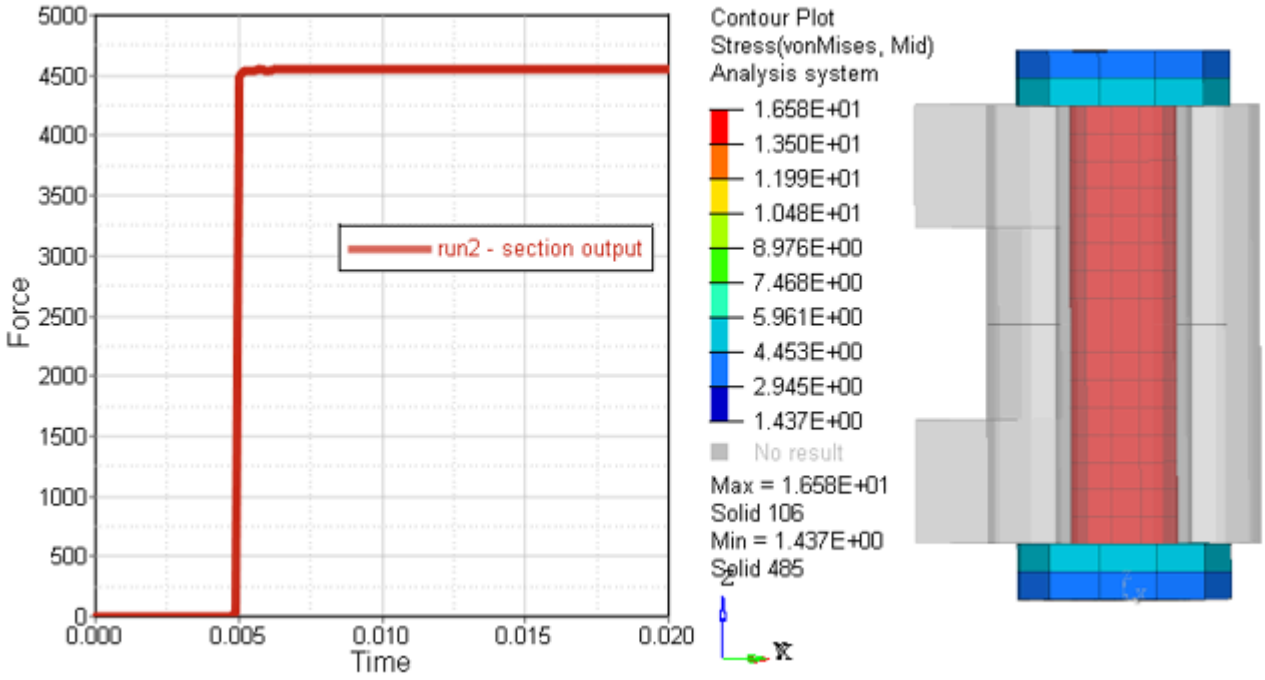


Figure 360: Section Output and Bolt Stress

Airbag Modeling

Airbags are modeled as monitored volumes `/MONVOL` in several different ways.

There are other monitored volume types used to model closed volumes filled with gas or liquid such as; tires, gas tanks (sloshing ignored), and LCD screens.

Monitored Volume (/MONVOL)

A monitored volume is defined with one or more shell (3-node or 4-node) parts.

The defined surface must be closed (Radioss Starter checks it and outputs an error message, if the surface is not closed). The shells normal must be oriented outside the volume. It is possible to reverse the shells normal for a given part, set by entering a negative part number (or by using an appropriate pre-processor to reverse them).

The following types of monitored volume may be used to model airbags:

- AIRBAG1: uniform pressure classical airbag
- AREA: a closed surface to output its volume and area of post-processing
- COMMU1: uniform pressure chambers with communication
- FVMBAG1: non-uniform pressure airbag solved by Finite Volume Method
- GAS: used to model tire pressure or a simple fuel tank
- LFLUID: linear fluid monitored volume to model an incompressible fluid
- PRES: a simple uniform pressure monitored volume

The properties and material laws used to model the fabric could be:

- `/MAT/LAW19` + `/PROP/SH_ORTH` - Used to model a fabric with a linear elastic behavior.
- `/MAT/LAW58` + `/PROP/SH_FABR` - Used to model the behavior of a nonlinear elastic fabric.

Recommendations:

- Fabric materials behave as a membrane and therefore, one integration point through the thickness is recommended, N=1.
- The fabric may be modeled with 3-nodes shell element, using the default Standard triangle (C0) element formulation which does not hourglass or with 4-nodes shell using the QBAT element, I_{shell} =12 with I_{smstr}=11.
- Although no longer recommended, if I_{shell} =4 is used, then to avoid hourglass, define the shell membrane hourglass coefficient *hm* and the shell out of plane hourglass *hf* to be 10%.
- To define the communicating surface for the COMMU1 type monitored volume, void materials and properties should be used `/MAT/VOID` and `/PROP/VOID`).
- The FVMBAG1 type may be used to model gas flow effects which is necessary with curtain airbags and Out Of Position (OOP) simulations.
- Generally, to define an interface self-impact contact, you may define one auto-impacting interface TYPE7 and one edge-to-edge interface contact TYPE11. If there are some initial penetrations, then set *Inacti* =6. These two contacts can also be replaced by one TYPE19 contact.

AIRBAG1

Uniform pressure is assumed inside the volume. Perfect gas law and adiabatic conditions are assumed. Injected mass (or mass flow rate) and temperature are defined as a time function using the injector property. A sensor can define the inflator starting time.

The deflation of the vent hole occurs after reaching a pressure (P_{def}) or time (t_{def}) criteria.

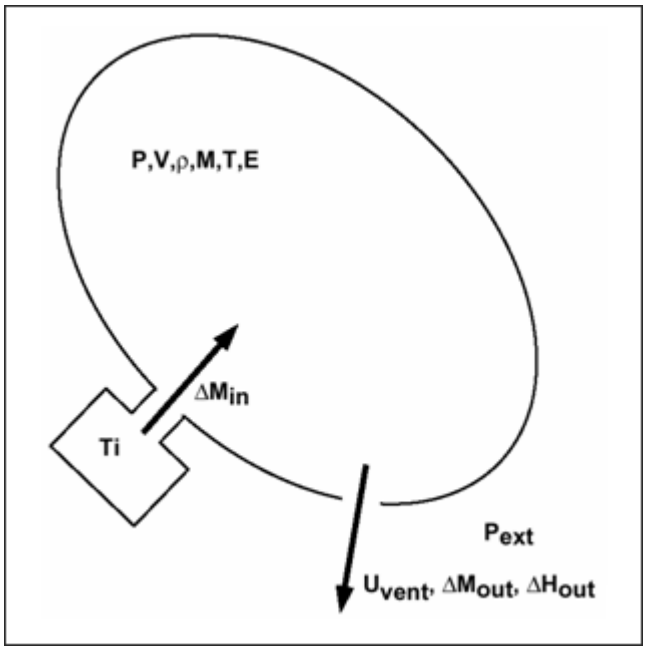


Figure 361:

Numerical Damping

Viscosity, μ can be used to reduce numerical oscillations.

If μ =1, a critical damping (shell mass and volume stiffness) is used. A viscous pressure, q is computed as:

$$q = -\frac{\mu}{A}\sqrt{\frac{PA\rho t}{V}}\frac{dV}{dt} \text{ if } \frac{dV}{dt} < 0$$

$$q = 0 \text{ if } \frac{dV}{dt} > 0$$

Where,

t	Fabric thickness
ρ	Density of the fabric
A	Bag surface

The applied pressure is:

$$P - P_{ext} + q \qquad (240)$$

Initial Conditions

To avoid initial disequilibrium and mathematical discontinuity for zero mass or zero volume, the following initial conditions are set at time zero (I_{equil} =0) or at the beginning of jetting (if I_{equil} =1).

- $P_{ext} = P_{ini}$ external pressure
- $T_0 = T_{ini}$ initial temperature (295K by default)
- If the initial volume is less than $10^{-4}A^{3/2}$, a constant small volume is added to obtain an initial volume: $V_{ini} = 10^{-4}A^{3/2}$
- Initial mass, energy and density are defined from the above values.

There is no need to define an injected mass at time zero.

Gases Definition

1. Initial and injected gas is defined with `/MAT/GAS`. Four types of gas (MASS, MOLE, PREDEF, or CSTA) could be defined. Then the specific capacity per unit mass at constant pressure for the gas is:
   - MASS type

     $$C_p = \left(C_{pa} + C_{pb}T + C_{pc}T^2 + C_{pd}T^3 + \frac{C_{pe}}{T^2} + C_{pf}T^4\right) \quad (241)$$

   - MOLE type

     $$C_p = \frac{1}{MW}\left(C_{pa} + C_{pb}T + C_{pc}T^2 + C_{pd}T^3 + \frac{C_{pe}}{T^2}\right) \quad (242)$$

     Where, MW is the molecular weight of the gas.
   - CSTA type

     User input C_p and C_V with the unit of $\left[\frac{J}{kgK}\right]$.
   - PREDEF type

     About 14 commonly used gases (N2, O2, Air, etc) predefined in Radioss.
2. Injected gas

   N_{jet} defines the number of injectors by monitored volume. The material of the injected gas is defined with `/MAT/GAS`. The injector properties (`/PROP/INJECT1` or `/PROP/INJECT2`) define the injected mass curve defined *fct_ID*$_M$ and injected temperature curve defined *fct_ID*$_T$.

   Injected mass curve and injection temperature can be obtained:
   - From the airbag manufacturer
   - From a tank test

   sens_ID is the sensor number to start injection.
3. Jetting effect I_{jet} is used only for `/MONVOL/AIRBAG1` or `/MONVOL/COMMU1`

   If $I_{jet} \neq 0$, the jetting effect is modeled as an overpressure ΔP_{jet} applied to elements of the bag.

   $$\Delta P_{jet} = \Delta P(t) \cdot \Delta P(\theta) \cdot \Delta P(\delta) \cdot \max(\mathbf{n} \cdot \mathbf{m}, 0) \quad (243)$$

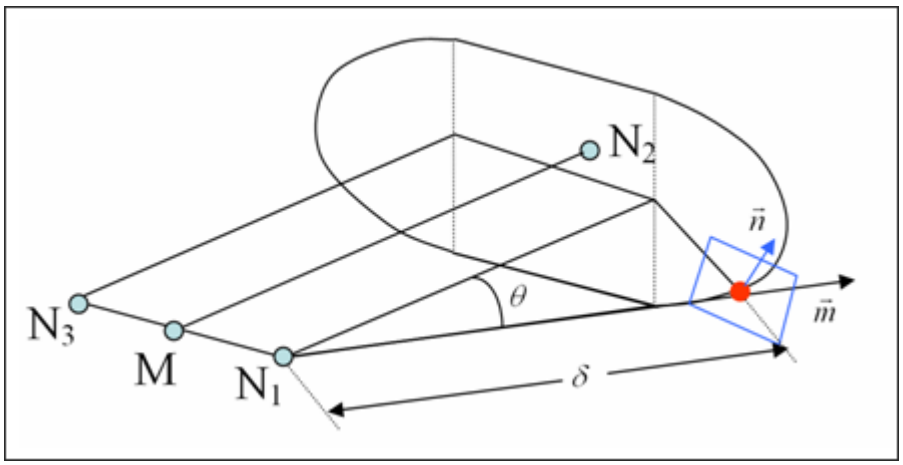


Figure 362:

N_1, N_2, and N_3 are defined based on the injector geometry (refer to the *Radioss Starter Input Manual*)

$\Delta P(t), \Delta P(\theta), \Delta P(\delta)$ are empirical functions provided by the user via fct_ID_{Pt}, $fct\_ID_{P\theta}$, and $fct\_ID_{P\delta}$

Vent Hole Definition

N_{vent} defines the number of vent holes used.

$surf\_ID_v$ is the surface identifier defining the vent hole

A_{vent} is the vent area (if $surf\_ID_v = 0$) or a scale factor ($surf\_ID_v \neq 0$)

B_{vent} = 0 (if $surf\_ID_v = 0$) or a scale factor on the impacted surface ($surf\_ID_v \neq 0$)

T_{stop} is a stop time for venting

T_{start} is the time at which leakage starts

ΔP_{def} is the relative vent deflation pressure

ΔtP_{def} is the time duration during which $\Delta P > \Delta P_{def}$

fct_ID_v is the function identifier $f_P(P - P_{ext})$ for Chemkin model (I_{form}=2)

$$vent\_holes\_surface = A_{vent} \cdot A_{non\_impacted} \cdot f_t(t) \cdot f_P(P - P_{ext}) \cdot f_A\left(\frac{A_{non\_impacted}}{A_0}\right) \quad (244)$$

If $fct\_ID_v \neq 0$, the outflow velocity, v is defined by Chemkin as:

$$v = Fscale_v f_v(P - P_{ext}) \quad (245)$$

Where, $Fscale_v$ is the scale factor of the function fct_ID_v.

and the outgoing mass is computed as:

$$\dot{m}_{out} = \rho \cdot A_{vent} \cdot \mathrm{f}_v(P - P_{ext}) \cdot Fscale_v \tag{246}$$

Or, with the conservation of enthalpy between airbag and vent hole, adiabatic conditions and unshocked flow, it is then possible to express outgoing mass flow through vent holes as a function of P_{ext}, ρ, P_{vent}, u_{vent} and A_{vent}.

$$\dot{m}_{out} = \rho_{vent} \cdot A_{vent} \cdot u = \rho\left(\frac{P_{ext}}{P}\right)^{\frac{1}{\gamma}} \cdot A_{vent} \cdot u \tag{247}$$

In the case of supersonic outlet flow, the vent pressure, P_{vent} is equal to external pressure, P_{ext} for unshocked flow. For shocked flow, P_{vent} is equal to critical pressure, P_{crit} and velocity, u is bounded to critical sound speed:

$$u^2 < \frac{2}{\gamma+1}c^2 = \frac{2\gamma}{\gamma+1}\frac{P}{\rho} \tag{248}$$

and

$$P_{crit} = P\left(\frac{2}{\gamma+1}\right)^{\frac{\gamma}{\gamma-1}} \tag{249}$$

$$P_{vent} = \max(P_{crit},\ P_{ext}) \tag{250}$$

The outgoing mass flow of gas i is:

$$\dot{m}_{out(i)} = \frac{V^{(i)}}{V}\dot{m}_{out} \tag{251}$$

Where, $V^{(i)}$ is the volume occupied by gas i and satisfies:

$$V^{(i)} = \frac{n^{(i)}}{n}V \tag{252}$$

from $PV^{(i)} = n^{(i)}RT$ and $PV = \left[\sum_i n^{(i)}\right]RT$.

Then,

$$\dot{m}_{out(i)} = \frac{n^{(i)}}{\sum_i n^{(i)}}\dot{m}_{out} \tag{253}$$

Porosity

The isenthalpic model is also used for porosity. In this case, you can define the surface for outgoing flow:

$$A_{eff} = C_{ps} \cdot Area_{ps} \tag{254}$$

or,

$$A_{eff} = \mathrm{C}_{ps}(t) \cdot \mathrm{Area}_{ps}(P - P_{ext}) \quad (255)$$

Where,

$\mathrm{C}_{ps}(t)$	Function of *fct_ID*$_{cps}$
$\mathrm{Area}_{ps}(P - P_{ext})$	Function of *fct_ID*$_{aps}$

It is also possible to define closure of the porous surface when contacts occurs by defining the interface option I_{bag}=1.

See Also
Monitored Volume (Theory Manual)

COMMU1

This option is used to simulate chambered airbags and may be used to unfold an airbag.

Each `COMMU1` type monitored volume works like an `AIRBAG1` type monitored volume with possible vent communication with some other `COMMU1` type monitored volume. A chambered airbag is therefore designed with two or more `COMMU1` type monitored volumes.

Each monitored volume can have an inflator and vent holes.

Case 1: Folded Airbag

To model a folded airbag, one `COMMU1` type monitored volume is used for each folded part. The boundary between two folded parts is closed with a void (dummy) property set. The area of communication is defined with this void property set. The pressure in each folded part will be different and the area of communication will increase during inflation. With this modelization the volume with inflator will inflate first and before the folded parts.

Volume 1: Prop. 1 + 4 + 5	Communication area: vol. 1 to 2: prop. 4 vol. 1 to 3: prop. 5
Volume 2: Prop. 2 + 4	Communication area: vol. 2 to 1: prop. 4
Volume 3: Prop. 3 + 5	Communication area: vol. 3 to 1: prop. 5

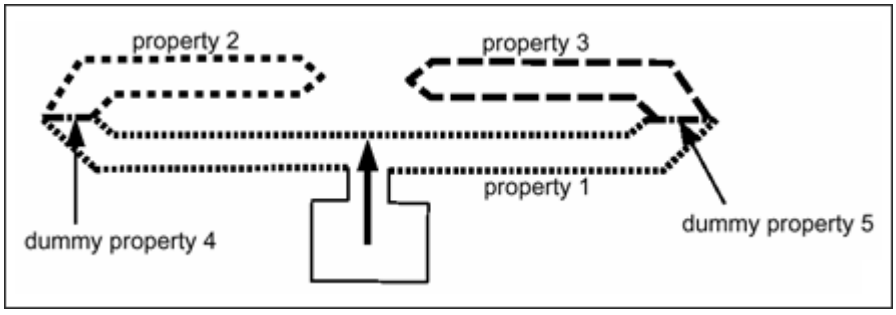


Figure 363:

Case 2: General Use

Monitored volume 1 can communicate with monitored volume 2 with or without communication from 2 to 1. Communication area, deflation pressure or time from 1 to 2 can be different than corresponding values from 2 to 1. That way it is possible to model a valve communication.

Two communication monitored volumes can have common nodes or common shell property set, but this is optional.

Volume 1 communicates with volume 2, and volume 2 with volume 1 and 3, but there is no communication from 3 to 2.

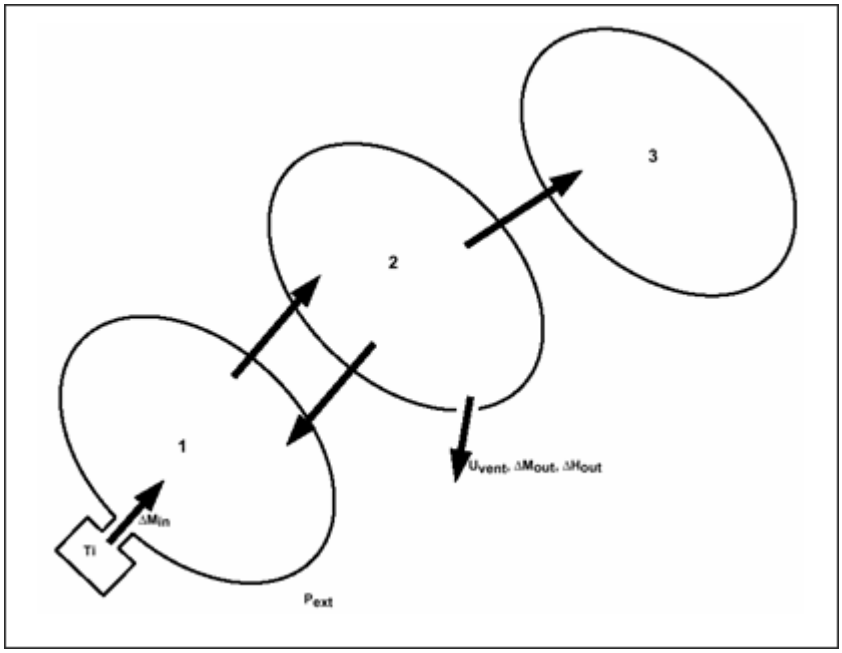


Figure 364:

General Equations

Same equations for `AIRBAG1` type monitored volume are used, but incoming and outgoing enthalpy and kinetic energies will take into account the communicating bags. For each communicating volume for which pressure is lower than in current volume, a mass and energy flow is computed with same equations for vent holes, the external pressure is just replaced by the pressure of communicating volume:

$$u^2 = \frac{2\gamma}{\gamma - 1}\frac{P}{\rho}\left(1 - \left(\frac{P_{vent}}{P}\right)^{\frac{\gamma-1}{\gamma}}\right) \quad (256)$$

with,

$$P_{vent} = \max\left(P_{crit}, P_{neighbor}\right) \quad (257)$$

$$\rho_{vent} = \rho\left(\frac{P_{vent}}{P}\right)^{1/\gamma} \quad (258)$$

$$\dot{m}_{out} = \rho_{vent} \cdot A_{vent} \cdot u = \rho \left(\frac{P_{ext}}{P} \right)^{\frac{1}{\gamma}} \cdot A_{vent} \cdot u \quad (259)$$

$$\dot{E}_{out} = \dot{m}_{out} \frac{E}{\rho V} = \left(\frac{P_{ext}}{P} \right)^{\frac{1}{\gamma}} A_{vent} u \frac{E}{V} \quad (260)$$

These mass and energy fluxes are removed from current volume and added to communicating volume at the next cycle.

Inflator, Vent Hole, Initial Conditions

Inflator, atmospheric vent holes and initial conditions are identical to `/MONVOL/AIRBAG1` type monitored volume.

Specific Input

The specific input for this type is:

μ is viscosity factor

P_{ext} is external pressure

ΔP_{def} is relative vent deflation pressure

A_{vent} is vent area ($surf\_ID_v = 0$) or discharge factor ($surf\_ID_v \neq 0$)

T_{start} is time to deflate vent hole

Initial gas and injected gas defined with `/MAT/GAS`.

Inject properties defined with `/PROP/INJECT1` or `/PROP/INJECT2`.

fct_ID$_M$ is injected mass curve (or mass rate)

Fscale$_M$ is scale factor for injected mass curve (or mass rate)

fct_ID$_T$ is injected temperature curve

Fscale$_T$ is scale factor for injected temperature curve

sens_ID is sensor number to start injection

Nvent is number of vent hole

Nbag is the number of communicating volume

For each communicating volume (1 to *Nbag*):

bag_ID is the identification of communicating volume

$surf\_ID_c$ defines the communication area

ΔP_{Cdef} is the relative communication deflation pressure

A_{com} is the communication area ($surf\_ID_c = 0$) or discharge factor ($surf\_ID_c \neq 0$)

T_{com} is the time to deflate communication area

In volume *j* input, the data for communication with volume *k* concerns only the flow from *j* to *k*. The data concerning the flow from *k* to *j* is defined in volume *k* input.

See Also
Monitored Volume (Theory Manual)

FVMBAG1

A non-uniform pressure is assumed inside the volume. The gas flow in `FVMBAG1` is solved using a finite volume integration scheme which allows the gas flow through the airbag to be correctly modeled.

Since the gas flow simulated, the unfolding and deployments behavior is correct which is needed for out of position occupant simulations and curtain airbag deployments.

Perfect gas law and adiabatic conditions are assumed. Injected mass, velocity and temperature are defined as a time function. The injected velocity is obtained using:

$$v_{in}(t) = \frac{dm(t)/dt}{S\rho_{in}(t)} \quad (261)$$

Where, S is the injector surface.

A sensor may define the inflator starting time.

Deflation of vent hole is available after reaching a pressure (P_{def}) or time (t_{def}) criteria.

In this version, heat gas capacity C_p is assumed constant.

More further detail, refer to FVMBAG1 Airbag Modeling Guidelines.

GAS

This monitored volume is typically used to model tire pressure or simple fuel tank.

For tire model V_i is zero and for fuel tank V_i is the fuel volume.

/MONVOL/GAS Examples

- A tire model:

  The inputs are:

  - γ = 1.4
  - μ
  - P_{ext}= 10^5 Pa
  - P_{ini} = initial tire pressure

    Then, the pressure in the tire is $P_{tire} = P_{ini} - P_{ext}$
  - $V_{inc} = 0$
- A fuel tank model if the sloshing effect is neglected

  Only if the sloshing effect is neglected, pressure in a partial filled fuel tank can be modeled with a type GAS monitored volume. Use the following input:

  - γ = 1.4
  - μ

- P_{ext} = 10^5 Pa
- P_{ini} = 10^5 Pa
- V_{inc} = volume of fuel

See Also
GAS Type

LFLUID

A linear fluid monitored volume which models a volume filled with fluid.

In the LFLUID monitored volume, the density and bulk modulus of the liquid are defined. There are optional curves to define the mass flow rate in and out over time. The bulk modulus can also be defined as a function of time.

Additional pressure can be included by defining a function for the pressure added versus time. Lastly, the maximum pressure in the volume can be defined.

PRES

A simple perfect gas monitored volume where the relative pressure is a function of relative volume or a function of time.

In the PRES monitored volume, the relative pressure is defined as P_{rel} = pressure inside of monitored volume - atmospheric pressure.

The relative pressure is defined using a function in one of two ways. By default, the relative pressure is defined as a function of the relative volume $P_{rel}\left(\frac{V_0}{V}\right)$, or the pressure can be a function of time $P_{rel}(t)$.

See Also
PRES Type

Monitored Volume Time Step

An "AIRBAG1" time step is estimated into the Engine, but this time step will never control the time step during the run. If that is the case, it means there is a non-physical airbag definition in the input deck.

Element Time Step Control

/DT/AIRBAG

Defines ΔT_{sca} and $\Delta T_{\min}$ for airbags
ΔT_{sca} $\Delta T_{\min}$

/DT/FVMBAG

Defines ΔT_{sca} and $\Delta T_{\min}$ for Finite Volume Method airbag
ΔT_{sca} $\Delta T_{\min}$

`/DT/FVMBAG` is only used for monitored volumes type `/MONVOL/FVMBAG1`. The ΔT_{sca} and $\Delta T_{\min}$ can be defined to insure stability and a minimum time step size by triggering additional finite volume merges when the `FVMBAG1` entity penalizes the computation. `/DT/FVMBAG/1` is used to merge finite volumes when $dt < \Delta T_{min}$.

FVMBAG1 Airbag Modeling Guidelines

Application Field

Present guidelines define modeling and validation requirements for numerical airbag models in Radioss with Finite Volume Method (FVM).

These guidelines are based on production Radioss versions 2017.2.5, 2019.2.4, 2020.1, 2021.2 double precision which are used by many airbag manufacturers. The single precision version of Radioss cannot be used for airbag calculations. The guideline requirements are based on the common practice of airbag modeling. Some of the requirements are not directly connected to the FVM but are recommended to create an airbag model which is easy to understand and incorporate in crash simulations.

- FVM - Finite Volume Method
- FV - Finite Volume
- UP - Uniform Pressure

General Airbag Model Requirements

Airbag numerical models are created for the numerical simulation of crash events related to occupant safety problems. Airbag models may differ in the amount of detail and accuracy depending on their intended application.

The airbag model should possess all the typical features of airbag hardware: gas generator, housing, retainers, interchamber components, vent holes, coated fabric definition, fabric porosity for non-coated fabric, tethers, seams, etc.

The geometry, mass, inertia properties, and materials of the airbag model should correspond to hardware and CAD data.

The different airbag parts such as airbag components, external surfaces, internal surfaces, and vents, should be organized in assemblies `/SUBSET`.

Each airbag model should represent a model to be run independently and used in other component or full vehicle models as include files.

The airbag model shall be provided in the design position. The airbag positioning inside a component model or full car model should be performed using the `/TRANSFORM` cards.

Meshing and Folding

The airbag fabric should be meshed in the non-folded state using tria elements with an average element size of 3 to 4mm. Mixed + tria mesh is allowed starting in version 2021.2. Larger element sizes are

only acceptable for vent holes and other components that use a void material (example, inter-chamber void component). All nodes of void related elements should be attached to the surrounding fabric components. Airbag fabrics have a low modulus of elasticity and therefore normally the elements size does not affect time step.

External airbag fabric components should form a closed volume with normal vectors directed outwards.

> **Note:** This limitation was released in version 2021, but it is still recommended to have outward normal.

Intersections between any airbag components (for example, external, internal, inflator, or housing) are not allowed.

Airbag folding should be performed using pre-simulation according to the CAD folding design. The folding model should use validated fabric material definitions and contacts. Radioss is the preferred solver for pre-simulation folding (Airbag Solutions). This guarantees that the folded model will work correctly with the airbag internal contact definition listed in corresponding sections.

Reference Geometry

Reference geometry should be represented as flat node based (`/XREF`) or element based (`/EREF`) geometry.

In /EREF, the reference geometry parts of the airbag can be separate parts that are not connected.

In some cases, /XREF can also be used to with reference geometry that is not connected. In this case, each part referenced in the card should not connect to itself in the actual airbag model.

Reference geometry should be parallel to global XY plane.

> **Note:** This is not a requirement of FVM approach but instead simplifies defining the orthotropic directions of airbag components.

For reference geometry that is a closed volume:

- the geometrical center of non-folded airbag should be at point X=0, Y=0, or Z=0
- normal directions of components in reference geometry should be the same as in folded airbag model
- intersections are not allowed in unfolded airbag reference geometry

For disconnected airbag reference geometry based on elements, `/EREF`:

- each component should be in the Z=0 plane
- the components normal direction in the reference geometry should be the same as in folded airbag model

During design iterations, the reference geometry can be modified or morphed in order to investigate the influence of minor airbag shape variations to airbag characteristics.

Note: When using reference geometry, the time step for the fabric is calculated based on the elements reference geometry size. Initial material anisotropic directions are also calculated based on the reference geometry.

Gas Generator

The gas generator model should represent all details available in CAD data: gas generator, injector openings, and retainers.

The gas generator model should represent all the details available in the CAD data: gas generator, injector openings, and retainers. The injector opening is represented by elements on the inflator surface. Inflator components are modeled as shell component with LAW2 material and `/RBODY` rigid body spider. Mass and inertia properties of inflator components should correspond to hardware.

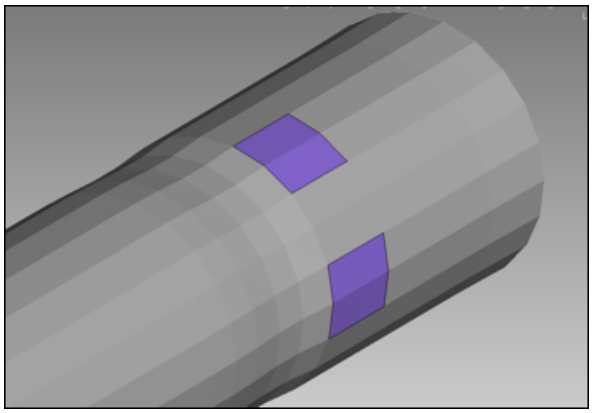


Figure 365: Inflator and Injector Elements

The injection direction is opposite to element's normal. Elements from different injectors should not share common elements. Radioss will assume that the element belongs to the last defined injector. Elements from a vent hole or porous surface also should not share common elements with injectors. Radioss automatically excludes the elements from vent or porous surface definition.

The injector surface can also be defined on an internal surface insideof airbag volume. In this case the gas flow direction is opposite to the internal surface element's normal.

The inflator property `/PROP/INJECT1` and `/PROP/INJECT2` define the parameters of the injected gas composition. Each of the gas components should be specified as a separate material card: `/MAT/GAS/MASS`, `/MAT/GAS/MOLE`, or `/MAT/GAS/PREDEF`. In the latter case, the gas material is built according to the keyword given in the `/MAT/GAS/PREDEF` card and units specified in the `/BEGIN` card.

Example of gas material definition in kg, mm, ms (predefined, related to mass, related to mole):

```
#---1----|----2----|----3----|----4----|----5----|----6----|----7----|----8----|----9----|---10----|
/MAT/GAS/MASS/1
INJECTOR MIXTURE
#                  MW
                .03300
#                 Cpa                 Cpb                 Cpc                 Cpd                 Cpe
             1070.00                   0                   0                   0                   0
#                 Cpf
                    0
```

```
#---1----|----2----|----3----|----4----|----5----|----6----|----7----|----8----|----9----|---10----|
```

The molecular weight and coefficients of specific heat coefficient are defined for each of injected gas components.

The heat capacity function.

$$C_p = \left(C_{pa} + C_{pb}T + C_{pc}T^2 + C_{pd}T^3 + \frac{C_{pe}}{T^2} + P_{pf}T^4 \right) \quad (262)$$

should be monotonically increasing for temperatures between ambient temperature and the maximum temperature value given in the temperature curves of the injector. When the function is not monotonically increasing, Radioss automatically corrects the $C_p(T)$ function to keep it monotonic. This can happen when C_{pe} > 0 or one of C_{pb}, C_{pc}, C_{pd}, C_{pf} coefficients are negative.

The composition of the injected gas, injection mass flow and temperature for each of the gas components are defined in the `/PROP/INJECT1` or `/PROP/INJECT2` cards.

Example of injector property definition:

```
#---1----|----2----|----3----|----4----|----5----|----6----|----7----|----8----|----9----|---10----|
/PROP/INJECT1/1
Inflator 1
#   Ngases     Iflow            Ascale_T
         5         1                   1
#   mat_ID   fct_IDM   fct_IDT                     Fscale_M           Fscale_T
         1         2         1                            0                  0
#   mat_ID   fct_IDM   fct_IDT                     Fscale_M           Fscale_T
         2         3         1                            0                  0
#   mat_ID   fct_IDM   fct_IDT                     Fscale_M           Fscale_T
         3         4         1                            0                  0
#   mat_ID   fct_IDM   fct_IDT                     Fscale_M           Fscale_T
         4         5         1                            0                  0
#   mat_ID   fct_IDM   fct_IDT                     Fscale_M           Fscale_T
         5         6         1                            0                  0
#---1----|----2----|----3----|----4----|----5----|----6----|----7----|----8----|----9----|---10----|
```

This card is then referenced in `/MONVOL/FVMBAG1` airbag card.

The injector Time to Fire (TTF) should be determined through a sensor in `/MONVOL/FVMBAG1`, not through a shift of the mass and temperature curves.

When the inflator is determined correctly and referenced in `/MONVOL/FVMBAG1`, the Radioss Starter outputs the list of elements used as the injector surface.

```
ELEM:       92 <-> SH3N :    55261587 - INFLATOR :         1
ELEM:      502 <-> SH3N :    55262048 - INFLATOR :         1
ELEM:      622 <-> SH3N :    55261581 - INFLATOR :         1
ELEM:      627 <-> SH3N :    55261568 - INFLATOR :         1
ELEM:      633 <-> SH3N :    55262068 - INFLATOR :         1
ELEM:      814 <-> SH3N :    55261601 - INFLATOR :         1
```

The inflator model should be validated using a tank test simulation. The tank test report should include the pressure and temperature measured in the test, injector mass flow for each of the gas components, and temperature curves. The pressure and temperature from the test could be compared to the average pressure and temperature in the simulation.

Initial Air inside of Airbag

Material for the air inside of the airbag should be specified either through `/MAT/GAS/MASS`, `/MAT/GAS/MOLE` or `/MAT/GAS/PREDEF`.

In the latter case, the gas material is defined in the `/MAT/GAS/PREDEF` card in the units defined in the `/BEGIN` card.

Generic material card for air (kg, mm, ms):

```
#---1----|----2----|----3----|----4----|----5----|----6----|----7----|----8----|----9----|---10----|
/MAT/GAS/PREDEF/4
Initial gas
AIR
#---1----|----2----|----3----|----4----|----5----|----6----|----7----|----8----|----9----|---10----|
/MAT/GAS/MASS/102
AIR, kg, mm, ms
#                  MW
#                 Cpa                 Cpb                 Cpc                 Cpd                 Cpe
              966.99                   0                   0                   0                   0
#                 Cpf
                    0
#---1----|----2----|----3----|----4----|----5----|----6----|----7----|----8----|----9----|---10----|
/MAT/GAS/MOLE/104
AIR, kg, mm, ms
#                  MW
             0.028013
#                 cpa                 cpb                 cpc                 cpd                 cpe
              27.088                   0                   0                   0                   0
#                 cpf
                    0
#---1----|----2----|----3----|----4----|----5----|----6----|----7----|----8----|----9----|---10----|
```

The same air material used in the airbag should also be used as air in the tank test simulation for validation of injector.

Airbag Fabric Material

The nonlinear anisotropic material LAW58 used be used as the airbag material.

The material should be tested and validated using the following tests: picture frame test, biaxial tension test, and uniaxial tension test in the weft and wrap directions.

A picture frame test determines the fabric material behavior in shear.

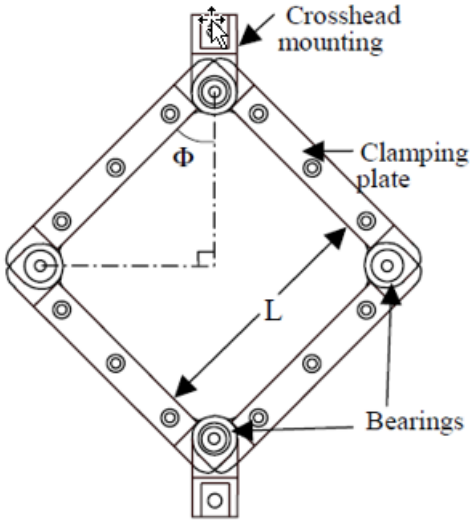


Figure 366: Picture Frame Shear Rig

The shear stress as a function of the anisotropy angle in degrees should be extracted from the force displacement curves from the picture frame test and used as input curve in material `LAW58`.

$$\tau = \frac{F}{2 \cdot t \cdot L \cdot \sin\left(\frac{\pi}{4} + \frac{\alpha}{2}\right)} \tag{263}$$

Where,

$\alpha = \left(\frac{\pi}{2}\right) - 2 \cdot \arccos\left(\frac{1}{\sqrt{2}} - \frac{D}{L}\right)$ *and* D

Vertical displacement of the upper corner of the specimen

L

Side length of the specimen

F

Measured force

t

Fabric thickness

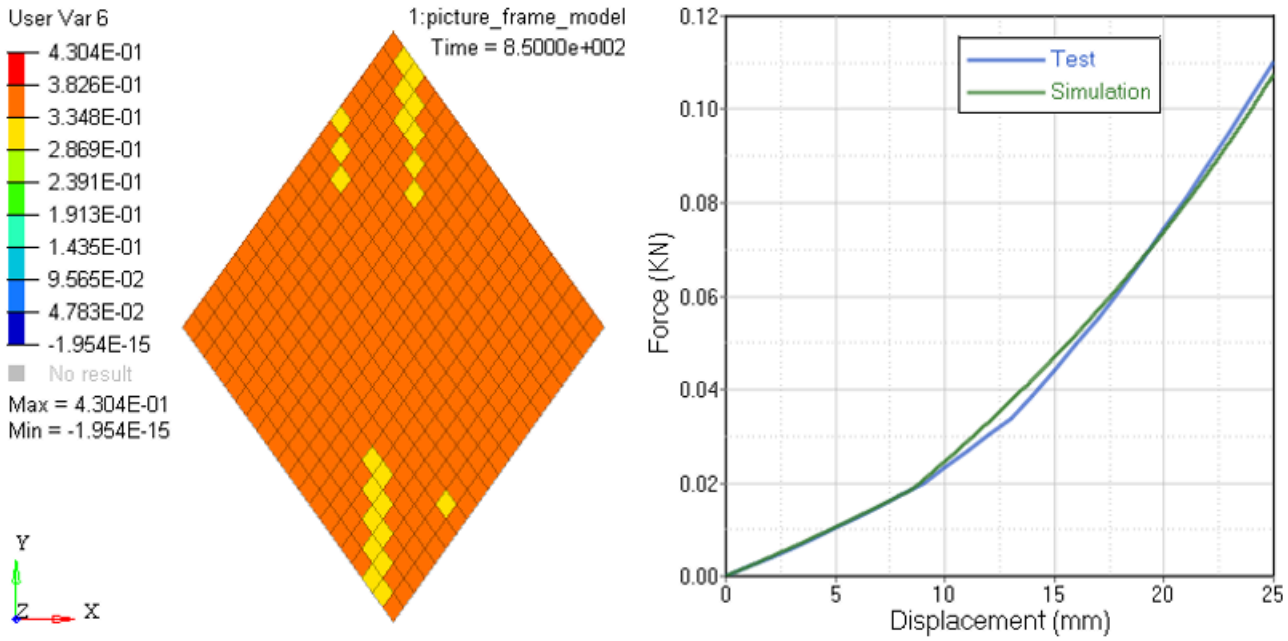


Figure 367: Picture Frame Simulation and Shear Stress as Function of Anisotropy Angle

Cyclic loading and unloading picture frame tests can be performed to extract the loading and unloading curves in shear. The curves can be directly used as input in LAW58 material. The curves should be specified both for the negative and positive values of anisotropic angle in degrees.

The biaxial test determines the `LAW58` stiffness in weft and wrap directions.

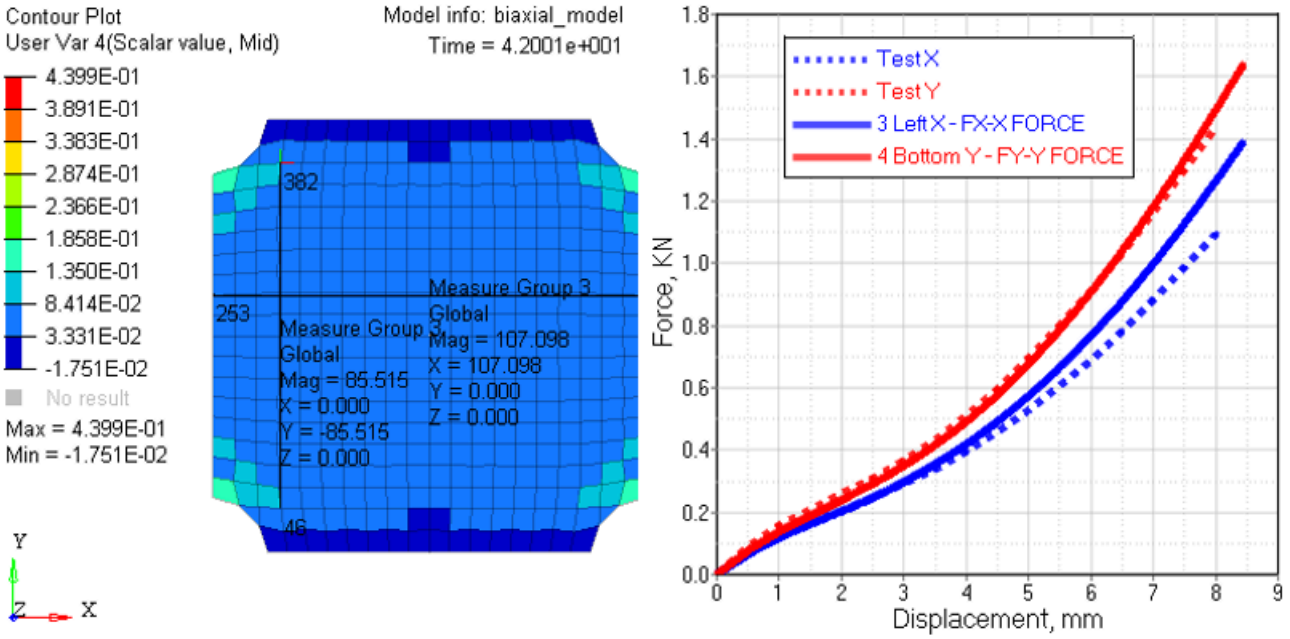


Figure 368: Biaxial Simulation and Forces Displacement Curves in Warp and Weft Directions

The force displacement curves for warp and weft direction can be recalculated into stress and strain curves using size and thickness of the specimen. Due to corner boundary effects, slight scaling or other tuning of the input curves may be necessary.

Before the woven warp and weft fibers become straight, uniaxial test data can be used to validate the initial fabric stiffness for LAW58. During this phase, the fabric is normally softer. The strain straightening parameters S_1 and S_2 define when the strain at when the fibers are straight. $Flex_1$ and $Flex_2$ are the scaling coefficients which reduce the stiffness of the warp and weft curves while the fibers are not straight. In many cases the phase is very short and it can be ignored by defining S_1=S_2=1e-03.

The $Flex_1$ and $Flex_2$ parameters are also used to scale down fabric stiffness in warp and weft direction in compression. By default, $Flex_1$=$Flex_2$=$Flex$=0.01.

Uniaxial test data for 0, 90 and 45 degree fiber orientations can be used to double check the airbag fabric LAW58 material data that was validated using the picture frame test and biaxial test.

Airbag tethers are normally loaded in uniaxial tension. Uniaxial tension tests should be completed and validated for these materials. Validation of the 45 degree oriented specimens is also important for these materials.

LAW58 materials can include the hysteresis effect in cyclic loading. The stress strain curves for loading and unloading should be extracted from the cyclic biaxial tension test.

Generic LAW58 materials in kg, mm, ms with loading and unloading option with loading and unloading option.

```
#---1----|----2----|----3----|----4----|----5----|----6----|----7----|----8----|----9----|---10----|
/MAT/LAW58/2
Altair test fabric law58
#              RHO_I               RHO_O
                8E-7                   0
#                E1                  B1                  E2                  B2                FLEX
               0.380                   0               0.380                   0                 1.0
#                G0                  GT              AlphaT0                 Gsh           sens_ID
            0.0035              0.0055               7.175                                     1
#                Df                  Ds               Gfrot                            ZEROSTRESS
                  0                   0                   0                                     1
#       N1         N2                  S1                  S2
          1          1                   0                   0
#    Dir 1
       500                1.00
#    Dir 2
       501                1.00
#    Dir12
       502                1.00
#  FCT_ID4   FCT_ID5             Fscale4             Fscale5   FCT_ID6             Fscale6
       600       600                   1                   1       502                   1
/FUNCT/500
stress-strain curve dir 1
#                 X                   Y
0.0000000000e+000                   0
1.0000000000e-002    2.9343543750e-003
2.0000000000e-002    5.2630337500e-003
3.0000000000e-002    6.9235412500e-003
4.0000000000e-002    8.4176677083e-003
5.0000000000e-002    1.0120216458e-002
6.0000000000e-002    1.2102790000e-002
7.0000000000e-002    1.4219608542e-002
8.0000000000e-002    1.6595997708e-002
9.0000000000e-002    1.9202795833e-002
1.0000000000e-001    2.2040856250e-002
1.1000000000e-001    2.5270391667e-002
1.2000000000e-001    2.8711702083e-002
1.3000000000e-001    3.2558656250e-002
1.4000000000e-001    3.6676625000e-002
1.5000000000e-001    4.1031145833e-002
1.6000000000e-001    4.5697987500e-002
```

```
/FUNCT/600
unloading stress-strain curve dir 1, 2
#                  X                   Y
0.0000000000e+000    0.0000000000e+000
2.0000000000e-002    5.0000000000e-004
4.0000000000e-002    1.0000000000e-003
6.0000000000e-002    2.0000000000e-003
8.0000000000e-002    5.0000000000e-003
1.0000000000e-001    1.0000000000e-002
1.2000000000e-001    3.0000000000e-002
1.4000000000e-001    7.0000000000e-002
#---1----|----2----|----3----|----4----|----5----|----6----|----7----|----8----|----9----|---10----|
/FUNCT/501
stress-strain curve dir 2
#                  X                   Y
0.0000000000e+000                     0
1.0000000000e-002    3.7475658333e-003
2.0000000000e-002    6.4746864583e-003
3.0000000000e-002    8.6433156250e-003
4.0000000000e-002    1.0769805833e-002
5.0000000000e-002    1.2923027083e-002
6.0000000000e-002    1.5325416250e-002
7.0000000000e-002    1.8090380208e-002
8.0000000000e-002    2.1231860417e-002
9.0000000000e-002    2.4870956250e-002
1.0000000000e-001    2.8884937500e-002
1.1000000000e-001    3.3246235417e-002
1.2000000000e-001    3.8027027083e-002
1.3000000000e-001    4.3127925000e-002
1.4000000000e-001    4.8519200000e-002
1.5000000000e-001    5.4217816667e-002
1.6000000000e-001    6.0077625000e-002
#---1----|----2----|----3----|----4----|----5----|----6----|----7----|----8----|----9----|---10----|
/FUNCT/502
stress-strain curve dir 12
#                  X                   Y
-1.8803485061e+001   -1.5770466522e-003
-1.8097254117e+001   -1.5114614261e-003
-1.7387852291e+001   -1.4413505922e-003
-1.6675208927e+001   -1.3674267252e-003
-1.5959251230e+001   -1.2915012899e-003
-1.5239904181e+001   -1.2138769816e-003
-1.4517090425e+001   -1.1332172636e-003
-1.3790730171e+001   -1.0497187778e-003
-1.3060741077e+001   -9.6747019039e-004
-1.2327038129e+001   -8.8635494868e-004
-1.1589533515e+001   -8.0307143684e-004
-1.0848136486e+001   -7.1778894110e-004
-1.0102753220e+001   -6.4282044814e-004
-9.3532866620e+000   -5.8468504129e-004
-8.5996363643e+000   -5.2563223081e-004
-7.8416983143e+000   -4.6541585980e-004
-7.0793647491e+000   -4.0614312591e-004
-6.3125239587e+000   -3.6168074948e-004
-5.5410600764e+000   -3.1690734331e-004
-4.7648528535e+000   -2.7145205177e-004
-3.9837774194e+000   -2.2537031107e-004
-3.1977040240e+000   -1.8050370403e-004
-2.4064977614e+000   -1.3620800862e-004
-1.6100182740e+000   -9.1442121733e-005
-8.0811943303e-001   -4.6243379765e-005
                   0                     0
 8.0811943303e-001    4.6243379765e-005
 1.6100182740e+000    9.1442121733e-005
 2.4064977614e+000    1.3620800862e-004
 3.1977040240e+000    1.8050370403e-004
 3.9837774194e+000    2.2537031107e-004
 4.7648528535e+000    2.7145205177e-004
 5.5410600764e+000    3.1690734331e-004
 6.3125239587e+000    3.6168074948e-004
 7.0793647491e+000    4.0614312591e-004
 7.8416983143e+000    4.6541585980e-004
 8.5996363643e+000    5.2563223081e-004
 9.3532866620e+000    5.8468504129e-004
 1.0102753220e+001    6.4282044814e-004
 1.0848136486e+001    7.1778894110e-004
 1.1589533515e+001    8.0307143684e-004
 1.2327038129e+001    8.8635494868e-004
 1.3060741077e+001    9.6747019039e-004
```

```
1.3790730171e+001    1.0497187778e-003
1.4517090425e+001    1.1332172636e-003
1.5239904181e+001    1.2138769816e-003
1.5959251230e+001    1.2915012899e-003
1.6675208927e+001    1.3674267252e-003
1.7387852291e+001    1.4413505922e-003
1.8097254117e+001    1.5114614261e-003
1.8803485061e+001    1.5770466522e-003
#---1----|----2----|----3----|----4----|----5----|----6----|----7----|----8----|----9----|---10----|
```

The input curves in LAW58 should increase monotonically. Stress strain curves in warp and weft direction should have exactly 2 intersection points: at strain 0 and some positive strain. Shear stress versus anisotropic angle curve should have 3 intersection points: at angle 0, one positive and one negative angle. All loading and unloading curves should be specified.

The material parameter *ZEROSTRESS*=1 should be set and a sensor should be used in the material LAW58 definition for the activation of airbag reference geometry. This should be the same sensor which is used to start the first injection in `/MONVOL/FVMBAG1`.

Property Cards for Airbag Fabric

Property `/PROP/TYPE16 (SH_FABR)` should be used with `LAW58`.

Generic `/PROP` card to be used with `LAW58`, kg, mm, ms for airbag with tria elements.

```
#---1----|----2----|----3----|----4----|----5----|----6----|----7----|----8----|----9----|---10----|
/PROP/TYPE16/1
Shell
#   Ishell    Ismstr     Ish3n
         4         4         2
#                hm                  hf                  hr                  dm                  dn
                  0                   0                   0                   0                   0
#        N   Istrain               Thick              Ashear              Ithick     Iplas
         1         0                 0.3                   0                   0         0
#                Vx                  Vy                  Vz       Isk      Ipos
                  1                   0                   0         0         0
#               Phi               Alpha               Thick                   Z       Mat
                  0                   0                 0.3                   0         1
#---1----|----2----|----3----|----4----|----5----|----6----|----7----|----8----|----9----|---10----|
```

For fabric there is no bending stiffness. Therefore the membrane formulation N=1 should be used for the airbag fabrics property. The airbags should be meshed with tria elements with these property options: I_{shell}=4 , I_{smstr}=4, and I_{sh3n}=2

Four node shells can be used from version 2021.2. I_{shell}=12, I_{smstr}=4 should be used in this case.

In the `/PROP/TYPE16 (SH_FABR)` property, the fabric material is defined for each layer. This should be exactly the same material used in the `/PART` card definition.

The initial material directions are determined by specifying global V_x, V_y, and V_s vectors. The vectors are projected on each element of the reference geometry of the airbag and are then rotated by the angle of *Phi*. This gives the first material direction. The second material direction is by default perpendicular to the first direction or can be specified using the *Alpha* input.

Note: The default parameters of the hourglass and damping coefficients h_m, h_f, h_r, d_m, and d_n work well when the quality of the airbag folding is good.

Finite Volume Method Airbag Modeling

In Radioss the standard method for airbag calculation is the Finite Volume Method (FVM).

The FVM models gas flow inside the airbag, including the interaction with internal airbag components which could include internal walls, baffles, etc. The airbag cards, `/MONVOL/FVMBAG1` and `/MONVOL/FVMBAG2` are used to set up the FVM airbag. Uniform pressure airbags can be used for debugging purposes to check unfolding, contacts, and global consistency of gas dynamic parameters. `/MONVOL/AIRBAG1` can be used to activate a uniform pressure airbag for debugging.

Generic /FVMBAG1 card:

```
/MONVOL/FVMBAG1/46
FVMBAG1
#surf_IDex                          Hconv
 666000001                              0
#            AscaleT             AscaleP             AscaleS             AscaleA             AscaleD
                  0                   0                   0                   0                   0
#   mat_ID                                             Pext               T0       Iequi      Ittf
 666400002                                             1E-4                296         1         3
#     Njet
         1
#inject_ID   sens_IDsurf_IDinj
 666400002 666000500 666200211
#fct_IDvel                     Fscalevel
 666000200                             1
#    Nvent  Nporsurf
         1         2
# surf_IDv     Iform               Avent               Bvent                              vent_title
 666000015         1                   0                   0
#             Tstart              Tstop               dPdef             dtPdef             IdtPdef
                  0               1E30                   0                  0                   0
#  fct_IDt   fct_IDP   fct_IDA                       Fscalet             FscaleP             FscaleA
         0         0         0                             1                   1                   1
# fct_IDt'  fct_IDP'  fct_IDA'                      Fscalet'            FscaleP'            FscaleA'
         0         0         0                             1                   1                   1
#surf_IDps   Iformps Iblockage                                                          surface_title
 666000004         2         0
#             Tstart              Tstop               dPdef             dtPdef             IdtPdef
                  0                   0                   0                  0                   0
#  fct_IDv                     Fscalev
 666000004                             1
#surf_IDps   Iformps Iblockage                                                          surface_title
 666000003         2         0
#             Tstart              Tstop               dPdef             dtPdef             IdtPdef
                  0                   0                   0                  0                   0
#  fct_IDv                     Fscalev
 666000003                             1
# frame_ID     kmesh              Tswitch
         0         4                  20
#                L1                  L2                  L3
                  0                   0                   0
#      Nb1       Nb2       Nb3 grbric_ID surf_IDin      Iref
         0         0         0         0         0         0
#   Igmerg                         Cgmerg              Cnmerg              Ptole
         0                        1e-04                1E-4                  0
#                qa                  qb                Hmin          delta_Tsca          delta_Tmin
                  0                   0                   0                   0                   0
#   Ilvout    Nlayer   Nfacmax    Nppmax    Ifvani
         1         0         0         0         0
```

Generic /FVMBAG2 card:

```
/MONVOL/FVMBAG2/46
FVMBAG2
#surf_IDex surf_IDin                Hconv
 666000001                              0
#   mat_ID                                             Pext                T0                  Ittf
 666400002                                            1e-04               296                     3
#     Njet
         1
#inject_ID   sens_IDsurf_IDinj
```

```
 666400002 666000500 666200211
#    Nvent  Nporsurf
         1         2
# surf_IDv     Iform               Avent               Bvent                              vent_title
 666000015         1                   0                   0
#             Tstart               Tstop               dPdef             dtPdef             IdtPdef
                   0                1E30                   0                  0                   0
#  fct_IDt   fct_IDP   fct_IDA                      Fscalet             FscaleP             FscaleA
         0         0         0                            1                   1                   1
# fct_IDt'  fct_IDP'  fct_IDA'                     Fscalet'            FscaleP'            FscaleA'
         0         0         0                            1                   1                   1
#surf_IDps   Iformps Iblockage                                                        surface_title
 666000004         2         0
#             Tstart               Tstop               dPdef             dtPdef             IdtPdef
                   0                   0                   0                  0                   0
#  fct_IDv                      Fscalev
 666000004                           1
#surf_IDps   Iformps Iblockage                                                        surface_title
 666000003         2         0
#             Tstart               Tstop               dPdef             dtPdef             IdtPdef
                   0                   0                   0                  0                   0
#  fct_IDv                      Fscalev
 666000003                           1
#              Cmerg              Tswitch             Iswitch            Pswitch
               1e-04                 20.0                   0               0.0
#                 CFL              DT_min
                  0.0                 0.0
```

The /MONVOL/FVMBAG2 card is available as of 2021.1. It is an exact replication of the /MONVOL/FVMBAG1 card. Obsolete and non-used input is removed from /MONVOL/FVMBAG2. This makes it simple and easy to use. Some differences in the results between the /MONVOL/FVMBAG1 and the /MONVOL/FVMBAG1 card may come when injection velocity in /MONVOL//FVMBAG1 is set different from 300m/s. This value is hard coded in the /MONVOL/FVMBAG2 card.

The closed airbag external surface should be defined using /SURF/PART and reference a set of shell components with normals directed outwards.

The scaling parameters, *Ascale_T*, *Ascale_P*, *Ascale_S*, *Ascale_A*, and *Ascale_D* are not used.

Atmospheric values are defined for initial air material, initial temperature, and initial pressure.

The parameter, I_{ttf} should be set to 3. In this case, vents are activated at TTF of the first injector given in the sensor referenced in /MONVOL/FVMBAG1 and all time dependent parameters and curves controlling vent openings and porosities will be shifted by the sense activation time of the first injector

The number of injectors is specified by the N_{jet} flag. Elements used as an injection surface should be place into separate PART component, which belongs either to the external or internal airbag surface. Different injectors cannot use same elements for injector surfaces.

Time to fire (TTF) to start gas injection should be defined in /SENSOR. Global parameter definition /PARAMETER can be used to parametrize input of the TTF and other airbag parameters.

The function for injection velocity should be set at a constant value of approximately the velocity of sound of injected gas. This function does not have significant impact to simulation results.

Option I_{equil}=1 should be used. The option provides a simplified FVM cycling before TTF.

The function for injection velocity should be set at a constant value of approximately the velocity of sound of injected gas. This function does not have significant impact to simulation results.

Internal surfaces (internal walls, baffles, etc.) shall be defined as a collection of components and referenced in the $surf\_ID_{in}$ field.

It is strongly recommended to close all important inter-chamber openings using separate void components with all nodes attached to fabric component. This should be done before the airbag folding to avoid intersections with other airbag components. By adding these void elements, the finite volumes which belong to different chambers of airbag during deployment simulation will not be merged during the simulation. This also allows the output and control mass flow between different chambers of airbag to be monitored using `/TH/SURF`.

Note: External and internal airbag components, inter-chamber voids and void components that represent venting should not have geometric intersections. The creation of the finite volume mesh is not possible when there is any intersection.

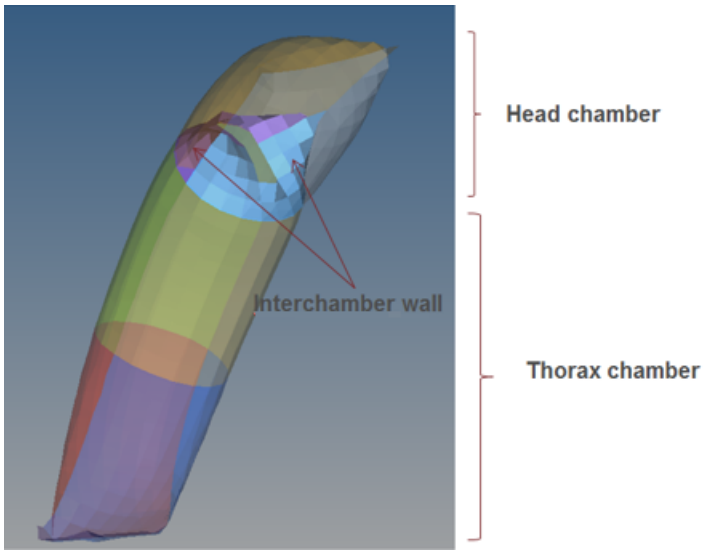


Figure 369: Simple sleeve Airbag with 2 chambers

The Initial Finite Volume mesh is created automatically when the option `/MONVOL/FVMBAG1` *Kmesh*=14 is used and no manually created initial tetra Finite Volume mesh is defined as described in the next section. Next, the airbag is checked to make sure no intersections exist in the airbag external and internal surfaces. If no intersections exist, the Finite Volume mesh is created. If the Finite Volume mesh cannot be created due to intersections in the airbag components, an error message is written with the possible node/element numbers and positions that are causing the problem with creation of Finite Volume mesh. When this happens, HyperMesh can be used to inspect and correct the issues with the airbag mesh. Use the **check 2Dmesh** option in the **tetramesh** panel, with default setting to see if the external and internal airbag components are acceptable for tetra mesh generation. Tetra mesh created by Radioss is output as Radioss include file `MONVOL`.

In case when four node shells are used (it is possible since 2021.2) Radioss performs so called "*smart split*" to split four node shells into three node shells without adding intersections. After splitting, Radioss creates tetra mesh as usual.

Note: "*smart split*" mesh is used only to create the tetra mesh. All simulation is performed using original four node shell and tria shell mesh in this case.

HyperMesh can also be used to manually generate the initial Finite Volume mesh as tetra elements. In this case, all external and internal components of the airbag should be defined when creating the mesh. The "**Fixed trias/quads to tetra mesh**" option should be used to make sure that the original surface mesh is not changed. Use the option tetramesh option "**check 2Dmesh**" with default setting allows to check whether the external and internal airbag components are acceptable for tetra mesh generation.

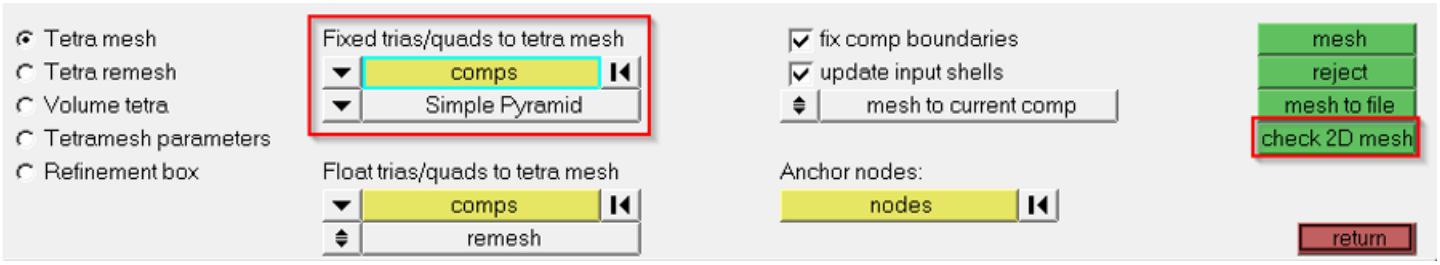


Figure 370:

The initial Finite Volume mesh `/TETRA4` elements should be place into a separate `/PART` and assigned the material `/MAT/VOID` and property `/PROP/VOID`. The mesh must completely fill the airbag volume.

Example of tetra component definition of initial Finite Volume mesh inside of airbag:

```
#---1----|----2----|----3----|----4----|----5----|----6----|----7----|----8----|----9----|---10----|
/PART/8
TETRA4
         8         8         0
/MAT/VOID/8
tetra
/PROP/VOID/8
tetra
#---1----|----2----|----3----|----4----|----5----|----6----|----7----|----8----|----9----|---10----|
```

To make sure that the tetra mesh fills the entire internal airbag volume, verify in the Starter output file that the INITIAL VOLUME OF MONITORED VOLUME = SUM VOLUME OF FINITE VOLUMES listed in the same file. Also, verify that the NUMBER OF FINITE VOLUMES (1568 in present case) is the same as the number of tetras created in HyperMesh.

```
INITIAL VOLUME OF MONITORED VOLUME. . .= 299199.9998912

VOLUME NUMBER          1
NUMBER OF SURFACE POLYGONS. . . . . .=      1200
NUMBER OF SURFACE TRIANGLES . . . . .=      1200
NUMBER OF COMMUNICATION POLYGONS. . .=      2536
NUMBER OF COMMUNICATION TRIANGLES . .=      2536
NUMBER OF FINITE VOLUMES. . . . . .=      1568
MIN FINITE VOLUME VOLUME. . . . . .=  66.66666700000         (FINITE VOLUME ID 292)
INITIAL MERGING VOLUME. . . . . .=  19.08163264612
SUM VOLUME OF FINITE VOLUMES. . . . .= 299199.9998912
SUM AREA SURFACE TRIANGLES. . . . . .=  230031.1439046
SUM MASS OF FINITE VOLUMES. . . . . .= 3.9428379122204E-04
```

If the tetra mesh created in HyperMesh does not completely fill the airbag volume, then Radioss will try to fill the extra volume. When this happens, the following warning message appears in the Starter output file.

```
WARNING ID :           631
** WARNING IN FVMBAG DEFINITION
DESCRIPTION :
-- MONITORED VOLUME ID : 1
-- MONITORED VOLUME TITLE : SAB
IN LOCAL FRAME DIRECTION 1
```

```
GIVEN LENGTH 0.000000000000 IS SMALLER THAN BOUNDING LENGTH 301.4996942325
IT IS RESET TO 304.5146911748
WARNING ID :          631
** WARNING IN FVMBAG DEFINITION
DESCRIPTION :
-- MONITORED VOLUME ID : 1
-- MONITORED VOLUME TITLE : SAB
IN LOCAL FRAME DIRECTION 2
GIVEN LENGTH 0.000000000000 IS SMALLER THAN BOUNDING LENGTH 100.0000000000
IT IS RESET TO 101.0000000000
WARNING ID :          631
** WARNING IN FVMBAG DEFINITION
DESCRIPTION :
-- MONITORED VOLUME ID : 1
-- MONITORED VOLUME TITLE : SAB
IN LOCAL FRAME DIRECTION 3
GIVEN LENGTH 0.000000000000 IS SMALLER THAN BOUNDING LENGTH 1.535460483274
IT IS RESET TO 1.550815088107
```

The quality of tetra mesh is not important because the finite volumes based on the tetra mesh will be merged according to the strategy provided by the merging parameters. I_{gmerg}, C_{gmerg}, and C_{nmerg} can be entered in `/MONVOL/FVMBAG1`. However, it is recommended to deactivate the merging process during Starter initialization by setting C_{gmerg} = C_{nmerg} =1e-04 and use an advanced merging algorithm in the Engine, where the same parameters can be defined in /FVMBAG/MODIF or /DT/FVMBAG/Iflag.

From version 2019.1, merging parameters and time step parameters can be input directly in the `/MONVOL/FVMBAG1` and `/MONOL/FVMBAG2` cards.

If `/MONVOL/FVMBAG1` I_{fvani} =1, then the animation file `*A000` shows the initial Finite Volumes.

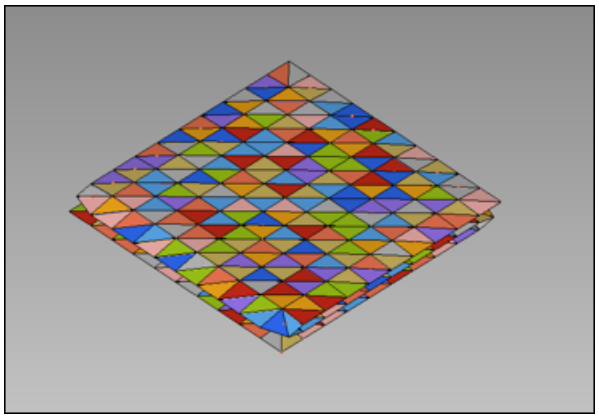


Figure 371: Representation of Initial Finite Volumes in `*A000` File

The number of tetra elements depends on the complexity and the type of airbag. For example, a side airbag can be about 30,000 – 50,000 tetras, and a curtain airbag 250,000 – 500,000 tetras.

The automatic volume meshing option `/MONVOL/FVMBAG1` *Kmesh*=1, should not be used. It is recommended to either use *Kmesh*=14 (HyperMesh tetra mesher) or manually create the tetramesh in HyperMesh.

Vent Holes

Each vent hole should be represented as a separate component in the same position as in the CAD geometry.

The vent hole is modeled using a void material and property. All nodes of the void vent components should be connected to a fabric component. Density, Young's modulus and thickness should be defined for the void components using the same values as the fabric material of the airbag. These values are important for the contact defined between void components which helps to maintain the internal airbag volumes.

Material and property for vent void component:

```
#---1----|----2----|----3----|----4----|----5----|----6----|----7----|----8----|----9----|---10----|
/MAT/VOID/2
Material void
#                RHO                   E
                8E-7                0.38
#---1----|----2----|----3----|----4----|----5----|----6----|----7----|----8----|----9----|---10----|
/PROP/VOID/2
Property void
#              Thick
                0.3
#---1----|----2----|----3----|----4----|----5----|----6----|----7----|----8----|----9----|---10----|
```

If an element is referenced in two different vent or porous surface definitions, then the element will be used in the last definition and removed from the first definition (this limitation was removed in 2017.2.3).

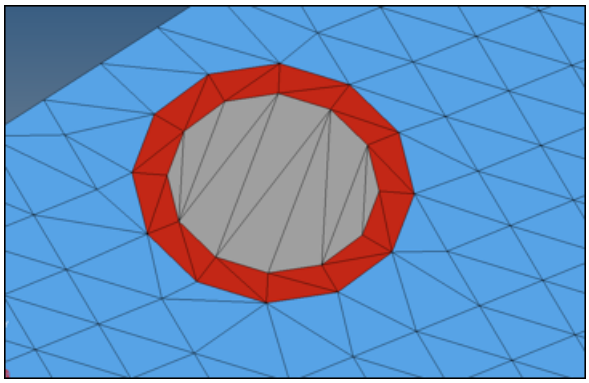


Figure 372: Blue- Airbag Fabric; Grey- Vent Hole; Red - Component That Separates Vent Hole from Rest of Airbag

Definition of isenthalpic vent hole and threshold function to activate the vent hole (kg, mm, ms)

```
/MONVOL/FVMBAG1
…….
# Sid_vent      Ivent                Avent                Bvent
 666000015          1                    1                    0
#              Tstart               Tstop                dPdef              dtPdef             Idtpdef
                1E30                   0                1E-06                   0                   0
#  fct_IDt    fct_IDP    fct_IDA                       Fscalet             FscaleP             FscaleA
         0        123          0                            0                   0                   0
# fct_IDt'   fct_IDP'   fct_IDA'                      Fscalet'            FscaleP'            FscaleA'
         0          0          0                            0                   0                   0
……..


/FUNCT/123
1 vent area scaling function
#                   X                    Y
                   -1                    0
```

```
                0                   0
             1e-06                  0
             2e-06                  1
                 1                  1
```

The vent hole can be activated at T_{start} or when the overpressure defined by ΔP_{def} is reached. To use only T_{start} define a large ΔP_{def} value. To use only ΔP_{def} define a large T_{start} value. If using ΔP_{def}it is recommended to use 1e-06GPa (1% atmospheric pressure).

The Starter outputs a list of elements which belong to each vent hole.

```
ELEM:    46991 <-> SH3N :   55089506 - VENT HOLE:        1
ELEM:    46992 <-> SH3N :   55089507 - VENT HOLE:        1
ELEM:    46993 <-> SH3N :   55089508 - VENT HOLE:        1
ELEM:    46994 <-> SH3N :   55089509 - VENT HOLE:        1
ELEM:    46995 <-> SH3N :   55089510 - VENT HOLE:        1
ELEM:    46996 <-> SH3N :   55089511 - VENT HOLE:        1
ELEM:    46997 <-> SH3N :   55089512 - VENT HOLE:        1
```

During the design process, the variation of the vent hole diameter can be simplified by creating several circular components to fill the vent hole area. The vent hole diameter can then be easily modified by changing the parts that define the vent hole surface. When using this method, the vent holes cannot be modeled using VOID properties because not all of the vent hole elements are attached to the airbag fabric. The simplified orthotropic LAW19 can be also used to represent the fabric material of the circular vents but with a reduced stiffness.

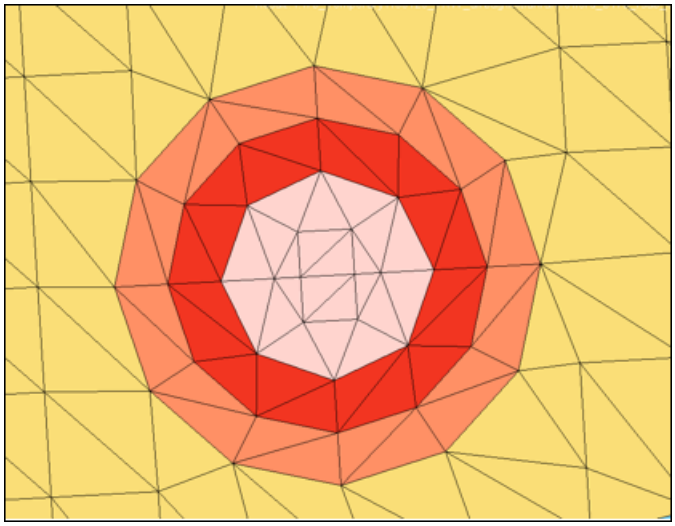


Figure 373: Vent Hole with Variable Diameter

The edges of the slit vent can be connected using stiff springs `/PROP/TYPE4` to provide proper folding of the airbag in the area near the slit vent. The springs can also be used to vary the length of the slit vent. Closed edges of the vent should be modeled using stiff springs to avoid opening.

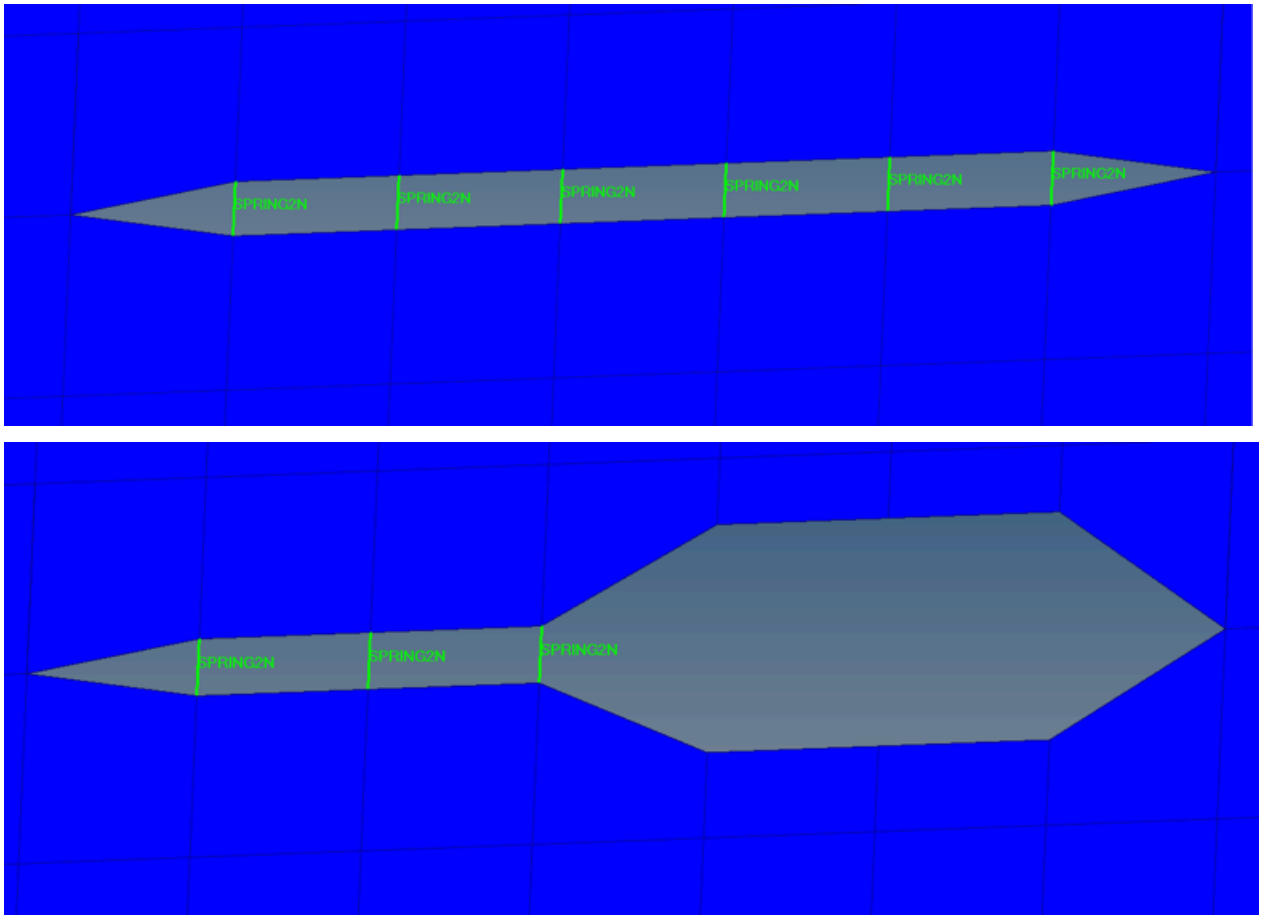


Figure 374: Slit Vent Modeling

Airbag Fabric Porosity

Porosity of airbag material is modeled by the addition of a porosity card `/LEAK/MAT` to `/MAT/LAW58` card.

Generic /TYPE7 and TYPE11 contacts between airbag and housing (kg, mm, ms)

```
#---1----|----2----|----3----|----4----|----5----|----6----|----7----|----8----|----9----|---10----|
/MAT/LAW58/1
Altair test fabric law58
#         Init. dens.          Ref. dens.
                8E-7                   0
#                 E1                  B1                  E2                  B2                FLEX
              0.380                   0               0.380                   0                 1.0
#                 G0                  GT              AlphaT                                    Sensor
            0.0035              0.0055               7.175                                      1
#                 Df                  Ds              Gfrot                                ZeroStress
                  0                   0                  0                                      1
#        N1        N2                  S1                  S2
           1         1                   0                   0
#     Dir 1
         500                1.00
#     Dir 2
         501                1.00
#     Dir12
         502                1.00
#  FCT_ID4   FCT_ID5             Fscale4             Fscale5   FCT_ID6             Fscale6
         600       600                   1                   1       502                   1
/LEAK/MAT/1
Fabric
```

```
#    Ileak             AscaleT             AscaleP
         1
#                AC' fct_IDAC'
#                 LC                  AC
#              1.0e-03             1.0e-03
#---1----|----2----|----3----|----4----|----5----|----6----|----7----|----8----|----9----|---10----|
```

The /LEAK/MAT specifies how the effective porous area of a component having this material will vary as a function time or pressure. It is determined by the value of *Ileakage*.

Ileakage =1:

$$A_{eff} = \sum_n LC \cdot AC \cdot Area_n \tag{264}$$

Ileakage =2:

$$A_{eff} = \sum_n \mathrm{LC}(t) \cdot \mathrm{AC}(P) \cdot Area_n \tag{265}$$

Ileakage =3:

$$A_{eff} = \sum_n \mathrm{LC}(t) \cdot \mathrm{AC}(P - Pext) \cdot Area_n \tag{266}$$

Ileakage =4:

$$A_{eff} = \sum_n \mathrm{LC}(Area_n / Area_0) \cdot \mathrm{AC}(P_{ext} / P) \cdot Area_n \tag{267}$$

Ileakage =5:

$$A_{eff} = \sum_n \frac{Area_0}{L^2}\left[(C_1 \Delta P^{C_2} - C_3)(L - R)^2 + C_3(L\lambda_1 - R/\sqrt{\lambda_2})(L\lambda_2 - R/\sqrt{\lambda_1})\right] \cdot \sin\alpha_{12} \tag{268}$$

$$\Delta P = P / P_{ext} - 1$$

Where, λ_1 and λ_2 are the stretches in warp and weft directions and α_{12} is the angle between warp and weft directions.

Ileakage =6:

$$A_{eff} = \sum_n Area_0(X_0 + X_1 r_s + X_2 r_p + X_3 r_s r_p) \tag{269}$$

with $r_s = Area_n / Area_0$ and $r_p = P_{ext} / P$.

Parts with this material are considered porous, when referenced via a surface ID using the /MONVOL/FVMBAG1 *surf_ID*$_{ps}$ option.

Example of porosity definition in /MONVOL/FVMBAG1.

```
#---1----|----2----|----3----|----4----|----5----|----6----|----7----|----8----|----9----|---10----|
/MONVOL/FVMBAG1/1
DAB
#      Isur
         4
#             Scal_T              Scal_P              Scal_S              Scal_A              Scal_D
                  0                   0                   0                   0                   0
#   Mat_ID                           Mu                Pext                Tphi      Iequi        Ittf
         4                          0.1          0.00010135                   0          0           3
```

```
#      Njet
         1
#    Inj_ID    Isensor surf_IDinj
        10          1         100
#     Nvent       Nps
         0          1
#Sur_id_ps  Iform_ps
         4          1
#              Tstart                Tstop               dPdef              DtPdef              ldtpdef
….
#---1----|----2----|----3----|----4----|----5----|----6----|----7----|----8----|----9----|---10----|
```

The mass flow though the porous material is defined using `/MONVOL/FVMBAG1` $Iform_{ps}$ and has the following formulations:

- $Iform_{ps}$ = 1 $\dot{m}_{out} = A_{eff}\sqrt{2P\rho}\,Q^{\frac{1}{\gamma}}\sqrt{\frac{\gamma}{\gamma-1}\left[1-Q^{\frac{\gamma-1}{\gamma}}\right]}$ (Isentropic - Wang Nefske)
- $Iform_{ps}$ = 2 $\dot{m}_{out} = A_{eff}\rho v(P-P_{ext})$

  Where, v is the outflow gas velocity (Chemkin)
- $Iform_{ps}$ = 3 $\dot{m}_{out} = A_{eff}\sqrt{2\rho(P-P_{ext})}$ (Graefe)

The $Iform_{ps}$ =2 option is normally used to represent fabric porosity. The outflow gas velocity as function of airbag overpressure comes from test data.

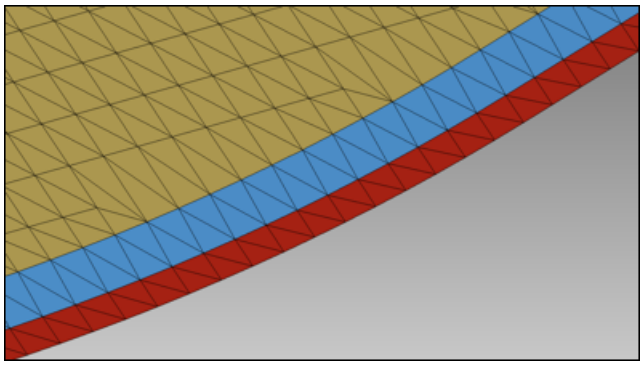


Figure 375: Porous Fabric (Yellow) is Separated from Permeable Seam (Red) by Blue Non-Vent Component

Venting and Fabric Porosity Activation

When there is no sensor to activate the gas injection, the vent holes and porosity become active when $T > T_{start}$, or if the pressure $P > \Delta P_{def}$ value longer than the time given in ΔtP_{def}, where all these values are input using `/MONVOL/FVMBAG1`.

When at least one of the injectors is activated by the sensor, then activation of venting and porosity options is controlled by I_{ttf}.

T_{inj} is the time of the first injector to be activated by the sensor.

I_{ttf} = 0

	Venting, Porosity
Activation	When $P > \Delta P_{def}$ longer than the time $\Delta t P_{def}$, or $T > T_{start}$
Deactivation	T_{stop}
Time dependent functions	No shift

I_{ttf} = 3

	Venting, Porosity
Activation	When $T > T_{inj}$ and $P > \Delta P_{def}$ longer than the time $\Delta t P_{def}$, or $T > T_{inj} + T_{start}$
Deactivation	$T_{inj} + T_{stop}$
Time dependent functions	Shifted by $T_{inj} + T_{start}$

All other related curves are active when corresponding venting, porosity or communication option is active.

Airbag Internal Contacts

External and internal airbag components including inter-chamber voids and void components used for vents should not have any geometrical intersections.

A combination of `/INTER/TYPE7` and `/INTER/TYPE11` contact should be used to represent internal airbag contact.

Generic `/INTER/TYPE7` airbag contact (kg, mm, ms):

```
#---1----|----2----|----3----|----4----|----5----|----6----|----7----|----8----|----9----|---10----|
/INTER/TYPE7/666700001
Airbag Selfcontact - 666700001
#  Slav_id   Mast_id      Istf      Ithe      Igap                Ibag      Idel      Icurv      Iadm
 666100103 666100101         4         0         0                   1         0          0         0
#          GAP_SCALE             GAP_MAX             Fpenmax
                   0                   0                   0
#              STMIN               STMAX          %MESH_SIZE               dtmin
                   1               1E30                   0                   0
#              STFAC                FRIC             GAP_MIN             Tstart               Tstop
                   1                   0                   1                   0                   0
#     I_BC                         INACTI               VIS_S              VIS_F              BUMULT
       000                              6                   0                   0                   0
#    Ifric    Ifiltr               Xfreq     Iform
          0         0                   0         2
/SURF/PART/666100101
Airbag External Surface by Part - 666100101
 666000001 666000002 666000003 666000004 666000005 666000006 666000007 666000008 666000009 666000013
 666000014 666000015 666000016 666000017 666000018 666000201 666000202 666000203
#---1----|----2----|----3----|----4----|----5----|----6----|----7----|----8----|----9----|---10----|
```

For `/INTER/TYPE7` contact:

- All external and internal fabric components including inter-chamber voids and void components for vents should be defined in a self-contact where the components are listed as secondary nodes and main surfaces
- I_{stf}=4 provides correct contact stiffness
- I_{bag}=1 provides vent closure due to contact
- St_{min}=1KN/mm limits smallest contact stiffness
- Gap_{min}=1mm provides a smooth increase of contact force as secondary nodes are approaching main segment
- Friction is set to 0
- *Inacti*=6 deactivates initial penetrations

Initial penetration can be checked with HyperCrash's penetrations checker. The airbag model should be depenetrated up to the physical thickness of airbag fabric. For example, if the thickness of airbag fabric is 0.3mm, the airbag should be depenetrated to 30% of Gap_{min} = 1 value.

Generic `/INTER/TYPE11` airbag contact (kg, mm, ms):

```
#---1----|----2----|----3----|----4----|----5----|----6----|----7----|----8----|----9----|---10----|
/INTER/TYPE11/666800001
Airbag Selfcontact Edge - 666800001
#  Slav_id    Mast_id     I_stf               I_gap    Multimp                 Idel
 666100102 666100102         4                   0         0                    0
#               STmin               STmax           MESH_SIZE               dtmin     Iform    Sens_Id
                    1                   0                   0                   0         2          0
#               STFAC                FRIC                 GAP              Tstart               Tstop
                    1                   0                 0.9                   0                   0
#      I_BC                         INACTI               VIS_S               VIS_F              BUMULT
       000                               6                   0                   0                   0
/LINE/SURF/666100102
Airbag Line by Surface - 666100102
 666100101
#---1----|----2----|----3----|----4----|----5----|----6----|----7----|----8----|----9----|---10----|
```

For `/INTER/TYPE11` contact:

- All external and internal fabric components including inter-chamber voids and void components for vents should be listed both on secondary and main side
- I_{stf}=4 provides correct contact stiffness
- St_{min}=1KN/mm limits smallest contact stiffness
- Gap_{min}=0.9mm provides a smooth increase of contact force as secondary nodes are approaching main segment. This contact gap is 90% of `/INTER/TYPE7` contact gap
- Friction is set to 0
- *Inacti*=6 deactivates initial penetrations

Initial penetration can be checked with HyperCrash penetrations checker. The Airbag model should be depenetrated till 90% of Gap_{min} value which would result in a maximum penetration of 10% of the Gap_{min} value.

Assuming there are no initial intersections in the model, the definition of a TYPE7 and TYPE11 contact prevent node-to-surface and edge-to-edge intersections during airbag simulation. If initial intersections do exist in the model, the model can be unstable because nodes can be stuck on the wrong side of a

main surface. In addition to this, the finite volume mesh cannot be created when airbag components have intersections. Therefore, it is important to remove all initial intersections.

When the `/DT/INTER/DEL` option is used in the Engine file, airbag secondary nodes can be removed from the internal contacts causing local intersections of airbag fabric and decreased accuracy of airbag simulation. When this option is used, define a local *dtmin*=1e-5ms in the `/INTER/TYPE7` and `/INTER/TYPE11` internal contacts definition which will override the global `/DT/INTER/DEL` *dtmin* value.

Note: Model time step should not be limited by the contact kinematic time step.

Airbag Housing

The airbag housing should represent all details available in CAD data.

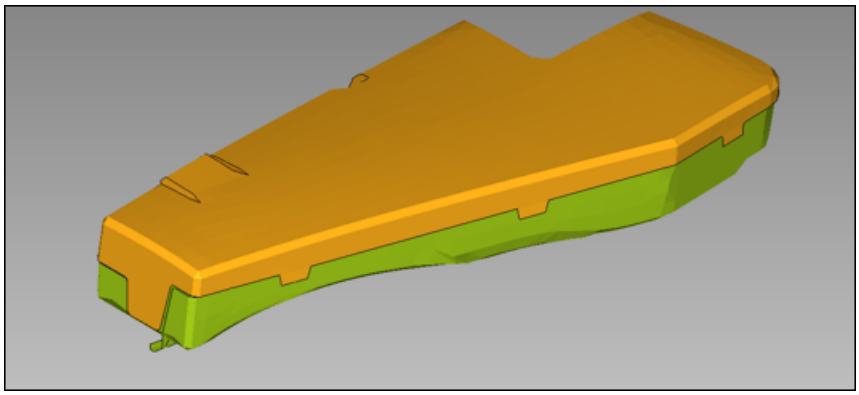


Figure 376: Airbag housing

When the airbag will be included in a full vehicle simulation, the mesh size and quality should be similar to the full vehicle. Material laws `LAW2` and `LAW36` can be used to model elasto-plastic materials. Materials with failure should be used to reproduce the opening of the airbag cover.

Solid foam components should be modeled using `LAW38` or `LAW70`, whereas hyperelastic components are modeling using `LAW42`.

Element properties for shell `/PROP/TYPE1` should use I_{shell}=24 to avoid hourglass effects. For solid foam or hyperelastic components, use `/PROP/TYPE14` with I_{solid}=24 and I_{smstr}=10.

Contact between Airbag and Environment

Contact between the airbag and environment should be separated into specific contacts:

- Airbag to inflator
- Airbag to housing
- Airbag to dummy
- Airbag to seat structure

Generic `/TYPE7` and `/TYPE11` contacts between airbag and housing (kg, mm, ms)

```
#---1----|----2----|----3----|----4----|----5----|----6----|----7----|----8----|----9----|---10----|
/INTER/TYPE7/666710001
Airbag vs. Housing
```

```
#  Slav_id   Mast_id      Istf      Ithe      Igap                Ibag      Idel     Icurv      Iadm
 666100103 666200201         4         0         0                   1         2         0         0
#          Fscale_GAP              GAP_MAX             Fpenmax
                   0                    0                   0
#               STMIN               STMAX          %MESH_SIZE               dtmin
                   1                 1E30                   0                   0
#               STFAC                FRIC              GAP_MIN              Tstart               Tstop
                   1                   .1                   1                   0                   0
#     I_BC                          INACTI               VIS_S               VIS_F              BUMULT
       000                              6                   0                   0                   0
#    Ifric    Ifiltr               Xfreq     Iform
          0         0                   0         2
#---1----|----2----|----3----|----4----|----5----|----6----|----7----|----8----|----9----|---10----|
/INTER/TYPE7/666710002
Housing vs. Airbag
#  Slav_id   Mast_id      Istf      Ithe      Igap                Ibag      Idel     Icurv      Iadm
 666100104 666200202         4         0         0                   1         2         0         0
#          Fscale_GAP              GAP_MAX             Fpenmax
                   0                    0                   0
#               STMIN               STMAX          %MESH_SIZE               dtmin
                   1                 1E30                   0                   0
#               STFAC                FRIC              GAP_MIN              Tstart               Tstop
                   1                   .1                   1                   0                   0
#     I_BC                          INACTI               VIS_S               VIS_F              BUMULT
       000                              6                   0                   0                   0
#    Ifric    Ifiltr               Xfreq     Iform
          0         0                   0         2
#---1----|----2----|----3----|----4----|----5----|----6----|----7----|----8----|----9----|---10----|
/INTER/TYPE11/666810001
Airbag vs. Housing
#  Slav_id   Mast_id     I_stf               I_gap   Multimp                Idel
 666100102 666200203         4                   0         0                   2
#               STmin               STmax           MESH_SIZE               dtmin     Iform   Sens_Id
                   1                   0                   0                   0         2         0
#               STFAC                FRIC                GAP              Tstart               Tstop
                   1                   0                 0.9                   0                   0
#     I_BC                          INACTI               VIS_S               VIS_F              BUMULT
       000                              6                   0                   0                   0
#---1----|----2----|----3----|----4----|----5----|----6----|----7----|----8----|----9----|---10----|
```

Normally two symmetric `/INTER/TYPE7` contacts and one `/INTER/TYPE11` edge to edge contact should be specified. Use the following contact settings for these contacts.

- I_{stf}=4, to provide proper contact stiffness
- I_{bag}=1, for vent closure
- I_{del}=2, to remove deleted elements from contact
- St_{min}=1KN/mm
- I_{form}=2

Time History and Animation Output

It is recommended to request the default (DEF) variable group for time history output `/TH/MONV` which includes:

- Global gas dynamic parameters: mass of gas, volume of airbag, area of airbag, pressure (average), temperature (average) and heat capacity coefficients (average).
- For each vent hole: vent area, outflow velocity, and outflow mass. (Default starting in 2017.2.3)
- Finite volume parameters: number of Finite Volumes (*NFV*) and smallest Finite Volume time step (*DTBAG*). For versions before 2017.2.4, the *NFV* and *DTBAG* options were not included in the default output and thus have to be defined when using older versions.

Generic time history output card with 5 vent holes

```
#---1----|----2----|----3----|----4----|----5----|----6----|----7----|----8----|----9----|---10----|
/TH/MONV/666000001
```

```
Airbag MonVol Time History - 666
#     var1      var2      var3      var4      var5      var6      var7      var8      var9     var10
DEF       NFV       DTBAG
#     Obj1      Obj2      Obj3      Obj4      Obj5      Obj6      Obj7      Obj8      Obj9     Obj10
 666000001
#---1----|----2----|----3----|----4----|----5----|----6----|----7----|----8----|----9----|---10----|
```

The local pressure should be measured in the model by creating a pressure `/GAUGE` using a node of the airbag fabric where the pressure was measured near the injector in the test.

Generic time history output card for pressure gauge

```
#---1----|----2----|----3----|----4----|----5----|----6----|----7----|----8----|----9----|---10----|
/GAUGE/1
FWD
#  node_ID                                        shell_ID                DIST
  50050421
#---1----|----2----|----3----|----4----|----5----|----6----|----7----|----8----|----9----|---10----|
/TH/GAUGE/1
TH GAUGE
#     var1      var2      var3      var4      var5      var6      var7      var8      var9     var10
DEF
#     Obj1      Obj2      Obj3      Obj4      Obj5      Obj6      Obj7      Obj8      Obj9     Obj10
         1
#---1----|----2----|----3----|----4----|----5----|----6----|----7----|----8----|----9----|---10----|
```

The mass flow through any permeable internal airbag surfaces can be output using the `/TH/SURF` card.

Generic time history output for area and massflow through a permeable internal airbag surface

```
#---1----|----2----|----3----|----4----|----5----|----6----|----7----|----8----|----9----|---10----|
/TH/SURF/666000002
Airbag MonVol Time History - 666
#     var1      var2      var3      var4      var5      var6      var7      var8      var9     var10
AREA      MASSFLOW
#     Obj1      Obj2      Obj3      Obj4      Obj5      Obj6      Obj7      Obj8      Obj9     Obj10
 666000001
#---1----|----2----|----3----|----4----|----5----|----6----|----7----|----8----|----9----|---10----|
```

The card should refer to surfaces defined by `/SURF`.

The following cards should be defined in the Engine file for the animation output of gas pressure, density, temperature, and fluid velocities in airbag surface nodes.

`/ANIM/NODA/P` or `/H3D/NODA/P`

`/ANIM/NODA/DENS` or `/H3D/NODA/DENS`

`/ANIM/NODA/TEMP` or `/H3D/NODA/TEMP`

`/ANIM/VECT/FVEL` or `/H3D/NODA/FVEL`

Time Step and Merging Control

The time step and number of finite volumes influence the run time of the airbag simulation model. This section describes the options available to influence the time step and number of finite volumes.

The time step for the finite volumes is based on the initial mesh and calculated as:

$$\Delta t_{fv} = \Delta T_{sca} \cdot \left(\frac{l_c}{v + C} \right) \tag{270}$$

Where,

ΔT_{sca}	Time step scale factor entered in `/DT/FVMBAG/1`, `/MONVOL/FVMBAG1`, or `/MONVOL/FVMBAG2`

l_c	Characteristic length which is the minimum edge length of the initial finite volume tetra mesh
v	Maximum gas velocity
C	Maximum sonic velocity

The characteristic length is calculated in the Starter based on initial FV mesh and does not change during the simulation. The Polyhedra output is for the *Kmesh*=1 mesh generation method which is no longer recommended. The minimum length information listed after the "`NUMBER OF ADDITIONAL BRICKS`" is for automatic mesh generation, *Kmesh*=14 or if the finite volume elements are created manually in HyperMesh.

```
FVMBAG: FINITE VOLUME MINIMUM LENGTH
     ------------------------------------
     VOLUME NUMBER  666000001
     TOTAL NUMBER OF FINITE VOLUMES.. . . . . .=     30610
     NUMBER OF POLYHEDRA . . . . .. . . . . . .=         0
         MINIMUM LENGTH USED FOR TIME STEP. . . .=  0.000000000000
         MINIMUM LENGTH BASED ON VOLUME . . . . .=  10000000000.00
         MINIMUM LENGTH BASED ON NODAL DISTANCE .= 1.0000000000000E+15
     NUMBER OF ADDITIONAL BRICKS. . . . . . . .=     30610
         MINIMUM LENGTH BASED ON VOLUME . . . . .= 0.7039415515349
         MINIMUM LENGTH BASED ON NODAL DISTANCE .= 0.5268862297298
         MINIMUM LENGTH BASED ON VOLUME/AREA. . .= 4.9347603553545E-02
```

Since there is no gas motion, the initial time step is the characteristic length divided by sonic velocity of air at a room temperature. As the injection starts, the gas velocity and gas sonic velocities increase and the time step normally drops during a short phase after the time to fire. In this phase the time step may become less than the target structural time step and thus control the time step of the run. However, after some time (normally 5-10ms) the gas velocity is decreasing and the time step and becomes higher than the structural time step. It is important to minimize the length of time that the airbag time step is the lowest time step in the simulation.

The characteristic length is underestimated in the simulation because it is based on the minimum edge length of the folded airbag. Thus, the time step can be increased by making ΔT_{sca} a value greater than 1 in `/DT/FVMBAG/1`.

Since version 2017.2.4, the FVM time step is output by default when `/MONVOL/FVMBAG1` is used in the model. For older versions, the variable *DTBAG* in `/TH/MONV` must be defined.

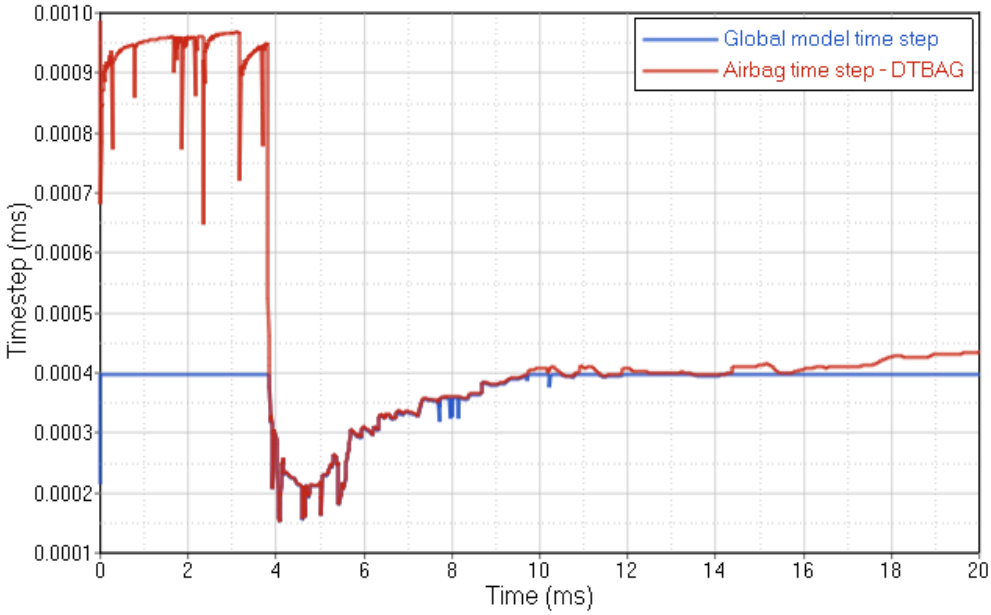


Figure 377: Typical time step evolution for a model with FVM Airbag (red is the global time step, blue is the FVM time step)

FVM airbag is controlling the time step from 3.75ms to 7.5ms)

During the simulation the FV are merged using the these methods.

- Stability merging: this is default merging. A FV is always merged when its volume becomes negative.
- Global merging: a FV is merged if its volume becomes less than factor *Cgmerg* multiplied by the average volume of all the finite volumes which is the airbag volume divided by the number of FVs. The parameter is specified in `/FVMBAG/MODIF` or in `/MONVOL/FVMBAG1`, `/MONVOL/FVMBAG2` (as of 2019.1).
- Neighborhood merging: a FV is merged if its volume becomes less than *Cnmerg* multiplied by the average volume of its neighbor finite volumes. The parameter is normally specified in `/FVMBAG/MODIF` card. This type of merging is difficult to control and therefore it is not recommended.
- Time step merging: a FV is merged if its time step < dtmin defined in `/DT/FVMBAG/1` or in `/MONVOL/FVMBAG1`, `/MONVOL/FVMBAG2` cards (as of 2019.1)

The number of finite volumes reduces quickly during a simulation. However, it is important to have a smooth reduction of number of finite volumes and a sufficient number (1-10% of the initial number) of finite volumes at the end of simulation. The easiest way to control the merging of finite volume is to adjust the *Cgmerg* option. Normally the value should be between 0.01 and 0.1.

Radioss outputs the final number of FVs and number of merged FVs at the end of simulation.

```
** FINITE VOLUME        24 MERGE STATISTICS **
 NUMBER OF REMAINING FINITE VOLUMES .....:      6999
 GLOBAL MERGE ..........................:     22730
 NEIGHBORHOOD MERGE ....................:         0
 STABILITY MERGE .......................:       885
 TIME STEP MERGE .......................:         2
```

Switch from FVM to Uniform Pressure (UP) Approach

The /MONVOL/FVMBAG1 and /MONVOL/FVMBAG2 T_{switch} parameter can be used to switch from a FVM to UP calculation. The UP calculation is less costly and thus saves simulation time. The switch should be performed at a time point when pressure inside of airbag stabilize which occurs when the locally measured pressure becomes the same as the average pressure.

Stability Run

A folded airbag model with the correct material, property, contact definition, and specified reference geometry should not move before activation of injector.

To check stability, set T_{delay}=50ms in /SENSOR/TIME referenced by /MONVOL/FVMBAG1 and run a simulation with the end time of 50ms. The airbag model should not show any significant increase of stresses, strains, and non-physical motion before TTF. If there is motion or stress it is usually caused by contact or some issue with the reference geometry. Do not use damping /DAMP, contact activation, or make the airbag model rigid to prevent airbag motion before TTF.

The airbag housing and inflator components should be included in the stability run along with the contact definition between the airbag and these components.

Uniform Pressure Run

A uniform pressure run should be performed to make sure that gas dynamic data, injector input, fabric materials, and contacts give physical results.

To create a uniform pressure test:

- Replace /MONVOL/FVMBAG1 with /MONVOL/AIRBAG1 keyword
- Remove injector surface from injector definition
- Comment off injector velocity curve input
- Keep all vents as in finite volume airbag
- Comment out information lines with input for finite volume bag meshing

Conversion of /MONVOL/FVMBAG1 to /MONVOL/AIRBAG1

```
#---1----|----2----|----3----|----4----|----5----|----6----|----7----|----8----|----9----|---10----|
/MONVOL/AIRBAG1/1
SAB
#     Isur
         4
#           Ascale_T            Ascale_P            Ascale_S            Ascale_A            Ascale_D
                  0                   0                   0                   0                   0
#   Mat_ID                                              Pext                T_0     Iequi      Ittf
      5000                                          1.0135E-4                296
#     Njet
           1
#inject_ID   sens_IDsurf_IDinj
#     10001        22        19
      10001        22
##fct_IDvel                      Fscalevel
#         4                             0
#    Nvent
           0
##   Iframe
#          0
##                11                  12                  13
#
```

```
##      Nb1       Nb2       Nb3grbric_ID surf_IDin      Iref
#         0         0          0        21        20         0
##   Igmerg                          Cgmerg               Cnmerg               Ptole
#         1                          1e-04                1e-04                   0
##                  qa                  qb                 Hmin
#                                                           0
##   Ilvout    Nlayer    Nfacmax    Nppmax    ifvani
#         0         0          0         0         1
/SURF/PART/4
AIRBAG EXTERNAL SURFACE
         1         2         4         5         6         7
#---1----|----2----|----3----|----4----|----5----|----6----|----7----|----8----|----9----|---10----|
```

Results of simulations should be inspected visually to check for realistic airbag deployment, absence of stuck or hanging nodes due to contact intersections, time step development, energy balance, and energy error.

The airbag parameters, mass, volume, pressure, and temperature should be plotted and compared with the airbag volume and mass of the injected gas physical specifications.

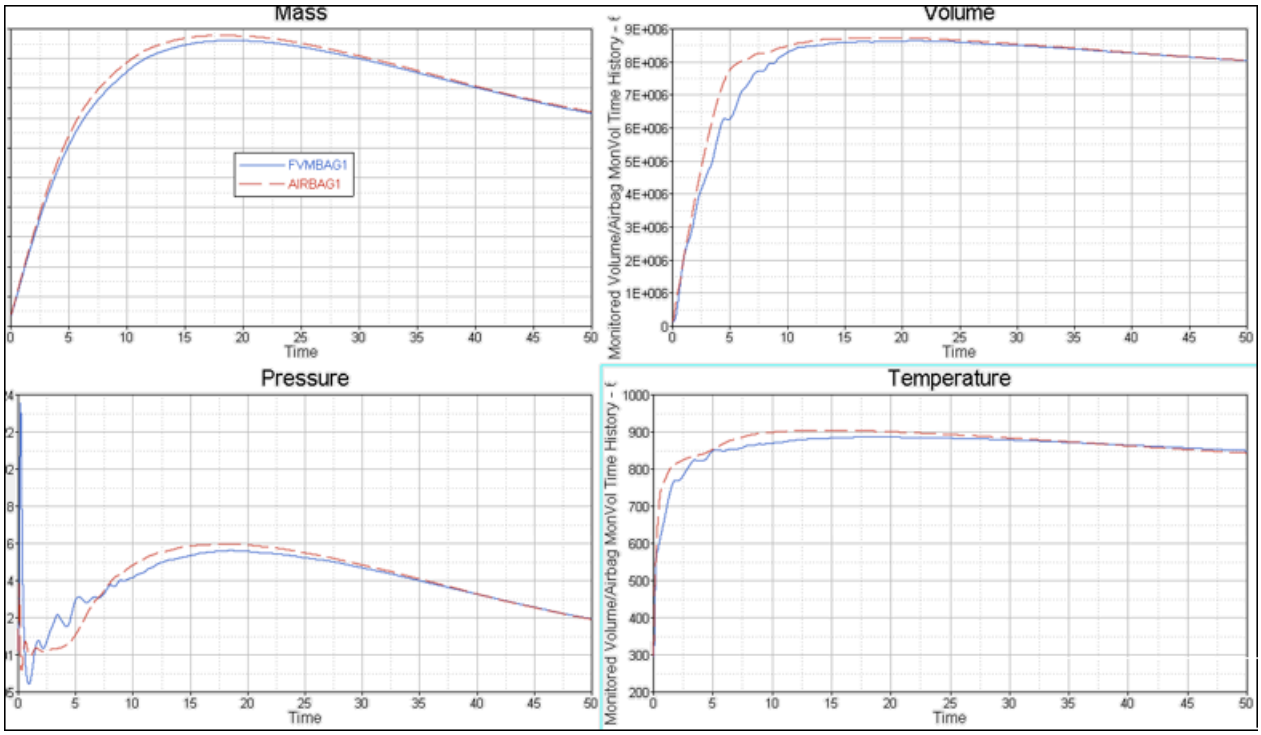


Figure 378: Airbag parameters to control for /AIRBAG1 run

For each vent, the vent area and vent outflow should be plotted and compared to the physical size of the actual vent hole in the airbag.

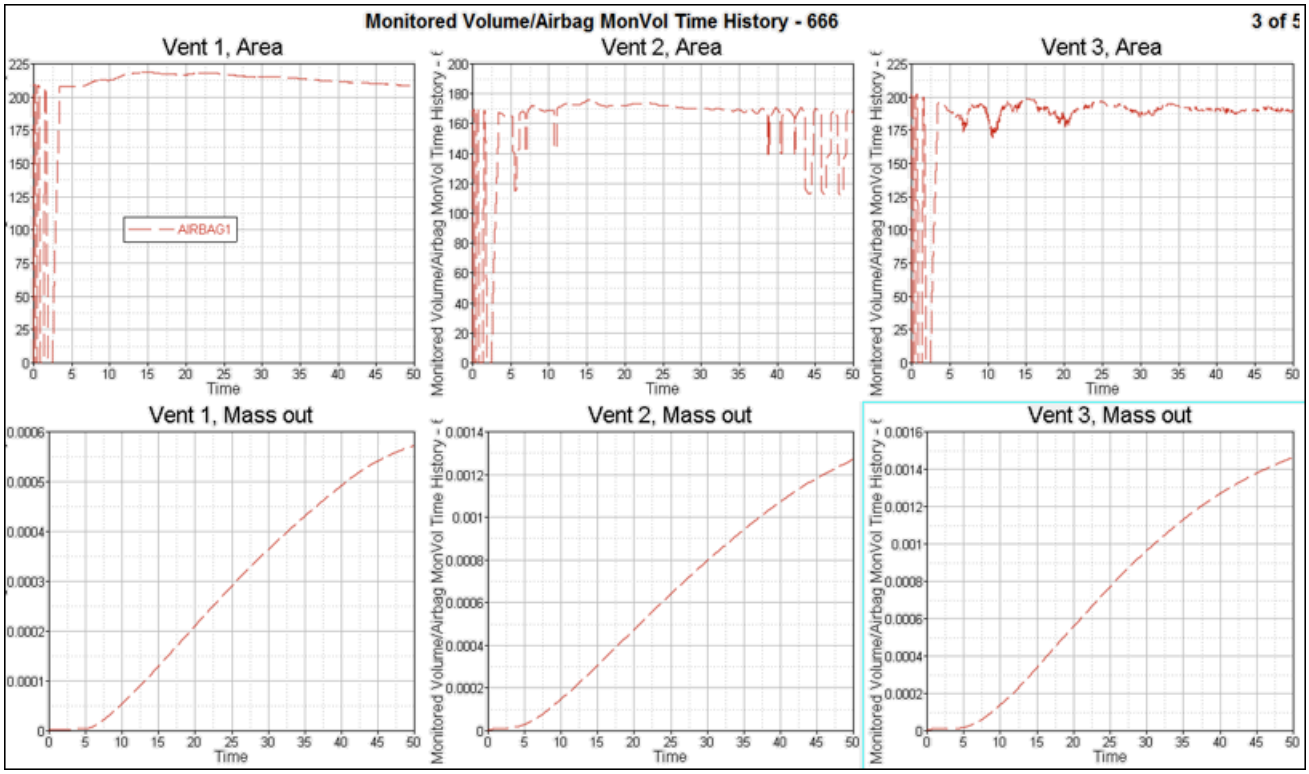


Figure 379:

This data should be used as a basic comparison to the average gas dynamic data from a `/MONVOL/FVMBAG1` simulation.

Standalone Run of FVM Airbag

FVM airbag should be run standalone to verify correct deployment behavior and airbag characteristics. The FV merging parameters and time step control can be modified to reduce simulation time while still providing realistic results.

Typical FV parameters are entered in the Engine file as (can be also added in `/MONVOL/FVMBAG1`, `/MONVOL/FVMBAG2` cards as of 2019.1):

```
/DT/FVMBAG/1
1.5 1.0e-05
/FVMBAG/MODIF
666000001
1  0.010  0.001  1
```

Variation of time step scaling coefficient and global merging parameter may be necessary to provide proper time step and number of FV at the end of the simulation.

Number of finite volumes and FV time step can be plotted in HyperGraph.

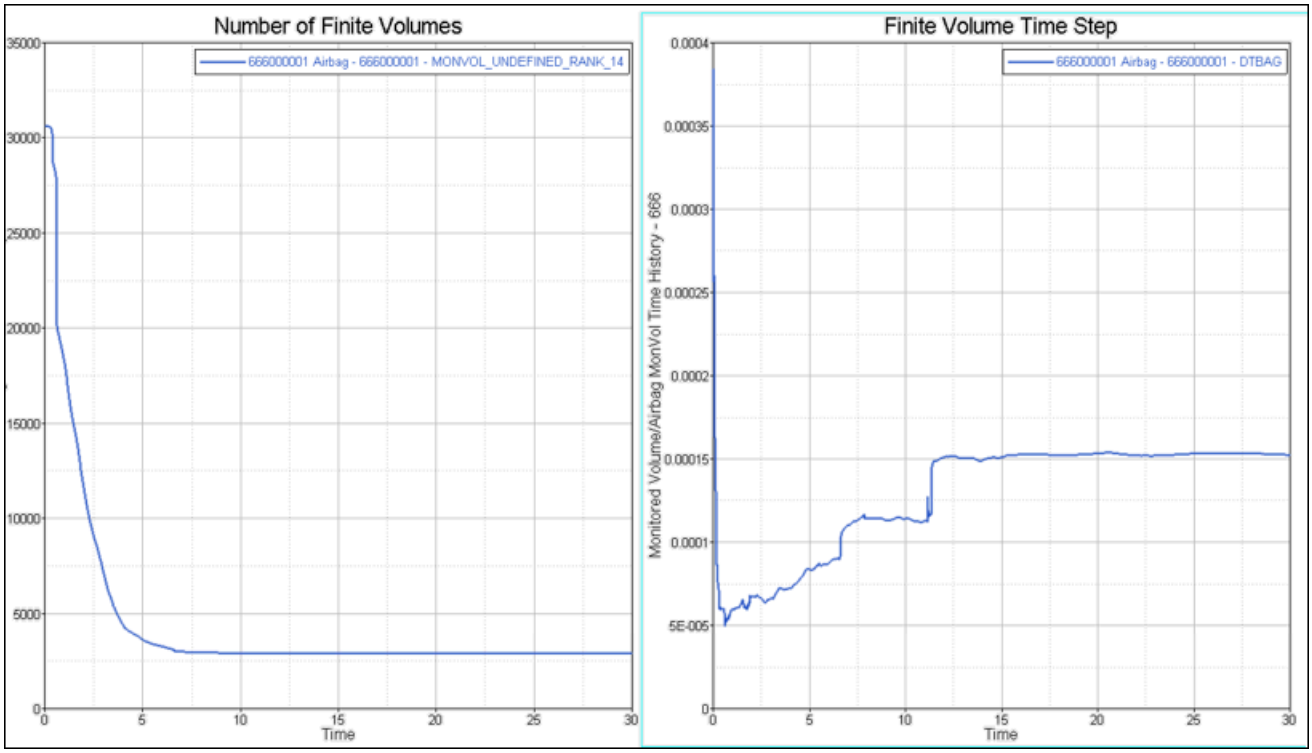


Figure 380:

The number of finite volumes should reduce smoothly to 2-10% of initial number of finite elements. The merging strategy should also provide a time step higher than the minimal time step required in crash simulations.

The merging of FV should checked in Engine output file:

```
    2  0.6688E-03  0.2863E-03 FVBAG  666000001    0.0%   2.739      0.1415E-14   0.000     -0.8103E-19
0.000
** MONITORED VOLUME ID: ********TIME STEP:  0.2878E-03 FINITE VOLUME:   21940

** MONITORED VOLUME ID: ******** - FINITE VOLUME MESH UPDATE **
   NUMBER OF FINITE VOLUMES :     30605
   SUM VOLUME FINITE VOLUMES :    155386.459        (VOLUME AIRBA  155386.459    )
   SUM AREA SURFACE POLYGONS :    227440.887        (AREA AIRBAG   227440.887    )
```

A reasonable number of finite volume are needed and finite volume merging should not cause the number to drop immediately to one finite volume.

The finite volume time step in the `/DT/FVMBAG` card should be less than the initial finite volume time step output in Starter output. Merging parameter *Cgmerg* in `/FVMBAG/MODIF` should not be very high to provide smooth FV merging history.

The sum of finite volume volumes should be equal to the actual airbag volume as:

```
** MONITORED VOLUME ID:         1 - FINITE VOLUME MESH UPDATE **
NUMBER OF FINITE VOLUMES :     1565
SUM VOLUME FINITE VOLUMES :   298121.912        (VOLUME AIRBA 298121.912)
SUM AREA SURFACE POLYGONS :   230093.158        (AREA AIRBAG   230093.158)
```

Every time finite volumes are merged, this message appears in the Engine output.

```
44  0.1320      0.2865E-02 FVBAG          1   1.0%   7.222      0.1104       0.000      0.4265E-01   0.000
** MONVOL ID        1 FINITE VOLUME MESH UPDATE - LOOPING - NUMBER OF FINITE VOLUMES :     1564
** MONVOL ID        1 FINITE VOLUME MESH UPDATE - LOOPING - NUMBER OF FINITE VOLUMES :     1563
** MONITORED VOLUME ID:        1TIME STEP:  0.4293E-02 FINITE VOLUME:     396
```

The gas motion inside the airbag should be reviewed by contour plotting the animation output of the temperature and velocity of the gas. The motion should clearly develop from the inflator into the airbag volume and account for the airbag topology (internal walls, baffles, vent holes, etc.).

Airbag gas parameters should be compared to uniform pressure airbag run. Significant differences should be inspected and explained.

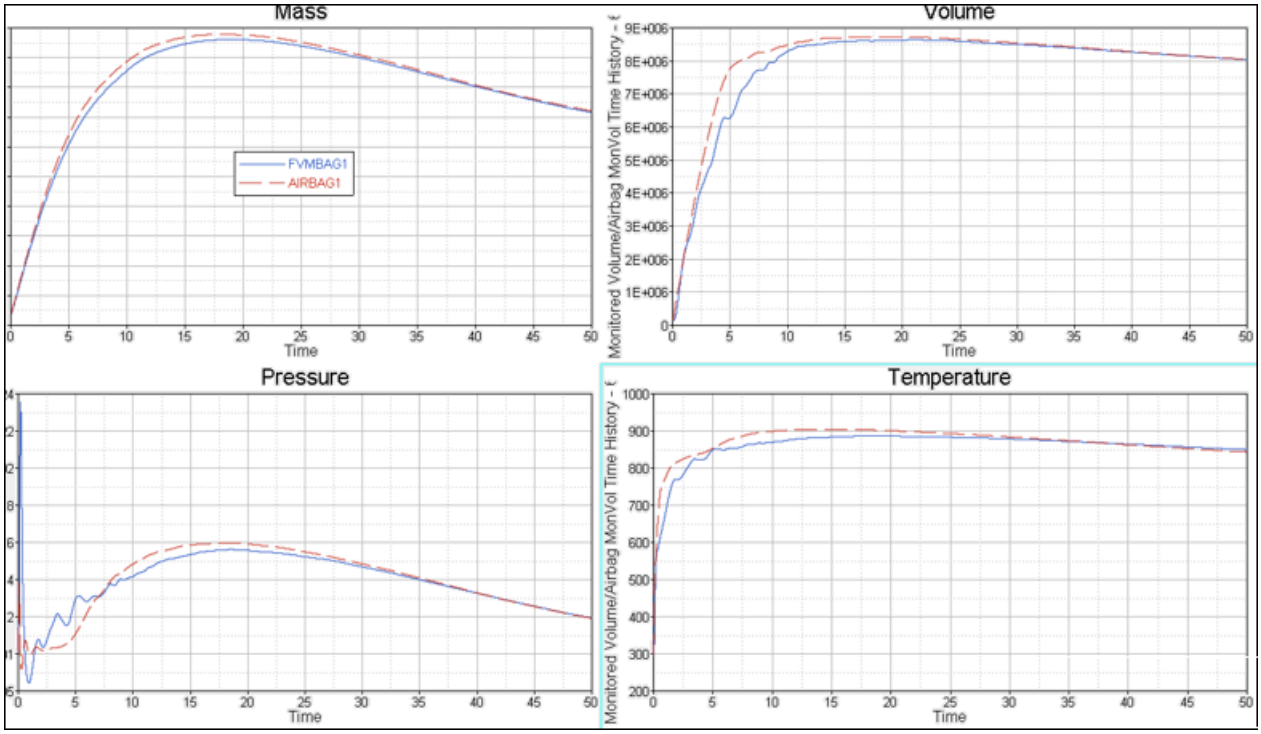


Figure 381: Comparison of airbag output for AIRBAG1 and FVMBAG1 simulations

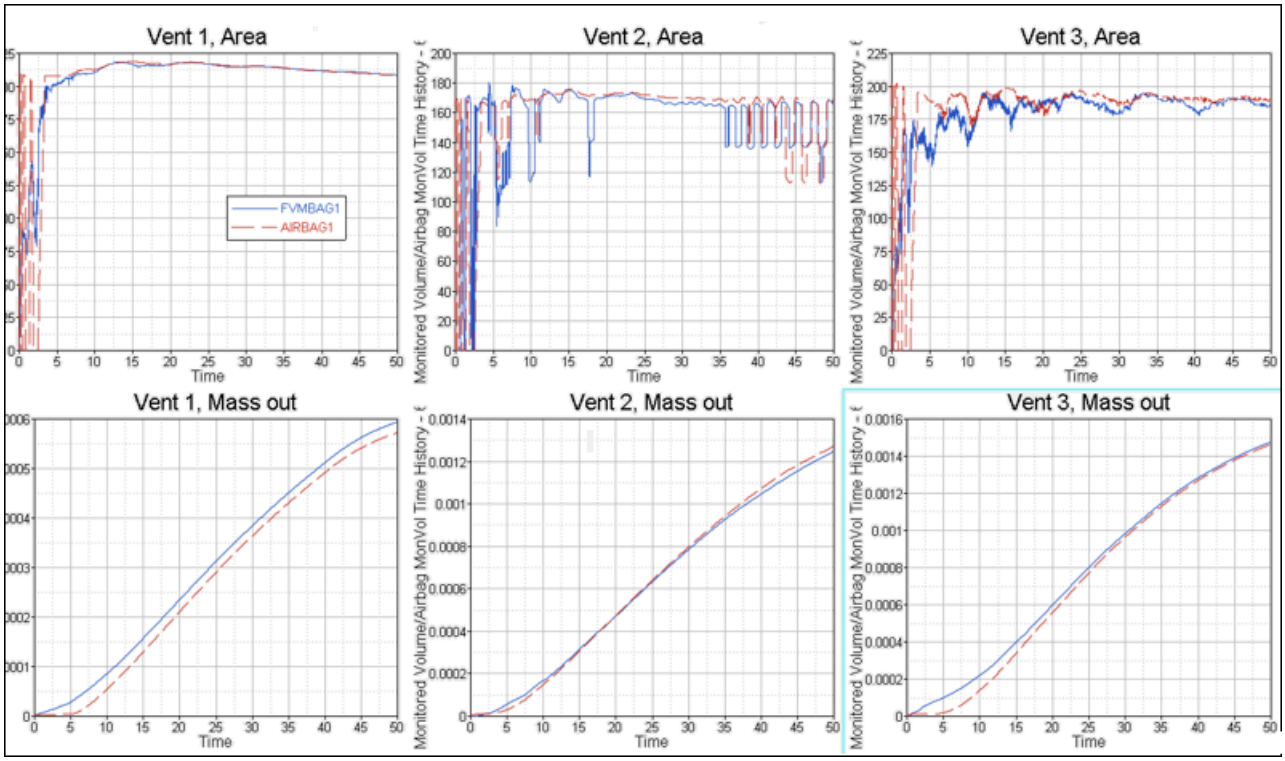


Figure 382: Comparison of venting parameters for AIRBAG1 and FVMBAG1 simulations

The vent area and vent outflow should be plotted for each vent and controlled and compared to uniform pressure run.

Elimination of Bad Volume

Initial tetra shape is not very important for FVM calculation. The volumes are merged with neighbors according to *Cmerge* global merging parameters and *Tscale* parameter I for FVM time step definition.

Checklist

Useful checklist for airbag modeling:

- All airbag fabric components are meshed with tria elements according to meshing requirements (3-4mm average size)
- Airbag is free from intersections
- If the airbag will be used in a larger simulation, the entity ID numbering is correct based on requirements of the larger model
- The airbag model is divided into `/SUBSET` representing the airbag external surfaces, vents, internal surfaces, housing, and inflator
- Mass and inertia of the model matches the physical part
- Fabric material data was validated using biaxial, picture frame and uniaxial tests
- Vent holes are modeled

- Porous fabric is modeled
- Gas material for each gas component is modeled
- Inflator is validated using a tank test simulation and comparing the mass flow and temperature curves to the tank test results
- Gas injections starts at TTF defined in `/SENSOR` T_{delay} not by offsetting massflow and temperature curves
- The mass flow curve start with non-zero values
- $C_p(T)$ function is increasing monotonically for each gas
- Reference geometry is present either as nodal based `/XREF` or element based `/EREF`
- The same `/SENSOR` is used to activate injection and activate reference geometry in LAW58
- Ambientand internal air material properties are specified
- Either the `/FVMBAG1` or `/FVMBAG2` card is specified
- Internal airbag contacts defined and HyperCrash penetration check shows no intersections and a maximum penetration amount of 10% of the contact gap
- Stability run is done to check for motion before TTF
- Uniform pressure run is completed, no anchoring of contact nodes
- Stand-alone FVM run is completed, no anchoring of contact nodes and realistic unfolding
- Model runs with required time step
- Number of FV does not reduce to one
- Visual development of the flow (temperature contour plot, fluid velocity vector) is realistic

Tank Test

By using data from a tank test output, it is possible to obtain the temperature and the mass flow of the gas supplied, which can be used as input to Radioss.

With a tank test it is possible to measure the pressure at the injection point or in the middle of the tank, the two values are equal so the pressure variation is well known. Also, the amount of gas supplied and the characteristics of the gas in the test are known.

For temperature, it is more difficult to get accurate test data because the temperature gauges are often not accurate enough. Therefore, the temperature from the tank test may be wrong.

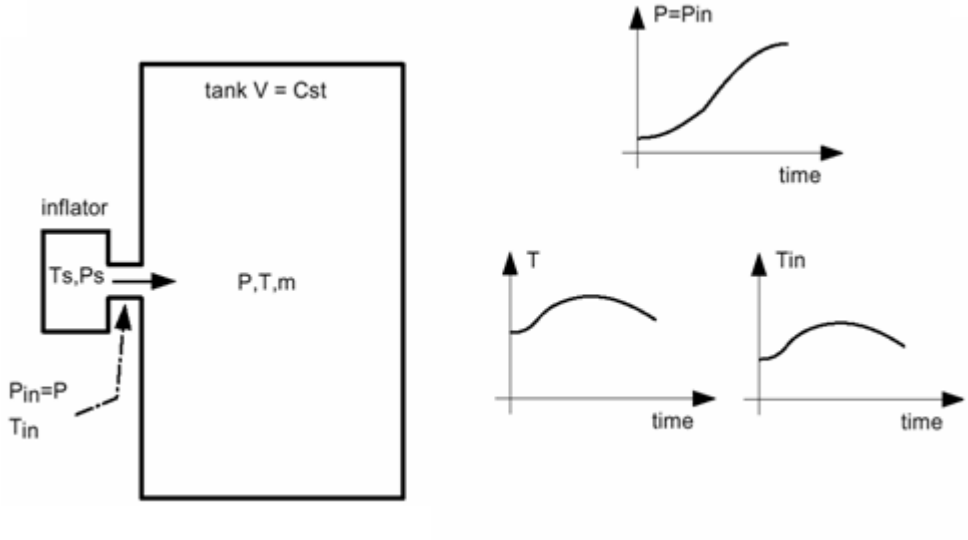


Figure 383: Unknown Mass Flow Rate

The following cases take into account that you may or may not know the injected temperature and the temperature in the tank.

In case the temperature at the injector and in the tank is not known:

- Initial and injected gas composition

  Knowing the molecular weight (MW_i) of each elements and the molar fraction (X_i), it is possible to define the molar weight of the gas (MW):

  $$MW = \sum X_i \cdot MW_i \tag{271}$$

  The average heat capacity, per mass unit of a mixture of gases is given by the Amagat-Leduc equation:

  $$C_p(T) = \frac{\sum m_i \cdot C_{pa}}{\sum m_i} + \frac{\sum m_i \cdot C_{pb}}{\sum m_i} T + \frac{\sum m_i \cdot C_{pc}}{\sum m_i} T^2 \tag{272}$$

Using the previous equation, the heat capacity coefficients ($C_p(T)$) of the initial and injected mixture may be defined.

Knowing the characteristics of the injected gas, the initial gas, and the mixture, it is possible to find the mass flow and the temperature for the inflator. The following basic equations are used to carry out the analysis.

- The perfect gas equation of state is:

  $$PV = nRT, \quad n = \frac{m}{MW} \tag{273}$$

  with $R = 8.314 \frac{J}{mole \cdot K}$

- And the adiabatic equation:

$$H = const. \tag{274}$$

with H being the total enthalpy of the system (inflator + tank).

From the conservation of the energy, the basic energy equation of the tank test can be written as:

$$dE_{airbag} = -PdV + dH_{in} - dH_{out} \tag{275}$$

Here dH_{out} =0, since the tank test is adiabatic. The constant volume of the tank test means dV =0.

Therefore, Equation 275 is summarized as:

$$U_f - U_0 = dH_{in} \tag{276}$$

$$\Leftrightarrow \int_0^{T_{mix}} m_{(in+air)} \cdot C_{V(in+air)} dT - \int_0^{T_{air}} m_{(air)} \cdot C_{V(air)} dT = \int_0^{T_{in}} m_{(in)} \cdot C_{P(in)} dT$$

$$m_{(mix)} T_{mix} \left(C_{pa(mix)} + C_{pb(mix)} \frac{T_{mix}}{2} + C_{pc}$$

$$\left(C_{pa(air)} + C_{pb(air)} \frac{T_0}{2} + C_{pc(air)} \frac{T_0^2}{3} - \frac{R}{MW_{air}} \right) = m_{(in)} T_{in} \left(C_{pa(in)} + C_{pb(in)} \frac{T_{in}}{2} + C_{pc(in)} \frac{T_{in}^2}{3} \right)$$

Inflator Temperature

In Equation 276, the unknown variable is only T_{ini}.

The other variables are known or could be determined using Equation 272 and Equation 273:

- $m_{(in)} m_{(air)} = MW_{air} \frac{P_0 V}{R T_0}$ and $n_{(mix)} = \frac{m_{(in)}}{MW_{in}} + \frac{m_{(air)}}{MW_{air}}$
- for i = a to c: $C_{pi(in)}$, $C_{pi(air)}$, $C_{pi(mix)}$ are calculated with Equation 272, and
- $T_{mix} = \frac{P_{tank} V_{tank}}{n_{(mix)} R}$

Therefore, Equation 276 finds the temperature at the injector T_{in} of the injected gas by iterating on T_{in}. First, the temperature is guessed and six iterations are sufficient to converge to the solution.

Mass Flow

Knowing the evolution of the pressure versus time at the top of the tank test, it is possible to determine the mass flow rate with:

$$\dot{m} = \frac{\partial m}{\partial P} \frac{\partial P}{\partial t} \cong \frac{\Delta M}{\Delta P} \frac{\partial P}{\partial t} \tag{277}$$

With,

ΔP	Total pressure variation during the experiment
ΔM	Total injected mass

Equation 277 may be written if the variation of mass versus the variation of the pressure is a function strictly growing, which is the case.

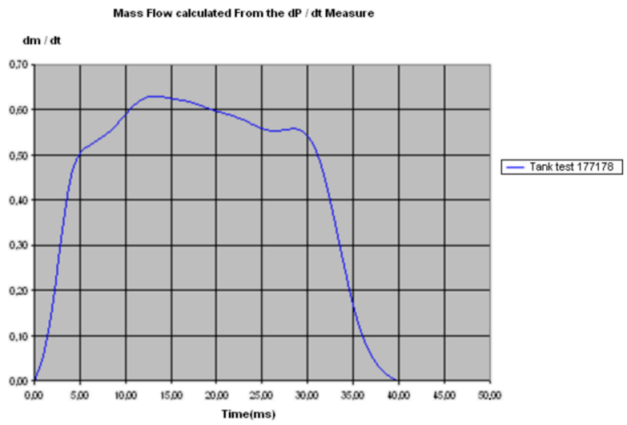


Figure 384: Mass Flow Curve - Evolution of the Mass Flow versus Time

Inflator Gas Velocity

Since the pressure is quickly uniform, the following equation may be written: $P_{in} = P(t)$, knowing T_{in} the density may be expressed as a function of T_{in} and $P(t)$:

$$\rho_{in} = \frac{P(t)}{T_{in} R / MW_{in}} \tag{278}$$

Besides, $V_{in}(t) = \frac{dm(t)/dt}{S\rho_{in}(t)}$, the sound speed into the gas is equal to:

$$c_{in}2 = \gamma \frac{R}{MW_{in}} T_{in} \tag{279}$$

and if $V_{in} > c_{in}$, then $V_{in} = c_{in}$ the velocity is major by the sound speed velocity.

1. If the injector temperature is known:

   $$\frac{dm}{dt} = \frac{v}{\gamma(\gamma-1)C_V T_{in}} \frac{dP}{dt}$$

2. If the temperature in the tank is known:

   $$\frac{dm}{dt} = \frac{V}{\gamma(\gamma-1)C_V T^2}\left(T\frac{dP}{dt} - P\frac{dT}{dt}\right) \text{ or } m = \frac{PV}{(\gamma-1)C_V T}$$

   $$T_{in} = \frac{T^2}{\gamma}\left(\frac{\frac{dP}{dt}}{T\frac{dP}{dt} - P\frac{dT}{dt}}\right) \quad \text{or} \quad T_{in} = \frac{T}{\gamma} + \frac{m}{\gamma}\frac{\frac{dT}{dt}}{\frac{dm}{dt}}$$

3. If the temperature in the tank is constant:

   $$T_{in} = \frac{T}{\gamma}$$

Airbag Setup Recommendation

Airbag Materials

Recommendations for fabric material model setup using LAW19.

LAW19 Option	Recommended Value	Recommended Option Details
R_E	$0.001 < R_E < 0.01$	If $R_E > 0.01$ the compression stiffness is too high.
ZeroStress	1	Airbag pre-strain is applied correctly.
sens_ID	Use the same sensor as used in `/MONVOL` card	Fabric pre-strain through reference geometry must be activated at time to fire.

Recommendations for fabric material model setup using LAW58.

LAW58 Option	Recommended Value	Recommended Option Details
B_1 and B_2	> 0	B_1 and B_2 values should be positive.
Flex	0.01	The default value is adequate.
$G_0 < G_T$	--	Shear parameters should be consistent.
α_T	Enter value in degrees	Shear locking angle shall be entered in degrees.
ZeroStress	1	Airbag pre-strain is applied correctly.
sens_ID	Same sensor that is used in `/MONVOL` card	Fabric pre-strain through reference geometry must be activated at time to fire.

Airbag Properties

Recommendation in fabric properties `/PROP/TYPE9 (SH_ORTH)` and `/PROP/TYPE16 (SH_FABR)`.

Property Option	Recommended Value (per material)		Recommended Option Details
	LAW19 [25]	**LAW58** [26]	
I_{sh3n} for trias	2	2	This is the default tria formulation.

Property Option	Recommended Value (per material)		Recommended Option Details
	LAW19 [25]	**LAW58** [26]	
I_{smstr} for trias	1	4	Best practice recommendation for 3 node trias
I_{shell} for 4-node shells	12	12	This formulation avoids hourglassing.
I_{smstr} for 4-noded shells	11	11	Must be used with I_{shell} =12, 4 node shells.
N	1	1	Enable membrane formulation
Thick	> 0	> 0	Thickness must be defined
I_{plas}	0	0	Not relevant for fabric
I_{thick}	0	0	Not relevant for fabric
Vx, *Vy*, *Vz*	Non-zero vector	Non-zero vector	This input defines a vector which determines first orthotropy direction. If not defined the Global X- axis is used.
I_{dril}	0	0	Not relevant for fabric

Airbag Self-Contact

If all intersections can be removed from the airbag, then surface to surface and edge contact and be defined using `/INTER/TYPE19`. In some advanced cases, it is better to define the two separate contacts, `/INTER/TYPE7` for node to surface and `/INTER/TYPE11` for edge contact.

Interface Options	Recommended Value for contacts TYPE7, TYPE11 and TYPE19	Comments
Intersections	No intersections	No intersections are allowed.
Penetrations	Penetrations < 95% gap	All penetrations > 95% gap need to be removed.
I_{stf}	4	Interface stiffness is the minimum of the main and secondary stiffness.
St_{min} (kN/mm)	1	Avoids contact that is too soft.
I_{del}	0	Not relevant for airbag self-contact.

Interface Options	Recommended Value for contacts TYPE7, TYPE11 and TYPE19	Comments
I_{gap}	0	Constant gap is used as defined by Gap_{min}.
Gap_{min}	1.0 mm	For TYPE11, use Gap_{min} =0.9mm
Inacti	6	Automatically remove initial penetrations by reducing the contact gap.
Fric	0	No friction used in airbag self-contact.
I_{bag}	1	Block vent holes and porous fabrics during contact.

If it is not possible to remove intersections, then `/INTER/TYPE23` can be used with these recommended values.

Interface Options	Recommended Value for TYPE23	Comments
Penetrations	Penetrations < 95% gap	All penetrations > 95% gap need to be removed.
I_{stf}	4	Interface stiffness is the minimum of the main and secondary stiffness.
I_{del}	0	Not relevant for airbag self-contact.
I_{gap}	0	Constant gap is used as defined by Gap_{min}.
Gap_{min}	Thickness of fabric material	This value determines both contact gap and contact depth, too high values will mean that intersected nodes will not be released due to too high contact depth.
Inacti	6	Automatically remove initial penetrations by reducing the contact gap.
Fric	0	No friction used in airbag self-contact.
I_{bag}	1	Block vent holes and porous fabrics during contact.

25. Fabric material LAW19 is only compatible with `/PROP/TYPE9 (SH_ORTH)`.

Airbag Deployment Debugging

The objective here is to provide some guidelines on how to troubleshoot a simulation where the airbag does not deploy properly or crashes because of the airbag.

There are many possible reasons for the deployment to fail or the simulation to crash because of the airbag. Here, the most common issues are the focus.

Unit System

Check the density and modulus of the different parts and verify that they are in the same unit system. For example, Young's modulus, for plastics and metallic parts should be in range of:

0.01 GPa < E < 210 Gpa

These values can be checked in HyperCrash from the contour check located in the **Quality** pull-down menu.

Incorrect Contact Definition

Missing Contacts

To prevent the airbag from getting caught/locked during the deployment, define the following contacts:

- Airbag self-contact for surface and edge contact.
  - TYPE19
  - Alternative for some models that require different settings for surface and edge contact define TYPE7 + TYPE11
  - Refer to Airbag Setup Recommendation for contact
- Airbag to environment (IP, Steering wheel…)
  - TYPE7: airbag is secondary, environment is main
  - TYPE7 symmetric contact: environment is secondary, airbag is main
  - TYPE11: airbag is secondary, environment is main
  - Alternative is to define a single TYPE19 with airbag as secondary and environment as main
  - Setting I_{del}=1 (critical) when failure is defined

Penetration and Intersection

Intersections will prevent correct deployment of the bag and should be removed. There should be no intersection on the self-impact contact of the bag or between the bag and the environment. Intersection can be checked in HyperMesh or HyperCrash.

Penetrations are common in the self-contact of a folded airbag. The ideal case would be to have no penetration at all. The flag *Inacti*=6 adjusts automatically reduces the gap to prevent initial contact force caused by penetrations. However, after the automatic reduction, there still needs to be an adequate gap. Therefore, the maximum penetration should be less than 95% of the contact gap. A penetration check can be done in the HyperCrash or HyperMesh penetration checking tool.

26. Fabric material LAW58 is only compatible with `/PROP/TYPE16 (SH_FABR)`.

Sensor and Reference Geometry

The sensor defining the injection start time for the airbag (TTF), `/MONVOL` *sens_ID* should also be referenced in the material LAW19 or LAW58 *sens_ID*. This sensor allows the solver to know when the reference geometry should become active.

If the same *sens_ID* is not used, then unrealistic deformation of the airbag, after the first few milliseconds of the deployment and erratic behavior, can occur.

Incorrect Material

Check the recommended values in Airbag Materials.

Incorrect Property

Since airbags geometry is very complex, trias (3-node shell) are also commonly used. Some recommendations for properties used for airbag are:

- `/PROP/TYPE9` (for LAW19)

  I_{sh3n}=2; I_{smstr}=1; N=1
- `/PROP/TYPE16` (for LAW58)

  I_{sh3n}=2; I_{smstr}=4; N=1

Incorrect Reference Geometry

If the simulation fails after the first few cycles and the initial internal energy is very high, it could be due to an incorrect or corrupted reference geometry. The reference geometry can be displayed in HyperCrash. Check that all elements of the airbag are included, and they do not look distorted.

If after 50ms the airbag is not smooth (Figure 385), check if the reference geometry is defined on all the parts of the airbag.

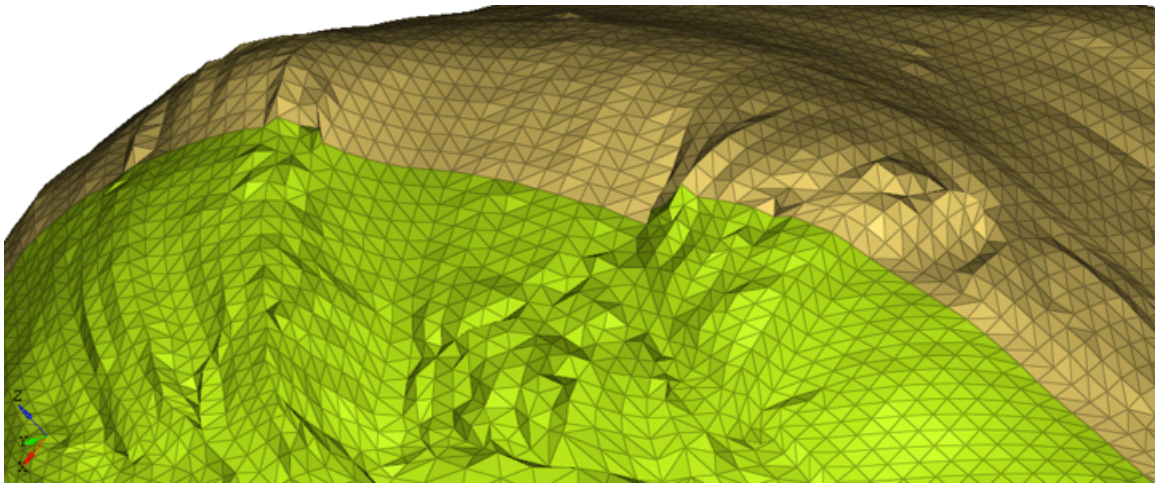


Figure 385:

Incorrect Boundary Condition / Initial Velocity

Check if there is a boundary condition applied on the canister of the airbag. In a sled/full vehicle simulation, this may prevent the airbag from moving with the vehicle. Those boundary conditions may come from another airbag model simulation.

When running a sled case with an initial velocity, remember to check that the airbag is included. A good practice in this case is to isolate the vehicle and verify that all the nodes have an initial velocity > 0 at t=0.

Internal Nodes in Parts with VOID Materials

Parts that have a void material assigned should not have any internal nodes as VOID material has no stiffness. Void material is used sometimes to define communication surfaces or vents.

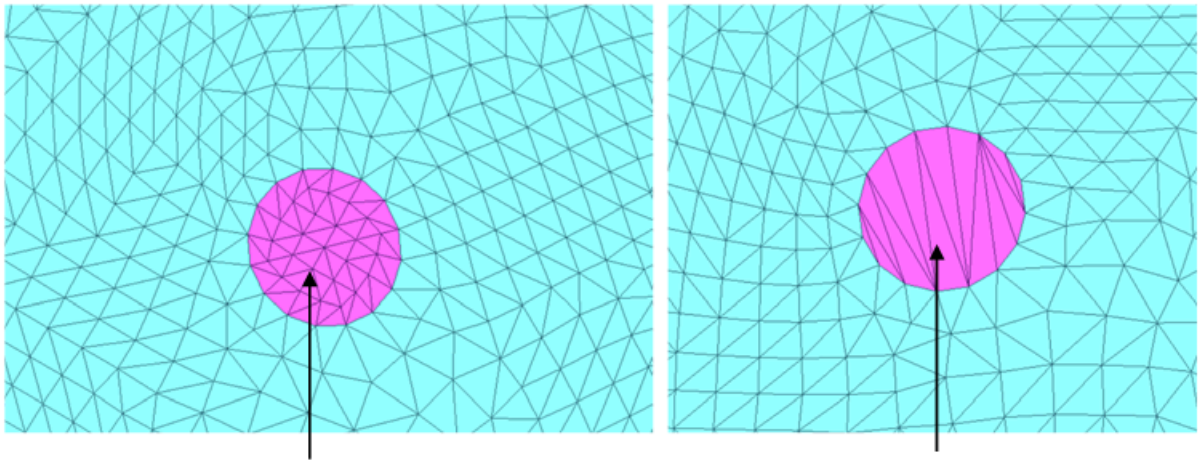


Figure 386: Left: Void Material with no Internal nodes; Right: Incorrect Void material has internal nodes

Incorrect Normal Orientation

For a monitored volume, the normal of all the segments which defined the surface must point outward. Otherwise, solver will print the error message in the Starter output file.

```
ERROR ID :           8
** ERROR IN MONITORED VOLUME DEFINITION
DESCRIPTION :
     --MONITORED VOLUME ID :     6620000
     --MONITORED VOLUME TITLE :     PAB
OPEN VOLUME OR WRONG ELEMENT ORIENTATION
SOLUTION :
   VOLUME MUST BE CLOSED AND NORMALS OUTWARD-ORIENTED
```

Check the normal of the elements that define the monitored volume in HyperCrash or HyperMesh and adjust appropriately.

> **Note:** When the reference geometry is defined with `/EREF`, as changing the normal of the elements may be dangerous, as it will create a disconnect between the element connectivity in the reference and folded/scaled mesh.

Instead of changing the normal of the elements, a better solution is to ask the solver to reverse the normal when generating the surface of a particular component.

In the example below, the solver will reverse the normal of Part ID = 1.

```
/SURF/PART/19
Airbag
       -1          2          3          5          8          9
```

Airbag Time Step, /DT/FVMBAG/1

In the Engine file, time step treatment `/DT/FVMBAG/1` should be present.

For example:

```
/DT/FVMBAG/1
0.9 0.0002
/FVMBAG/MODIF
#monvol_ID
2000
# igmerg cgmerg cnmerge ilvout
1  0.01000   0.001000   1
```

Static Deployment Test

Before applying the airbag model in your vehicle model, it is recommended to first run a static deployment test.

For this test, use:

- The time step Engine file parameters (`/DT/NODA/CST`; `/DT/INTER/DEL`, …) that you will be using in the full vehicle/sled simulation
- The version of the solver that you will be using in the full vehicle/sled simulation

Verify that the airbag deploys properly. Check mass added, volume, pressure, gas mass and gas mass out to verify that the airbag deploys as expected. If the airbag does not deploy as expected here, there is little chance it could deploy well in full vehicle.

Debugging Guidelines

Car crash models are generally complex with a great number of elements, materials, properties, springs, interfaces, and so on. If the numerical results obtained do not match the physical behavior, or if the computation stops with error messages, it is not always easy to find the origin of the modeling problem. A method is needed to check models when troubles occur.

The following guidelines help to correct a majority of problems; but not all of them. Complex models with simplifying modeling assumptions are difficult to correct and require an experienced user.

Potential problems in complex models can be tracked and resolved in three steps:

- Before computation
- After Radioss Starter compiling
- During or after computation by help of post-processors

The following sections describe the debugging method.

Model Preparation

Dedicated Altair pre-processor tools for crash integrate a large number of model checkers. These tools also include automatic correction algorithms. The most common checks before running a model are related to:

- Mesh quality
- Spring definition
- Intersections and penetrations in interface definition
- Checking the incompatible kinematic conditions:
  - No common secondary nodes between rigid bodies
  - Only "free" main node
  - Spherical inertia for the small ones
- Mass and center of gravity
- Thickness of the parts
- Initial velocity value and direction
- Rigid wall normal
- Material law: units system consistency
- Free nodes
- Monitored volume:
  - Closed volume
  - Positive volume
- Connectivity between parts
- Mesh sizes between connected parts

Some basic rules to create a valid model are presented.

Mesh Quality

The Mesh Recommendations discussed in Finite Elements should be respected. The shell mesh must be as homogenous as possible. It is not recommended to use different shell formulations for a given physical part. For very fine meshes, it is recommended to use fully-integrated elements or a physical stabilization method (`/PROP/SHELL`). Triangle shells are avoided. If the mesh includes triangles due to a difficult mesh, it is recommended to reduce its number to 5% per part.

Material Check

Some common material check questions asked are:

1. Is the material in the right unit system?
   - For metallic parts:

     40 GPa < Modulus < 210 GPa

     1.8e-6 kg/mm^3 < Density < 7.8e-6 kg/mm^3
   - For plastic parts (PP, PC-ABS, PP GF30%...):

     0.9 GPa < Modulus < 13 GPa

     9e-7 kg/mm^3 < Density < 1.6e-6 kg/mm^3
   - For foam:

     1e-8 kg/mm^3 < Density < 1e-7 kg/mm^3
2. Is it a proven material from the material database?

   Realistic material data needs be input.
3. Check for these common problems in material.
   - Negative slopes in stress-strain curve
   - Elastic material being assigned to deformable parts
   - Unrealistic yield stress (> 2 GPa)
   - Failure is not defined and elements of plastic or metallic parts are stretching unrealistically (plastic strain > 1)

   Some of the checks mentioned above can be performed in HyperCrash.
   - Unit consistency can be done with the contour check in the quality panel.
   - The check "`Part is not integrated in rbody for LAW1`" finds parts which are deformable and using material LAW1, which can become unstable, if the deformation becomes large.

Figure 387:

Solid and Shell Definition

Some materials it is recommended to use a special option in shell or solid.

1. For parts using elastic-plastic material law (LAW2, LAW27, LAW36, etc.).
   - I_{shell}=24, I_{smstr}=4, I_{plas}=1, I_{thick}=1, N=5
2. For solid parts using elastic-plastic material law (LAW2, LAW27, LAW36, etc.).
   - For hexahedral elements:

     I_{solid}=24, I_{smstr}=4
   - For first order tetra elements:

     I_{solid}=1, I_{tetra}=0 or 1
   - For second order tetra elements:

     I_{solid}=1
3. For solid parts using hyperelastic material (LAW42, LAW69, etc.)
   - For hexahedral elements:

     I_{solid}=24, I_{smstr}=10, I_{cpre}=1, I_{HKT}=2
   - For tetra elements:

     I_{solid}=1, I_{smstr}=10
4. For foam:
   - Material LAW38:

     I_{solid}=24, I_{smstr}=10
   - Material LAW70:

     I_{solid}=1 or 17, I_{smstr}=1 or 11
   - For foam modeled with tetra elements, I_{tetra} should not be set to 1 as foam is highly compressible.
5. For fabric:
   - LAW19 tria elements:

     I_{shell}=4, I_{smstr}=1, d_m=0.2, N=1

- LAW58 tria elements:

  I_{shell}=4, I_{smstr}=4, d_m=0.2, N=1

This can also be checked with HyperCrash Model Checker.

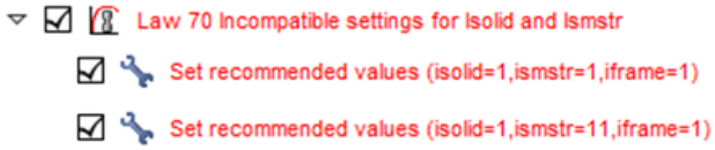


Figure 388: LAW70

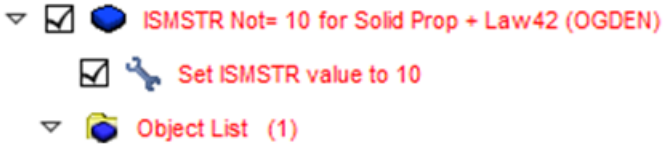


Figure 389: LAW42

Spring Definition

Non-physical or bad definition of springs is a common problem in crash models. In fact, the properties of springs must be consistent in mass, inertia, stiffness and length. Actually, a spring must have a physical behavior:

- With regard to mass distribution over a one-dimensional bar, the inertia will have upper and lower limits:

  $$\frac{ml^2}{12} \le I \le \frac{ml^2}{4} \quad (280)$$

  The lower limit is the inertia of a uniform mass distributed bar. The upper limit gives the inertia of a bar with two extremity masses m/2. As the springs are also used to model bolts and spot welds, you can go beyond these limitations and write:

  $$0.01ml^2 \le I \le 100ml^2 \quad (281)$$

- The cross-section of a spring can be computed either by $S = \frac{Kl}{E}$ or $S = \frac{m}{\rho l}$. If the ratio between the two computed values is greater than 100, the inconsistency may result some trouble.
- In the spring property, for a negative strain the force must be negative and for a positive strain, it must be positive. Otherwise, the spring generates energy during computation.
- If the tangent stiffness is negative for a nonlinear elastic spring, there is a risk of instability especially if vibrating in the negative slope zone (energy error may become positive and increase).
- If the maximum slope of the curve (so the maximum stiffness) is greater than the initial stiffness, unloading in the zone of maximum slope will be false (see Modeling Tools).
- Very high stiffness in spring elements can cause low timestep and instability.

  For example, the following check in HyperCrash allows to identify the spring element that have unusually high stiffness in tension (1000 kN/mm) and rotation (10000 kN.mm/rad).

Note: Property that is referenced in parts of the dummy / barrier models should be ignored.

Figure 390:

Beam Consistency

In addition to the beam assumption length/depth > 10, the following recommendations define a consistent beam:

$$L > \sqrt{A} \tag{282}$$

$$0.01A^2 < I_y < 100A^2$$
$$0.01A^2 < I_z < 100A^2$$
$$0.1(I_y + I_z) < I_x < 10(I_y + I_z)$$

Intersections and Penetrations in Interfaces

Initial mesh intersections create unrealistic connections. Moreover, this can cause locking situations leading to computation failure.

Initial penetrations are due to the interface gap definition. They lead to unrealistic and uncontrolled internal forces which may cause local plastic strains in the beginning of the computation. The structure is then locally less stiff.

Altair pre-processor tools can be used to detect and remove intersections and initial penetrations. A model containing many initial penetrations cannot be considered as valid. After corrections, if a few number of initial penetrations remain, the flag *Inacti* =1 can be activated in `/INTER/TYPE7`. In this case, the interface stiffness for the nodes initially penetrated is deactivated.

If only small initial penetrations remain (less than 5% of the gap), the variable gap in time can be used by setting the flag *Inacti*=5; which is better than *Inacti*=1.

- Best practice (recommended):
  - For interface TYPE7, 11 and 19:
    - I_{stf}=4, St_{min}= 1, I_{del} = 1 or 2, *Inacti*=6, I_{form}=2, Gap_{min} ≥ 0.49
    - There is no intersection in the areas where the simulation fails
    - Penetrations are not too deep (check residual distance in HyperCrash/HyperMesh). It should not be below 0.1mm.
  - For interface TYPE2:

- Use $Spot_{flag}$= 28 if main is shell or secondary node has rotational DOF and main does not (solid)
- Use $Spot_{flag}$= 27 if both secondary and main do not have rotational DOF (solid)
- I_{del} should be set to 1 for all cases

Rigid Body Definition

The following recommendations should be pointed out:

- Never use a node of the mesh as a main node for a rigid body
- A rigid body made of two secondary nodes requires using the spherical inertia flag (I_{spher} =1 in `/RBODY`)
- Be aware that the main node may move when initializing the rigid body in Radioss Starter

Most of rigid body definition problems can be detected and corrected using Altair pre-processor tools.

Incompatible Kinematic Conditions

Incompatible conditions may results in an increase of the total energy and non-reproducibility of results in parallel computation even if `/PARITH/ON` is used. Radioss Starter detects the potential incompatible conditions that must be controlled by you.

Parallel Computation

The option `/PARITH/ON` must be used when using multi-processors. If it is not used, the problems will be difficult to reproduce and may appear and disappear.

Comments

1. Using *Inacti* =1 makes the model softer.
2. Never put free nodes in the interfaces. The selection of surfaces in interface definition must be done with shells and solids elements (or parts) and not nodes.

Unit Consistency

In Radioss, data for any unit system can be provided, but it is very important to keep the unit consistency. If a model does not have unit consistency, it will lead to incorrect results (unexpected behavior) or may lead to an error in the calculation.

Basic Units

	SI			CGS	Hydro	US	Japanese
Length	m	mm	mm	cm	cm	in	mm
Mass	kg	Mg(Ton)	kg	g	g	lb	kg
Time	s	s	ms	s	µs	s	ms
Plane angle	rad	rad	rad	rad	rad	rad	rad

	SI			CGS	Hydro	US	Japanese
Temperature	K	K	K	K	K	K	K
Frequency	Hz	Hz	Hz	Hz	Hz	Hz	Hz
Gravity	9.81	9.81E+03	9.81E-03	9.81E+02	9.81E-10	386	9.81E-03

SI Unit Example

	SI Unit Example
Length	$[\mathrm{m}]$
Mass	$[\mathrm{kg}]$
Time	$[\mathrm{s}]$
Plane angle	$[\mathrm{rad}]$
Temperature	$[\mathrm{K}]$
Frequency	$[\mathrm{Hz}]$
Rotational velocity	$\left[\frac{\mathrm{rad}}{\mathrm{s}}\right]$
Area	$[\mathrm{m}^2]$
Volume	$[\mathrm{m}^3]$
Moment of area (inertia)	$[\mathrm{m}^4]$
Consumption	$[\mathrm{m}^2]$
Speed	$\left[\frac{\mathrm{m}}{\mathrm{s}}\right]$
Acceleration	$\left[\frac{\mathrm{m}}{\mathrm{s}^2}\right]$
Tension	$\left[\frac{\mathrm{m}}{\mathrm{s}^2}\right]$
Lineic mass	$\left[\frac{\mathrm{kg}}{\mathrm{m}}\right]$
Surface mass	$\left[\frac{\mathrm{kg}}{\mathrm{m}^2}\right]$

	SI Unit Example
Volume mass	$\left[\frac{\mathrm{kg}}{\mathrm{m}^3}\right]$
Mass flow	$\left[\frac{\mathrm{kg}}{\mathrm{s}}\right]$
Volume flow	$\left[\frac{\mathrm{m}^3}{\mathrm{s}}\right]$
Quantity of movement	$\left[\frac{kg \cdot m}{s}\right]$
Kinetic moment	$\left[\frac{kg \cdot m}{s}\right]$
Moment of inertial (I)	$[\mathrm{kg} \cdot \mathrm{m}^2]$
Moment of force	$[\mathrm{N} \cdot \mathrm{m}]$
Force	$[\mathrm{N}]$
Linear force	$\left[\frac{\mathrm{N}}{\mathrm{m}}\right]$
Stiffness	$\left[\frac{\mathrm{N}}{\mathrm{m}}\right]$
Rotational stiffness	$\left[\frac{\mathrm{N} \bullet \mathrm{m}}{\mathrm{rad}}\right]$
Rotational damping	$\left[\frac{\mathrm{N} \bullet \mathrm{m} \bullet \mathrm{s}}{\mathrm{rad}}\right]$
Torsion damping	$\left[\frac{kg \cdot m^2}{s \cdot rad}\right]$
Viscous damping	$\left[\frac{\mathrm{kg}}{\mathrm{s}}\right]$
Damping for bending	$\left[\frac{\mathrm{N} \bullet \mathrm{s}}{\mathrm{m}}\right]$
Quadratic bulk viscosity	$[Pa^{\lambda} \cdot s]$
Dynamic viscosity	$[\mathrm{Pa} \cdot \mathrm{s}]$
Kinematic viscosity	$\left[\frac{\mathrm{m}^2}{\mathrm{s}}\right]$

	SI Unit Example
Density	$\left[\frac{\mathrm{kg}}{\mathrm{m}^3}\right]$
Power	$[\mathrm{W}]$
Energy	$[\mathrm{J}]$
Enthalpy	$[\mathrm{J}]$
Entropy	$\left[\frac{\mathrm{J}}{\mathrm{K}}\right]$
Strain rate	$\left[\frac{1}{\mathrm{s}}\right]$
Time relaxation	$[\mathrm{s}]$
Thermal expansion	$\left[\frac{1}{\mathrm{K}}\right]$
Thermal conductivity	$\left[\frac{\mathrm{W}}{\mathrm{m}\cdot\mathrm{K}}\right]$
Thermal resistance	$\left[\frac{\mathrm{W}}{\mathrm{m}^2\cdot\mathrm{K}}\right]$
Specific heat (Cp, Cv)	$\left[\frac{kg}{s^2\cdot m\cdot K}\right]$
Specific heat capacity (Cp)	$\left[\frac{\mathrm{J}}{\mathrm{m}^3\cdot\mathrm{K}}\right]$

Verify Consistency

Use basic units Mass, Length, or Time so you can get all other units you need.

$$Force = Mass \cdot Acceleration = \frac{Mass \cdot Length}{Time^2}$$

$$Pressure = \frac{Force}{Area} = \frac{Mass}{Length \cdot Time^2}$$

$$Energy = Force \cdot Length = \frac{Mass \cdot Length^2}{Time^2}$$

$$Density = \frac{Mass}{Volume} = \frac{Mass}{Length^3}$$

$$Acceleration = \frac{Length}{Time^2}$$

$$Volume = Length^3$$

For example, using base unit $[\mathrm{kg}]$, $[\mathrm{mm}]$, or $[\mathrm{ms}]$, will provide the following units force, pressure or density.

$$Force = \frac{Mass \cdot Length}{Time^2} = \frac{[kg] \cdot [mm]}{[ms]^2} = 10^3 \frac{[kg] \cdot [m]}{[s]^2} = [kN]$$

$$Pressure = \frac{Mass}{Length \cdot Time^2} = \frac{[kg]}{[mm] \cdot [ms]^2} = 10^9 \frac{[kg]}{[m] \cdot [s]^2} = [\mathrm{GPa}]$$

$$Energy = \frac{Mass \cdot Length^2}{Time^2} = \frac{[kg] \cdot [mm]^2}{[ms]^2} = \frac{[kg] \cdot [m]^2}{[s]^2} = [J]$$

$$Density = \frac{Mass}{Length^3} = \frac{[kg]}{[mm]^2} = 10^6 \cdot \frac{[kg]}{[m]^2}$$

Check Units

Property Card	Check the thickness unit in property if it is shell
Material Card	Check density unit
	Check E-modulus
	Check stress unit if possible
Load	Check force unit
	Check gravity unit, if using `/GRAV`
Length	Measure the geometry length unit with HyperCrash or HyperMesh

All units must be consistent.

Most Popular Units (with steel examples)

Mass	Length	Time	Force	Energy	Stress	Density	Young's module	Gravity	Yield stress
kg	m	s	N	J	Pa	7.8e+03	2.1e+11	9.81e +00	2.06e +05
g	mm	ms	N	mJ	MPa	7.8e-03	2.1e+05	9.81e-03	2.06e +02
kg	mm	ms	KN	J	GPa	7.8e-06	2.1e+02	9.81e-03	2.06e-01

Mass	Length	Time	Force	Energy	Stress	Density	Young's module	Gravity	Yield stress
Mg (ton)	mm	s	N	mJ	MPa	7.8e-09	2.1e+05	9.81e +03	2.06e +02
g	cm	micros	10^7N	10^5J	Mbar	7.8e+00	2.1e+00	9.81e-10	2.06e-06

Model Instrumentation

The outputs and storage frequencies must be set to get enough information for post-processing results.

Variables To Save

To find the origin of problems, you need output files with enough information, especially for the most potentially problematic entities as springs, interfaces and rigid bodies.

Springs	Save all spring and beam elements (except for spring of spot welds) `/TH/SPRING` Measure displacements and forces inside specific spring Example: engine mount, subframe, fixation, etc.
Interfaces	Save all contact interfaces (except for TYPE 2) `/TH/INTER` Output the forces of the specific interfaces Example: dummy/belt
Rigid Bodies	Measure the displacement and force inside specific rigid bodies Example: subframe fixations, engine mount, etc.
Section, Rigid Wall and Monitored Volume	Save all TH outputs

Set the Radioss Engine deck to save variables in animation files:

Mass	`/ANIM/MASS`
Nodal mass variation	`/ANIM/NODA/DMAS`
Plastic Strain	`/ANIM/ELEM/EPSP`
von Mises	`/ANIM/ELEM/VONM`
Hourglass	`/ANIM/ELEM/HOURG`

Specific energy	`/ANIM/ELEM/ENER`
Velocity	`/ANIM/VECT/VEL`
Contacts	`/ANIM/VECT/CONT`
Force on rigid wall	`/ANIM/VECT/FOPT`

In addition, `/ANIM/VECT/ACC` and `/ANIM/VECT/FINT` may be useful in case of a problem.

Note: Be sure to get an animation file when a run fails; you can use the `/STOP` option in the Engine file. If the computation stops, run another restart and write an animation in the first cycle.

Model Checking by Radioss Starter

Radioss Starter checks for consistency between format lines and prompts errors and warnings, if they exist. The output file `runname_0000.out` contains useful information helping to debug models.

The following checks are useful in the output file:

Errors and Warnings Messages
All errors must be corrected
Warning to be corrected:

- Degenerated elements
- Initial penetrations
- Real incompatible kinematic conditions

Check Time Step (element and nodal)
How many elements control time step? If a few, then modify the mesh to optimize the time step.

Check Mass and Inertias
Mass scaling causes instability. It is worthwhile to check the center of gravity computed by Radioss. That should fit the physical center of gravity.

Note: Small elements, beams, springs and spot welds need to be closely observed.

Computation Checks

This is a method to check the quality of simulation results. Even if the run does not fail, it needs to be verified that three fundamental conservation laws are respected. The time step variation and the qualitative evaluation of results may help to avoid modeling errors.

Energy Balance

Taking into account the external works, the total energy must remain constant or decrease slightly. The total energy can increase at the end of the computation, during the spring back or at the beginning during the first cycles.

Internal energy + Kinematic energy + Hourglass energy + Contact Energy + … = Variation of the External Work

If under-integrated elements are used, the total hourglass energy must remain lower than 10% of the total energy. If this is not the case, the mesh should be reworked or elements with physical stabilization method should be used.

The contact energy is not really physical. For each subset and for each part the following limitation is recommended:

$$\frac{E_h + E_c}{E_t} \leq 15\% \tag{283}$$

Where,

E_h	Hourglass energy
E_c	Contact energy
E_t	Total energy

The energy error is computed by Radioss as:

$$\%Error = 100\left(\frac{E_k + E_k^r + E_i}{E_{k,1} + E_{k,1}^r + E_{i,1} + E_{wk} - E_{wk,1}} - 1\right) \tag{284}$$

Where,

E_k	Translational kinetic energy
E_k^r	Rotational kinetic energy
E_i	Internal energy
E_{wk}	External work
$E_{k,1}$	Initial translational kinetic energy
$E_{k,1}^r$	Initial rotational kinetic energy
$E_{i,1}$	Initial internal energy

$E_{wk,1}$ Initial external work

The error must be negative and decreasing (except for the first cycle or in spring-back stage). The error must be less than 15% at the end of computation (5% for a good model). If the error increases a little, then decreases may be normal. If the error increases, this means that a problem has occurred. Then, the error may grow to 99.9% with computation failure message.

Mass Balance

If the mass increases, its variation must remain smaller than 1% for each subset and for each part ($\frac{dM}{M} < 1\%$). If the mass variation is between 1% to 3%, check if the nodes with the added mass are moving or not. If this is the case, the added mass results in an increase in kinetic energy. For more than 3% of variation, the results are probably bad.

Momentum Balance

The dynamic equilibrium of each node is satisfied by the Newton law at the end of each cycle. As Radioss resolves the equilibrium equations at each cycle, normally the momentum balance is satisfied. However in case of a problem, a cross-check between nodal accelerations and the impactor forces (interface, rigid wall, barrier, etc.) can help to better understand problems.

$$F_{wall} = \frac{Monentumvariation}{dt} \quad (285)$$

Time Step Evolution

If the time step decreases and then it goes up quickly, there is not a problem. If it varies greatly from one cycle to another, it may be due to the interface stiffness.

If the time step remains low, a problem has occurred. In this case, find the node (or element) controlling the time step and try to understand why the decrease occurred.

Visual Inspection of Deformed Parts

After getting some animation files, the overall deformation of the structure can be compared to the physical behavior. Inspect the deformed shape to verify:

- There are no flying nodes (parts)
- The deformed shape is smooth
- The chord angle between adjacent elements is sufficiently small
- There are no intersections

Good physical behavior is obtained when:

- The plastic strain is less than 30%

$$\frac{E_k + E_k^r}{E_t} \leq 1\% \quad (286)$$

Meaningless Results

Contact forces, von Mises stress, nodal velocities, and accelerations have to be checked carefully. If the values are meaningless (for example: von Mises stress = 1 GPa), the first check may be the unit system consistency. Refer to the Appendix for more information.

See Also
Results Checking (FAQ)

Model Debugging

Here are some guidelines to find the origin of computation problems.

First, note that Radioss Explicit Solver allows resolving high nonlinear dynamic problems; a computation may be wrong even if the run does not fail. If the calculation goes to the end, it illustrates a good numerical behavior but not inevitably a good physical response. However, it is required to receive the message "`Normal termination`" at the end of the Engine output file, to validate numerical resolution procedures.

The validity of results can be demonstrated by satisfying the following conditions:

- Numerical stability
- Physical behavior
- Physical reliability

The numerical stability is ensured if a message "`Normal termination`" and energy and mass balance are verified.

If the prepared model does not really represent the physical problem, the wrong results may be obtained. To understand the problem, first you need to ask good questions, which should be answered to put in evidence the reliability of results:

- How dependent is the result on friction?
- How dependent is the model on rupture phenomena?
- How dependent is the result on unknown material parameters?
- How dependent is the model on other phenomena that are difficult to simulate?

If the results are highly dependent to a given parameter, the experimental test must be realized to use high precision values for computation.

> **Note:** The best model is that for which you know the values of physical parameters!

Divergence

Divergence occurs when one of the following conditions is observed:

- Positive energy error (except for the first cycle)
- Negative energy error by more than 15% (except for the first cycle)
- Kinematic time step activation in interface TYPE7
- Time step given by a rigid body

- Unexplained changes in time step
- Quick increase of mass

There are three types of divergence:

Quick divergence
Energy error increase is often exponential. The calculation fails in few cycles.
Potential causes are:

- Incompatible kinematic conditions
- Negative stiffness in spring
- Negative stiffness in tabulated material law
- Secondary nodes too far from the main surface in interface TYPE2

Late divergence
Time step is too low. The structure is distorted and high penetrations in a lot of interfaces are observed.
Potential cause is the mesh quality

Slow divergence
The final error is not necessarily the cause of divergence.
Potential causes:

- For a linear divergence, the cause can be the existence of incompatible kinematic conditions.
- For a sinusoidal divergence, it is typically liberated or generated energy (example: initial penetrations and spring stiffness functions).
- Too soft of a material can also be the cause.

It is important to find which event triggered the problem. The event just before the divergence needs to be checked. If a strange behavior is observed for a given part, the connected parts and previous events also have to be studied.

Run Problems

Run Stops At Cycle 0
The data is not written in the Engine output file `runname_0001.out`. This is generally due to bad running procedure when the Restart file cannot be read properly.

Run Stops After Few Cycles
The data is written in the Engine output file `runname_0001.out`. The origin of the problem can either be the incompatible kinematic conditions (for example: rigid bodies with a common secondary node) or out of bounds values in material or element properties; although, initial penetrations may be the cause.

Run Stops During Computation
First check the required disk space, then the behavior just before and after divergence can be studied. The time step evolution and the energy error need to be observed.

- An animation file before run terminates. This is very useful to identify areas:
  - Where the velocity is out of range (>> 100 m/s)
  - Mass is getting added

- Deformation is excessive (check plastic strain)
- Stress is unrealistic (> 2 Gpa)

  In HyperView, review the animation and check visually for any obvious issues:

  - Velocity is not applied to all the nodes of the vehicle
  - Incorrect boundary conditions
  - Missing connections (seatbelt is not connected properly, etc.)
  - Review basic contours:

    1. von Mises Stress: check for values that are out of bounds
    2. Added Mass: check for very big mass
    3. Velocity: check for very big velocity In many cases, the nodes with the highest velocity or high mass added will be in the same area.

- A point in the `T01` file. This is useful to identify:
  - The part which have sudden increase in internal/kinetic energy. Plot the internal and kinetic energy of all the parts in HyperGraph (you need to put all the parts in time history).
  - Plot global energy balance to identify, if the divergence is due to the internal energy, kinematic energy, external work force.
- Information at the end of the listing file (`_0001.out`).

  For example, in the case below, the information provided at the end of the `_0001.out` is very useful in trying to understand the area of the model causing the simulation to stop.

  The solver reports:

  - Which parts have the highest kinetic (1) / internal energy (2)
  - Which nodes have the highest velocity (3) / mass change (4)

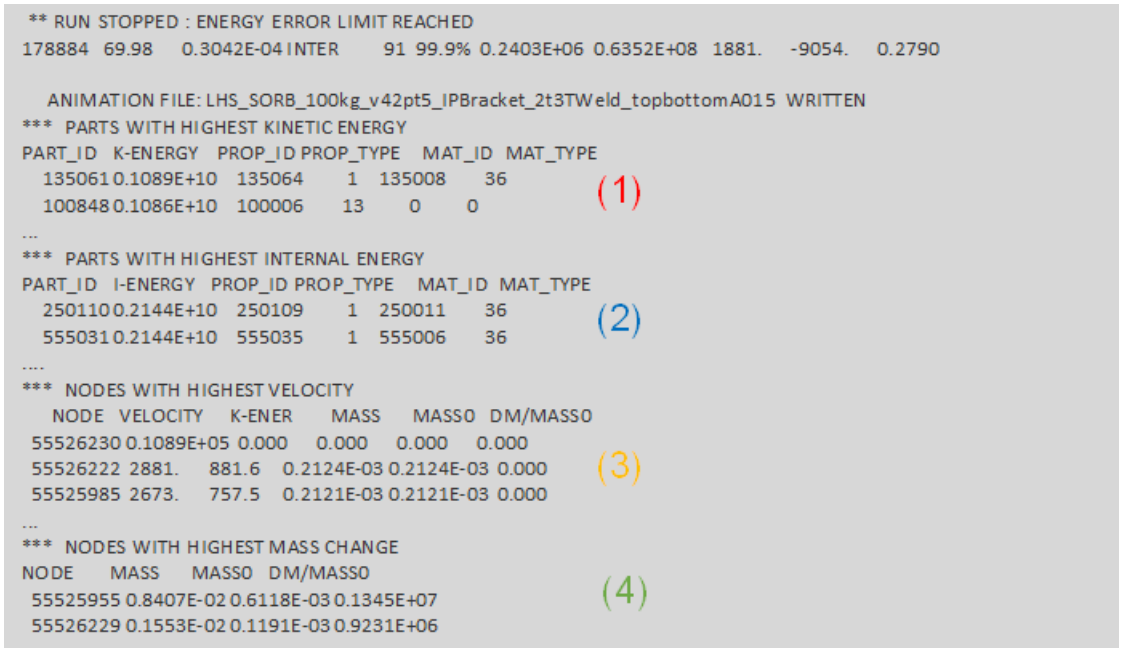


Figure 391:

For example, in the case below, before the simulation fails with an error termination, the following message may be printed in the `_0001.out`:

```
CYCLE    TIME       TIME-STEP  ELEMENT          ERROR  I-ENERGY    K-ENERGY T
 K-ENERGY R  EXT-WORK MAS.ERR
62100   18.53     0.2985E-03    SOLID      230440718     -1.3%    0.3484E+05
 0.1313E+06     109.7   1523.
RUPTURE OF SOLID ELEMENT NUMBER 230025298
 DELETE CONNECTION SOLID ELEMENT 110010956, ENERGY= 4.732011155E-05
RUPTURE OF SOLID ELEMENT NUMBER 230024559
**WARNING MINIMUM TIME STEP 0.5976E-05 IN INTERFACE NB 2300040(DTMIN=
 0.1000E-04)
SECONDARY NODES NB       230373002  230373019
MAIN NODES NB            230373003  230373004
 DELETE SECONDARY LINE FROM INTERFACE
```

An initial intersection causes high distortion and eventually ERROR termination.

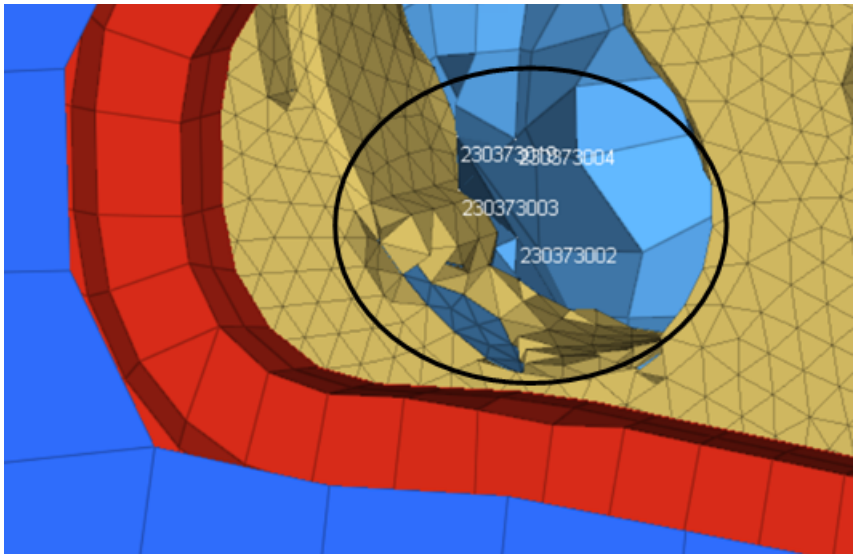


Figure 392:

For example, in the case below, the run is killed, due to contact with a rigid causes the ERROR termination. The output file prints:

```
339840    67.61      0.5696E-09 INTER       2157  99.9%  0.1571E+06  0.6329E+13  0.1612E+09  -1661.
 WARNING INTERFACE NB         2157
 GAP=  0.630000000000000      PENE=  0.629999999957717
 LINE      1237096     1237097 DE-ACTIVATED FROMINTERFACE

MESSAGE ID :          207

 ** RUN KILLED : TOTAL MASS ERROR LIMIT REACHED
```

Figure 393:

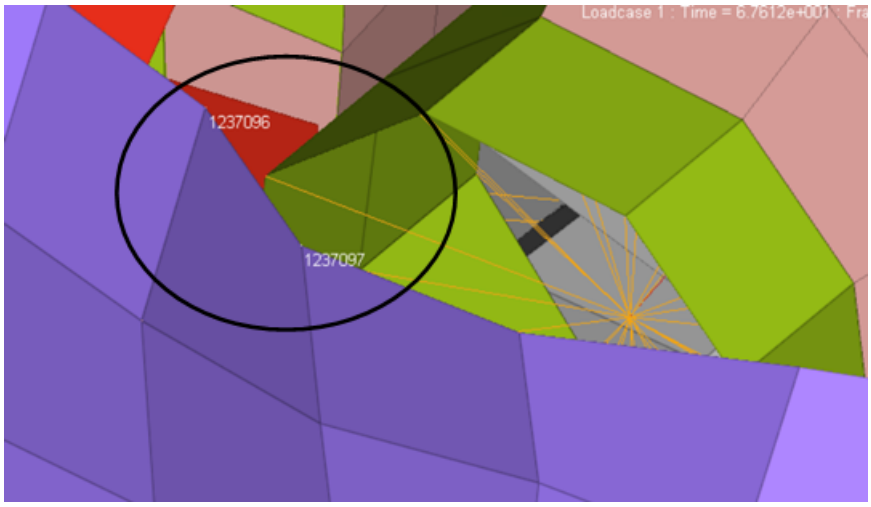


Figure 394:

Negative Volume Message

This is mainly due to high deformation of solid meshes. Fully-integrated brick elements are especially affected by this problem which may be caused by a bad interface behavior or bad material definitions. In any case, the use of co-rotational formulation is recommended to avoid bad shear deformation response.

The Stress-Strain Computation Options (/PROP) assumption can be used with `/DT/BRICK/CST` to avoid negative volumes (refer to Time Step and Finite Elements for more details about this option).

Appendix

Basic Relations

	E, ν	E, G	E, B	G, ν	G, B	B, ν	λ, μ
E	E	E	E	$2(1+\nu)G$	$\frac{9BG}{3B+G}$	$3(1-2\nu)B$	$\frac{(3\lambda+2\mu)\mu}{\lambda+\mu}$
$G=\mu$	$\frac{E}{2(1+\nu)}$	G	$\frac{3EB}{9B-E}$	G	G	$\frac{3(1-2\nu)B}{2(1+\nu)}$	μ
$B=K$	$\frac{E}{3(1-2\nu)}$	$\frac{EG}{9G-3E}$	B	$\frac{2(1+\nu)G}{3(1-2\nu)}$	B	B	$\frac{3\lambda+2\mu}{3}$
ν	ν	$\frac{E-2G}{2G}$	$\frac{3B-E}{6B}$	ν	$\frac{3B-2G}{6B+2G}$	ν	$\frac{\lambda}{2(\lambda+\mu)}$
D_{11}	$\frac{E(1-\nu)}{(1+\nu)(1-2\nu)}$	$\frac{(4G-E)G}{3G-E}$	$\frac{3B(3B+E)}{9B-E}$	$\frac{2G(1-\nu)}{1-2\nu}$	$\frac{3B+4G}{3}$	$\frac{3B(1-\nu)}{1+\nu}$	$\lambda+2\mu$
$D_{12}=\lambda$	$\frac{E\nu}{(1+\nu)(1-2\nu)}$	$\frac{(E-2G)G}{3G-E}$	$\frac{3B(3B-E)}{9B-E}$	$\frac{2G\nu}{1-2\nu}$	$\frac{3B-2G}{3}$	$\frac{3B\nu}{1+\nu}$	λ
C_{11}	$\frac{E}{1+\nu^2}$	$\frac{4GG}{4G-E}$	$\frac{36BE}{36-(3-E/B)^2}$	$\frac{2G}{1-\nu}$	$\frac{4G(3B+G)}{3B+4G}$	$\frac{3B(1-2\nu)}{1-\nu^2}$	
C_{12}	$\frac{E\nu}{1+\nu^2}$	$\frac{(E-2G)2G}{4G-E}$	$\frac{6E(3-E/B)}{36-(3-E/B)^2}$	$\frac{2G\nu}{1-\nu}$	$\frac{2G(3B-2G)}{3B+4G}$	$\frac{3B(1-2\nu)}{1-\nu^2}$	

Hook Law 3D (principal stress and strain)

$\boldsymbol{\sigma = D\varepsilon}$

$\sigma_1 = D_{11}\varepsilon_1 + D_{12}\varepsilon_2 + D_{13}\varepsilon_3$

$\sigma_1 = (\lambda+2\mu)\varepsilon_1 + \lambda(\varepsilon_2+\varepsilon_3)$

$\sigma_1 = \lambda(\varepsilon_1+\varepsilon_2+\varepsilon_3) + 2\mu\varepsilon_1$

$\sigma_1 = K\varepsilon_{kk} + 2\mu e_1$ with $\varepsilon_{kk} = \varepsilon_1+\varepsilon_2+\varepsilon_3$ and $e_1 = \varepsilon_1 - 1/3(\varepsilon_1+\varepsilon_2+\varepsilon_3)$

Hook Law 2D (plane stress)

$\boldsymbol{\sigma = C\varepsilon}$

$\sigma_1 = C_{11}\varepsilon_1 + C_{12}\varepsilon_2$

Unit Systems

Length	Time	Mass	Force	Pressure	Velocity	ρ	Energy	G
m	s	Kg	Kg m/s2	N/m2	m/s	Kg/m3	Kmg2/s2	9.81
m	s	Kg	N	Pa	m/s	m Kg/l	J	9.81
m	s	g	mN	mPa	m/s	μ Kg/l	mJ	9.81
m	s	Mg (ton)	KN	KPa	m/s	Kg/l	KJ	9.81
m	ms	Kg	MN	MPa	Km/s	m Kg/l	MJ	9.81e-6
m	ms	g	KN	KPa	Km/s	μ Kg/l	KJ	9.81e-6
m	ms	Mg (ton)	GN	GPa	Km/s	Kg/l	GJ	9.81e-6
mm	s	Kg	mN	KPa	mm/s	M Kg/l	mJ	9.81e+3
mm	s	g	mN	Pa	mm/s	K Kg/l	nJ	9.81e+3
mm	s	Mg (ton)	N	MPa	mm/s	G Kg/l	mJ	9.81e+3
mm	ms	Kg	KN	GPa	m/s	M Kg/l	J	9.81e-3
mm	ms	g	N	MPa	m/s	K Kg/l	mJ	9.81e-3
mm	ms	Mg (ton)	MN	TPa	m/s	G Kg/l	KJ	9.81e-3
cm	ms	g	daN	10^5Pa bar	dam/s	Kg/l	dJ	9.81e-4
cm	ms	Kg	10^4 N (KdaN)	10^8Pa (Kbar)	dam/s	K Kg/l	hJ	9.81e-4
cm	ms	Mg	10^7 N	10^{11} Pa	dam/s	M Kg/l	10^5 J	9.81e-10

Length	Time	Mass	Force	Pressure	Velocity	ρ	Energy	G
		(ton)	(MdaN)	(Mbar)				
cm	μ_s	g	10^7 N (MdaN)	10^{11} Pa (Mbar)	10^4 m/s	Kg/l	10^5 J	9.81e-10

Filtering

Often it is useful to filter results in a material or failure law to remove numerical noise. The most common filter is an exponential moving average filter. This is especially important for material models that include strain rate effects.

In most materials, the flag F_{smooth} = 1 must be defined to enable the filtering and the cutoff frequency entered using F_{cut}. For the case of filtering strain rates, use:

$$\dot{\varepsilon}_{filtered}(t) = \alpha\dot{\varepsilon}(t) + (1-\alpha)\dot{\varepsilon}_{filtered}(t-dt) \quad (287)$$

Where,

$\dot{\varepsilon}_{filtered}(t)$
: Filtered strain rate.

$\dot{\varepsilon}(t)$
: Strain rate at the current timestep before filtering.

α
: Degree of weighting decrease, a constant smoothing factor between 0 and 1. A higher value discounts previous values faster which results in less filtering.

dt
: Timestep of the simulation.

$\dot{\varepsilon}_{filtered}(t-dt)$
: Filtered strain rate at the previous time step.

For materials laws where F_{cut} can be entered.

$$\alpha = 2\pi \quad dt \quad F_{cut} \quad (288)$$

Where, F_{cut} is the cutoff frequency.

Thus,

$$F_{cut} = \frac{\alpha}{2\pi \quad dt} \quad (289)$$

The cutoff frequency is a function of the model timestep. Experience shows that the speed of the deformation is important also. For slower speeds, like a car crash, 1 – 10 kHz (1000 – 10,000 Hz) is a good value, but for high-speed events, like ballistic, less filtering should be used - so 1 – 10 GHz

is appropriate. Good engineering judgment should be used to determine a reasonable value for each simulation. Refer to RD-E: 1102 Strain Rate Effect for an example of strain rate filtering.

Implicit Structural Finite Element Analysis

In Radioss, both implicit and explicit time-integration schemes are available to you. In an explicit scheme, velocities and displacements are obtained by direct integration of nodal accelerations. With this approach, the cycle time step is often small, due to stability considerations. Therefore, for static or slow dynamic computations where the duration of the study is long, many cycles are necessary to carry out the simulation. An alternative to explicit method in such cases is the implicit time-integration scheme. It can be shown that the implicit scheme is unconditionally stable, resulting in a larger cycle time step vis-a-vis the explicit method. However, in an implicit scheme, a global stiffness matrix should be assembled and inverted, leading to higher cost and greater memory requirements, per loading step.

An explicit approach is more appealing for fast dynamic problems with highly nonlinear geometric and material behaviors. As all quantities may be treated as vectors, low storage is needed. The number of cycles to achieve simulation may be quite large, but global efficiency is generally observed. The implicit method, on the other hand is more efficient for static applications and/or long duration simulations.

The optimal solution is to have both implicit and explicit methods readily available in the same code. In addition, implicit and explicit solvers in Radioss share the same routines for internal force computation; thus enabling seamless switching between the two schemes without an out-of-balance issue during the transition. Therefore, the two schemes can be coupled to run multi-stage analyses. The two uses for the implicit solution are spring-back (implicit) after sheet metal forming (explicit), or gravity loading/initial state computations (implicit) before crash simulations (explicit). It is recommended to use OptiStruct for all other implicit simulations.

Although implicit scheme is unconditionally stable, it is not unconditionally convergeable (nonlinear case). In fact, it is generally less robust than explicit scheme and requires more user involvement.

In this section, available implicit features for different implicit analyses are first presented; followed by a list of keywords required to activate implicit simulation. In addition, some details are given to better understand and to efficiently use the various implicit options. Finally some Output Messages and Troubleshooting techniques are listed.

Implicit Features and Compatibility

Not all explicit features are available for implicit analyses; actual available features are presented in this section.

Element Formulations

In Radioss implicit, the element formulations are used in two main computations.

1. Stiffness Matrix which will determine the displacements:

   Available element formulations for this computation are given in the table below.
2. Strains, Stress and Internal Nodal Forces:

   Implicit and explicit analyses share the same coding and all formulations are available for this computation.

/BRICK	/TETRA4	/TETRA10	/BRIC20	Thick Shell /BRICK, /PENTA6
I_{solid}=24 I_{solid}=14 (recommended)	I_{tetra4}=0	$I_{tetra10}$=0	I_{solid}=16	I_{solid}=14, 15

/SHELL	/SH3N	/SPRING	/BEAM	/TRUSS
I_{shell}=24 (QEPH) I_{shell}=12 (QBAT)	I_{sh3n}=1 I_{sh3n}=2 (default)	/PROP/TYPE4, /PROP/TYPE8, /PROP/TYPE12, /PROP/TYPE13	/PROP/TYPE3 (BEAM)	/PROP/TYPE2 (TRUSS)

Monitored Volume TYPE3, Orthotropic shell (TYPE9, 10, 11 and 17) are also available.

Orthotropic solid (TYPE6) is also available from version 11. However, the elements in the above table are strongly recommended, especially in a nonlinear analysis.

Material Compatibility

Radioss uses only the elastic part of material parameters to build the stiffness matrix (or Jacobian matrix for nonlinear analyses) for most of the material laws.

For the hyper-elastic type laws (42, 62, 69, and 82), special treatments have been done to avoid the divergence. The use of elastic matrix (called initial stress method) results out slow convergence (consisting tangent matrix development is ongoing); two Nonlinear Solution methods are recommended

in this case: Modified Newton and Quasi-Newton (BFGS). For the viscous type laws used for quasi-static analysis, a long duration time is recommended for the simulation (as there is no physical sense for time in this case). In this way, there is no compatibility issue with any Radioss material, but converging issues might be experienced, especially for the high nonlinear materials.

Kinematic Conditions

Several kinematic conditions are supported.

The following available kinematic conditions are supported in Radioss Implicit:

- Boundary conditions `/BCS`
- Imposed displacement `/IMPDISP` (velocity `/IMPVEL`, and acceleration `/IMPACC`)
- Rigid bodies `/RBODY`
- Interface TYPE2 `/INTER/TYPE2` (not available with rupture or penalty methods)
- Rigid walls `/RWALL` (not recommended due to performance reasons. Use contact interface, if possible.)
- Adaptive meshing link `/ADMESH`
- Interpolation constraint element `/RBE3`
- Rigid element `/RBE2`

Contact Interfaces

Only contact interface Types 5, 7, 10 and 11 are available and are supported in Radioss Implicit.

> **Note:** Some options of interface TYPE 7, such as $I_{fric} > 0$, I_{adm} are currently not available. Only the SMP version is available with interface TYPE 24 in implicit.

Contact interfaces are treated through a penalty method (with fictitious springs) and the contact stiffness will be stored in an independent stiffness matrix. Information regarding this second stiffness matrix can be printed out (if `/IMPL/PRINT/NONLIN/n` is defined) once contacts have been detected.

Loads

Concentrated load `/CLOAD`, pressure `/PLOAD`, and gravity load `/GRAV` are available. The loading definition is the same as in an explicit simulation.

> **Note:** Additional attention should be brought to time-based loading functions in nonlinear implicit computations as this might influence the iteration convergence. In such computations, the use of increasing time versus load function instead of a constant function is recommended even for the gravity load and the imposed displacements.

Thermal Analysis

In Radioss not all thermal options are currently available for implicit analysis.

Implicit Analysis Types

The following analysis modules (types) are available:

- Model Check: with `/IMPL/CHECK`
- Static or Quasi-static linear
- Buckling mode analysis
- Static or Quasi-static nonlinear
- Dynamic nonlinear
- Modal analysis

Implicit Analysis Activation

The keywords required to launch a Radioss Implicit Analysis are defined in the Engine file and begin with `/IMPL`.

The only exception is `/EIG`, which is defined in Radioss Starter for running Modal Analysis. Most Radioss output terms have the same meaning for Implicit and Explicit Analysis.

For static problems, a time step is used. This has no physical meaning and only serves to define the increments for loading and displacements; that is the cycle number here will correspond to the increment number.

Therefore, there the change is minimal in the Engine input file. Review the output, like `/PRINT` (for example: `/PRINT/-1` will print out at each increment) and some output file frequencies, as large time step is used.

Note: Keywords marked with an asterisk (*) in the following section are required.

Linear Analysis

The basic keywords for linear analyses are:

1. Linear Static:

   /IMPL/LINEAR: Static linear computation

   /IMPL/SOLVER/n: Linear solver method, where n indicates the type of solver

   /IMPL/PRINT/LINEAR/Nprint: Printout frequency for linear resolution

   The printout frequency keyword is used mainly for iterative solver (or mixed). When used with a direct solver, the relative residual will be printed.

   /IMPL/LINEAR/INTER: Contacts can be taken into account in a Linear Analysis through a simple two-step linear approach: A linear first step that evaluates penetrations and generates interface reaction springs followed by a linear second step.

2. Linear Quasi-static:

   /IMPL/QSTAT: Quasi-static Analysis

   /IMPL/QSTAT/DTSCAL: Scaling factor of adding inertial stiffness

   When */IMPL/QSTAT/DTSCAL* is used with */IMPL/LINEAR/INTER*, the scaling factor is only applied to the second step.

3. Euler Buckling:

   /IMPL/BUCKL: Euler buckling modes

   /IMPL/LINEAR: Static linear computation

It becomes */IMPL/BUCKL/1* and Lanczos solver (BCSLIB-EXT) is used for buckling mode extraction. Buckling from restart or with pre-stresses is also available (*/IMPL/BUCKL/2*).

Animation output files (**Annn*) are special in the case of a Buckling Analysis. Each file contains one buckling mode (where the critical loading is presented by the time value). If a linear computation precedes a Buckling Analysis, then the first two animation files (**A001*, **A002*) contain results of linear

structure computation and the buckling modes are available starting from the third animation file (`*A003`).

Nonlinear Analysis

Iterations are needed in an implicit Nonlinear Analysis. The simulation is typically divided into several increments (steps) to manage the nonlinearity that will be represented by the time step control.

A nonlinear solver (Newton-based method) is used to find the balanced forces on the updated geometry (large displacement, large strain) at each step.

Two types of control parameters should be defined:

- Nonlinear solver (using `/IMPL/NONLIN/n`)
- Steps (using `/IMPL/DT/n`, `/IMPL/DTINI` and `/IMPL/DT/STOP`)

The basic keywords for Nonlinear Analysis are:

*`/IMPL/NONLIN/n`: Nonlinear solver method, where, n indicates the type of nonlinear solver

`/IMPL/SOLVER/n`: Linear solver method, where, n indicates the type of linear solver

`/IMPL/PRINT/NONLIN/Nprint`: Printout frequency for nonlinear iterations

*`/IMPL/DT/n`: Time step control, where, n indicates the time step control method

*`/IMPL/DTINI`: Initial time step

`/IMPL/DT/STOP`: Min-max time step values

`/IMPL/QSTAT`: Quasi-static implicit

`/IMPL/SPRBACK`: Spring-back computation

Implicit Solvers

Implicit Solvers with Parallel Version Compatibility Table

	Option	Radioss SMP	Radioss MPP SPMD
Implicit Resolution	/IMPL/SOLVER	PCG	
	/IMPL/SOLVER/3	BCS	MUMPS
	/IMPL/SOLVER/5	BCS+PCG	MUMPS+PCG
Buckling Modes	/IMPL/BUCKL/1 with *Isolv*=1	SuperLU	MUMPS
Eigen Modes	/EIG (Starter)	SuperLU	MUMPS

Acronyms:

PCG:	Iterative Preconditioned Conjugate Gradient
BCS:	Boeing Solver
MUMPS:	Massively Parallel Multi-Frontal Solver

Linear Solvers

Direct, Iterative and Mixed

Linear solver will be used in both Linear and Nonlinear Analyses, so it is very important to choose an appropriate solver for your application.

The PCG (Preconditioned Conjugate Gradient) iterative solver has been available from the first version of Radioss Implicit. Direct solvers are also available. The default solver is the PCG with Factored Approximate Inverse preconditioning method.

Choosing the appropriate solver depends on the application model. In general, an iterative solver is suited for well-conditioned models with homogeneous stiffness (for example: solid element models); but computationally more expensive for an ill-conditioned model with heterogeneous stiffness. On the other hand, direct solvers provide more accurate results and are less sensitive to matrix quality but require more memory storage. When out-of-core memory is used, the performance of direct solvers could be greatly reduced.

If you are not sure as to which solver to use for a particular application, it is recommended to try a direct solver first, provided that memory is not an issue. For large simulations, such as Full-Vehicle Analysis, where memory might be an issue, the PCG method with higher quality preconditioner (this is set using `/IMPL/PREPAT/n`, for example: n=2) could be used instead.

For a Nonlinear Analysis, it is worth comparing the two methods on your model by running a simple Linear Analysis before launching the actual analysis.

Mixed solvers may provide better performance for simulations with contact where the contact stiffness DOF is much lower than the total DOF.

Nonlinear Solvers

Modified Newton and Quasi-Newton Methods

Once again, the choice of solver depends on the type of analysis. Generally, the Quasi-Newton method is more suitable for an analysis with a high degree of nonlinearity, but it requires more memory and costs more per iteration.

Fluid-Structure Interaction Analysis with Radioss and AcuSolve

Note: Support for Direct Coupled Fluid-Structure Interaction (DC-FSI) Radioss and AcuSolve has been discontinued. It is now recommended to use OptiStruct coupled with AcuSolve as described in Fluid-Structure Interaction in the *OptiStruct User Guide*.

Radioss and AcuSolve are fully-integrated to perform a Direct Coupled Fluid-Structure Interaction (DC-FSI) Analysis based on a partitioned staggered approach. Radioss and AcuSolve are both time domain simulation codes that break the coupled simulation into a number of time steps. Because the governing equations of both Radioss and AcuSolve are nonlinear, sub-iterations are typically required within each time step. At each sub-iteration of FSI analysis, the fluid tractions in AcuSolve are converted into nodal forces which are then transferred to the structural interface mesh of Radioss. These forces are used to calculate the deformation of the structure using Radioss. Note that in addition to the load from the fluid flow tractions, additional structural loads can also be applied. The resulting deformed shape of structure is passed back to AcuSolve as the new fluid boundary. This FSI cycle is shown below in Figure 395.

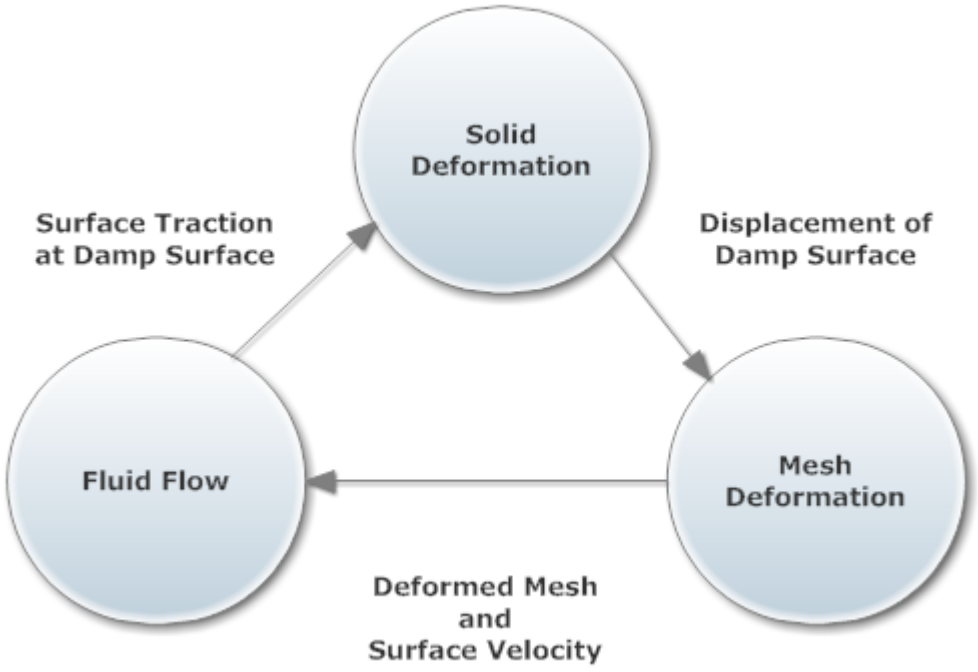


Figure 395: Direct-Coupled Fluid-Structure Interaction (DC-FSI) Cycle

Target Applications

The DC-FSI capability aims at simulations of compliant structural response to fluid flow and its complex interrelationship. This capability is particularly suited for problems exhibiting nonlinear structural response. It may also be used for linear structural response; however, for linear structural response the P-FSI solution offered by AcuSolve may be more effective in solving the linearized structural response with the nonlinear flow solution. For further information on the P-FSI method, refer to the AcuSolve Command Reference Manual. Most Radioss features including nonlinear materials, nonlinear geometric effects, and contact are available for use. The interface domain can be modeled with solid, shell, and

beam elements. Most of the features in AcuSolve can be used in DC-FSI simulations. These features may include, namely laminar or turbulent flow, heat transfer, a rich set of material options, free-surface and multi-species driven flows. The moving boundary problems are modeled using Arbitrary Lagrangian-Eulerian (ALE) and sliding mesh technologies. For further details, consult the AcuSolve *Command Reference Manual*.

Primary applications targeted by this capability include:

- Automotive: Hydraulically damped rubber mounts, door seals, shock absorbers, design of valves and rubber diaphragms and antilock braking systems
- Oil/Gas: Long marine risers, moorings, free spans and drilling risers
- Aerospace: Wing aero-elasticity, UAV and MAV
- Wind turbine: Large deformation of blades
- Consumer Goods: design and packaging
- Bio-medical

See Also
Fluid-Structure Interaction

DC-FSI using Radioss and AcuSolve

The following workflow is strongly recommended for solving DC-FSI problems:

1. Develop Radioss structural model and an AcuSolve-only fluid model, and ensure that the uncoupled analyses run successfully.
2. Prepare the stand-alone models that include DC-FSI commands.
3. Run the coupled analysis.
4. Post-process the FSI solution.

This workflow ensures that both the Radioss and the AcuSolve models are defined properly prior to performing a coupled simulation. Radioss and AcuSolve do not require that the analysis be run with a particular unit system, but both analyses need to use a consistent unit system. As a rule, all quantities exchanged between the two solvers will be in dimensional form, and the components of all vector quantities will be resolved in the inertial frame. For consistency, identical inertial frames must be selected for Radioss and AcuSolve.

Radioss Model

Preparing the Radioss model for DC-FSI involves three steps:

1. Create input deck for Radioss
2. Identify the interface region and the solution quantities exchanged
3. Define the communication scheme

The Radioss FSI input data that has the Engine keyword `/IMPL/DYNA/FSI` is:

```
/IMPL/DYNA/FSI
ESET SSET PORT MINX MAXX TAG DATA MSG WAIT INIT
FTOL DTOL TTOL XTOL
```

Each of the data items are discussed below under their relevant topic.

Damp Surfaces

The damp surface of the structural mesh must be specified in the Radioss FSI input data. The damp surface can either be specified by a group of elements or a surface. Note that the structural mesh on the damp surface does not have to match the interfacing fluid mesh. AcuSolve will internally project the CFD tractions from the fluid interface mesh onto a non-matching damp surface structural mesh. The mapping of nodal forces is also supported for structural beam elements. For example, a rod, pipe or blade can be modeled with simple beam elements in the structural mesh. The corresponding fluid mesh will contain the actual three-dimensional geometry of these beam elements.

If the damp surface is specified by a group of elements (`ESET`), use the `/GRBEAM/BEAM/`, `/GRSHEL/SHEL/`, or `/GRSH3N/SH3N/` data for beam, 4-node, and 3-node shell elements, respectively. It is possible to specify the damp surface of a solid by coating it with shell elements and then assigning these elements zero material properties using the `/MAT/LAW0` data. If the damp surface consists of beam elements, then these must be specified using the beam element group data. The ID of the element group is the first item of the `/IMPL/DYNA/FSI` data.

The damp surface (SSET) can also be specified by the Radioss `/SURF` data. The `/SURF` data has many options that can be used to specify a surface. The surface can be either the surface of a shell structure or the surface of a solid mesh. The surface ID is the second item of the `/IMPL/DYNA/FSI` data. Beam elements cannot be specified using the `/SURF` data.

> **Note:** The damp surface can be defined by a combination of an element group and a surface.

The matching fluid surface is specified in the AcuSolve data using the `EXTERNAL_CODE_SURFACE` command.

In some cases a portion of the structural damp surface must be "tagged" so that AcuSolve can associate it with the AcuSolve boundary surface. This is done by specifying the Radioss part ID as the `external_code_tags parameter` in the `EXTERNAL_CODE_SURFACE` data.

Example:

```
EXTERNAL_CODE_SURFACE( "Drill" ) {
    ...
        external_code_tags = {"10"}
        ...
}
```

Data is used to specify that this surface corresponds to the Radioss part with ID `10`. Note that each individual surface must consist of elements with the same part ID, and each separate surface must consist of elements with separate part ID's. In order to apply a tag, the surface of a solid structure, the solid must be coated with shell elements and the property ID of these shell elements used as the surface tag. To specify the use of surface tags, set item `10` of the `/IMPL/DYNA/FSI` data (TAG) to `1`.

AcuSolve Model

This section provides an overview for preparing the AcuSolve model for DC-FSI. For detailed information on the commands, refer to the *AcuSolve Command Reference Manual*. The three steps to prepare the AcuSolve model are:

1. Set the analysis parameters to include an external field.
2. Define the solution strategy.

3. Define the external surface definition.

Analysis Parameter Setup to include an External Field

Use the EQUATION command to specify the solution fields available or the system of equation that are present in the problem. To include a field that is computed with an external solver, for Radioss, set the external_code parameter to 'ON'.

For example, when conducting a fluid-structure analysis in which AcuSolve computes the flow and Radioss computes displacements, turn ON the external_code parameter and allow for arbitrary mesh movement by setting the mesh parameter to arbitrary_lagrange_eulerian.

```
EQUATION {
flow = navier_stokes
mesh = arbitrary_lagrange_eulerian
external_code = on
}
```

Define Solution Strategy

In conjunction to the EQUATION command, which specifies the existence of solution fields in the problem, you must use TIME_SEQUENCE and STAGGER commands to define time stepping and staggering strategy. The preferred method is to use the AUTO_SOLUTION_STRATEGY command to have AcuSolve generate the solution strategy commands. In order to highlight the solution strategy commands, highlight the TIME_SEQUENCE and STAGGER commands.

The TIME_SEQUENCE command specifies the time stepping and staggering strategy. In the following example, the external field is solved first for structural displacements, which is then followed by the mesh displacement (ALE) in which the fluid elements are adjusted. Then the flow and turbulence equations are solved.

```
TIME_SEQUENCE {
   min_time_steps                    = 1
   max_time_steps                    = 15
         :
   min_stagger_iterations            = 1
   max_stagger_iterations            = 20
   staggers                          = { "external_code",
"mesh_displacement",
"flow",
"turbulence" }
}
```

For each of the fields being solved (each stagger defined in the TIME_SEQUENCE command), you need to define the nonlinear iteration and linear solver parameters, using the STAGGER command. For example, the STAGGER commands are shown for external_code, mesh_displacement, and flow equations.

```
STAGGER ( "external_code" ) {
   equation                          = external_code
   min_stagger_iterations            = 1
   max_stagger_iterations            = 10
}

STAGGER( "mesh_displacement" ) {
   equation                          = mesh_displacement
   min_stagger_iterations            = 1
   max_stagger_iterations            = 5
   convergence_tolerance             = 0.1
   linear_solver                     = conjugate_gradient
   min_linear_solver_iterations      = 10
   max_linear_solver_iterations      = 1000
   linear_solver_tolerance           = 0.001
   projection                        = off
```

```
}

STAGGER( "flow" ) {
   equation                          = flow
   min_stagger_iterations            = 1
   max_stagger_iterations            = 2
   convergence_tolerance             = 0.1
   lhs_update_frequency              = 1
   linear_solver                     = gmres
   linear_solver_tolerance           = 0.1
}
```

Define Fluid-Structure Interface

Use the `EXTERNAL_CODE_SURFACE` command to define the interface between the fluid and structure. The command specifies the surface topology, as well as the interface proprieties.

In the following example, the interface surface topology is read from the file `wall.ebc` located in the `MESH.DIR` directory and consists of 3-node triangles. The `velocity_type`, `temperature_type`, and `mesh_displacement_type` define the interface properties (boundary conditions) and are:

```
EXTERNAL_CODE_SURFACE( "wall" ) {
   surfaces = Read( "MESH.DIR/wall.ebc" )
   shape             = "three_node_triangle"
   element_set = "tet fluid"
   velocity_type = wall
   temperature_type = tied
   mesh_displacement_type   = tied
   gap = 0
   gap_factor = 0
   external_code_tags = {}
}
```

The interface properties or boundary conditions are specified through the `mesh_displacement_type`, `velocity_type`, and `temperature_type` parameters. The `mesh_displacement` parameter defines whether the fluid mesh is tied to the solid mesh or allowed to slip against the solid mesh surface. Set the `mesh_displacement_type=tied` to tie the fluid mesh to the solid mesh, or `mesh_displacement_type=slip` to allow the fluid mesh to slide against the solid surface, which acts as a guide surface.

The `velocity_type` specifies how the fluid velocity behaves in relation to the structural mesh velocity. Set `velocity_type=wall` to tie the fluid velocity to the mesh velocity, or set `velocity_type=slip` for the normal component of the fluid velocity to be tied to the solid mesh velocity.

There are four possible interface combinations based on the settings of the `mesh_displacement` and `velocity_type` parameters. These are summarized in Table 23.

Table 23: Direct-Coupled Fluid-Structure Interaction Boundary Conditions

Fluid-Solid Interface Conditions		**Mesh Displacement**	
		Tied	**Slip**
Velocity Type	Wall	$x_{fluid} = x_{solid}$ $u_{fluid} = \left(x^{n+1}_{fluid} - x^{n}_{fluid}\right) / \Delta t$	$x_{fluid} \in S_{solid}$ $u_{fluid} \bullet n = \left(x^{n+1}_{fluid} - x^{n}_{fluid}\right) / \Delta t \bullet n$

Fluid-Solid Interface Conditions		Mesh Displacement	
		Tied	Slip
			$u_{fluid} \bullet t_i = \left(x^{n+1}_{solid} - x^{n}_{solid}\right) / \Delta t \bullet t_i \, i = 1, 2$
	Slip	$x_{fluid} = x_{solid}$ $u_{fluid} \bullet n = \left(x^{n+1}_{fluid} - x^{n}_{fluid}\right) / \Delta t \bullet n$	$x_{fluid} \in S_{solid}$ $u_{fluid} \bullet n = \left(x^{n+1}_{fluid} - x^{n}_{fluid}\right) / \Delta t \bullet n$

When the fluid is allowed to slide along the solid mesh, neighborhood searches between the fluid and solid meshes are continuous performed. The `gap_factor` parameter specifies a non-dimensional (with respect to the length of the element face) maximum allowable gap and the gap parameter specified a dimensional maximum gap distance between each quadrature point of the AcuSolve surface to the closest surface given by Radioss to check for gaps. If the distance is greater than the gap, the computation stops with an error message.

Communication between Radioss and AcuSolve

Radioss and AcuSolve can be run on heterogeneous and remote platforms which are located on the same network domain. The communication between Radioss and AcuSolve is via sockets. To start a co-simulation between Radioss and AcuSolve, one of the analysis software needs to initiate the communication process, while the other analysis software needs to connect to the initiated communication process.

In Radioss the socket port number (PORT) is specified by the third item of the `/IMPL/DYNA/FSI` data. The default port number is `10000`. This same port number must be specified in the `EXTERNAL_CODE` data in the `AcuSolve.inp` file. If the machine that Radioss is running on is named `linux_mach`, then the `EXTERNAL_CODE` will look like:

```
EXTERNAL_CODE {
communication = socket
socket_initiate = no
socket_host = "linux_mach"
socket_port = 10000
}
```

AcuSolve and Radioss are started independently. AcuSolve will wait for Radioss to initiate the socket connection. The time that AcuSolve will wait is determined by the `WAIT` parameter of the `/IMPL/DYNA/FSI` data. The default value is `3600` seconds.

Before the start of time step loop, basic information about the interface must be exchanged between the two codes. First, a number of parameters controlling the interface strategy must be set for both codes. Second, the physical parameters of the interface must be defined. Typically, both AcuSolve and Radioss start at the same time step; however, if there is a desire to have AcuSolve start first (to allow for the flow to develop), the `INIT` data of the `/IMPL/DYNA/FSI` data can be set to the initial AcuSolve time step at which the coupled analysis will start.

Time Steps and Data Exchanges

Both Radioss and AcuSolve need to use the same time step size and the total number of time steps should be the same. Also, the size of the time step needs to remain constant in both Radioss and AcuSolve. For example, for `1000` time steps of `0.001` seconds the AcuSolve and Radioss data are:

```
AUTO_SOLUTION_STRATEGY {
initial_time_increment = 0.001
max_time_steps = 1000
min_stagger_iterations = 1
max_stagger_iterations = 20
}
```

and

```
/RUN/wing/1
1.0
/IMPL/DTINI
0.001
/IMPL/DT/2
        12   0.0   120   1.0    1.0
```

For the Radioss input data `/IMPL/DT/2`, items 4 and 5 specify the time step scale factors. These need to be set to `1.0` in order to keep the Radioss time step constant.

At each time step, the forces and displacements are exchanged between Radioss and AcuSolve until they converge to a certain tolerance. Once convergence is achieved, the analysis continues on to the next time step. These exchanges are called "staggers" in AcuSolve. The minimum number of staggers (exchanges) should be set to `1` in both sets of input data. This is specified by the `min_stagger_iterations` parameter in the AcuSolve `AUTO_SOLUTION_STRATEGY` data and the `MINX` data in the Radioss `/IMPL/DYNA/FSI` data.

The maximum number of exchanges can be controlled by a combination of force and displacement convergence criteria and a hard limit on the maximum number of exchanges. The maximum number of exchanges is set by the `max_stagger_iterations` parameter in the AcuSolve `AUTO_SOLUTION_STRATEGY` data and the `MAXX` data in the Radioss `/IMPL/DYNA/FSI` data.

Force and displacement convergence tolerances are used to reduce the number of exchanges to the number needed to receive stable and accurate results. This can dramatically reduce run times while guaranteeing accurate results. The force and displacement tolerances are specified by the `FTOL` and `DTOL` data in the Radioss `/IMPL/DYNA/FSI` data. These are tolerances on the L2 norms of the force and displacement vectors. If these tolerances are set high, the solution time will be reduced, but solution accuracy may also be reduced.

Typically, only forces and displacements need to be exchanged. In special cases, such as a spinning beam, you can request that moments and rotations are also exchanged. This is done by setting the `DATA` parameter to `2` of the Radioss `/IMPL/DYNA/FSI` data.

Data Exchange Verification

In order to verify that the correct data is being exchanged between Radioss and AcuSolve, the `MSG` parameter can be set to a non-zero value of the Radioss `/IMPL/DYNA/FSI` data. The higher the value, the greater the amount of information that is written to the Radioss `.out` file for verification. This data includes the data communication control parameters, as well as the actual values of the forces and moments being exchanged.

Examples

Typically, you only need to define the damp surface to run Fluid-Structure Interaction Analysis with Radioss and AcuSolve. If the damp surface has a surface ID of `10`, the input data would be:

```
/IMPL/DYNA/FSI
 0 10 0 0 0 0 0 0 0 0
 0 0 0 0
```

In this case, all of the other FSI parameters are set to their default values. In Radioss, `0` means the default value and not explicitly zero (unless the default value is actually zero).

In order to reduce run times, you could set an upper bound on the number of exchanges, as well as setting higher values for the force and displacement convergence tolerances. For example, you could set the maximum number of exchanges to `5` and the force and displacement convergence tolerances to `0.01` and `0.005` respectively with the FSI data:

```
/IMPL/DYNA/FSI
0 10 0 0 5 0 0 0 0 0
0.01 0.005 0 0
```

In the case of a beam problem with tagged elements and the exchange of moments and rotations, the input data would be:

```
/IMPL/DYNA/FSI
20 0 0 0 0 1 1 0 0 0\
0 0 0 0
```

Where, `20` is the element set that contains all the beam elements. The beam element tags are based on the beam property ID's.

Radioss Structural FSI Data Setup

The structural analysis type is implicit dynamics and is solved using the General Newmark method, so the Radioss Engine Keyword is:

```
/IMPL/DYNA/2
```

Suggested Radioss time step convergence criteria are `0.001` for energy and force and `0.01` for displacement, as set on the Implicit Nonlinear Solution Control data:

```
/IMPL/NONLIN/1
123 0.001 0.001 0.01
```

Since the time step must remain constant, the Time Step Control method must allow for many time iterations without changing the time step and the scale factors on the time step must be `1.0`. The Time Step Control method 2 data should be:

```
/IMPL/DT/2
 12 0.0 120 1.0 1.0
```

Also, the number of allowable divergence iterations must be increased so that the time step is not adjusted. This is done by setting the Divergence Criterion to `5` with:

```
/IMPL/DIVERG/5
```

Finally, to write the applied loads from the fluid tractions and the resulting displacements to the Radioss results animation files, use:

```
/ANIM/VECT/FEXT
/ANIM/VECT/DISP
```

Model Locations

The Radioss and AcuSolve models must be co-located. In other words, fluid boundary mesh and the corresponding structural damp surface mesh match each other as physical boundary. The gap between these meshes should be small. To verify the conformity of exchange surfaces in AcuSolve, consult the `.Log` file for the structural mesh information, as shown below. Verify the number of damp surface elements and the minimum and maximum coordinates of the damp surface, as well as for matching bounding box and areas to help identify position, size and orientation mismatch:

```
acuSolve: No. external code nodes = 15050
acuSolve: No. external code tri3s = 0
acuSolve: No. external code quad4s = 15000
acuSolve: Min/Max x-coord = -3.812500e-02 3.812500e-02
acuSolve: Min/Max y-coord = -3.805000e-02 3.805000e-02
acuSolve: Min/Max z-coord = 0.000000e+00 3.812500e+00
acuSolve: Total surface area = 9.126732e-01
acuSolve: No. fluid damp surfaces = 5637
acuSolve: Min/Max x-coord = -3.812499e-02 3.812500e-02
acuSolve: Min/Max y-coord = -3.812494e-02 3.812498e-02
acuSolve: Min/Max z-coord = 0.000000e+00 3.812500e+00
acuSolve: Total fluid surface area = 9.128247e-01
```

At each time step the damp surface deforms, due to the fluid flow load. At the same time, the fluid mesh is moved to match the new damp surface. The average and maximum gap between the meshes is shown for each time step as:

```
acuSolve: Ave distance = 2.842759e-05
acuSolve: Max distance = 1.182390e-04
```

The gap should remain small for a valid solution.

Coupled Simulation Post-processing

There are a couple of options to post-process the coupled simulation:

- Post-process the structural solution and interface response using HyperView and the fluid solution and interface response using a post-processor compatible with AcuSolve, such as HyperView.
- Use a third-party post-processing software to post-process the combined structural and fluid solution.

Scaling of Quantities

You may apply a multiplier function in AcuSolve to the forces and/or moments imported from Radioss. Scaling fields may be useful when starting a fluid-structure interaction simulation with high inertial effects. Specify a multiplier function on the `EXTERNAL_CODE` command.

```
MULTIPLIER_FUNCTION("ramp" ) {
type = piecewise_linear
curve_fit_values = { 1, 0.0 ; 10 , 1 }
curve_fit_variable = time_step
  }

    EXTERNAL_CODE {
     …
```

```
multiplier_function = "ramp"
…
  }
```

Parallel Execution of the Solvers

AcuSolve and Radioss may be run in parallel on distributed memory clusters.

Platform Support

AcuSolve runs on Windows (64-bit) and Linux (64-bit).

Limitations

- If the damp surface contains beam elements, no other elements can be used to define the damp surface. In other words, beam elements cannot be mixed with other elements to define the damp surface.
- Quadratic and modified tetrahedral elements are not supported
- There must be a constant locked time step increment between two solvers
- Thermal Analysis is not supported at this time

Quasi-static Analysis Notes

When running a Quasi-static Analysis (`/IMPL/QSTAT`), the positive definite property of stiffness matrix will be reinforced by including the inertia matrix. The extra stiffness is a function of masses, inertia and the time step. Smaller time steps will add more stiffness. In addition, a scaling factor can be defined for the added matrix through the keyword `/IMPL/QSTAT/DTSCAL` (this factor is inversely proportional to this added matrix just like time step). For Nonlinear Analysis, this will only modify the convergence speed; whereas for Linear Analysis, time step (one step) should be chosen carefully (neither too small, as this will change the result significantly; nor too big, as this might lead to a non-positive definite matrix).

This option is quite suitable for a model that consists of parts connected only by contact interfaces and has failed with other analysis types. A Linear Quasi-static Analysis can also be used for model checking of high level (even for Explicit Analysis). When using a time step that is not too large, results can always be obtained, whether the model is well constrained or not.

As mentioned in Implicit Analysis Activation, when `/IMPL/QSTAT/DTSCAL` is used with `/IMPL/LINEAR/INTER` (two steps), the scaling factor is only applied in the second step. By carefully choosing this scaling factor and the termination time, the correct contact in the first step can be found and minimizes the error (due to added stiffness) on the final step.

A typical example with this methodology is the initial state simulation under gravity of a full car with dummies, in which dummies link the car only by contact. A short stop time is defined, so that a large displacement between the parts does not occur during the contact research on the first step, and a large quasi-static scaling factor is applied for the second step to get the static solution.

Nonlinear Implicit Analysis Notes

In Nonlinear Implicit Analyses, when the element formulations are different between the stiffness matrix building and the internal force calculation (same as explicit one), convergence issues are often observed; this is especially true for the reduced integration elements with perturbation hourglass controls. On the other hand, it is possible to obtain converged results with elements of similar formulations such as `QEPH` and `QBAT` for shells, or `HA8`, `HC8`, `S8` and `HEPH` for solids. In any case, same formulations are preferred.

As mentioned before, Radioss uses only the elastic part of the material parameters to build the stiffness matrix which allows working with all available Radioss materials. Another advantage of using the elastic stiffness matrix is that convergence issues are avoided with certain materials (such as elastoplastic). Some example tests have shown that the elastic stiffness matrix associated with line-search provides reasonable convergence for nonlinear material computations. Of course, the Newton-Raphson (using tangent matrix) method could also be added in the future.

The only exception to using the elastic part of material is in the case of spring elements, which are not associated with any material laws. When a nonlinear function is defined in the spring property for Implicit Nonlinear Analyses, the tangent stiffness elementary matrix will be computed. However, to avoid the convergence issues in Nonlinear Analysis, the elastic stiffness matrix is still used for elastoplastic springs (H>0). In Linear Analyses with spring elements, if nonlinear stiffness functions have been defined, a linear stiffness is calculated taking only the first points (besides 0, 0) of this function.

Convergence issues are frequently encountered in nonlinear analyses, in which the control parameters play an important role. These parameters are problem-dependent and the input values could determine the convergence or failure of a computation. The best values are often a good compromise between the quality and the performance.

First, control parameters for the nonlinear solver should be input.

The format for this keyword is:

```
/IMPL/NONLIN/n
L_A Itol Tol
```

Two nonlinear solvers are available:

- Modified Newton method (n=1)
- Quasi-Newton (BFGS) method (n=2)

L_A This parameter sets the maximum number of iterations for reforming the stiffness matrix; a value of 3 for iterative solver and a value of 6 for direct or mixed solver is recommended.

Itol Criteria used for run termination.

1	Relative residual in energy
2	Relative residual in force

Tol	Tolerance value; iteration is considered to be converging if the relative residual value (residual value versus reference value) is equal to or less than the tolerance.

The reference values are computed at zero iteration (Output Messages).

With Radioss, updated stiffness matrix has the sense mainly in geometry (using elastic parameter for nonlinear materials); therefore, a small value for *L_A* will provide faster convergence, depending on the displacement increment level, but with more computation cost for stiffness matrix reforming; that is especially true with direct solver.

Usually, stop criteria in force (*Itol* =2) provides a better result on balance, which is the default criteria. Criteria in energy might converge easily and is better suited for the simulation of elasto-plastic materials under monotonically increasing loading; because in this case, energy increases more strongly than the force (which is yield by the plastic admissible stresses), the convergence becomes easier due to the same evolutions of reference values (the higher the reference value easier the convergence).

Second, time step control should be chosen when using the keywords `/IMPL/DT/n` and `/IMPL/DTINI`. If a time step control method is not defined, constant time step is used during the nonlinear simulation (not recommended).

To manage time step more efficiently, two automatic time step control methods are available (n=1 or 2). The first method is a line-search method that works only according to converged iteration numbers. The second method is an arc-length like displacement control method and is recommended for general use.

When automatic time step control is used at each step (cycle), the Radioss solver automatically adjusts the time step with the input scale factors. If the iteration diverges, Radioss will scale down the time step and then restart the iteration.

The default scaling factors are 0.67 (for decreasing factor) and 1.1 (for max increasing that is maximum factor by which time step will be increased).

Using a max increasing factor closer to one, for example: 1.01, will enable easier convergence for difficult analyses, like buckling.

One should also define an initial time step using the keyword `/IMPL/DTINI`. This provides an approximate number of steps needed for the simulation.

An adequate number of steps may be needed (typically > 100) to run nonlinear simulations because:

1. Models with geometric or contact nonlinearity converge easier with small steps.
2. Some material behavior, especially path-dependant ones, need small steps to be accurately integrated.

It is also very useful to set the Min and Max limits for the time step using `/IMPL/DT/STOP`.

DT_min	For stopping the computation in the event of divergence.
DT_max	Sets a ceiling on the time step. The computation will not stop upon reaching this value; but it is useful for convergence of highly nonlinear simulations or just for the output need.

Restarting Radioss is also available with implicit analysis. In this case, different parameters can be defined for converging strategies.

Nonlinear Analysis with Contact

The relative force residual (default) is recommended for Nonlinear Analyses with contact.

Like Explicit Analysis, a larger gap for contacts leads to better convergence. This is especially true for interface TYPE 11, where a sufficient gap value has to be set in order to avoid termination of computation due to decreased time step. For simulations with contact (for example: initial state of stamping under gravity, parts constrained by contacts) using interface TYPE 7 with a small initial gap, defining Gap_{min} slightly larger than the initial gap will lead to better convergence. The initial penetrations will recover quickly.

If contact is defined with friction, then incremental stiffness formulation (I_{form} =2) is recommended.

Since it is difficult to choose an appropriate reference value in analyses with contacts (especially in simulations involving rigid parts with imposed displacement impacting a deformable part), Radioss makes an exception by modifying the reference value in the first iteration. If the relative residual becomes too large (>>1), Radioss will restart the iteration with a smaller time step and a higher reference value. The modified reference value is sometimes arbitrary and too high, so you should examine this value (printout at zero iteration at each step), as this may lead to convergence with the wrong result. This could also be true when high initial penetrations are present in an interface definition.

When interface TYPE 7 is used, the contact stiffness plays an important role in convergence and therefore, I_{stf} = 4 (which takes the minimum of main and secondary stiffness's for contact) is recommended. This is due to the reason mentioned in the previous paragraph and also because the penalty contact force will be balanced with the internal force of deformable impacted part. That means the stiffness near the effective stiffness one will converge easier than a higher one. Sometimes, a stiffness with a scaling factor reduction (for example: *Stfac* 0.01) or reduction in impactor thickness (if rigid one) might reduce unbalanced forces and improve convergence, particularly in shell structures under bending where the effective stiffness is much lower than membrane stiffness; but it should be noted that too low of a value could also lead to divergence.

Fluid-Structure Interaction Analysis with Radioss and AcuSolve

Note: Support for Direct Coupled Fluid-Structure Interaction (DC-FSI) Radioss and AcuSolve has been discontinued. It is now recommended to use OptiStruct coupled with AcuSolve as described in Fluid-Structure Interaction in the *OptiStruct User Guide*.

Radioss and AcuSolve are fully-integrated to perform a Direct Coupled Fluid-Structure Interaction (DC-FSI) Analysis based on a partitioned staggered approach. Radioss and AcuSolve are both time domain simulation codes that break the coupled simulation into a number of time steps. Because the governing equations of both Radioss and AcuSolve are nonlinear, sub-iterations are typically required within each time step. At each sub-iteration of FSI analysis, the fluid tractions in AcuSolve are converted into nodal forces which are then transferred to the structural interface mesh of Radioss. These forces are used to calculate the deformation of the structure using Radioss. Note that in addition to the load from the fluid flow tractions, additional structural loads can also be applied. The resulting deformed shape of structure is passed back to AcuSolve as the new fluid boundary. This FSI cycle is shown below in Figure 396.

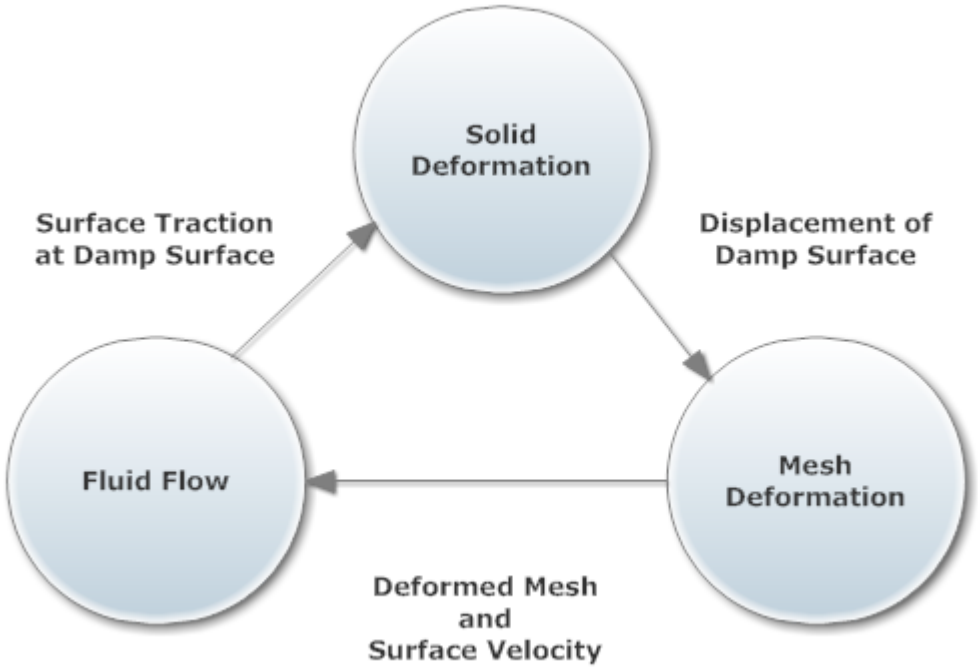


Figure 396: Direct-Coupled Fluid-Structure Interaction (DC-FSI) Cycle

Target Applications

The DC-FSI capability aims at simulations of compliant structural response to fluid flow and its complex interrelationship. This capability is particularly suited for problems exhibiting nonlinear structural response. It may also be used for linear structural response; however, for linear structural response the P-FSI solution offered by AcuSolve may be more effective in solving the linearized structural response with the nonlinear flow solution. For further information on the P-FSI method, refer to the AcuSolve Command Reference Manual. Most Radioss features including nonlinear materials, nonlinear geometric effects, and contact are available for use. The interface domain can be modeled with solid, shell, and

beam elements. Most of the features in AcuSolve can be used in DC-FSI simulations. These features may include, namely laminar or turbulent flow, heat transfer, a rich set of material options, free-surface and multi-species driven flows. The moving boundary problems are modeled using Arbitrary Lagrangian-Eulerian (ALE) and sliding mesh technologies. For further details, consult the AcuSolve *Command Reference Manual*.

Primary applications targeted by this capability include:

- Automotive: Hydraulically damped rubber mounts, door seals, shock absorbers, design of valves and rubber diaphragms and antilock braking systems
- Oil/Gas: Long marine risers, moorings, free spans and drilling risers
- Aerospace: Wing aero-elasticity, UAV and MAV
- Wind turbine: Large deformation of blades
- Consumer Goods: design and packaging
- Bio-medical

See Also
Fluid-Structure Interaction

Output Messages

The following is part of a printout (in step 2 of the Engine output file `_0001.out`) when the keyword `/IMPL/PRINT/NONLIN/n` is used in the Engine file (`_0001.rad`):

```
CYCLE    TIME    TIME-STEP  ELEMENT  ERROR  I-ENERGY   K-ENERGY T  K-ENERGY R  EXT-W
-------
   2   2.380      0.4802     SHELL   0.0%   5.117      0.000           0.000  5.117
    --SUPPLEMENTARY CONTACT STIFFNESS MATRIX IS CREATED--
     WITH DIM. : ND   =     141 NZ  =    1050 NB_MAX =      15
    NL_ITERATION=    0   INITIAL RESIDUAL NORM= 0.5913E+01 0.4613E+04 0.1600E+05
    --STIFFNESS MATRIX IS REFORMED --
    --SUPPLEMENTARY CONTACT STIFFNESS MATRIX IS CREATED--
     WITH DIM. : ND   =     150 NZ  =    1059 NB_MAX =      15
    NL_ITERATION=    1  RELATIVE RESIDUAL NORM= 0.1000E+01 0.8593E-01 0.5005E-01
    --SUPPLEMENTARY CONTACT STIFFNESS MATRIX IS CREATED--
     WITH DIM. : ND   =     156 NZ  =    1101 NB_MAX =      15

  CONVERGED WITH 2 ITERATIONS, |du|/|u|,|r|/|r0|,|dE|/|E|= 0.5915 0.4747E-01 0.1166
```

Each iteration contains a printout of the normalized (L2 Norm) initial values (iteration zero) or relative one of displacement, force residual and energy. So `|du|/|u|` is relative residual in displacement, `|r|/|r0|` is relative residual in force and `|dE|/|E|` is relative residual in energy.

Stiffness Matrix Setup

At the start of every computation, the following message is printed:

```
   *--------- STIFFNESS MATRIX SETUP ---------*
SYMBOLIC DIM : ND =  164202 NZ =   6047160 NB_MAX =      96
FINAL    DIM : ND =  154605 NZ =   5652417 NB_MAX =      96
```

Where,

ND	Dimension of the stiffness matrix (number of DOFs)
NZ	Number of non-zero values of the strict triangular matrix (only `ND+NZ` values are stored due to the symmetry of matrix)
NB_MAX	Maximum number of connected nodes in the model

Radioss eliminates all dependent nodes (this process is called 'condensation') before resolution, and so the `SYMBOLIC` and `FINAL` dimensions correspond respectively to the matrix dimension before and after condensation.

If contact is taken into account, the stiffness matrix of penalty spring is stored independently and information on this is printed (if `/IMPL/PRINT/NONLIN/n` is defined) once contacts are detected:

```
--SUPPLEMENTARY CONTACT STIFFNESS MATRIX IS CREATED--
     WITH DIM. : ND   =      42 NZ  =      42 NB_MAX =      16
```

Note: Only information of the final (after condensation) matrix information is printed out.

In the Radioss SPMD version, the sub-matrix of the main stiffness matrix stored on each machine is printed; and for the supplementary contact matrix, only the total final one is printed.

- CPU information

If the keyword `/MON/ON` is defined in the Radioss Engine input file (`_0001.rad`), some detailed information regarding CPU processing time information for the implicit scheme is printed in the `.out` file:

```
#PROC  [K]SETUP [M]SETUP IMPL.SOLV IMPL.TOTAL
   1   .2484E+01 .1528E+02 .6742E+02 .8520E+02
 #PROC  CONT.SORT  CONT.  F ELEMENT  MAT KIN.COND.  INTEGR.     I/O     TASK0  ASSEMB.    RESOL
   1   .0000E+00 .0000E+00   .1875     .3125E-01     .0000  .0000E+00  .2188 .2188  .0000   .8561E+02
----------
 ELAPSED TIME :     89.32 s
```

Where,

$[K]$	Main stiffness matrix
$[M]$	Pre-conditioning matrix (if iterative solver is used)
IMPL.SOLV	For the resolution of {x}=[K]-1{b}

The numerical values represent time.

Output Solving for each Iteration (/IMPL/SOLV and /IMPL/NONLIN/SOLVINFO)

Output global solve information at each solving iteration. This command creates three additional files:

Filename_implicit.info	For largest residuals (force and moment), largest displacement and rotation increments and largest corrections at each iteration and each increment.
Filename_implicit.hist	Contains all data for plotting purposes in Altair Simulation.
Filename_implicit.tpl	A `tpl` script for plotting in Altair Simulation the data from `.hist` file.

Troubleshooting

When the Radioss stops with the one of following error messages:

If this error message is printed, two cases are possible.

```
STIFFNESS MATRIX IS NOT DEFINITE
```

1. The stiffness matrix $[K]=[K0]+[Kg]$ (geometrical) is not definite. This can happen when some of the DOF values in stiffness matrix are zero (for example: one of the rotational stiffness of a spring TYPE 8 is defined as zero), or elastic modulus for a material is too small (often related to inconsistencies in the units used in the model).
2. The model still has rigid body modes even after boundary conditions have been applied. This is a more common source of this error. Examples include two parts that are connected only by contact interfaces or a 6 degrees of freedom spring connect to a 3 degrees of freedom solid element.

   Simulations fail due to the implicit resolution (matrix inverse), even with an iterative (PCG) solver.

   Radioss implicit will automatically fix these free degrees of freedom in the first case (see *Radioss Engine Input* for more details).

   To check the rigid body modes in the second case, employ one of the following methods by running:

   - A model check (with `/IMPL/CHECK` in the Engine Input file). The results of this check will be listed in the `*n.out` file.
   - A Linear Quasi-static Analysis (a direct solver is preferred) with a sufficiently large time step. The free part can be found by reviewing the animation files.

Some possible workarounds for eliminating such rigid body modes are:

- Adding artificial springs with (small stiffness) to connect the free parts.
- Creating slight initial penetration in the contacts, if interfaces are defined between the free parts.
- Run a Quasi-static Analysis.

The error message below occurs when no loading has been defined in the model or when loading is zero during the first step in a nonlinear analysis.

```
IMPLICIT STOPPED DUE TO LOADING DATA
```

The run terminates with the following Radioss message:

```
**RADIOSS IMPLICIT STOPPED DUE TO LOADING DATA**
ISTOP =       0
```

A few suggested workarounds are:

1. Check loading in the model.
2. Increase the initial time step.
3. Creating a slight contact penetration in cases where loading is due to contact.

If the error message shown below is printed:

```
TIMESTEP REDUCED DUE TO DE-ACTIVATION
```

The following warning could be printed in nonlinear simulations with contact:

```
**TIMESTEP WILL BE REDUCED TO AVOID DE-ACTIVATION IN INTERFACE**: 14
```

In such a case, Radioss will try to restart the iteration with reduced time step and eventually the run might fail, if the time step drops below the prescribed minimum. This is usually seen in models with interface TYPE 11 (edge to edge contact) with small gap.

```
** ERROR : RADIOSS IMPLICIT STOPPED DUE TO MODELING DATA **
     ISTOP=      -1
```

This message is generally following different modeling error messages: "STIFFNESS MATRIX IS NOT DEFINITE", "...TOO LARGE ENERGY VALUE".

Often, it was about the constraint conditions or too large time step.

```
**ERROR: RADIOSS IMPLICIT STOPPED DUE TO TIMESTEP LIMIT**
    ISTOP =      -2
```

When the time step is smaller than the input minimum one (see /IMPL/DT/STOP), Radioss will stop the run with this message.

The reason is generally due to the convergence issue (DT_min), if DT_min was already small enough or not. There is no general workaround for convergence issue, as outlined below:

1. Adding more printouts in out files (/PRINT/n, for example, *n*=1, /IMPL/PRINT/NONLIN/n, for example, n=1), and analyzing the reason of failing.
2. Check the model, constraint conditions, materials, contact interface defines and so on.
3. If the fail was due to instability, try /IMPL/DT/3 (Riks method) for snap-thru type simulations or /IMPL/QSTAT for general instability case.
4. Switching other convergence criteria (refer to *Itol* in /IMPL/NONLIN) or losing the tolerance, if possible.
5. Try /IMPL/NONL/SMDISP to see if the issue came from geometric nonlinear.

```
IMPLICIT STOPPED DUE TO SOLVER ERROR
```

The run terminates with the following Radioss message:

```
"**ERROR: RADIOSS IMPLICIT STOPPED DUE TO SOLVER ERROR**
     ISTOP =      -4"
```

Two suggested workarounds are:

1. Check which solve method is used in /IMPL/SOLVER. Has the direct solver (BCS) been used in SMP run? If it is, then use another solve method.
2. Another frequent case is when the model is relatively large and the available free disk is not large enough. In that case, check the working space free volume (in ./ if TMPDIR is defined). If the issue came from memory management, you could also force Radioss to use out-of-core resolution (/IMPL/SBCS/OUTCORE).

If this error is still there with the above workarounds, contact your local Altair Simulation support team.

If BCS or mix solver (using BCS) was used, BCS error IDs will printout on the screen of running shell, report them to support team.

Appendix

Examples of Engine File for Implicit Analysis

Linear Static Analysis

```
# RADIOSS ENGINE
/TITLE
Implicit linear computation
/RUN/PLABUCK/1
0.1
/VERS/140
/TFILE
0.0001
/PROC/1
/PRINT/-1
/ANIM/DT
0. 0.01

/IMPL/PRINT/LINEAR/-100
#Linear solver method (Direct solver MUMPS)
/IMPL/SOLVE/2
# Iprec It_max Itol Tol
5 0 3 0.0
#    1  0  0  0.
/IMPL/BUCKL/1
1.0e-10 0. 20 2 0 1.e-4

/ANIM/VECT/DISP
/ANIM/ELEM/ENER
/ANIM/ELEM/EPSP
/ANIM/ELEM/VONM
/ANIM/TENS/STRESS/MEMB
/ANIM/TENS/STRESS/BEND
/ANIM/GZIP
/MON/ON

Nonlinear Static Analysis
# RADIOSS ENGINE
/TITLE
Implicit computation
/VERS/140
/RUN/pcompp_rad_s1/1
0.1000000E+01
/ANIM/DT
0.0000000E+00 0.1000000E+00
/TFILE
0.1000000E-01
/RFILE
5000
/PRINT/-1

/MON/ON
/ANIM/VECT/DISP
/ANIM/ELEM/EPSP
/ANIM/GZIP

#---------------------------------------------------------
# STATIC Non-Linear IMPLICIT OPTIONS #
#---------------------------------------------------------
# Printout frequency for nonlinear computation:
/IMPL/PRINT/NONL/-1
# Static nonlinear computation (modified Newton method):
/IMPL/NONLIN/1
# L_a Itol Tol
0 2 0.5000000E-01
# Linear solver method (Mix):
/IMPL/SOLVER/5

#Initial timestep:
/IMPL/DTINI
0.1000000E+00
```

```
# Min-max values for timestep:
/IMPL/DT/STOP
# Dt_min Dt_max
0.1000000E-04 0.3000000E+00
# Timestep control method 2 (arc-length + line-search):
/IMPL/DT/2
# It_w L_arc L_dtn Scal_dtn Scal_dtp_max
6 0.0000000E+00         20 0.6700000E+00 0.1100000E+01

/END/ENGINE
```

Nonlinear Static Analysis

```
#-------------------------------------------------------
# STATIC Non-Linear IMPLICIT OPTIONS #
#-------------------------------------------------------
# Printout frequency for nonlinear computation:
/IMPL/PRINT/NONL/-1
# Static nonlinear computation (modified Newton method):
/IMPL/NONLIN/1
# L_a Itol Tol
0 2 0.5000000E-01
# Linear solver method (Mix):
/IMPL/SOLVER/5

#Initial timestep:
/IMPL/DTINI
0.1000000E+00
# Min-max values for timestep:
/IMPL/DT/STOP
# Dt_min Dt_max
0.1000000E-04 0.3000000E+00
# Timestep control method 2 (arc-length + line-search):
/IMPL/DT/2
# It_w L_arc L_dtn Scal_dtn Scal_dtp_max
6 0.0000000E+00         20 0.6700000E+00 0.1100000E+01

/END/ENGINE
```

Fluid and Fluid-Structure Simulation

In this section, fluid and fluid-structure simulation is presented.

Arbitrary Lagrangian-Eulerian (ALE) and Computational Fluid Dynamics (CFD)

With Arbitrary Lagrangian-Eulerian (ALE) and Computational Fluid Dynamics (CFD) Simulation, the following phenomena can be modeled:

- Transient Analysis
- Laminar and turbulent flow ($k-\varepsilon$ model, LES Smagorinsky)
- Compressible and semi-incompressible flow
- Conductive heat transfer
- Fluid/structure coupling
- Multi-material

The most used fields of application are:

- Classical fluid flow analysis
  - Internal flows
  - Aerodynamics
  - Free surface
  - Open channel with obstacles
  - High speed tooling
- Fluid/structure interaction
  - Valves
  - Exhaust noise source prediction
  - Laser welding
  - Nuclear accidents
  - Defense

Summary of ALE Keywords

Usual Radioss keywords will act on material at related grid points. All the parameters starting with `/ALE` will directly act on the grid points.

Description

To activate ALE formulation with a given material law, add:

- /ALE/MAT

ALE formulation requires defining a grid formulation:

- /ALE/GRID/DONEA
- /ALE/GRID/DISP
- /ALE/GRID/SPRING
- /ALE/GRID/STANDARD (recommended)
- /ALE/GRID/ZERO

The Eulerian formulation can be defined in two ways. In this case, grid points remain fixed.

- /ALE/MAT + /ALE/GRID/ZERO
- or /EULER/MAT

A modeling problem is well posed with boundary conditions on both material and grid velocities:

- /BCS (constraining material velocity on nodes)
- /ALE/BCS (constraining grid velocity on nodes)
- /EBCS (constraining elementary state)

It is also possible to define Lagrangian nodes within an ALE part by using `/ALE/BCS` or merging a Lagrangian element.

Define links on grid velocities with:

- /ALE/LINK/VEL

This is often used in FSI modeling to window a moving free structure with an ALE grid. For example, in ditching, a limited ALE domain can be linked to free interfaced structure. The advantage is that the water and air inlets are constant, since it is initial state with no material velocity.

The specific ALE interfaces are:

- /INTER/TYPE1 (ALE nodes on a Lagrangian surface)
- /INTER/TYPE9 (interfacing a Lagrangian surface with a given ALE free surface)
- /INTER/TYPE12 (enables the transmission of flow between 2 ALE surfaces (main and secondary side))
- /INTER/TYPE18 (coupling with Lagrange structure)
- /INTER/TYPE22 (cut cell method, BETA software release)

There are seven material laws compatible with the ALE or EULER formulation:

Multi-material laws
/MAT/LAW37 (BIPHAS) (bi-material liquid gas)
/MAT/LAW20 (BIMAT) (general 2D bi-material law)
/MAT/LAW51 (MULTIMAT) (general 3D multi-material law, scattered scheme)

Multi-Phase material law
/MAT/LAW26 (SESAM) (SESAME-Johnson-Cook material)
/MAT/LAW151 (MULTIFLUID) (general 3D multi-material law, collocated scheme)

Boundary material law
/MAT/LAW11 (BOUND) (Inlet/Outlet material)

Thermal law
/MAT/LAW18 (THERM) (Thermal material)

Besides the above ALE materials, the following material laws can be used in ALE/EULER analysis:

- /MAT/LAW2 (PLAS_JOHNS)
- /MAT/LAW3 (HYDPLA)
- /MAT/LAW4 (HYD_JCOOK)
- /MAT/LAW5 (JWL)
- /MAT/LAW6 (HYDRO or HYD_VISC)
- /MAT/LAW10 (DPRAG1)
- /MAT/LAW11 (BOUND)
- /MAT/LAW16 (GRAY)
- /MAT/LAW21 (DPRAG)
- /MAT/LAW37 (BIPHAS)
- /MAT/LAW41 (LEE_TARVER)
- /MAT/LAW46 (LES_FLUID)
- /MAT/LAW49 (STEINB)
- /MAT/LAW51 (MULTIMAT)
- /MAT/LAW75 (POROUS)
- /MAT/LAW80

The second order MUSCL scheme for reconstruction of volumetric fractions can be invoked through:

- /ALE/MUSCL

To define inlet/outlet:

- /EBCS
- /MAT/LAW11 (BOUND) (general case, except if using LAW37 or LAW51)
- /MAT/LAW37 (BIPHAS) (see elementary formulation when using multi-material LAW37)
- /MAT/LAW51 (MULTIMAT) (see elementary formulations when using multi-material LAW51)

The initial velocity for material located at grid point is defined with:

- /INIVEL (scattered scheme)

The initial velocity for collocated scheme (LAW151) can be defined with:

- /INIVEL/FVM

Non-reflecting outlet boundary condition for material LAW151 can be setup with:

- /EBCS/FLUXOUT

Inlet boundary condition for material LAW151 can be setup with:

- /EBCS/INLET

The hydrostatic pressure field, due to gravity, can be initialized with:

- /INIGRAV

The initial volumetric fractions of materials LAW51 and LAW151 sub-materials can also be defined with:

- /INIVOL

Velocities and thermodynamic values can be mapped into the ALE domain by using:

- /INIMAP1D (1D to 2D or 1D to 3D mapping)
- /INIMAP2D (2D to 3D mapping)

Specific rigid wall for ALE is:

- /RWALL/THERM

Finite Volume formulation for internal force calculation can be activated with:

- /ALE/SOLVER/FINT *Iform*=2

It provides more accurate results for non parallelepipedic shapes. See default formulation for under-integrated formulation in Reduced Integration Method in the *Theory Manual*.

For detonics applications, the following keywords are available:

High Explosive material laws

/MAT/LAW5 (JWL)
/MAT/LAW41 (LEE_TARVER)
/MAT/LAW51 (MULTIMAT)
/MAT/LAW151 (MULTIFLUID)

High Explosive Ignitions

/DFS/DETLINE (detonation line)
/DFS/DETPLAN (planar wave)
/DFS/DETPOINT (detonation point)
/DFS/WAV_SHA (wave shaper)

For laser matter interaction, the following keywords are available:

- /DFS/LASER (LASER beam)
- /MAT/LAW26 (SESAM) (SESAME-Johnson-Cook material - only compatible material)

The following equation of states are available in Radioss:

- /EOS/COMPACTION
- /EOS/OSBORNE
- /EOS/LINEAR
- /EOS/POLYNOMIAL
- /EOS/GRUNEISEN
- /EOS/PUFF
- /EOS/TILLOTSON
- /EOS/NOBLE-ABEL
- /EOS/IDEAL-GAS or /EOS/IDEAL-GAS-VE
- /EOS/STIFF-GAS
- /EOS/LSZK
- /EOS/MURNAGHAN
- /EOS/SESAME

Specified parts can be activated or deactivated from ALE computations with:

- /ALE/ON

- /ALE/OFF

Choice of Formulation

The kinematical description of the continuum determines the relationship between the deforming continuum and the mesh of computing domain. The studies of continuum mechanics usually make use of two classical descriptions of motion:

- Euler description
- Lagrangian description

The Arbitrary Lagrangian-Eulerian description was developed later to combine the advantages of the above classical kinematical descriptions, while minimizing their respective drawbacks as much as possible.

Euler Formulation

The Eulerian formulation is classical in fluid mechanics. The mesh is fixed and the material flows through the mesh. Equations are modified with respect to Lagrangian formulation in order to take into account the convective terms.

It can be activated for a specific part by a flag in material data:

`/EULER/MAT/mat_ID`

Where,

mat_ID	Identification number of the material to be set Eulerian

The treatment of moving boundaries and interfaces is difficult with Eulerian elements. The Eulerian formulation cannot be used in many cases where the boundaries of the domain move.

Lagrangian Formulation

The Lagrangian formulation is classical in structural analysis. The mesh is tied to the material points and follows the material deformation. No sliding between material (structure) and mesh is allowed. Loads and boundary conditions can easily be applied to the material points (nodes).

The Lagrangian description allows easy tracking of free surfaces and interfaces between different materials. However, when the structure is severely deformed, Lagrangian elements become similarly distorted since they follow the material deformation. Therefore, in those cases the accuracy and robustness of the Lagrangian simulations deteriorates severely.

This is the default formulation in Radioss, that is if a material is not defined as Eulerian (`/EULER/MAT/mat_ID` option), nor as ALE (`/ALE/mat_ID` option), this material is Lagrangian.

ALE Formulation

ALE stands for Arbitrary Lagrangian-Eulerian formulation. Material flows through an arbitrary moving mesh. Both the material and the mesh move with respect to the laboratory. It looks like a combination of Lagrangian and Eulerian formulations.

This formulation can be activated in Radioss for a specific part by a flag in the material data:

`/ALE/MAT/mat_ID`

Where,

mat_ID	Identification number of the material to be set in ALE

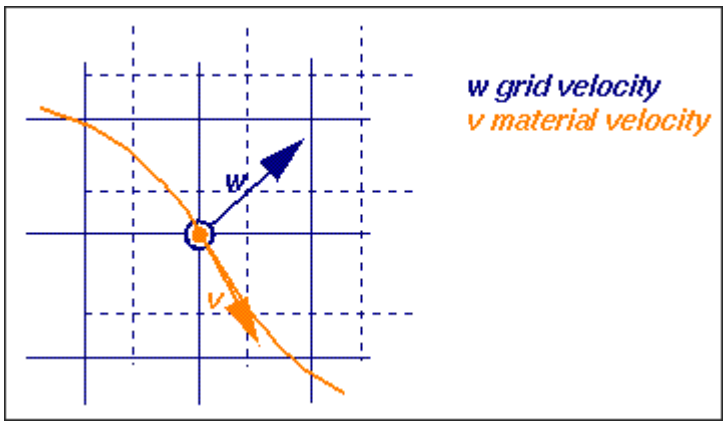


Figure 397: Arbitrary Grid Velocities and Displacements

In practice, built-in algorithms determine smooth grid deformation according to displacements of the ALE domain boundaries. Several algorithms are available (DONEA, SPRINGS, DISP, and ZERO).

It is worthwhile to note that ALE formulation can be degenerated in Lagrangian (w=u: the grid velocity is equal to the material velocity) or in Eulerian (w=0: the grid velocity is set to zero).

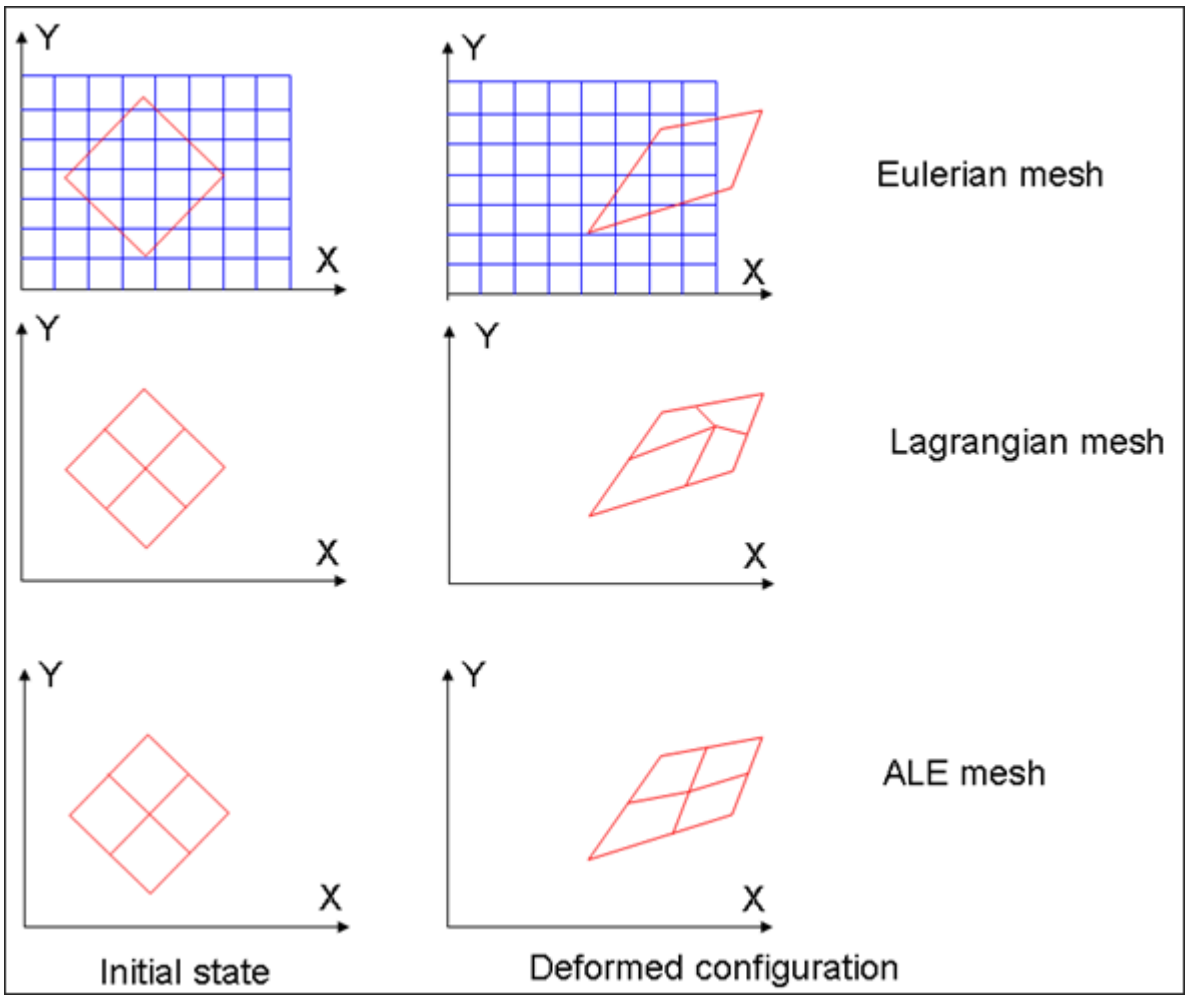


Figure 398: Eulerian, Lagrangian and ALE Meshes

Boundary nodes between ALE and Lagrangian materials must be set to Lagrangian: grid and material velocities are equal. Boundary nodes between ALE and Eulerian materials must have their grid velocity set to zero.

Both conditions are set using the `/ALE/BCS` option: You can specify extended boundary conditions for ALE nodes (grid velocity components can be set to 0 or to the material velocity). The grid velocities can be imposed with ALE links to any nodes in a similar manner to classical kinematic conditions (option `/VEL/ALE` in Engine).

Note: When a solid element is connected to a shell element, the connected nodes are automatically set as Lagrangian.

Fluid/Structure Connection

At least one row of ALE elements should be used when fluid is in contact with shells.

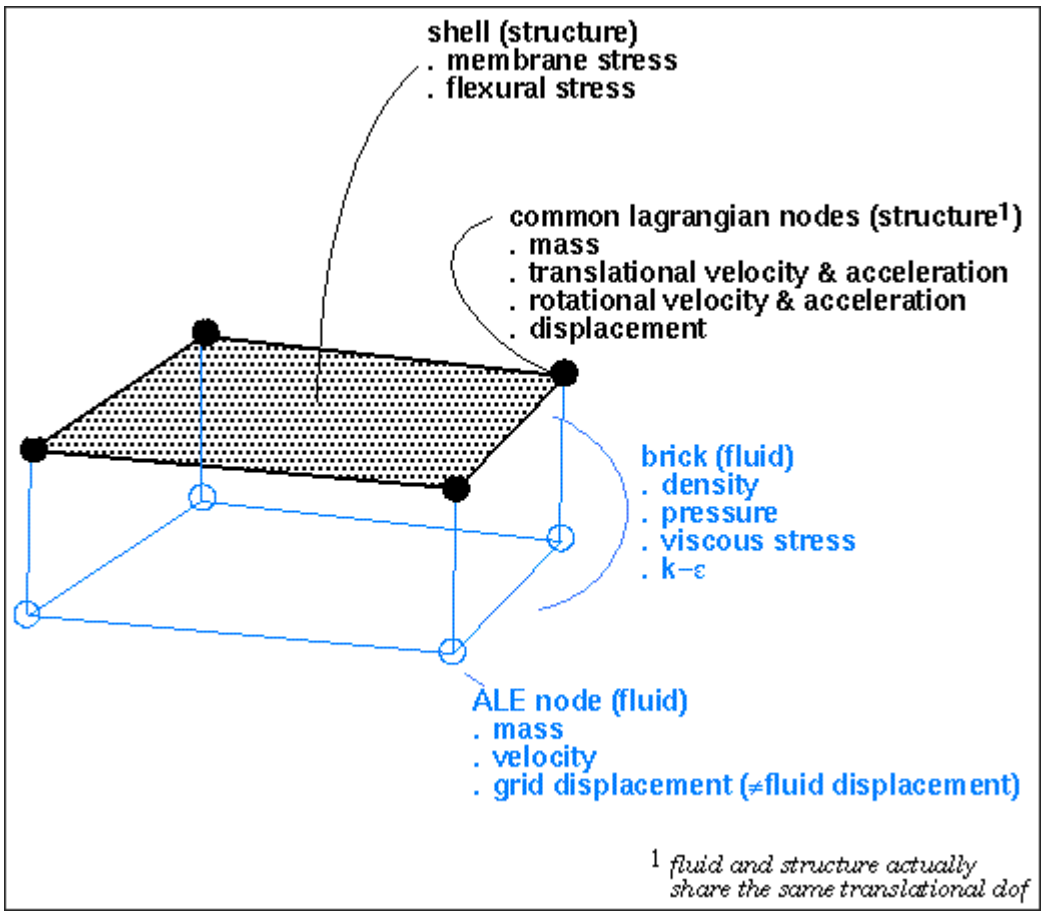


Figure 399: Mesh for Fluid-Structure Interaction Example

Methodology

The purpose of this section is to show the several steps for the generation of a proper Radioss ALE/CFD model.

CAD Cleaning

Usually, one receives a CAD model according to which the model should be built. The first task is to clean this CAD and perform simplifications:

- Add surfaces to close volumes used by automatic tetrahedral mesh
- Add surface to control mesh progression
- Patch surfaces to prepare mesh zones
  - Remove details whose size is smaller than what can be solved
  - Remove line constraints on surfaces whenever possible

Mesh Generation

Only hexahedron elements (or quad elements in 2D analysis) are compatible with the ALE or Eulerian formulations. To create the mesh, the usual technique consists in meshing at first outer surfaces of each considered domain with triangles, then generating automatically an internal volume tetrahedron mesh and finally split the tetra's in four hexa each (watch closely the number of elements). Triangle mesh size should be 3.5 times larger than the planned mesh size for tetrahedral.

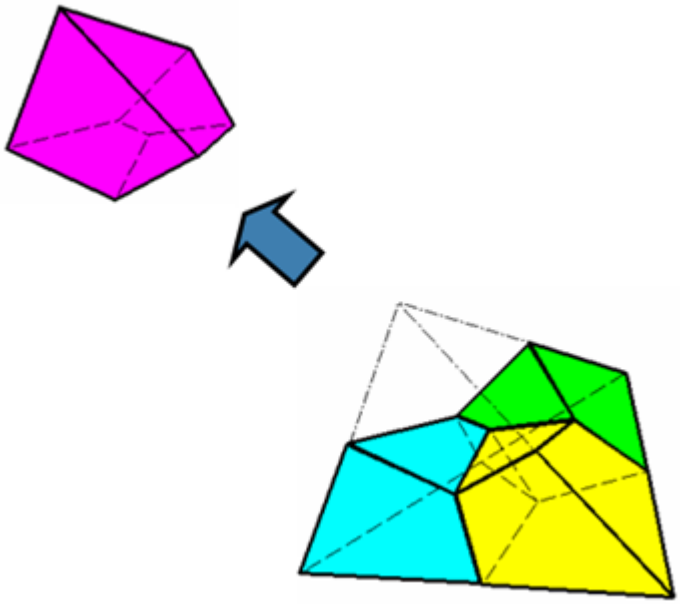


Figure 400: Tetrahedron Transformed to Hexahedron Mesh

This remesh function is found in HyperWorks under “split”. To remesh tetra element to hexahedron element, select **split into hexas**.

Figure 401:

Some extrusions are added whenever necessary, for example:

- Outlet
- Inlet
- Non-reflective frontiers (NRF)
- Tubes

Of course, any other technique is suitable if generating hexahedron elements.

For boundary layers, the wall element size is determined as explained in Wall Element Size. The mesh for boundary layer is generally obtained by meshing the wall surface using triangles. Then, by extruding

the surface mesh, a solid pentahedron mesh is created. The pentahedron elements can afterward be split easily to hexahedrons, as shown in Figure 402.

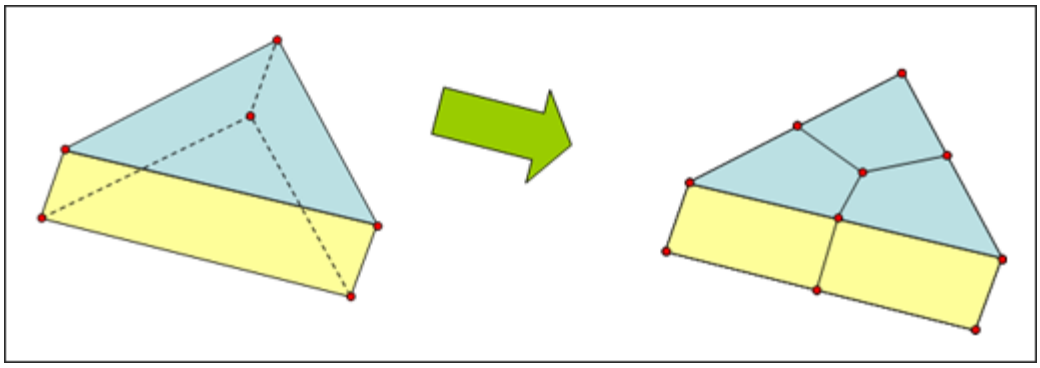


Figure 402: Split of a Pentahedron Element to Three Hexahedrons

Define Mesh Characteristics

Two aspects are involved when considering the mesh size:

- Velocity and pressure gradients: Near turbulent walls the mesh size is governed by the y+ value, which can range from 100 to 1000 or even higher (In pipes, y+ values as high as 8000 provide accurate results). From there, geometric progression is generally used toward coarser regions.
- Acoustic accuracy: A maximum size can be derived from the minimum wave length of interest. Generally use of 12 elements per wave length is acceptable.

Obviously the first condition will be governing region close to the obstacle or wall and the second will constrain the maximum size in the whole computational domain.

To build up a mesh, some trade-off is generally needed. The total time, $Ttot$, to be simulated can be evaluated as:

$$Ttot \sim 20 \cdot L / c$$

Where,

L	Largest model size

The total CPU needed, which is the major criterion to establish a trade-off between feasibility and accuracy can be estimated as:

CPU = Ttot/dtc * Number of element * cpu/el/cycle

The generation of the mesh in order to perform the desired simulation has to be carefully defined according to two criteria and a trade-off between feasibility (CPU time available) and precision.

Criteria 1: Advection

- 10 elements minimum per vortex to be solved
- Local Strouhal number not exceeding 1/6 for the frequency range of "interest" in regions of acoustic sources:

  Str = f h/v < 1/6

  For example: fmax =600 Hz, v=30m/s ≥ h ~ 8 mm

Criteria 2: Acoustic Propagation

- Critical in region away from acoustic sources
- Six elements per wave length along direction of propagation

  For example: fmax = 600 Hz, c=300 m/s ≥ h ~ 8cm

Trade-off

Between the available CPU total time to be simulated and the critical time step; the time step will automatically be set to satisfy Courant's condition:

dt = Min (h) / c

Total time should be a multiple of the smallest frequency present in the model. Total CPU time is proportional to the number of cycles (final time divided by time step), the number of elements and the cost of each element (depends on the computer):

T = Ncycle * Nelem * Cost / cycle / elem

Practically, it is advised to perform a timing on a couple of cycles (do not forget to subtract the time of initialization) in order to know the cost per cycle of the simulation before launching a large simulation.

Consider the case of airflow passing over an obstacle is studied (Figure below) where the aim of the simulation is to measure the noise generated by this flow in any location of the mesh. The mesh should have at least four different regions (refer to Mesh Generation):

- Coarse mesh (all the computational domain except the immediate surroundings of the obstacle)
- Refined mesh (close to the obstacle)
- Inlet element (one row of elements)
- Outlet elements (one row of elements)

With l being the characteristic size of the obstacle in the flow, there are typically three different mesh zones:

Zone	Typical Size of Elements	Note
very fine	a	Chosen such as the obstacle is descretized with a minimum of 20 cells in each direction.
fine	2a to 3a	
course	4a to 6a	Make sure the course cells are appropriate for convection of the highest frequencies of interest.

If the highest frequency of interest in the problem is f, then no cell in the mesh should be bigger than:

Size of element < C / 10.f

Where,

C	Speed of sound in the fluid

Inlet and outlet element thickness should be 1/10 of the neighboring elements of the computational domain.

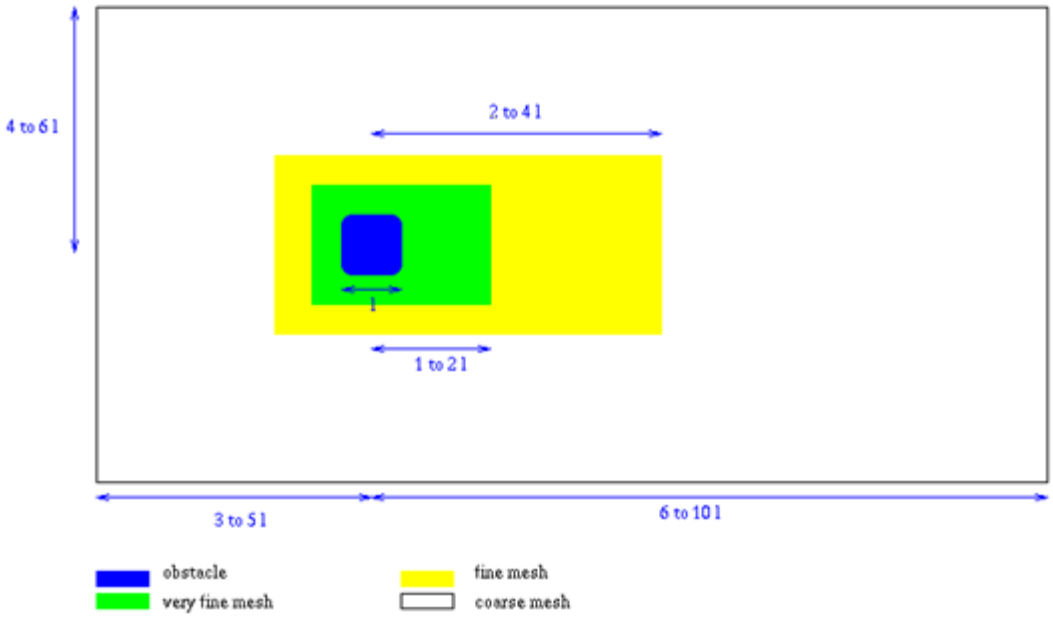


Figure 403: Number of Elements for Typical Mesh for CAA Problems (may vary for specific applications)

For problems with low Mach numbers (lower than 0.2), satisfactory results can be obtained under the quasi-compressibility assumption. This will save time on the computation. The compressibility implies that Navier-Stokes equations include wave equation. Then, acoustics and fluid flow can be treated at the same time, providing high numerical accuracy. Quasi-compressibility means, transport terms except in the momentum equation are neglected. Therefore, by reducing C_1 in hydrodynamic material laws, the sound speed decreases and the time step increases (for example by dividing the C_1 value by 10, the time step can be multiplied by 3).

Time Integration

Explicit time integration based on Newmark method is conditionally stable. The Courant condition also takes into account the flow of material through the elements gives the stability condition as:

$$\Delta t \leq S_f * \frac{\Delta l}{c + v - w} \tag{290}$$

$$v - w = \sqrt{\frac{1}{N}\sum_{i=1}^{3}\sum_{l=1}^{N}\left(v_i^l - w_i^l\right)^2} \tag{291}$$

Where,

N	Number of nodes for considered elements

The scale factor should be reduced to 0.7 and 0.5 for multi-physics laws.

Space Integration

Overall, the equation of motion for translational velocities with anti-hourglass and contact forces is written as:

$$\mathbf{M}\frac{\partial v}{\partial t} = \mathbf{F}_{ext} - \mathbf{F}_{\text{int}} + \mathbf{F}_{bod} + \mathbf{F}_{hgr} + \mathbf{F}_{cont} + \mathbf{F}_{trm} \quad (292)$$

Where,

$\mathbf{F}_{trm}$ Transport momentum vector given as

$$\mathbf{F}_{trm} = \sum_{elements} \mathbf{f}^{trm} \quad (293)$$

$$\mathbf{f}_{iI}^{trm} = \left(1+\eta_I\right)\cdot\int_v \rho\cdot\Phi_I\left(w_j - v_j\right)\cdot\frac{\partial v_i}{\partial x_j}dV \quad (294)$$

All matrices and vectors defined in the above equation are integrated over the spatial domain. The mass matrix is not constant in time, since the density and domain vary with time. It is shown that the solution of the equation is oscillatory in space when a mesh parameter, known as the Peclet number exceeds a critical value. This spatial stability can be avoided by adding numerical diffusion to the scheme which otherwise is generally under-diffusive and thus unstable. The momentum upwind coefficient is defined in material input. The full upwind case (*coef*=1) is the default value in Radioss and is generally used.

One upwinding technique is available in Radioss:

- Streamline Upwinding by Petrov-Galerkin method (SUPG):

  SUPG consists in modifying the shape functions to take into account the momentum convection terms. Since version 2018, SUPG is activated by default. SUPG can be turned off by using the Engine file option:

  ```
  /UPWM/SUPG
  ```

ALE Grid Calculation

In the ALE formulation, the freedom of moving the mesh is very appealing as it helps to combine the respective advantages of Lagrangian and Eulerian formulations. However, it is not easy to specify a grid velocity well-suited to the particular problem under consideration. As a consequence, the practical implementation of the ALE description requires that an automatic mesh-displacement prescription algorithm be supplied.

In Radioss, the following automatic grid computations exist.

/ALE/GRID/DONEA

This is the standard method applicable to the most of problems. It is based upon a combination of the material and grid velocities of the neighboring nodes:

$$W_I(t+\Delta t/2)=\frac{1}{N}\sum_J W_J(t-\Delta t/2)+\frac{1}{N^2}\frac{\alpha}{\Delta t}\sum_J L_{IJ}(t)\sum_J\frac{u_J(t)-u_I(t)}{L_{IJ}(t)} \quad (295)$$

Where,

$$1-\gamma\le\frac{W}{v}\le 1+\gamma \quad (296)$$

N	Number of nodes connected to node
L_{IJ}	Distance between node I and node J
α and γ	The adimensional factors given in input

Therefore, the grid displacement is given as:

$$u(t+\Delta t)=u(t)+w\left(\frac{t+\Delta t}{2}\right)\Delta t \quad (297)$$

/ALE/GRID/DISP

The average displacement formulation calculates the average velocity to determine the average displacement.

$$w(t+\Delta t)=\frac{1}{N}\sum_J W_J(t) \quad (298)$$

/ALE/GRID/SPRING

Each grid node is connected to neighboring grid nodes through a nonlinear viscous spring, similar to that shown in Figure 404.

The stiffness of each spring is given by $M\Delta t_{02}$ (where, M is the mass of the node and Δt_{02} is a user input typical time step), a viscosity and a ratio between shear spring stiffness and traction-compression stiffness of the springs can be defined.

It must be noted that those springs only affect the grid node velocity; they have no influence on the material velocity.

This method is very accurate and robust, but highly expensive in terms of CPU time.

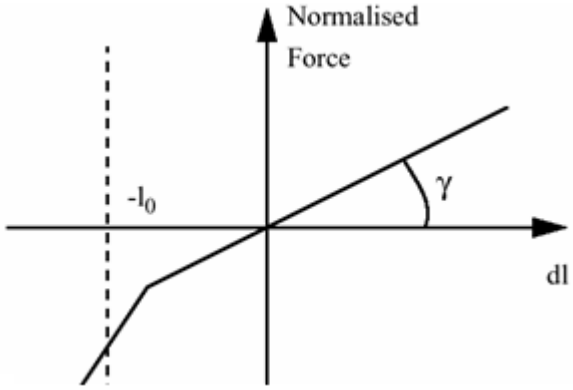


Figure 404: Spring Force Graph

/ALE/GRID/ZERO

No automatic grid calculation is performed for the grid. The grid velocity is either constant (0 if no initial grid velocity is specified, the formulation is therefore Eulerian) or imposed by Property TYPE15 for parts with a rigid body movement.

ALE/CFD Materials

The following material laws are commonly used for ALE/CFD simulations with Radioss:

- Newtonian or turbulent viscous fluid with a $k-\varepsilon$ model (activate LAW6 with /MAT/K-EPS)
- Viscous fluid for computational domain of Large Eddy Simulations (LES) (activate LAW46 with /MAT/LAW46 (LES_FLUID))
- Elementary boundary conditions (activate LAW11 with /MAT/LAW11 (BOUND))
- Elementary boundary conditions for turbulent flow (activate LAW11 with /MAT/B-K-EPS)
- Purely thermal materials (activate LAW18 with /MAT/LAW18 (THERM))
- Bimaterial (activate LAW20 with /MAT/LAW20 (BIMAT))
- Hydrodynamic bi-material liquid-gas (activate LAW37 with /MAT/LAW37 (BIPHAS))
- Multi-material solid, liquid, and gas (activate LAW51 with /MAT/LAW51 (MULTIMAT))
- Multi-fluid solid, liquid, and gas (LAW151 activated with /MAT/LAW151 (MULTIFLUID))

Many parameters are already defined by default and provide accurate results. You do not need to redefine these parameters, but you can adjust physical properties relevant to the material and units you are using.

Turbulence with $k-\varepsilon$ Model

The basic idea of turbulence (Reynolds equations) is to split the actual fluid velocity into its average component and its fluctuations; velocity fluctuations are assumed small with respect to average velocity. The isotropic turbulence theory then demonstrates the equivalence of these assumptions and the introduction of an additional viscous term to turbulent viscosity (Boussinesq approximation).

L.E.S. Material

Turbulent models based on the Reynolds equations separate the average quantities from the fluctuations and by design, force a steady state solution.

Whenever a flow contains large structures, fluctuations cannot be neglected when compared to average quantities. Then the Reynolds equations cannot handle accurately such situations.

The Large Eddy Simulation (LES) is different. No steady state or average flow is searched. The grid mesh is assumed to partly resolve the turbulence. At least the macro structures, whereas the smaller scales are replaced by a viscous term (sub-grid scale), supposed to model the mainly dissipative effects of the micro turbulence.

Material LAW46 is designed for Large Eddy Simulations. Radioss can be used with Smagorinsky's Sub-Grid-Scale model: (I_{sgs}=1 or 2) or without: (I_{sgs}=0) (MILES approach). The equation of state is linear compressible (constant compressibility), well suited for subsonic cases.

All elements connected to a node, either fixed or Lagrangian, will automatically be considered as wall elements and their viscosity will be set consistently with a logarithmic velocity profile.

Density, speed of sound, molecular kinematic viscosity, sub-grid scale model type (TYPE2 is recommended), and sub-grid scale constant (default is 0.1) must be provided.

The table below outlines different materials available for different usage.

		Multi-Material Laws		LAW20	LAW51	LAW151
Mach Number		Low (M<<1)		√	√	
		High (M>>1)		√	√	√
Solid Material Involved		Deviatoric stresses	(Sand, steel,…)	√	√	
Analysis	2D	Quad			√	√
		Tria	√		√	
	3D	Tetra				√
		Hexa			√	√
Numerical Scheme				Staggered	Staggered	Collocated
Resolution Method				Hybrid FEM/FVM	Hybrid FEM/FVM	Full FVM
Total Energy Conservation						√
Phases Discretization				Phases Interface Tracking	Second Order	Second Order

	Multi-Material Laws	LAW20	LAW51	LAW151
			(MUSCL algorithm)	(MUSCL algorithm)

√ : yes; blank: no

See Also
Materials/Failure (FAQ)

Quasi-Compressible Option

For a quasi-compressible option, density and energy are assumed to be constant.

No convection-diffusion is computed:

$$\frac{\partial \rho}{\partial t} = -\rho \nabla \mathbf{u} \qquad (299)$$

Therefore, the performance is improved and less computation time is needed.

The quasi-compressible option can be activated in the Radioss Engine file by `/INCMP`.

Using this option, the acoustic waves propagate as accurately as in the compressible case. However, it cannot be combined with the $k-\varepsilon$ turbulence model.

Turbulent Wall

Turbulent wall elements define the closure of the system with $k-\varepsilon$ model diffusive equations.

Turbulent wall conditions are set automatically in elements having at least one node with fixed boundary conditions (111 ... ...) or with Lagrangian conditions (... ... ... 111).

Lagrangian conditions must be explicitly declared in Radioss input for free nodes connected to shells; although these nodes are obviously Lagrangian.

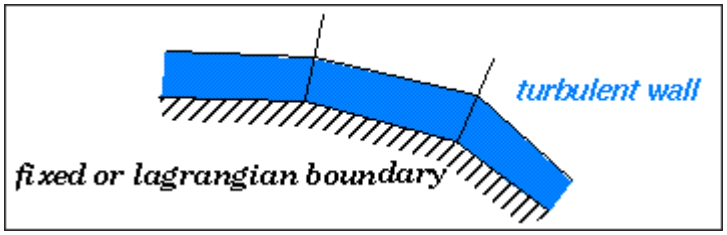


Figure 405:

Neglecting to declare nodal boundary conditions on the wall boundaries of the system can lead to indetermination and to divergence. Using the $k-\varepsilon$ model implies you must have at least one turbulent wall in your problem.

Wall Element Size

Usually, the value of y_2^+ is considered as critical and for tubes, you get good results for values of y_2^+ as high as 3000. Reasonable values are between 100 and 1000.

Where,

$$y_2^+ = \left(u^* y_2\right)/v \tag{300}$$

Where,

u^*	Friction velocity
y_2	Element size perpendicular to wall
v	Laminar kinematic velocity

In Radioss, y_2^+ is also equal to:

$$y_2^+ = 4.77\left(\frac{v_t}{v}\right) \tag{301}$$

Where, v_t is the turbulent kinematic viscosity.

Generally, meshes are built up with respect to affordability with some kind of trade-off on the value of y_2^+. If this value is too high, it means that the boundary layer will be poorly integrated. For open cases where the boundary layer often detaches, the mesh size in the wake of the body is mainly governed by the size of eddies. In this case, one can think accuracy is not governed by the wall element size, but by the quality of the algorithm.

Nodal Boundary Conditions

By default, kinematic constraints act on material velocities and accelerations. In Radioss, you can define a wide variety of such constraints. For multi-physic and fluid applications, options are:

- Fixed and full slip boundary conditions
- Imposed velocities (imposed flux at inlet)
- Rigid links (temporary adds during restarts)
- Rigid bodies to model rigid structures and connections and also to compute drag and lift forces (fluid impulse on rigid body is stored in time history database)

Grid constraints act only on grid velocities. Specify:

- Fixed and full slip grid conditions
- Lagrangian conditions (grid and material velocity are equally set)
- ALE links to maintain regular distribution of nodes
- Imposed grid velocities (moving inlet and outlet)

If fixed node with `/BCS`, then it means grid and material are fixed. If `/ALE/BCS` is used, then it is different. First 3 codes defined grid DOF, the second 3 codes defined if node act like Lagrangian in the corresponded direction.

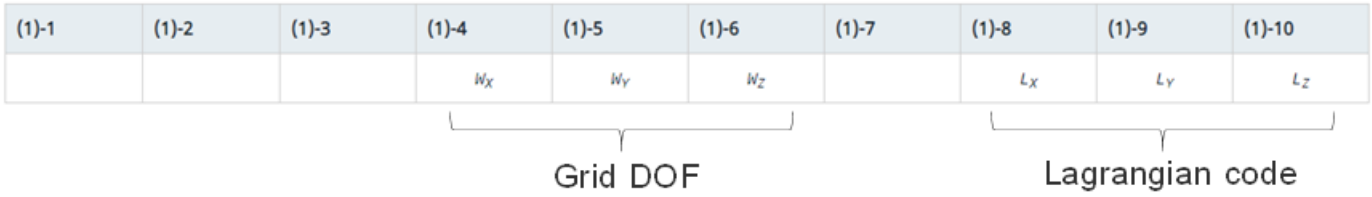


Figure 406:

Example:

If set *Grilag*=010 000, it means just fix grid in Y direction, material could move in Y direction.

If set *Grilag*=000 010, it means nodes are like Lagrangian in Y direction, gird and material are tied on these nodes in Y direction.

If set *Grilag*=010 010, it means gird and material are tied on these nodes in Y direction and fixed in Y direction. It is like trarot=010 000 in `/BCS`.

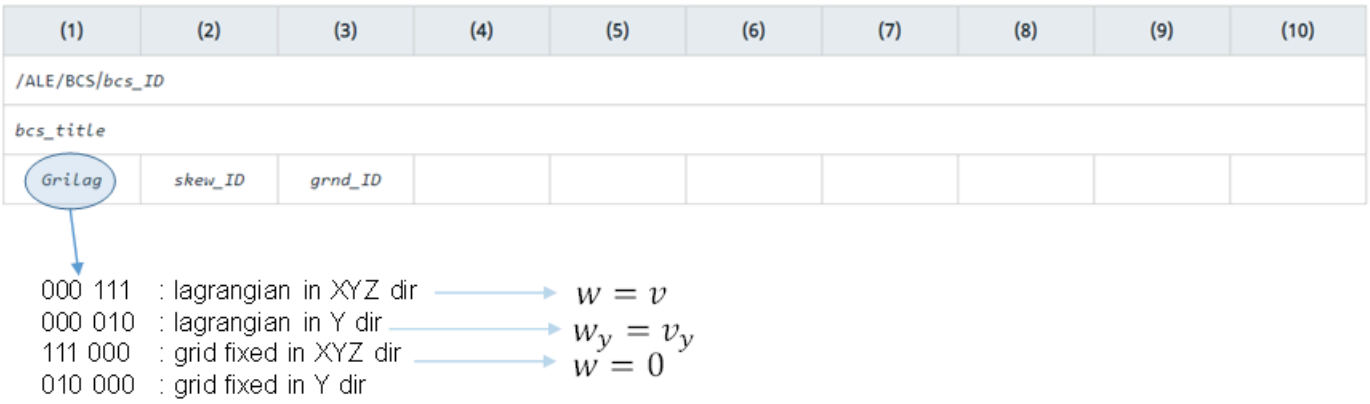


Figure 407:

Elementary Boundary Conditions

Boundary elements allow prescription of element values at domain boundaries. They can be specified with two ways:

One with boundary elements (quads in 2D analysis and solids in 3D analysis) with material:

- `/MAT/LAW11`
- `/MAT/LAW51`
- `/MAT/LAW18` (in Purely Thermal Material cases)

The other way is with `/EBCS` assign on surface of boundary elements.

For each variable P, rho, T, k, epsilon, internal energy, the following can be prescribed:

- imposed varying conditions according to user function
- continuity

- smoothly varying predefined function

With Radioss ALE/CFD any combination of the above options can be specified; on the counterpart, the closure of the various convection and diffusion equations has to be verified carefully by you.

Generally, the following types of elementary boundary condition are used:

- Inlet:

  Flux is imposed using imposed velocities `/IMPVEL`; density, energy, turbulent energy (that is: k) are imposed as constants. Continuity is imposed for pressure (display purposes only) and for epsilon. Turbulent energy, rho k is set to zero for external flows and to 1.5*rho*(0.06 V_{in})^2 for internal flows.
- Outlet:

  Continuity for all variables except pressure, which is imposed. When using the Non-reflective frontiers (NRF) option, you provide a value for sound speed and a typical relaxation length, which must be greater than the largest wave length of interest.
- Sides:

  Continuity for all variables with the Non-reflective frontiers (NRF) option or slip conditions without boundary elements. It ensures free field impedance to pressure and velocity fields.

If an element does not exist at boundary, continuity is assumed; but kinematic conditions are necessary to disallow fluxes; otherwise the convection equation is not closed and the program can diverge.

To specify the elementary variables at the boundary of the computational domain.

- In Material LAW11 below options used:
  - *Ityp* = 0 specifies stagnation conditions for perfect gas (Bernoulli inlet).
  - *Ityp* = 1 specifies stagnation conditions for a linear compressible material (Bernoulli inlet).
  - *Ityp* = 2 imposes values (inlet/outlet).
  - *Ityp* = 3 is for non-reflective frontiers (outlet).
- In Material LAW51 below options used:
  - *Iform*= 2 enables to impose sub-material states (density, energy, and volumetric fraction) which are also used to compute global material state. (inlet).
  - *Iform*= 4 gas inlet conditions for multi-material ALE laws (Gas inlet stagnation).
  - *Iform*= 5 liquid inlet conditions for multi-material ALE laws (Liquid inlet stagnation).
  - *Iform*= 6 This boundary material enables you to simulate a non-reflecting outlet boundary for multi-material law `/MAT/LAW51` (NRF outlet).

For example, in the input deck, density and energy are imposed constant at the inlet. Non-reflective frontiers are imposed at the outlet. The flux is then injected at inlet through imposed velocities at nodal points.

Non-Reflective Frontiers (NRF)

Ityp=3 in LAW11 and *Iform*=6 in LAW51 are used to prevent outgoing wave reflections on the boundaries of the domain.

Two possibilities are:

- An average pressure is imposed via a function. A relaxation term is added to let the average pressure converge toward the imposed value. This is well suited for outlets.
- An average pressure is calculated from the neighboring element pressure and the pressure converges toward this always changing value.

The impedance of the boundary is exactly the wave impedance of a monopole radiating at distance $2l_c$ from the boundary, where l_c is specified in the input data for this law.

This non-reflective frontiers (NRF) is not effective when velocities are imposed or when nodes are fixed.

Artificial Damping for Shock Waves

Quadratic bulk viscosity q_a and linear bulk viscosity q_b are, by default, set to values relevant for hydrodynamics and high speed impact problems involving shock waves.

For fluid dynamics and acoustics, these coefficients are not required, and most generally set to very small values to override the default.

Kinematic Constraints

Two kinds of kinematic conditions are available:

- Conditions applied on the material velocities
- Conditions on grid velocity (ALE)

By default, all kinematic conditions are applied to the material, not to the grid. For ALE applications, options of interest are:

- Fixed and full slip boundary conditions
- Imposed velocities (for example: imposed flux at inlet)
- Rigid links (temporary adds during restarts)
- Rigid bodies to model rigid structures and connections and also to compute drag and lift forces (that is, fluid impulse on rigid body is stored in time history database).

Boundary Conditions

ALE boundary conditions are activated by using `/ALE/BCS` keyword in the input deck. Grid constraints act only on grid velocities. If a boundary condition is not specified for grid velocity, nodes may move in any arbitrary manner (ALE).

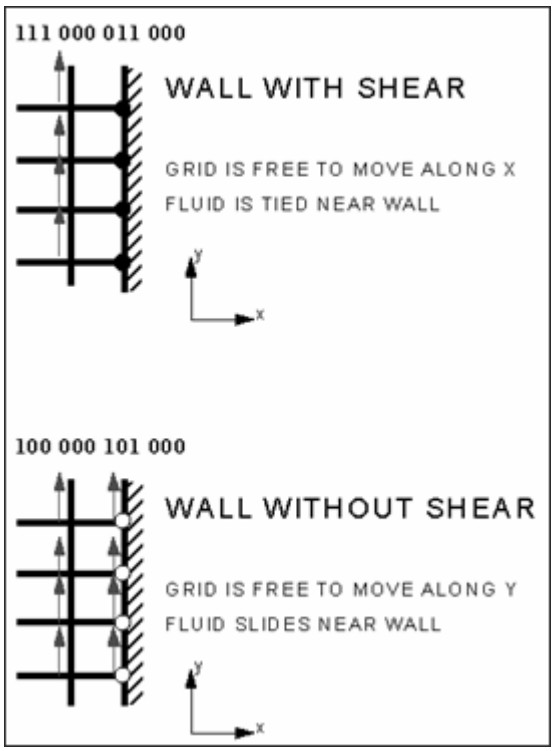


Figure 408: ALE Boundary Condition Applications

Rigid Body (/RBODY)

A rigid body is defined by a main node M and a set of secondary nodes. Rigid bodies are used in CFD application to compute reactive forces on structures.

ALE Links (VEL/ALE)

ALE link is a set of n nodes with grid velocity controlled by two main nodes M1 and M2. Input data is done at each restart (temporary add). Three options are available:

- Velocity is linearly interpolated with respect to input order
- Velocity is set to maximum absolute velocity of main nodes
- Velocity is set to minimum absolute velocity of main nodes

Interfaces

Available contact algorithms in Radioss are presented in Table 24.

Table 24: Radioss Interfaces Face-to-Face

Type	Description	Application	Contact Treatment
1 & 9	ALE / LAG with sliding	Fluid-structure interaction	Main-Secondary

Type	Description	Application	Contact Treatment
2	Tied interface	Change of mesh density (solid)	Main-Secondary or LM
3 & 5	Contact impact between two parts	Use of type 7 is recommended	Penalty
6	Contact impact between two rigid bodies	User-defined contacts	Penalty
7	General purpose contact impact between two parts	Solid contact impact at all speeds	Penalty or LM
8	Drawbead contact	Stamping applications	Penalty
10	Similar to type 7, but tied contact	Special purpose interface	Penalty
11	Impact between two lines	For beams, bars or springs	Penalty
12	Fluid / fluid contact	Fluid to fluid contact	Penalty
16 & 17	Contact between nodes to quadratic shape solids and solid-shells or between quadratic shapes	Meshes with 8-node or 16-node thick-shell or 20 bricks	LM
18	CEL Lagrange / Euler interface	Fluid-structure interactions	Penalty
22	Lagrange surface and fluid domain	Fluid-structure interactions	Penalty

The specific interfaces for Radioss ALE/CFD applications are:

- TYPE1: to contact an ALE part to a Lagrangian part
- TYPE12: for ALE/Euler bricks to contact moving or fix meshes
- TYPE18 and TYPE22: to treat fluid-structure interactions

Fluid/Fluid Interface

Interface TYPE12 is designed to connect two non-conforming meshes. Its use is restricted to a main/secondary approach, that is, no more than one main node is connected to a secondary segment.

> **Note:** The main/secondary search algorithm may fail where there are sharp angles. In this case, you must split the interface into several interfaces.

You also might detect an error when the closest main node of a given secondary node does not belong to the segment in front a secondary node. In this case, adjust the mesh locally to avoid ambiguous situations.

The interface is available for quasi-compressibility and compressibility options.

Select Main and Secondary Sides in Fluid/Fluid Interface (Type 12)

Rule 1:

Every secondary node needs a main segment.

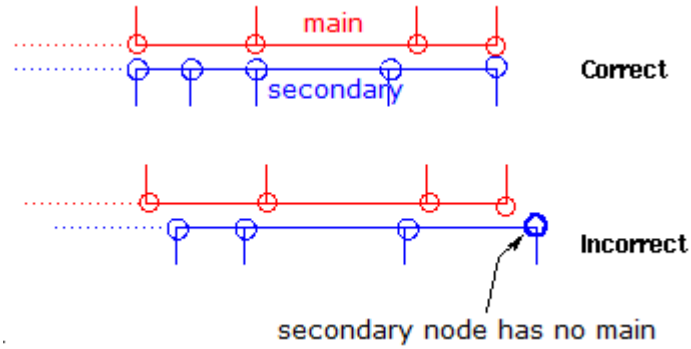


Figure 409:

This must remain true when interface is moving.

If a node has no projection on the main surface, the algorithm issues a warning but still works, because the secondary node remains associated to the nearest main segment.

Rule 2:

Every main node must receive contribution from at least a secondary node.

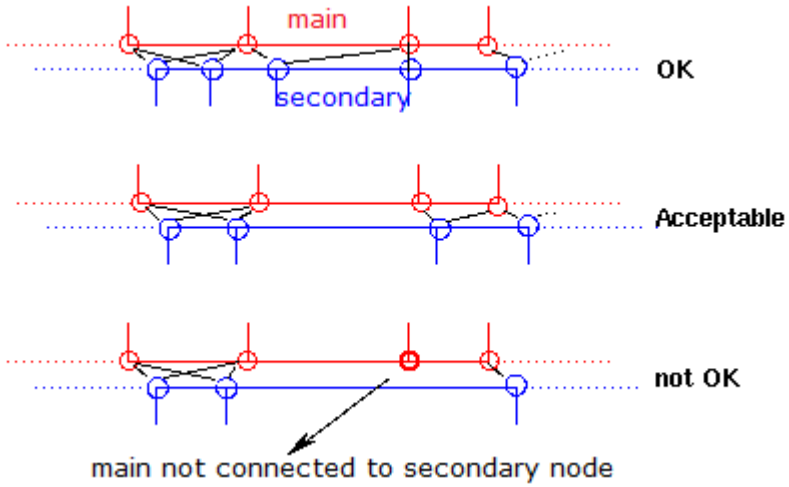


Figure 410:

If a main segment does not have a secondary but its neighbors do, the algorithm still works, because the above rule is satisfied. In principle, it is nevertheless recommended that main segments have at least one secondary node.

You can say that the secondary grid must be finer or equal to the main.

Rule 3:

When compressible, any segment (main or secondary) must be a surface of one and only one ALE brick element.

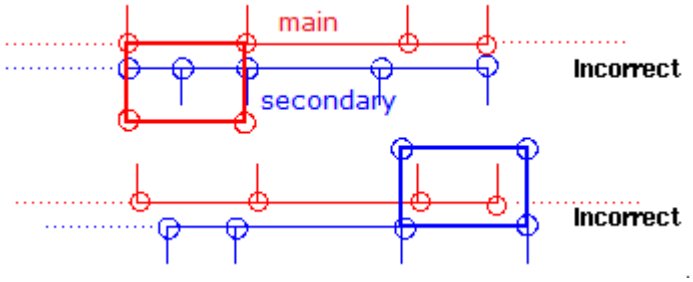


Figure 411:

Rule 4:

Kinematic conditions on the secondary side are allowed only if consistent with an identical condition on the main side.

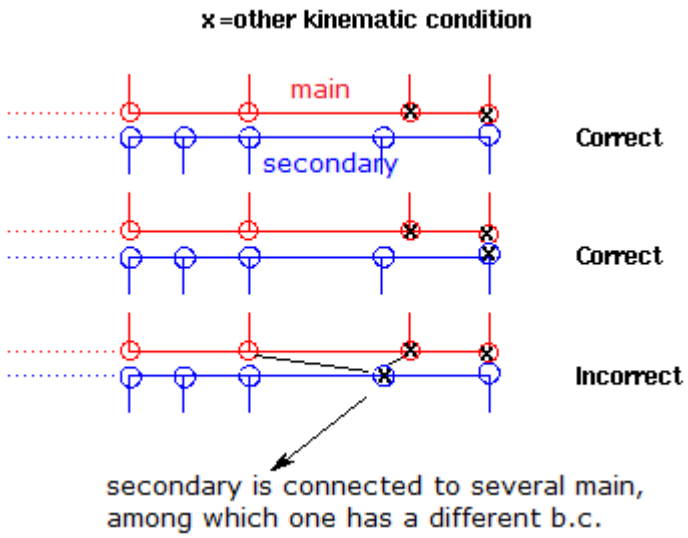


Figure 412:

This has to remain true when interface slides.

Fluid/Structure Interaction (ALE/Lagrange Interface)

Three kinds of interfaces are available.

Interface TYPE1

Interface TYPE1 is used to define boundaries between ALE and Lagrange domains. Full slip conditions are applied at the boundary between the two domains, like full sliding wall. Normal material velocities of Lagrange and ALE nodes are equally set.

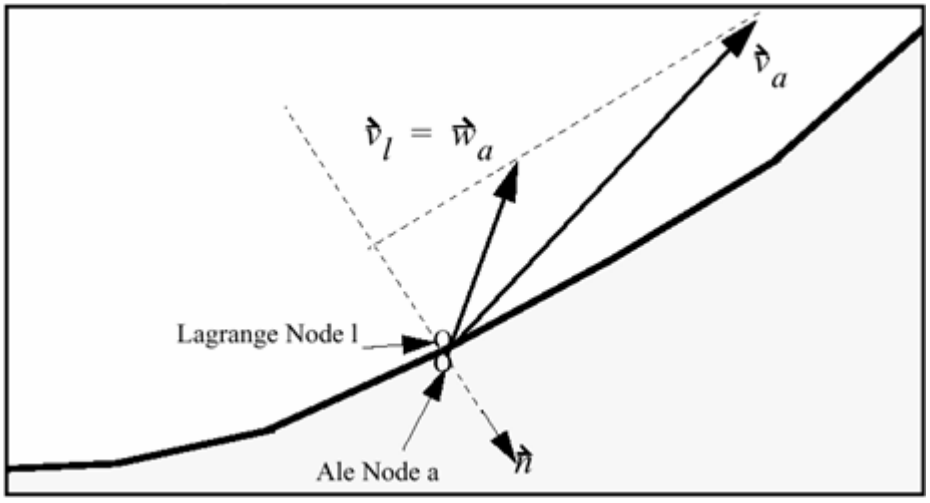


Figure 413: Fluid-Structure Interaction with Interface TYPE 1

Interface TYPE9

This is an ALE/Lagrange interface with void opening and free surface (that is, if the main and secondary sides are not in contact, there is no interaction between them). An ALE part is defined as secondary and the Lagrangian part is defined as main. The grid velocity is equal to the material velocity in normal direction.

Interface TYPE18 (CEL)

Interface TYPE18 creates contact between Euler/ALE nodes and a Lagrangian main surface. Visco-elastic penalty method is used to impose a relative material (fluid) velocity tangent to the main surface. The grid velocity is not modified by this interface. The interface allows the simulation of the penetration of a Lagrangian part (structure) into an Euler/ALE mesh, for example in the case of ship slamming.

The quality of simulation results using this interface depends strongly on the interface parameters and especially on the stiffness of the interface.

A recommended stiffness factor of the interface for fluid structure interaction problems can be obtained by:

$$Stfac = \frac{\rho \cdot v^2 \cdot S_{el}}{Gap} \tag{302}$$

Where,

ρ	The (highest) fluid density
v	Velocity of the phenomenon (speed of the sound or higher for supersonic phenomenon)
S_{el}	Surface of the Lagrangian elements

For aerodynamic problems, the recommendations are:

- $Gap = 1.5L_c$ with L_c being the length of fluid element

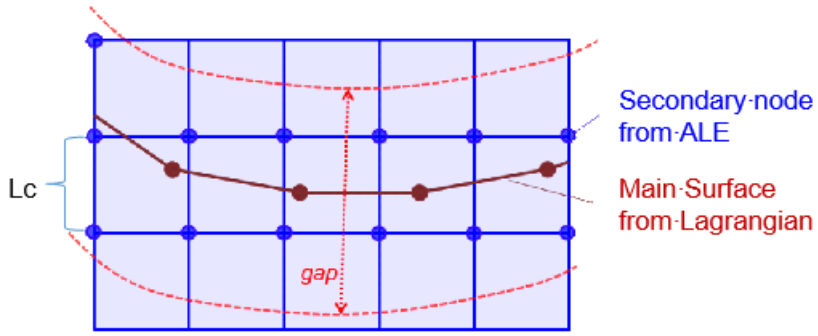


Figure 414:

- Constant stiffness with $K = g \cdot P \cdot L_c$
- Viscosity related to the interface stiffness:

$$v_s = 2\alpha \cdot Gap^2 \cdot \sqrt{\gamma P \rho} \quad (303)$$

With $\alpha = 0.5$ or $\alpha = \gamma$

P and ρ are the air characteristics

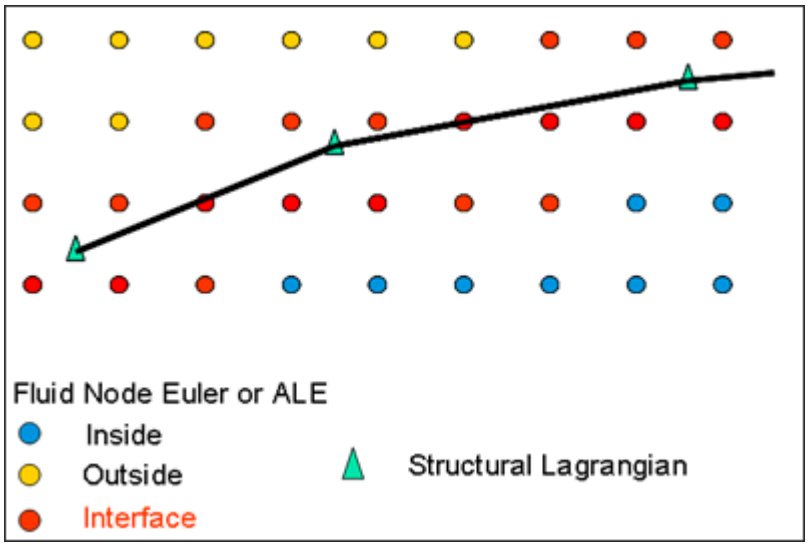


Figure 415: Fluid-Structure Interaction with Interface TYPE 18

The recommended gap value is equal to 1.5 by the element size.

Note: The interface needs a mesh fine enough to provide appropriate results.

Units

Radioss has built-in unit system for you to provide data in any consistent system.

To verify consistency, check that pressure = density x length^2 / time^2

Most popular units are (with steel examples):

Length	Time	Stress	Mass	Force	Energy	Density	Young's Module	Gravity	Yield Stress
m	s	Pa	kg	N	J	7.8e+03	2.1e+11	9.81e +00	2.06e+05
mm	ms	MPa	g	N	mJ	7.8e-03	2.1e+05	9.81e-03	2.06e+02
mm	ms	GPa	kg	KN	J	7.8e-06	2.1e+02	9.81e-03	2.06e-01
mm	s	MPa	Mg (ton)	N	mJ	7.8e-09	2.1e+05	9.81e +03	2.06e+02

Length	Time	Stress	Mass	Force	Energy	Density	Young's Module	Gravity	Yield Stress
cm	micros	Mbar	g	10^7N	10^5J	7.8e+00	2.1e+00	9.81e-10	2.06e-06

See Also
/UNIT (Starter)

Engine Input

As an example, the following Engine deck options can be used in Radioss CFD.

Define Optional Commands to Control Time Step

```
/DT/BRICK/STOP
1. 1.E-6
/DT/SHELL/CST
0.9 1.E-5
```

Define Commands to Determine Your Output Choices Time History File

```
/TFILE                  Output interval
1.e-4
```

Engine Output File

```
/PRINT/100              Cycle interval for output
```

Animation

```
/ANIM/DT                time settings
0.09 0.000833           time for first animation file, interval
```

Elementary Variables Choice

```
/ANIM/ELEM/K             Turbulent energy
/ANIM/ELEM/TVIS          Turbulent viscosity
/ANIM/ELEM/P             Pressure
```

Nodal Vector Choice

```
/ANIM/VECT/VEL  velocity
/ANIM/VECT/DISP displacement
```

Filtered Sampled Output

```
/@TFILE
0. 1.E-4 Start time, interval nodes
2009 2010 2011 2012 2017 2018 2023 2024 2029 2030
2035 2036 2041 2042 2047 2048 2053 2054 2059 2060
2065 2066 2071 2072 2077 2078 2153 2154 2157 2158
2161 2162 2165 2166 2169 2170 2173 2174 2177 2178
```

```
.....#
```

Troubleshooting

The program fails typically whenever:

- It stops with a floating exception error
- Time step is dropping rapidly

In most cases program failure is caused by:

- Divergence
- Hourglass excitation
- Incompatible boundary condition

Looking at time history curves like kinetic energy, turbulent energy, rotational energy, hourglass energy, material variables, etc., try to identify the time when an unexpected behavior can be observed.

If Starter does not go through with a clean message, open the output file and look for the word "`ERROR`", which will define a more detailed explanation of the problem.

If Starter crashes, open the Starter output (`Rootname_0000.out`) file and go to the end. Review the data to see if it makes sense or not - most likely 'not'. Then go backwards until you find data that does makes sense. Go to your input deck and inspect the area close to the switch between what makes sense and what does not make sense. There might be too many or not enough cards there.

> **Note:** Any modification in the input deck must be reflected in the control cards, if necessary.

If Engine stops with a negative density or time step problem or crashes:

- Rerun a restart between your last restart file and "just before the divergence"
- Run a new restart until the divergence with the following options in the Engine file:

  ```
  /PRINT/-1
  /ANIM/DT
  Ti 1.E-30 (where, Ti is the initial time of this run + epsilon)
  /TFILE/
  1.E-30
  ```

  You will then have a printout in the listing, an animation file and a time history sample for every cycle. Use them to investigate your problem. Very often looking at the velocity field in the animation shows some irregularity; which can easily be connected with a hole in the boundary conditions.

Divergence

Divergence means some variable is getting too large. This might occur if your problem is poorly defined (mathematically), or if the time step is bigger than the theoretical critical time step (numerical).

Try to locate the problem in time and space by running the time history and animation programs. Look for anomalous velocities, turbulent energy and viscosity.

Check your boundaries:

- Kinematic constraints
- Elementary boundaries (material TYPE11)

Verify classical assumptions of fluid mechanics.

Look for leaks in your mesh (non-coincident nodes, non-plane symmetries, omitted boundary conditions). If the turbulence model is being used, verify you have at least one turbulent wall.

Look for hourglass velocity patterns. This might occur if you have concentrated fluxes, elements with bad aspect ratios, or if the hourglass coefficient is set to a very small value (see material parameters).

Check to make sure you do not have a shell thicker than the element size: flexural stability is a priority assumed in the code (not calculated) and this assumption is conservative only as long as the thickness does not exceed the length of the shell.

Smooth Particle Hydrodynamics (SPH)

The Smooth Particle Hydrodynamics method formulation is used to solve the equations of mechanics, when particles are free from a meshing grid.

It is specially adapted to simulate phenomena with a very substantial deformation, that is a range of applications where the Finite Element method, with ALE and Lagrangian formulation, loses' its efficiency and accuracy.

The SPH method built in the Radioss code is compatible with most functions. For instance, it is possible to cause two objects to interact, one discretized by finite elements and the other by particles.

You can put the SPH formulation in an ALE model, only if the boundary between SPH and ALE is Lagrangian. The SPH formulation is only available in 3D analysis.

See Also

RD-E: 1300 Shock Tube Experiment

RD-E: 2200 Ditching

RD-E: 4900 Bird Strike on Windshield

RD-E: 5500 Fan Blade Rotation Initialization and Impact

SPH Cell Distribution (Theory Manual)

SPH Cells Distribution

The particles should be created in a hexagonal compact, face centered cubic or a cubic net packing.

The hexagonal compact net and face centered cubic are recommended for use in Radioss and give similar results. A face center cubic particle distribution can be created using the HyperMesh SPH panel. A HyperMesh Tcl macro to generate hexagonal compact net is available from the Altair Connect website.

Hexagonal Compact Net

A cubic centered faces net realizes a hexagonal compact distribution, which can be useful to build the net.

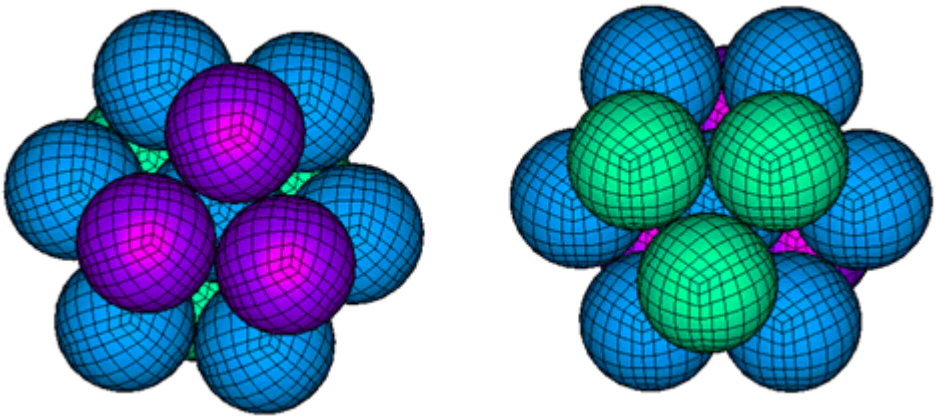


Figure 416: Local Views of the Hexagonal Compact Net

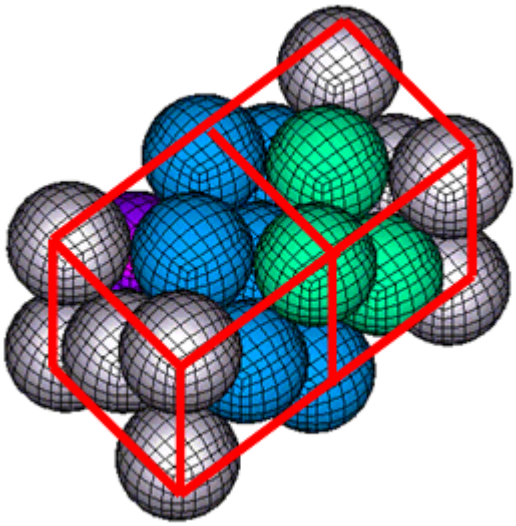


Figure 417: Perspective View of the Cubic Centered Faces Net

The hexagonal compact net distribution can be created in HyperMesh using a Tcl script available by searching in Altair Connect for the Hexagonal Compact Net Tcl script. When using this script, the pitch or distance between any particle and its closest neighbor is entered as h_0.

The mass of the particle m_p is defined in the property `/PROP/SPH`.

The SPH particle mass relates to the material density ρ and the pitch h_0 of the hexagonal compact net. This particle mass can be calculated as:

$$m_p \approx \frac{h_0^3}{\sqrt{2}}\rho \qquad (304)$$

Since the space can be partitioned into polyhedras surrounding each particle of the net, each one with a volume:

$$V_p = \frac{h_0^3}{\sqrt{2}} \quad (305)$$

Due to discretization differences in the volume, the mass can be more accurately represented by:

$$m_P = \frac{\rho V}{n} \quad (306)$$

Where,

V
: Total volume filled by the particles.

n
: Total number of particles distributed in the volume.

For hexagonal compact net, the recommended smoothing length h in `/PROP/SPH` is the pitch h_0 which is the smallest distance between the particles. A smoothing length smaller than is this can only be used when there is no tension physical problems material. If the material does include tensile behavior, then a smoothing length larger than h_0 can be used to increase stability but there will be an increase in the computational cost.

The following table shows the number of neighbors with when different smoothing lengths h in `/PROP/SPH` are used. The accuracy and computational cost of the simulation improves as the smooth length increases.

Table 25: Number of Neighbors in a Hexagonal Compact Net

Distance d	Number of particles at distance d	Number of particles within distance d
h_0	12	12
$\sqrt{2}h_0$	6	18
$\sqrt{3}h_0$	24	42
$2h_0$	12	54
$\sqrt{5}h_0$	24	78

Face Centered Cubic

The face centered cubic arranges the particles in groups of 14, forming the corners and the center of each face of a cube.

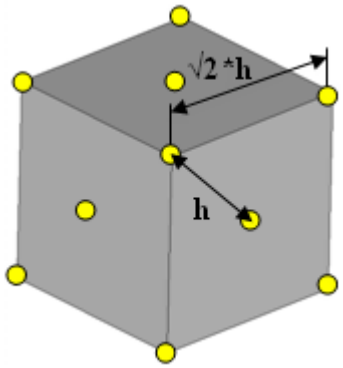


Figure 418:

Similar to hexagonal compact net, each particle has 12 neighbors and the mass of a particle is:

$$m_P = \frac{\rho V}{n} \quad (307)$$

For face centered cubic the recommended smoothing length *h* in */PROP/SPH* is the pitch entered when created the sph mesh in HyperMesh. The pitch h_0 is the smallest distance between the particles. A smoothing length smaller than h_0 can only be used when there is no tension physical problems material. If the material does include tensile behavior, then a smoothing length larger than h_0 can be used to increase stability but there will be an increase in the computational cost.

Simple Cubic Net

Let, *c* the side length of each elementary cube into the net.

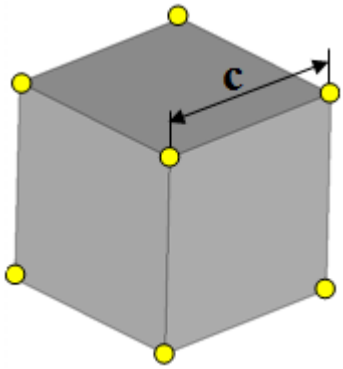


Figure 419:

The mass of the particles m_p should relate to the density of the material ρ and to the size *c* of the net, with respect to the following equation:

$$m_p = c^3\rho \quad (308)$$

Table 26: Number of Neighbors in a Cubic Net

Distance d	Number of particles at distance d	Number of particles within distance d
c	6	6
$\sqrt{2}c$	12	18
$\sqrt{3}c$	8	26
2c	6	32
$\sqrt{5}c$	24	56
$\sqrt{6}c$	24	80
$\sqrt{2}c$	12	92
3c	6	98

From experience using cubic net, a higher smoothing length compared to face centered cubic or hexagonal compact net is needed to solve the tension instability. This higher smoothing length increases the computational cost since more neighbor particles have to be included in the calculation for each particle.

For cubic net, a smoothing length *h* between 1.25c and 1.5c is recommended in `/PROP/SPH`.

Maximum Stored Number of Neighbors

When more than *Nneigh* neighbors are found within the security distance, the program retains the only *Nneigh* closest points and decreases the value of α_{sort}.

If all true neighbors lying inside the influence sphere of all particles are still retained, then the results do not change.

In the other case, a message such as "`Warning SPH Computation`" is sent to the Radioss output file. Note that in the case of such a message, the computation time increases since it becomes necessary to sort closest particles at each cycle. Moreover, this kind of situation has to be analyzed carefully since it is often put into an evidence (local) instability.

SPH Symmetry Conditions

Multiple SPH Symmetry Conditions

An axi-symmetry condition can be modelized through the use of two conditions with respect to two planes intersecting at the axis of symmetry. A spheric symmetry condition can be modelized through the use of three conditions with respect to three planes intersecting at the center of symmetry.

Nevertheless, these kinds of symmetries are not treated the same way.

For instance, in case of an axi-symmetry condition, not all ghost particles are built around the axis of symmetry. The only symmetric particles of real particles with respect to the two symmetry planes are built.

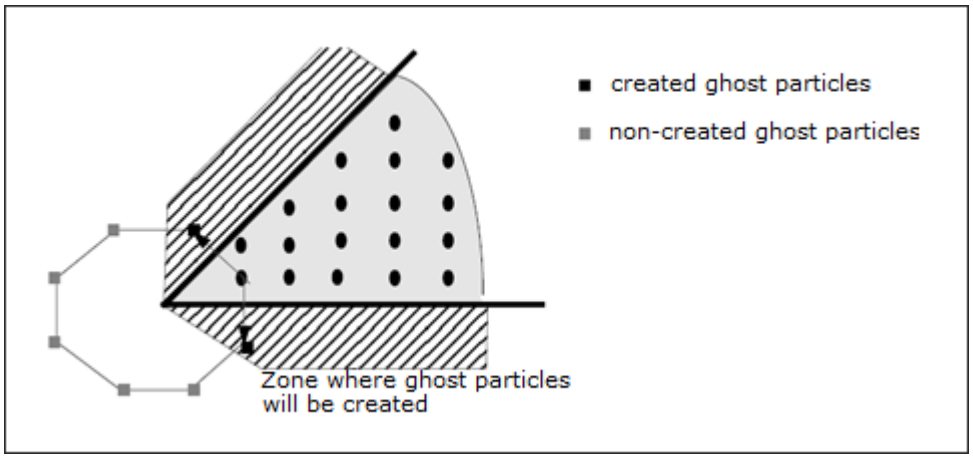


Figure 420: Incomplete Multiple Symmetries

Therefore, some characteristics of axi-symmetry (respectively spheric symmetry) conditions can be closed to the axis of symmetry (respectively the center of symmetry).

Nodes closed to the axis of symmetry (resp. the center of symmetry) and lying on a symmetry plane (P) can get a normal to (P) velocity which is non-zero, since their neighborhood is not symmetric with respect to plane (P).

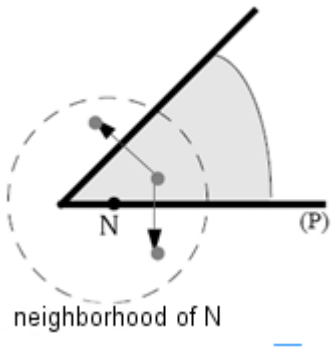


Figure 421: Neighborhood of Particle N is not Symmetric with Respect to Plane (P) and N's Normal Velocity to Plane (P) can be Non-zero

Kinematic Boundary Condition

With respect to the previous discussion: adding the kinematic boundary condition an explicit way allows to enforce it.

A kinematic boundary condition will be added to the nodes belonging to the nodes group specified into the `/SPHBCS` option, so that:

- If "Slide" type, the velocity of the node in direction "Dir" is set to zero
- If "Tied" type, the velocity of the node in all directions is set to zero

In case of several kinematic boundary conditions applied to the same node through different SPH symmetry conditions, the kinematic boundary conditions are composed automatically, even if the kinematic boundary conditions are applied into non-orthogonal directions.

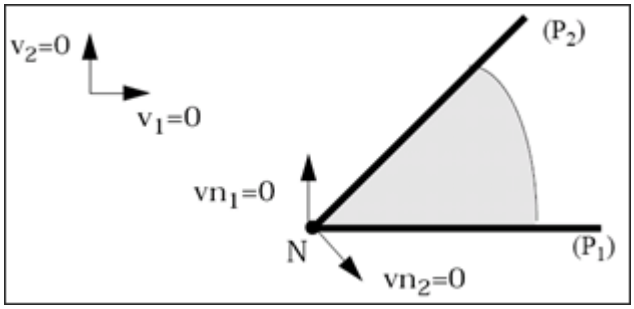


Figure 422: Combination of Kinematic Boundary Conditions from Different SPH Symmetry Conditions

Figure 422 indicates that if two kinematic boundary conditions are applied to N through two symmetry conditions with respect to planes (P_1) and (P_2), the two boundary conditions are modified so that the velocity in the plane normal to common axis of (P_1) and (P_2) will remain zero. Note that if one of the two symmetry conditions is a type "Tied" condition, the velocity of N in all directions is set to zero.

It also allows application to the same node, a kinematic boundary condition through a SPH symmetry condition (`/SPHBCS`) and a standard boundary condition (`/BCS`) at the same time, as long as the standard boundary condition is not given in a moving skew system, but a fix skew system or the global skew system. The two conditions are then composed the same way.

Part Mass

You must be advised that when a particle lies on a symmetry plane at time t=0, the mass and the initial volume considered for the particles are respectively:

$$m = \frac{m_p}{2}, \quad V = \frac{m_p}{2\rho} \tag{309}$$

Where, m_p is the mass specified into property set.

When a particle lies on n symmetry planes at time t=0,

$$m = \frac{m_p}{n}, \quad V = \frac{m_p}{n_\rho} \tag{310}$$

Ghost particles built from this particle will get the same initial volume and mass.

When $n > 2$, the previous equation may provide an error on mass and energies output for the part the particles belong to, with respect to the physical model.

Formulation Level

When a symmetry plane is defined, and even if a kinematic condition is set for all particles lying on the symmetry plane, particles lying at time zero inside the domain are theoretically able to cross the symmetry plane.

This is specific to SPH for which stiffness between particles does not increase to an infinite value when particles collapse. So it can occur when the particles, which lie on the symmetry plane let the particles which were inside the domain go through the symmetry plane.

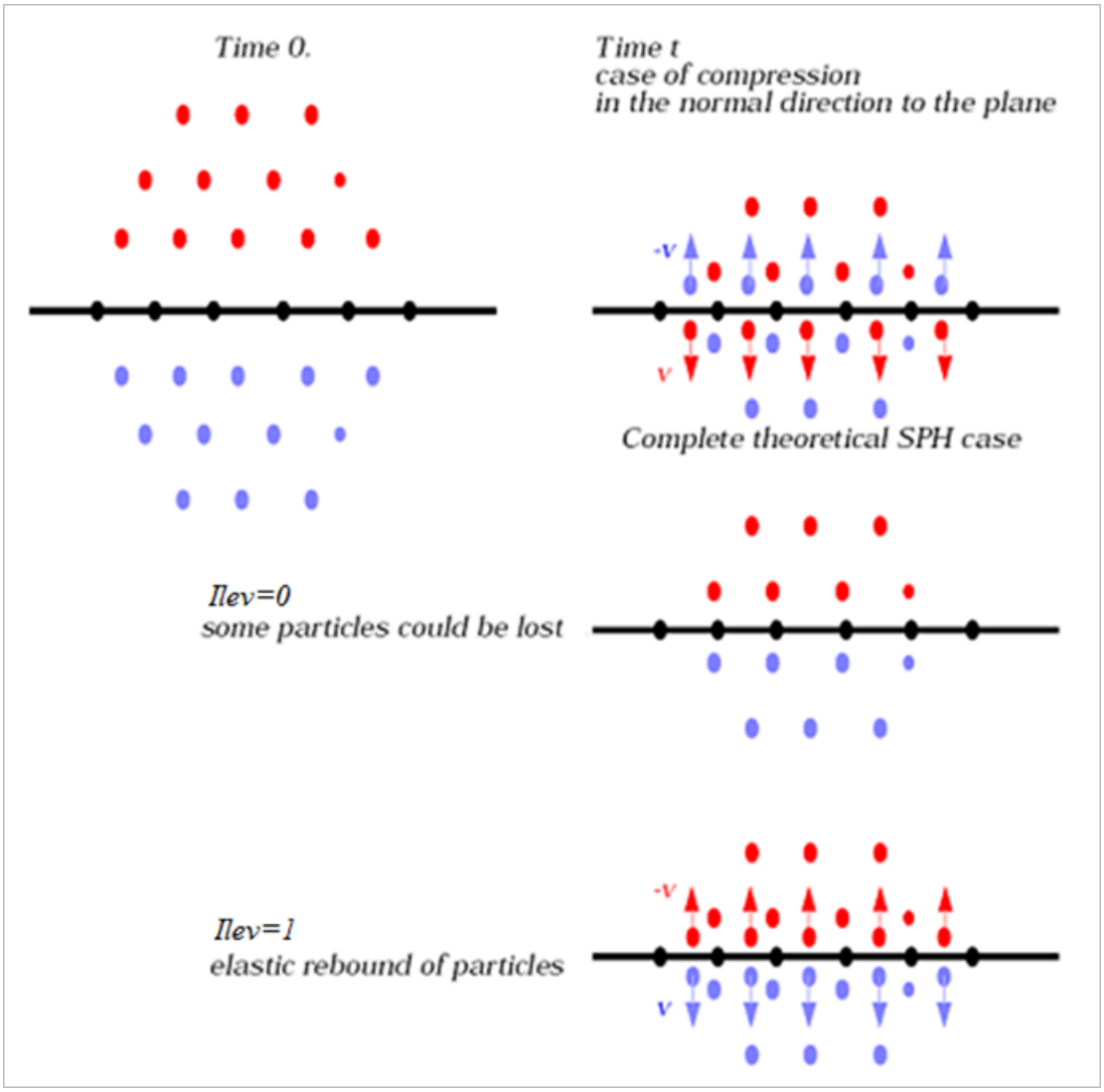


Figure 423: Formulation Level

If *Ilev*=0, particles crossing the symmetry plane will be (progressively) not taken into account anymore in the computation, neither than their symmetric particles which then lie inside the domain.

If *Ilev*=1, particles which have crossed the symmetry plane rebound an elastic way upon the symmetry plane: their velocity in the normal direction to the plane is set the opposite.

Note: When *Ilev*=1, it is strongly recommended to associate kinematic condition to all particles lying on symmetry plane at time zero, for computational time reasons.

Maximum Created Number of Ghost Particles

Ghost particles are created at each search for neighbors time within the security distance, and then destroyed when a new search occurs (a new set of ghost particles is then created).

At any search time, all ghost particles which are inside the security distance of any real particle are created.

In practice, some more particles, strictly necessary, are created: a symmetric particle G_i to particle N_i is created, with respect to symmetry plane P, if $\exists\, j$ neighbor of i:

$$d(N_i,(P))^2 \le (1+\alpha_{sort})\cdot \max(d_i+d_j)^2$$

With d_i and d_j are the smoothing lengths related to particle i and j.

As long as no real particles cross the symmetry plane (all real particles lie on the same side of the symmetry plane), this criteria is sufficient to get all ghost neighbors of all real particles inside the security distance, since:

$d(N_i,(P)) \le d(G_i,N_i)$ for $\forall\,(i,j)$

And, $d(G_i,N_i) \le \sqrt{1+\alpha_{sort}}\cdot \max(d_i+d_j)$

$\Rightarrow d(N_i,(P)) \le \sqrt{1+\alpha_{sort}}\cdot \max(d_i+d_j)$

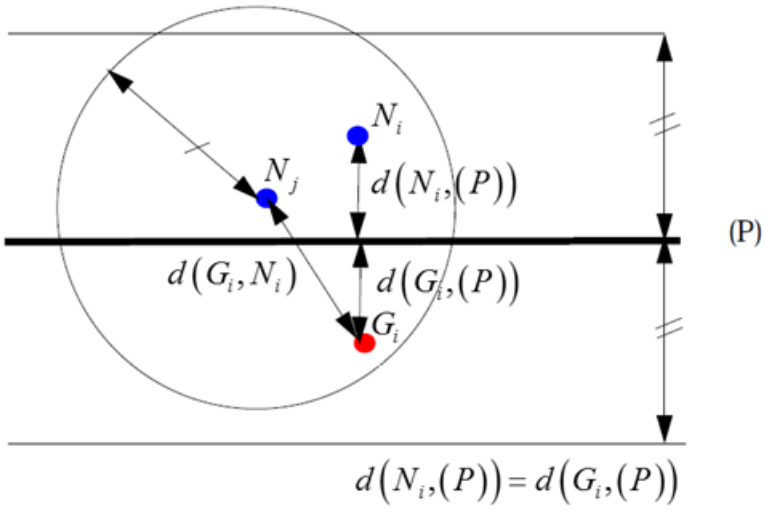


Figure 424: Ghost Particles to be Created

Particles, which one can expect to remain far from the symmetry plane all along the simulation, will never be symmetrized. This gives a way to over-estimate the number of particles which will be symmetrized at one time.

When a particle N_i has to be symmetrized with respect to n conditions, the particle N_i gives birth to n ghost particles. The following quantity must remain less than *Maxsph* (since v14.0.220, "*Maxsph*" is ignored and the memory is dynamically allocated).

$$\sum_{n=1}^{n} n_{particles} \tag{311}$$

Where,

n	Number of conditions
$n_{particles}$	Number of particles to be symmetrized with respect to condition n

Anyway, the default value which is the number of SPH symmetry conditions multiplied by the number of particles will be enough to treat any problem.

Solid to SPH Options (Sol2SPH)

The solid to SPH option (Sol2SPH) enables you to turn a solid element into particles either in order to increase the time step/robustness of a Lagrangian calculation, while not changing the physics.

Time Step

Two SPH time step methods are available in Radioss:

- Particle time step (`/DT/SPHCEL`)
- Nodal time step (`/DT/NODA`)

In particle time step, stable time step is computed as:

$$\Delta t = \Delta T_{sca} \cdot \min_i \left(\frac{d_i}{c_i \left(\alpha_i + \sqrt{\alpha_i 2 + 1} \right)} \right) \tag{312}$$

Where,

d_i	Smoothing length related to particle i
c_i	Sound speed at location i

$$\alpha_i = q_a + \frac{q_a \cdot \bar{\mu}_i \cdot d_i}{c_i} \tag{313}$$

For time step scale factor ΔT_{sca}, it is recommended to set it to 0.3.

In nodal time step, stable time step is computed as:

$$\Delta t = \Delta T_{sca} \cdot \sqrt{\frac{2m}{K^*}} \tag{314}$$

Where,

m	Mass for particles
K^*	Stiffness based on SPH interaction

For time step scale factor ΔT_{sca}, it is recommended to set it to 0.67.

Thermal Analysis

Heat transfer is now available between SPH particles and finite elements with I_{the}=1 in `/INTER/TYPE7` and `/INTER/TYPE21`; and with `/THERM_STRESS/MAT`, thermal expansion in SPH is also possible.

Solid to SPH Option (Sol2SPH)

The solid to SPH option (Sol2SPH) enables you to turn a solid element into particles in order to increase the time step/robustness of a Lagrangian calculation, while not significantly changing the physics.

It is compatible with the solid property options `/PROP/TYPE14 (SOLID)` and `/PROP/TYPE6 (SOL_ORTH)` and is limited to:

- I_{solid} = 1, 2 or 24
- I_{frame} = 1 or 2
- 8-node bricks (`/BRICK` including degenerated) and 4-node tetras (TETRA4), but not 10-node tetras.

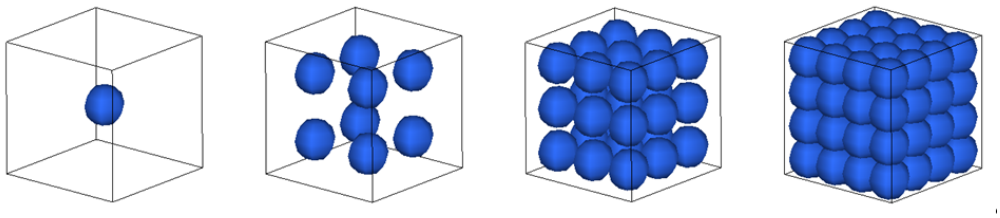
The input is:

```
#---1----|----2----|----3----|----4----|----5----|----6----|----7----|----8----|----9----|---10----|
/PROP/SOLID/3
Prop example
#    Isolid    Ismstr               Icpre  Itetra10     Inpts   Itetra4    Iframe                  dn
         1         2                   0         0         0         0         1                   0
#                q_a                 q_b                 h            LAMBDA_V                MU_V
                   0                   0                 0                   0                   0
#             dt_min   istrain      IHKT
                   0         0         0
#      Ndir  SPH_part
          1        10
#---1----|----2----|----3----|----4----|----5----|----6----|----7----|----8----|----9----|---10----|
```

SPH_part
: Part number defining the SPH properties

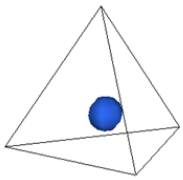
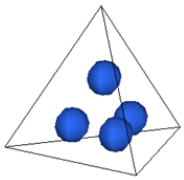
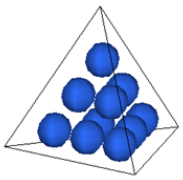
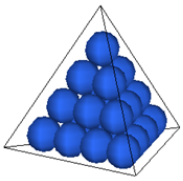
Ndir
: Number of particles created for each direction, the maximum value is 9

Table 27: 8 Nodes Brick

			
Ndir = 1: 1 particle is created	*Ndir* = 2: 8 particles are created	*Ndir* = 3: 27 particles are created	*Ndir* = 4: 64 particles are created

The particles are created according to a cubic net.

Table 28: 4 Nodes Tetra

			
Ndir = 1: 1 particle is created	*Ndir* = 2: 4 particles are created	*Ndir* = 3: 10 particles are created	*Ndir* = 4: 22 particles are created

The transition from solid elements to SPH particles can be triggered by contact or by brick deletion.

Sol2SPH Particle Behavior

All particles are created at the beginning of the calculation.

As long as their main element is not deleted, those particles are unreleased.

When the main solid element is deleted, the particles are released and behave like any other SPH genuine particle.

Unreleased particles kinematic is interpolated from their main element. In turn, the forces applied to those particles are transmitted to their corresponding solid nodes.

Those forces are:

- Contact forces if they are secondary of a `TYPE7` interface or rigid walls
- Interaction forces with released particles or genuine particles.

Particle Activation Process

Particles are fully activated based on the following criteria.

- Their main brick is deleted either because:
  - The minimum time step has been reached (`/DT/BRICK/DEL`)
  - The failure criteria associated to the brick material law has been met
- In case of contact:
  - If an unreleased particle is secondary of a `TYPE7` interface and enters the gap of the main surface, the brick is deleted, and the associated particles are released provided the I_{bag} flag of the interface is set to 1.
  - If an unreleased particle is within interaction distance of any particle belonging to a different part (including unreleased particles), the main brick is deleted, and the associated particles are released. Therefore, no contact interface is required between two Sol2SPH parts.

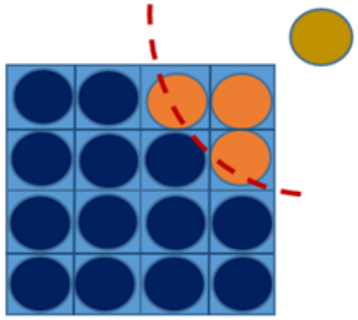


Figure 425: Orange Particles are Released as they are within Interaction Distance of the Yellow Particle (which belongs to another part)

- If *Isol2sph* is set to 2 in `/SPHGLO`, then particles are not released if the 2 parts belong to the same subset. This option can be useful to prevent unwanted activation of particles at the junction of parts belonging the same physical entities.

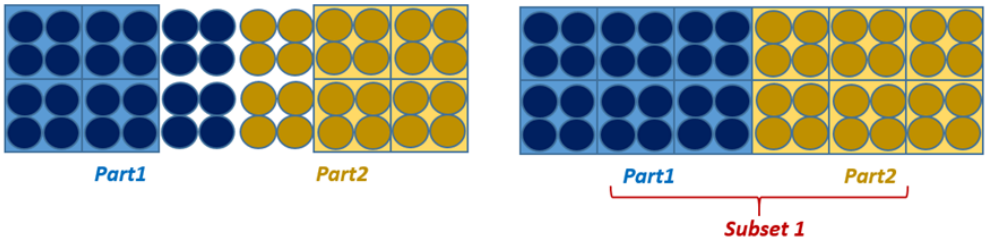


Figure 426: Isol2sph = 1: Unwanted particles activation / Isol2sph = 2 + subset definition

Particle properties are defined using `/PROP/SPH` option.

Typical Property for Sol2SPH Particles:

```
#---1----|----2----|----3----|----4----|----5----|----6----|----7----|----8----|----9----|---10----|
/PROP/TYPE34/10
SPH prop example
#                  mp                  qa                  qb            alpha_cs    skew_ID      h_ID
                    0                   0                   0                   0          0         0
#    order                   h             xi_stab
           0               125.0                   0
#---1----|----2----|----3----|----4----|----5----|----6----|----7----|----8----|----9----|---10----|
```

The mass of the particles is automatically recalculated from the main element to ensure mass conservation. For this type of SPH part, the input mass may be 0.0.

The smoothing length must be input. A good value for a regular mesh is:

$$h = \frac{1.5 \cdot l}{Ndir} \tag{315}$$

Where,

h	Smoothing length
l	Size of the brick elements

$Ndir$	Number of particles per direction for each solid elements

The material law can be `/MAT/LAW2`, `/MAT/LAW3`, `/MAT/LAW4`, `/MAT/LAW6`, `/MAT/LAW12`, `/MAT/LAW21`, `/MAT/LAW24`, `/MAT/LAW36`, `/MAT/LAW49`, or `/MAT/LAW81`.

Both material laws for solid and SPH must be of the same type.

The material parameters can be different, but make sure there is consistency.

It is advised to keep the same parameters and change only the failure values (failure criteria, max strain, Spalling pressure, ...).

Contact between Sol2SPH and Other Parts

Sol2SPH versus Finite Element Part

The contact must be defined through interface `TYPE7`.

The Sol2SPH part must be secondary and should be finer than the main part.

The contact must be defined between the main (FE) part and the SPH particles of the Sol2SPH part (the part number defined for instance in *SPH_PARTID*).

Self-impacting interfaces are not possible, but because particles are interacting with each other even when not activated, self-contact of a Sol2SPH part is always taken into account.

The stiffness factors and gap definition follow the same rule as normal SPH particles.

SOL2SPH particles may be automatically activated and their corresponding brick element deleted when entering gap distance, if the I_{bag} flag of the interface is set to 1.0.

SOL2SPH versus SPH or SOL2SPH

Contact is handled automatically, and no interface has to be created.

Other Modeling Guidelines

Limitations

Sol2SPH is not compatible with SPHBCS and inlet/outlet options.

Mesh Sizes and Other SPH Parts

All SPH and Sol2SPH parts should have the same pitch.

As Sol2SPH particle configuration is cubic, SPH parts should be also use a cubic net.

/BRICK/DT/DEL

Using `/DT/BRICK/DEL` is the most CPU efficient way of using Sol2SPH.

The minimum time step should be around 20-30% less than the natural time step.

It is recommended to use nodal time step for better SPH particles stability.

After the deletion of a brick, the time step may be as low as:

$$\Delta t = \frac{\Delta T_{\min}}{Ndir} \quad (316)$$

Where,

$\Delta T_{\min}$	Deletion brick time step as defined in `/DT/BRICK/DEL`
$Ndir$	Number of created particle per direction

Multi-Domain Technique

The objective of the Multi-Domain technique (also referred to as RAD2RAD) is to optimize the computing performances of large scale Radioss models.

The objective of the Multi-Domain technique (also referred to as RAD2RAD) is to optimize the computing performances of large scale Radioss models that meet certain criteria:

- Possibility of sub-dividing the whole model into a number of distinct subdomains with clearly defined interfaces/connections between them.
- Different subdomains should be characterized by different mesh sizes and consequently very different minimal time step.

The goal is to improve prediction accuracy at reasonable, possibly advantageous, computation time for models with domains of very different time step sizes.

For example, it is appealing to use Multi-Domain technique to compute large models that have one or more parts finely meshed to capture specific local phenomena such as cracks localization/propagation.

It is even more appealing to use Multi-Domain technique to compute large fluid-structure interaction models, as in aircraft ditching or landing simulations, where fluid elements with high time steps are numerous compared to Lagrange structure elements with very small time steps.

In the explicit integration scheme used by Radioss crash solver, the time step of the global model is penalized by the elements having the smallest time step. The concept is to replace this global model with physically equivalent subdomains, separating parts with different minimal time step. Each subdomain is resolved as a distinct Radioss model using its own time step, the force and momentum transfers between them being calculated by a separate main program (RAD2RAD), assuring stability constraints.

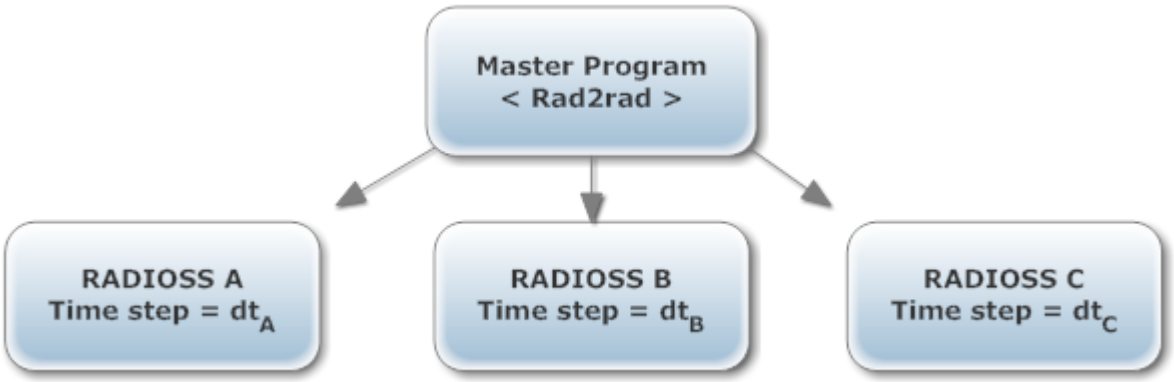


Figure 427:

Multi-Domain efficiency is based on two types of discrepancies:

- Time step sizes
- Domains relative sizes

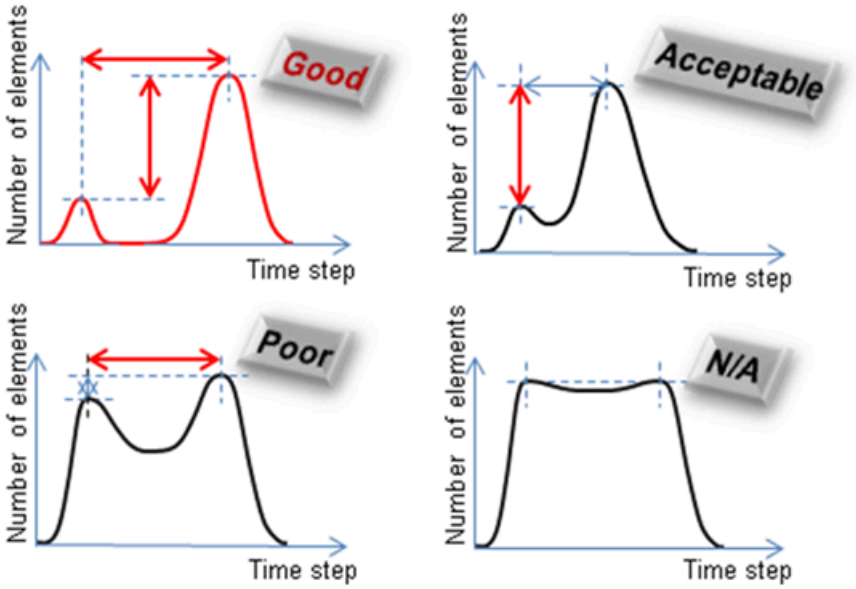


Figure 428:

It is particularly adapted to models with main and subdomains of:

- very different mesh densities hence very different time steps
- different formulations like Lagrange, ALE or Lagrange, and SPH, provided the ALE or SPH domain is larger than the Lagrange domain

Theoretical Speedup

This efficiency of the multi-domain method can be measured by the speedup coefficient. That is the ratio between the elapsed time of the original computation and the elapsed time obtained with the multi-domains method.

If the CPU cost of the main program (RAD2RAD) and time spent in communications are negligible, and that the time step and the cost per cycle of each domain are constant during the computation, an estimation of the speedup can be computed in order to determine if the use of multi-domains is relevant or not.

In the case of 2 domains, A and B, A being the domain with the smallest time step, the speedup can be obtained using the following formula:

$$Speedup = \frac{T_{MONO}}{T_{MULTI}} = \frac{(Ne_A C_A + Ne_B C_B) Nc_A}{Ne_A C_A Nc_A + Ne_B C_B Nc_B} \quad (317)$$

Where,

Nc	Number of cycles for each domain
Ne	Number of elements of each domain

C Average cost per element and per cycle for each domain

The formula can be rewritten as:

$$Speedup = \frac{\xi + (1-\xi)\cdot\psi}{\xi\cdot\gamma + (1-\xi\cdot\gamma)\cdot\psi} \quad (318)$$

Where,

$\xi = \frac{C_A}{C_B}$ Average cost per cycle ratio between domains

$\gamma = \frac{Nc_A}{Nc_B} = \frac{dt_B}{dt_A}$ Time step ratio

$\psi = \frac{Ne_A}{Ne_A + Ne_B}$ Percentage of elements in the domain with the smallest time step

If the average cost per cycle is the same in the 2 domains then the formula becomes:

$$Speedup = \frac{1}{\gamma + (1-\gamma)\cdot\psi} \quad (319)$$

Therefore, the speedup is very high when γ and ψ are close to zero, meaning that domain A is small compared to B and time step ratio is high. This is what is depicted in Figure 428.

Model Setup

The two possible model setups are:

- Single Input File Format
- Multiple Input File Format

The Radioss runs are completely independent and do not communicate directly with each other. Each subdomain uses its own time step as outlined below:

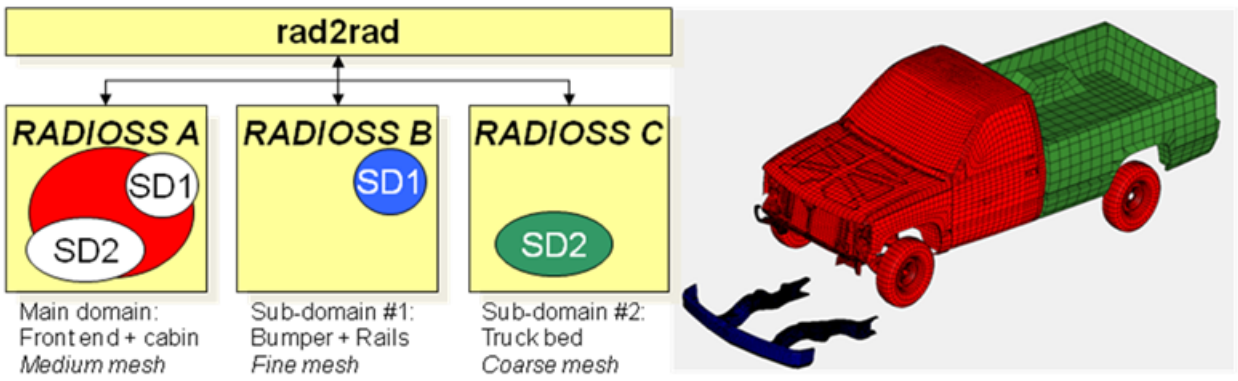


Figure 429:

The time step of each subdomain is arbitrary; but to allow the best gain in terms of speedup they should be significantly different from each other.

The best manual decomposition will be obtained by dividing global model into parts with a large number of elements having a big time step on one side, and a small number of elements with small time step on the other side.

All communication, data transfers, time step synchronization, equilibrium and stability conditions on the domain frontiers are assured by the main program.

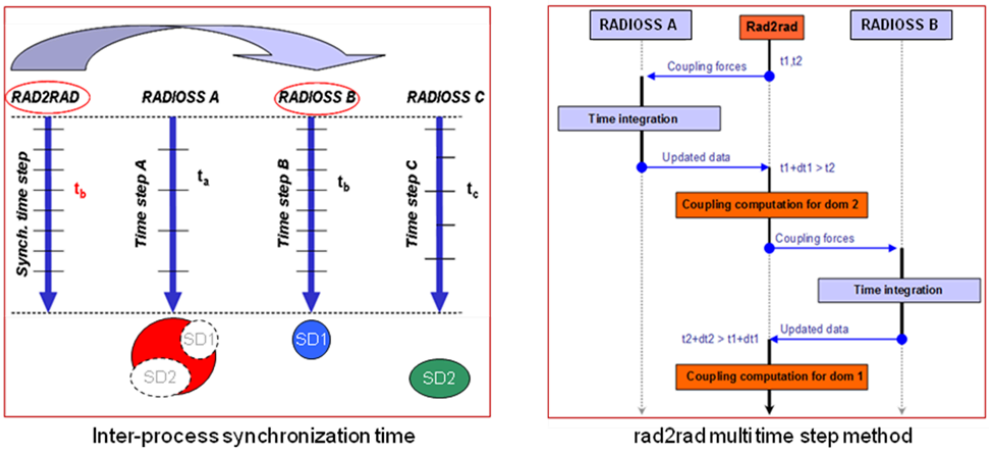


Figure 430:

New in 14.0

As of 14.0, Fluid-Structure Interaction simulations, including ALE and/or SPH are possible.

New in 13.0

As of 13.0, Multi-Domain is fully Hybrid-MPP compatible, i.e. enabling multiple threading for RAD2RAD.

Hints

For better computational performances:

- The subdomain should be less than 30% of the full domain (main _ sub-domains) in terms of nodes or elements and the natural (or acceptable mass scaling) time step ratio between subdomain and main domains should be below 1/3.
- Apply the same rule for pure Lagrange or FSI models, which is to put the parts with a significantly smaller time step in the subdomain.
- For pure Lagrange Multi-Domain models, the RAD2RAD workload must be minimized as much as possible by reducing the communication flow between domains, with this aim, contact interfaces and connections between sub and main domains should be minimized as much as possible.
- It is important to check that the subdomain is not part of a global contact interface, but has its own subdomain self-impact and to create cross domain contacts not larger than needed.
- It is strongly recommended to symmetrize each created cross domain contact interfaces.
- For Multi-Domain with ALE or SPH, there is no need to minimize the contact interfaces between fluid and structure.

- For models with a subdomain having a relatively small amount of elements, the number of allocated CPU should respect the empirical rule of at least 2000 element per CPU and it is advised to use the maximum possible number of threads per node of the used cluster.
- In one model, scattered parts with low time steps do not necessarily justify multiple subdomains. If these parts have similar low time steps, they must be put into one subdomain.
- If connections between subdomains and main domains also have a small time step, these connections should be part of the subdomain.
- Subdomains essentially composed of Lagrange 3D elements, make the Multi-Domain less efficient than if it was composed of 2D and 1D elements.
- For Multi-Domain FSI, it is advised to use the fluid in the main domain and defined `/DT/ALE` with an elementary free time step with a scaling factor of 0.5 for ALE. In the Langrange subdomain a nodal time step `/DT/NODA` with a scaling factor of 0.9 should be used.

Single Input File Format

Model Setup

A multiple input format setup was first introduced in Radioss Multi-Domain technique. The main drawback of this setup is that it implies a lot of work for the user who has to manually build several independent input files. It can be very long, difficult and the source of mistakes in the case of small domains extracted from very large and complex models.

The purpose of the single input file setup is to simplify the task of the user by building subdomains automatically. Only one Starter input file is required that includes the entire model, like a classic computation. Only the parts of the model to place in subdomains have to be specified. Then, the Starter automatically extracts the specified domains from the full model and generates one restart file for each domain. When using this method there can only be one subdomain but that subdomain can be made of multiple parts.

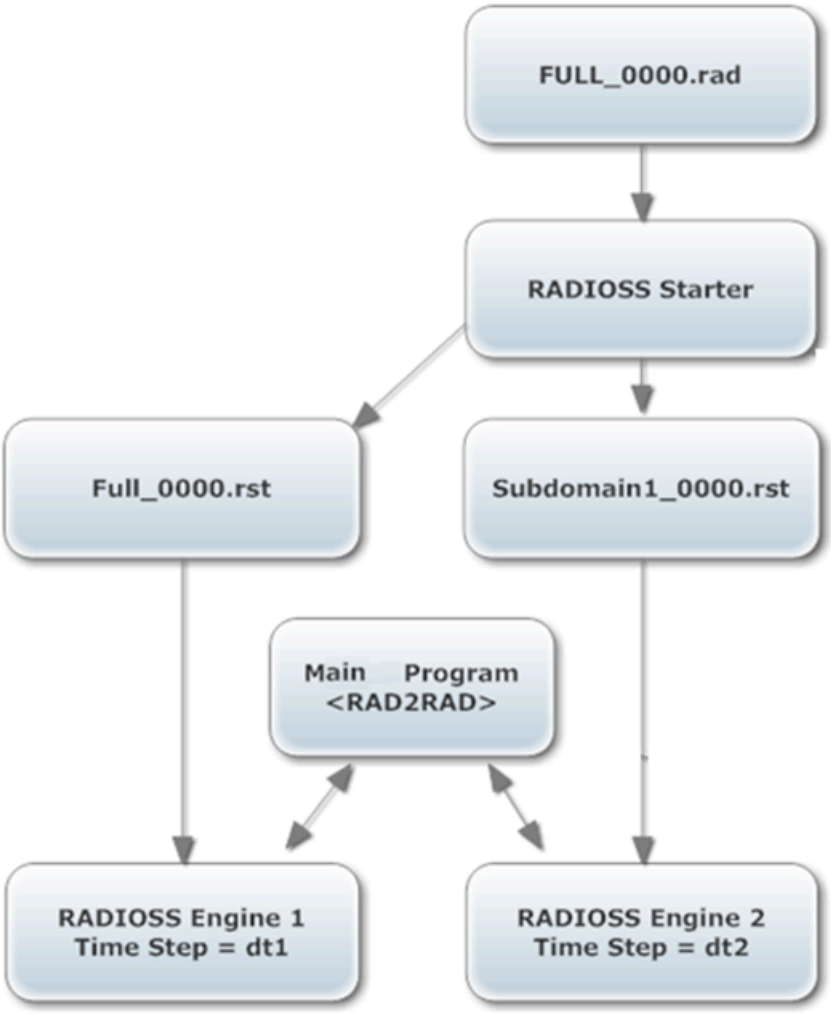


Figure 431: Subdomain Setup Architecture

Currently, only one domain can be specified, but this limitation can be extended in the future.

Automatic Creation of Subdomain

The subdomain is simply defined by a list of parts.

Split Model

The first step of the creation of a subdomain is the splitting of the full model. This is achieved by launching one Starter child process per domain. Each process only keeps the part associated to it and the corresponding nodes and elements. Entity groups, such as `/GRNOD`, `/GRPART` or `/SURF` are split as well, allowing a split of a lot of options (if an option refers to nothing in one domain it is suppressed). Some options are more complex to split and need a specific treatment which often implies a modification of the domain definition. Some other options can not be split. For more information about options incompatible with Multi-Domains, refer to Current Limitations.

As a result, a subdomain is most of the time composed of the parts specified by the user (with their related nodes and elements) and also by some other elements or nodes that are automatically added because of the split of some options.

Connections between Domains

The second step is the detection of the interactions between domains. The first type of interaction is the direct connection. The Starter automatically detects the common nodes between domains and creates Node-to-Node connections.

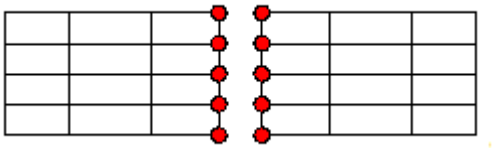


Figure 432:

Contact between Domains

The second type of interaction between domains is the contact. The computation of contact forces between domains is based on the artificial skin method. It means that the contacts are not computed by the external program RAD2RAD but inside one of the two domains called main domain.

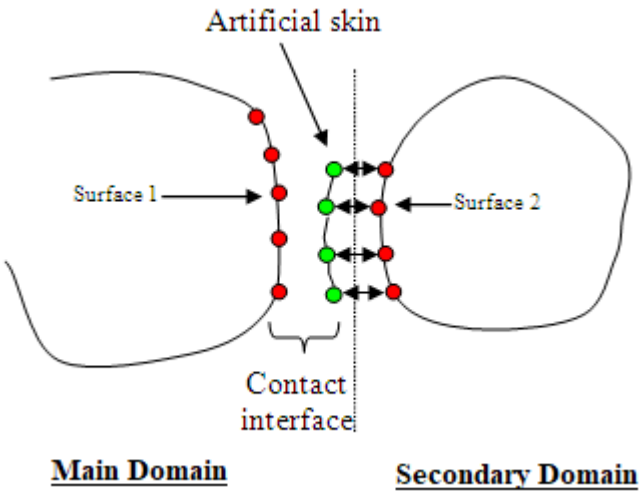


Figure 433:

With the single input file setup the contacts between domains are always computed in the subdomain mainly because the quality of the coupling is far better when the contact is treated in the domain with the smallest time step.

Therefore, the surfaces of the main domain concerned by the contact with a subdomain are automatically duplicated in this subdomain with a void material having the same density and Young's modulus. A cross-domain node-to-node coupling is then established between the nodes of the artificial skin in the subdomain and the ones of the original surface in the main domain.

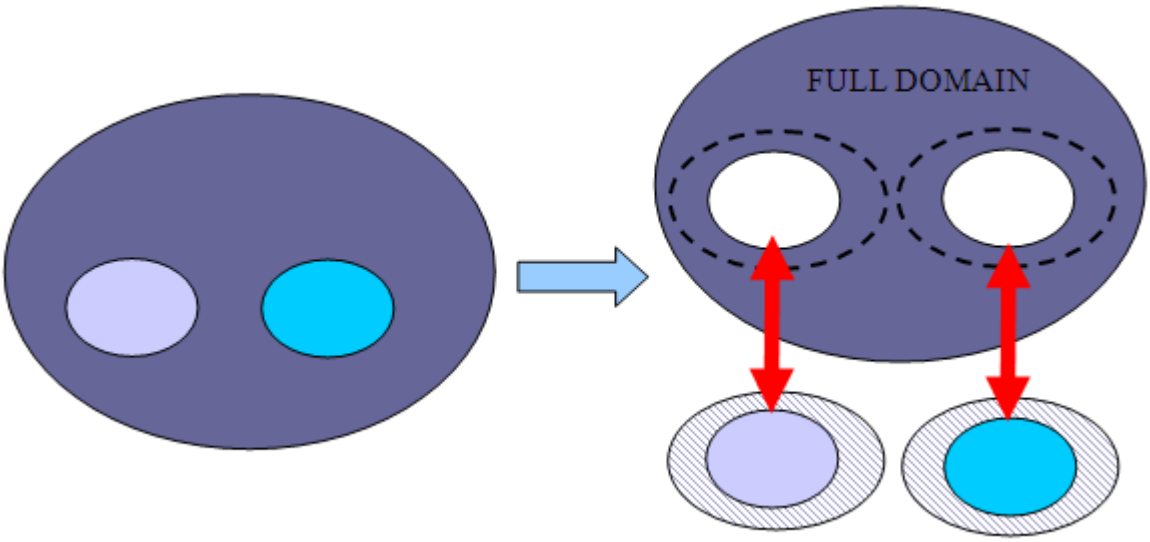


Figure 434:

This implies that in order to get good performances, all the contacts between main and sub-domain must be put in specific contact interfaces with as small as possible contact surfaces but without missing potential contacts. For example, if only one self-contact interface is used in a model with one sub-domain, the subdomain can potentially impact every parts of the main domain, then, the main domain will be fully duplicated with void material inside the sub-domain. The cost of the RAD2RAD will be huge and the performances very poor.

Note: If contact interfaces are badly defined, a warning message `multidomain interface is too big` is issued if the size of the duplicated part is less than 50% of the size of the full model. If this percentage is bigger than 50% an error message is issued.

Furthermore, it is strongly recommended to not have asymmetrically defined cross domain contact interfaces where the subdomain is only defined on either the secondary or the main side of the interface. In this case, only a portion of the contact interface is computed in the subdomain, what remains being computed in the main domain. In order to avoid asymmetrical contact interfaces it is recommended to systematically symmetrize the cross domain contact interfaces (by defining the subdomain in both: main and secondary sides of the cross domain contact or correcting already existing asymmetrical contacts by adding a symmetric contact interface).

Note: If a cross domains contact interface is split, a warning message `multidomain split contact interface` is issued.

Radioss contact interfaces that are compatible with Multi-Domain are:

- `/INTER/TYPE5`
- `/INTER/TYPE7`
- `/INTER/TYPE10`

- /INTER/TYPE11
- /INTER/TYPE18
- /INTER/TYPE24, if ($surf\_ID_1 > 0$, $surf\_ID_2 > 0$) or ($surf\_ID_1 > 0$, $surf\_ID_2 = 0$). Not if ($grnd\_ID_s > 0$, $surf\_ID_1 = 0$, $surf\_ID_2 > 0$).

User-defined Connections between Domains

Some kinematic conditions can also create connections between domains, if they are defined across the Multi-Domains interface.

- Rigid bodies: in the case of rigid bodies that connecting two domains, a specific treatment is applied. The rigid body is split in two, and a computation of mass, inertia matrix and position of the center of gravity is performed for each domain. Then, the two mai nodes of the two parts of the rigid body are coupled by RAD2RAD using a formulation similar to the one used for classical node-to-node coupling but adapted to non-spherical inertia.
- Tied interface: in the case of a tied interface TYPE2 with main elements on one domain and secondary nodes the other one, a different strategy is used. This strategy is similar to what is done for the contacts. The main elements are duplicated with void material in the domain containing the secondary nodes in order to have the tied interface fully defined in this domain. Then, the Multi-Domain coupling is only applied on the nodes of the main elements. If both domains contain secondary nodes, the duplication of main elements is performed on both sides.
- Rigid Links, RBE3 and Cylindrical Joints: the same idea is applied to rigid link and RBE3. If one of these options has secondary nodes on two domains, all the missing secondary nodes are duplicated on both sides and all the secondary nodes are coupled by the RAD2RAD. The option is then computed on both sides.

Other connection types cannot be split. It means that these connections can only be used inside one domain and away from the coupling zone. These options are:

- /MPC
- /RBE2
- /GJOINT

Data Input

Starter Input File

The sub-domains are specified by parts.

```
/SUBDOMAIN/subdomain_ID
subdomain_title
Part1    Part2    ...    Partn
```

Where,

Part$_n$	Identifier of the parts that belong to the subdomain
subdomain_ID	Domain identifier
subdomain_title	Subdomain name (that will be the rootname of one Engine file)

Engine Input File

One Engine input file is required for each domain and in order to activate the coupling, these files must contain the following directive:

```
/RAD2RAD/ON
```

One Engine file name comes from the Starter input file rootname: `"full_model_rootname"_0001.rad` and the other Engine input file name relating to the subdomain comes from the subdomain_title: `subdomain_title_0001.rad` given to the `/SUBDOMAIN` card in the Starter input file.

RAD2RAD Input File

The RAD2RAD input file is a text file defining some additional information required by the RAD2RAD program. With the sub-domain setup the RAD2RAD input file is automatically generated by the Starter. One can access it and modify it in order to change the parameters of the Multi-Domain computation before launching the RAD2RAD and Engine processes. The name of this file is based on the full model rootname `"_model_rootname"_0000.r2r`.

> **Note:** For more information concerning the RAD2RAD input file, refer to the online documentation of Multi-Domains.

Data Output

Starter Output Files

Separate Starter output files are generated for each domain.

Time History Files

A single Time History file is generated containing all information from both domains. This file has the same rootname as the Starter input file. This file content is equivalent to what is obtained following a classical monodomain computation.

All parameters for Time History output (type of time history, output frequency, format, and so on) must be specified in the Engine file of the main domain. If the parameters for time history are defined in the sub-domain Engine input file, they will be ignored.

As the frequency for the printout of the TH file is defined by the main domain, the minimum time interval allowed between two prints of a TH file is the time step of the main domain. For better accuracy, it is recommended to use a time frequency that is much higher than the time step of the main domain.

ABF Files

One ABF file is generated by each Radioss Engine. Therefore, to plot the whole model global variables, each domain's global variable of must be added up.

Output File

Global variables of the whole model can be computed by simply summing up the global variables printed in the output file of each domain (each Engine output).

There are two exceptions:

- Energy error: The energy balance is computed in each domain independently from the other domain. It means that for each domain, the Multi-Domain coupling forces are considered as external forces and their work is added to the work of the external forces. This work is only used internally for the energy balance computation. It is not included in the value of the work of the external forces printed in the output file or in the time history.
- Mass change: The mass change is also computed locally, meaning that it is the ratio of the added mass in the selected domain over the mass of this domain.

Animation Files

A set of animation files is generated by each Engine. With HyperView it is possible to visualize the two domains simultaneously by simply making an overlay.

RAD2RAD Output File

The output file `rad2rad.out` is generated by the RAD2RAD executable. This file contains useful information about the connections between domains (number of common nodes, type of coupling, and so on).

SPEEDUP Estimation

As of version 14.0, an estimation of the speedup is computed in the Starter in order to determine the potential efficiency of the multi-domains method. The value is printed in the Starter output file of the main domain. So, if during the computation the time steps change drastically in one domain, the speedup estimation will be irrelevant.

Time step control options defined in the Engine input file (`/DT/NODA/CST`, `/DTIX`, ...) accounted for in the time step estimations in the Starter.

CPU Allocation

The Radioss domains are treated sequentially, which means that only one Radioss process is run at a time. The total CPU resource is automatically allocated to the running process and the others are put in a no CPU consuming waiting mode. With the subdomain setup, the same number of SPMD domains is automatically allocated to all domains. For better performances, the same number of SMP threads per SPMD domain should be used for each domain when running in Hybrid-MPP.

As of version 12.0.210, the RAD2RAD executable is fully parallelized. It means that RAD2RAD must be launched exactly like the Engine executables (same `mpi` options) and that the same number of SPMD domains must be used for both Engine and RAD2RAD processes.

Launch a Multi-Domain Analysis

There are two ways to launch a Multi-Domain computation: using the Altair Compute Console or manually.

1. The easiest way to launch a Multi-Domain computation is to use the Altair Compute Console.

a) Select the multi-domain Starter file as the input file and define the number of cores to be used in the simulation.
The Compute Console will then run the Starter, Engine, and the RAD2RAD process. Refer to Altair Compute Console (ACC) for more details about using the Altair Compute Console with Radioss.

2. To launch a Multi-Domain computation, use the command line to browse to the working directory containing the input files.
3. Launch the Starter in a terminal using the command:

Option	Description
Linux	`<install_dir>/hwsolvers/bin/linux64/starter_version -i "rootname"_0000.rad`
Windows	`<install_dir>\hwsolvers\bin\win64\starter_version -i "rootname"_0000.rad`

4. Launch RAD2RAD in a terminal using the command:

Option	Description
Linux	`<install_dir>/hwsolvers/bin/linux64/r2r_version "rootname"_0000.r2r`
Windows	`<install_dir>\hwsolvers\bin\win64\r2r_version "rootname"_0000.r2r`

RAD2RAD will then wait for Radioss connections from the individual domains.

Note: The file `"rootname"_0000.r2r` is automatically created by the Starter.

5. Launch Engine for each domain in separate terminals.

Option	Description
Linux	`<install_dir>/hwsolvers/bin/linux64/engine_version -i "rootname"_0001.rad`
Windows	`<install_dir>\hwsolvers\bin\win64\engine_version -i rootname_of_the_subdomain"_0001.rad`

All the Radioss processes will then connect automatically to RAD2RAD.

6. Launch an SMP manual script.

```
Linux : run_linux_SMD
./s_<version>_linux64 -i FULL_0000.rad

./e_<version>_linux64 -nt 4 -i FULL_0001.rad > out_1 &
./e_<version>_linux64 -nt 4 -i SUBDOM_0001.rad > out_2 &
./r2r_<version>_linux64 -nt 4 FULL_0000.r2r
```

```
Windows : run_win_SMD.bat
E:\Rad2rad\s_<version>_win64.exe -i FULL_0000.rad
```

```
set KMP_STACKSIZE=64M
start /B E:\Rad2rad\e_<version>_win64.exe -nt 4 -i FULL_0001.rad > out1
start /B E:\Rad2rad\e_<version>_win64.exe -nt 4 -i SUBDOM_0001.rad > out2
start /B E:\Rad2rad\r2r_<version>_win64.exe -nt 4 FULL_0000.r2r
```

```
Windows (cygwin) : run_win_SMD
./s_<version>_win64.exe -i FULL_0000.rad;

./e_<version>_win64.exe -nt 4 -i FULL_0001.rad > out1&
./e_<version>_win64.exe -nt 4 -i SUBDOM_0001.rad > out2&
./r2r_<version>_win64.exe -nt 4 FULL_0000.r2r;
```

7. Launch an SPMD manual script.

```
Linux : run_linux_SPMD
./s_<version>_linux64 -np 4 -i FULL_0000.rad

mpiexec -n 4 ./e_<version>_linux64_impi -i FULL_0001.rad > out_1 &
mpiexec -n 4 ./e_<version>_linux64_impi -i SUBDOM_0001.rad > out_2 &
mpiexec -n 4 ./r2r_<version>_linux64_impi FULL_0000.r2r
```

```
Windows : run_win_SPMD.bat
E:\Rad2rad\s_12_main_win64.exe -np 4 -i FULL_0000.rad

set KMP_STACKSIZE=64M
start /B mpiexec -n 4 E:\Rad2rad\e_<version>_win64_impi.exe -i FULL_0001.rad> out1
start /B mpiexec -n 4 E:\Rad2rad\e_<version>_win64_impi.exe -i SUBDOM_0001.rad> out2
start /B mpiexec -n 4 E:\Rad2rad\r2r_<version>_win64_impi.exe FULL_0000.r2r
```

```
Windows (cygwin) : run_win_SPMD
./s_<version>_win64.exe -np 4 -i FULL_0000.rad;

mpiexec -n 4 ./e_<version>_win64_impi.exe -i FULL_0001.rad > out1&
mpiexec -n 4 ./e_<version>_win64_impi.exe -i SUBDOM_0001.rad > out2&
mpiexec -n 4 ./r2r_<version>_win64_impi.exe FULL_0000.r2r;
```

Current Limitations

Only one subdomain can be defined.

These options are not compatible with `/SUBDOMAIN`:

- `/DFS/DETPOINT/NODE`
- `/FX_BODY`
- `/SPHBCS`

The following connections can only be used inside one domain, but cannot be used in the coupling zone or across the Multi-Domain interface.

- `/GJOINT`
- `/MPC`
- `/RBE2`

Multi-Domain is incompatible with all kinematic conditions based on Lagrange multipliers, due to incompatibility with the coupling formulation.

Multi-Domain is not yet compatible with `AMS` (Advanced Mass Scaling), Rayleigh Damping (`/DAMP`), Dynamic Relaxation (`/DYREL`), unless nodes are not part of the cross-domain interface or contact, interfaces TYPE1.

The GAUGE, INTER and RWAL type sensors are not yet synchronized between domains. Meaning that if a sensor and all its associated features are not confined in one domain, the behavior of this sensor may be incorrect. Nevertheless, sensors of type DIST, ACCE and TIME are fully compatible with Multi-Domain and synchronized between domains.

Multiple Input File Format

Model Setup

The multiple input format setup was the first to be introduced in Radioss Multi-Domain technique. Each subdomain is built as a separate complete Radioss model, using its own complete input files. The coupling between the Radioss Engine computations is ensured by the RAD2RAD program based on the connections between domains defined in each Radioss model.

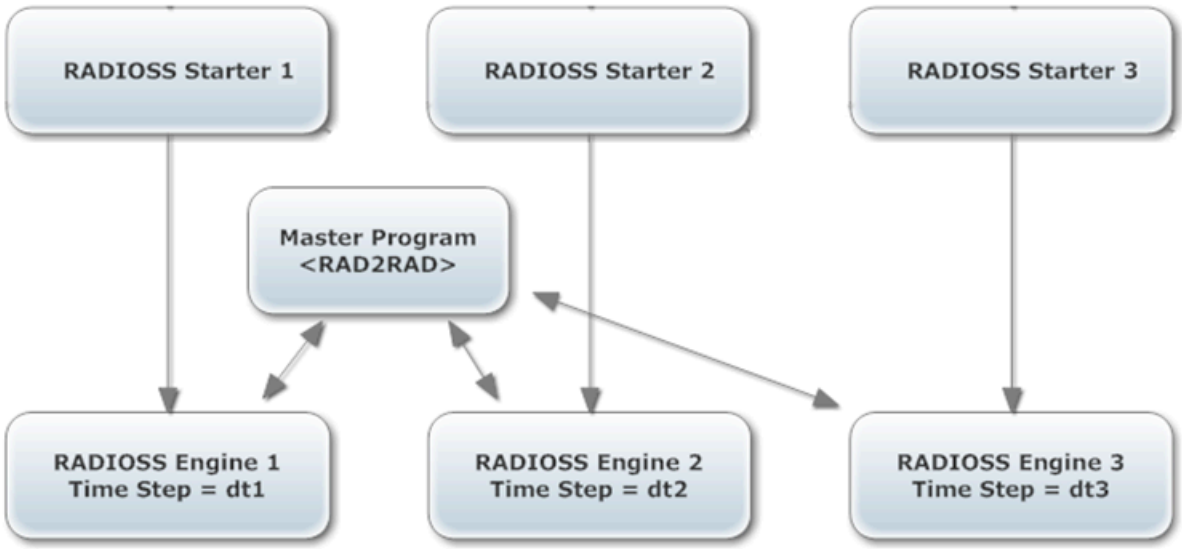


Figure 435:

Connection between Domains

A connection (or link) between 2 domains is always defined by 2 groups of nodes relating to the two corresponding meshes to be connected to each other and the type of this connection. Below is a list of available types of connections in RAD2RAD.

TYPE 1: Shell to Shell (Schur Dual Method)

It may be used to link shell element meshes through Lagrange multipliers. The meshes may be compatible or incompatible and the interfaces may be open or closed.

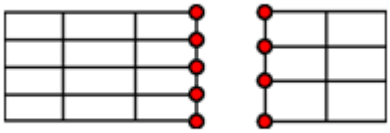


Figure 436:

> **Note:** In the case of incompatible meshes and open interface, the beginning and the ending point of the two faces must coincide. Heading the same way, in the case of closed interface, there must be at least one common point between the two faces.

TYPE 2: Shell to Beam (Schur Dual Method)

This type of link is very similar to TYPE 1; but it is adapted to beam/shell connection.

> **Note:** The restrictions on the shell part are the same as for TYPE 1 in the case of incompatible meshes.

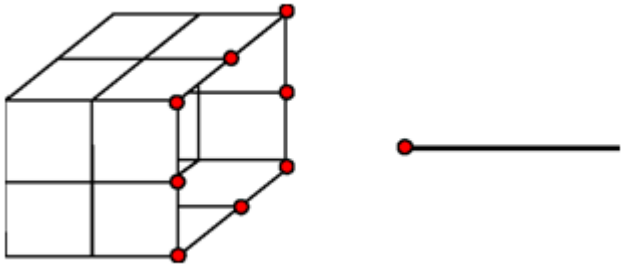


Figure 437:

TYPE 4: Node to Node (Schur Primal Method)

In the case of compatible meshes, the connection between domains can be reduced to simple node to node relations.

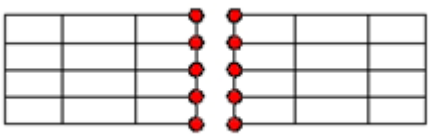


Figure 438:

This method is strictly equivalent to TYPE 1 for compatible meshes; but consumes less CPU time, as no matrix assembly and no inversion have to be performed. Furthermore, in this case, since only nodal data is needed, it can be used to connect any type of nodes (attached to any type of elements).

Note: In the case of shell to solid connection, the continuity condition is applied only on translational degree of freedom.

Contact between Domains

TYPE 5: Connection

It is also possible to compute contact between domains using the artificial skin method. The contacts are not computed by RAD2RAD, but inside one of the two domains called main domain. Therefore, the part of the secondary domain concerned by the contact must be duplicated in the main domain with a void material having same density and Young's modulus. The contact is then treated normally by Radioss inside the main domain and a node-to-node specific connection TYPE 5 (similar to connection TYPE4) must be used to connect the nodes of the artificial skin and the corresponding nodes of the secondary domain.

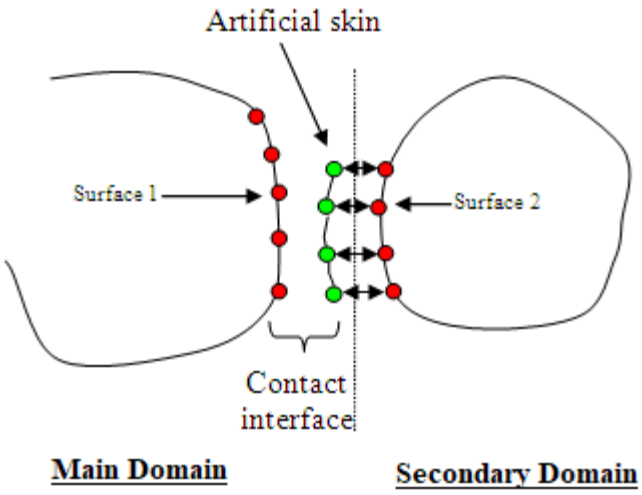


Figure 439:

In the Multi-Domain main input file, the definition of connection TYPE5 is similar to other connections; except that the nodes of the main domain (nodes of the artificial skin) must be declared first:

```
/LINK/TYPE5
Main_Domain Link_Id1
Second_Domain Link_Id2
```

Radioss contact interfaces that are compatible with Multi-Domain are:

- /INTER/TYPE5
- /INTER/TYPE7
- /INTER/TYPE10
- /INTER/TYPE11
- /INTER/TYPE18

- /INTER/TYPE24, if ($surf\_ID_1 1 > 0$, $surf\_ID_2 > 0$) or ($surf\_ID_1 > 0$, $surf\_ID_2 = 0$). Not if ($grnd\_ID_s > 0$, $surf\_ID_1 = 0$, $surf\_ID_2 > 0$).

> **Note:** The RAD2RAD connection TYPE5 is incompatible with contact interfaces using Lagrange multipliers formulation.
>
> Mass and nodal stiffness are transferred by RAD2RAD connection TYPE5 from the secondary domain to the artificial skin. Therefore, modifications of mass and nodal time step may be observed in the main domain at the beginning of the computation.

In the Multi-Domain main input file, the definition of connection TYPE5 is similar to other connections; except that the nodes of the main domain (that is nodes of the artificial skin) must be declared first:

```
/LINK/TYPE5
Main_Domain Link_Id1
Second_Domain Link_Id2
```

If kinematic constraints are applied on the part of the secondary domain that is duplicated in the main domain, they must be duplicated, in case of:

- A rigid body duplication, the main node must be specified in the RAD2RAD connection TYPE5.
- Interface TYPE2 duplication, an additional flag /TIED must be specified in the RAD2RAD input file.

> **Note:** For rigid bodies or interfaces TYPE2, the secondary nodes can be removed from the RAD2RAD connection. Their behavior is defined only by their main nodes or main elements. That is why Multi-Domains will not have any effects on them.
>
> The RAD2RAD is incompatible with Lagrange multipliers kinematic conditions, such as interface /INTER/LAGMUL/TYPE2 and /RWALL/LAGMUL.

Data Input

Starter Input File

Definition of connections (or links) between domains:

```
/EXTERN/LINK/Link_ID
title
 gr_ID
```

Where,

gr_ID	Nodal group identifier defining frontier nodes with other domain(s)
Link_ID	Link identifier

The number of external links in a Radioss model is not limited. The only restriction is that different links should not contain common nodes. Each link defines a distinct interface between current model and the external world.

Engine Input File

One Engine input file is required for each domain. In order to activate the coupling, all the Engine files must contain the following directive:

```
/RAD2RAD/ON
```

The RAD2RAD input file is a text file defining some additional information required by the RAD2RAD program. The names of the domains to connect and the connections between domains must be specified in the RAD2RAD input file. For each connection the identifier of the links that are connected and the used connection types must be specified. Some Multi-Domain specific parameters also have to be entered.

Note: For more information concerning the RAD2RAD input file, refer to the online documentation of Multi-Domain.

CPU Allocation

The Radioss domains are treated alternately; which means that only one Radioss process is running at a time. The full CPU resource is automatically allocated by the RAD2RAD to the running Engine process and the other processes are put in a no CPU consuming waiting mode. It means that all the same available CPU resources must be allocated to each domain during its computation.

Launch a Multi-Domain Analysis

There are two ways to launch a Multi-Domain computation, manually and through a script.

SMP Script Example
SPMD Script Example

1. To manually launch a Multi-Domain computation, use the command line to browse to the working directory that contain the input files of the individual domains (`_0000.rad` and `_0001.rad`) and the Multi-Domain main input file (`input.dat`).
2. Launch Starter for each domain in a terminal using the command:

Option	Description
Linux	`<install_dir>/hwsolvers/bin/linux64/s_version -i "filename"`
Windows	`<install_dir>\hwsolvers\bin\win64\s_version -i "filename"`

3. Launch RAD2RAD in a terminal using the command:

Option	Description
Linux	`<install_dir>/hwsolvers/bin/linux64/r2r_version input.dat`

Option	Description
Windows	`<install_dir>\hwsolvers\bin\win64\r2r_version input.dat`

RAD2RAD will then wait for Radioss connections from the individual domains.

4. Launch Engine for the individual domains in separate terminals, as follows:

Option	Description
Linux	`<install_dir>/hwsolvers/bin/linux64/e_version -i "filename"`
Windows	`<install_dir>\hwsolvers\bin\win64\e_version -i "filename"`

All the Radioss processes will then connect automatically to RAD2RAD.

An easier way to launch a Multi-Domain computation is to use a script.

5. Launch Starter for each domain the same way as described above.
6. Add a script called `run_SMP` or `run_SPMD` in your working directory.
7. In the script, change name of the input files, location of executables (MPI, RAD2RAD and Radioss Engine) and the number of processors, if necessary.
8. Run the script in a terminal.
   SMP Example

```
Linux : run_linux_SMP
./s_<versin> < PART1_0000.rad
./s_<version> < PART2_0000.rad
./e_<version> < PART1_0001.rad > out_1 &
./e_<version> < PART2_0001.rad > out_2 &
./r2r_<version> input.dat
```

```
Windows : run_win_SMP.bat
E:\Rad2rad\s_<version>.exe -i PART1_0000.rad
E:\Rad2rad\s_<version>.exe -i PART2_0000.rad
set KMP_STACKSIZE=64M
start /B E:\Rad2rad\e_<version>.exe -i PART1_0001.rad > out1
start /B E:\Rad2rad\e_<version>.exe -i PART2_0001.rad > out2
start /B E:\Rad2rad\r2r_<version>.exe input.dat
```

```
Windows (cygwin) : run_win_SMP
./s_<version>.exe < PART1_0000.rad;
./s_<version>.exe < PART2_0000.rad;
./e_<version>.exe < PART1_0001.rad > out1&
./e_<version>.exe < PART2_0001.rad > out2&
./r2r_<version>.exe data4.dat;
```

SPMD Example

```
Linux : run_linux_SPMD
./s_<version> < PART1_0000.rad
./s_<version> < PART2_0000.rad
/opt/hpmpi/bin/mpirun -np 4 ./e_<version> < PART1_0001.rad > out_1 &
/opt/hpmpi/bin/mpirun -np 4 ./e_<version> < PART2_0001.rad > out_2 &
../exec/r2r_<version>_impi data4.dat
```

```
Windows : run_win_SPMD.bat
E:\Rad2rad\s_<version>.exe -i PART1_0000.rad -n 4
```

```
E:\Rad2rad\s_<version>.exe -i PART2_0000.rad -n 4
set KMP_STACKSIZE=64M
start /B mpiexec -n 4 E:\Rad2rad\e_<version>.exe -i PART1_0001.rad > out1
start /B mpiexec -n 4 E:\Rad2rad\e_<version>.exe -i PART2_0001.rad > out2
start /B E:\Rad2rad\r2r_<version>.exe input.dat
```

```
Windows (cygwin) : run_win_SPMD
./s_<version>.exe < PART1_0000.rad;
./s_<version>.exe < PART2_0000.rad;
mpiexec -n 4 ./e_<version>.exe < PART1_0001.rad > out1&
mpiexec -n 4 ./e_<version>.exe < PART2_0001.rad > out2&
./r2r_<version>.exe data4.dat;
```

RAD2RAD Parallelization

If the sizes of the Multi-Domain interfaces are big, the time spent in RAD2RAD for computing the coupling forces can be long. It is now possible to reduce this time by splitting the task of computing the coupling among several processors. Contrary to single input file setup, for the multiple input file setup only SMP parallelization of the RAD2RAD is available. The number of processors that can be used must be less than the number of cores available on the hardware node where RAD2RAD is launched. For example, a good value is 6 for a machine with 8 cores. The number of threads for the RAD2RAD can be specified using `-nt` in the command line like for the Engines or directly in the RAD2RAD input file using the keyword `/PROC/nthread`.

Note: The affect of the parallelization of the RAD2RAD on the CPU time is negligible if the number of nodes that are treated by the RAD2RAD is low (< Nb of Cpus x 1000).

Note: As the connected Radioss Engines are sleeping when the RAD2RAD is active, the RAD2RAD parallelization is independent from the Engines parallelization. It means that one processor can be assigned to all the Engines as well as one processor is assigned to the RAD2RAD. They will use it one after the other.

MPI Settings

When using Multi-Domains and SPMD Engine executables, the following MPI environment variables should be used in order to improve the performances:

- HP-MPI: *MPI_FLAGS* set to y0
- Open-MPI: *OMPI_MCA_mpi_yield_when_idle* set to 1
- Intel-MPI: Default settings can be used

Output Files

- Radioss: Separate output files are generated by each Radioss process.
- RAD2RAD: A specific output file named `rad2rad.out` is generated by RAD2RAD.

Version Restrictions

- Each sub-domain is constructed as a complete. Separate Radioss model, using its own complete input files.

- The data transfers between Radioss processes and main program use the pipe system method of communication. All Radioss Engines and the RAD2RAD must be started on the same hardware node.
- Kinematic conditions on common nodes between domains are compatible only with connection TYPE5.
- The RAD2RAD program is not parallelized for multiple input files setup. This may lead to reduced performance in Multi-Domain computations with large subdomains running on a large number of CPU's.
- Currently, the Multi-Domain calculus is not automatically executed by the main program. All Radioss Starters must have been executed before the RAD2RAD program is launched. Although, batch execution is possible, RAD2RAD and all Radioss Engine processes must be launched independently and in parallel, on the single hardware node.
- For each Radioss domain, the number of links is limited to 15 and the number of CPU's is limited to 128.

Example: RAD2RAD Input File

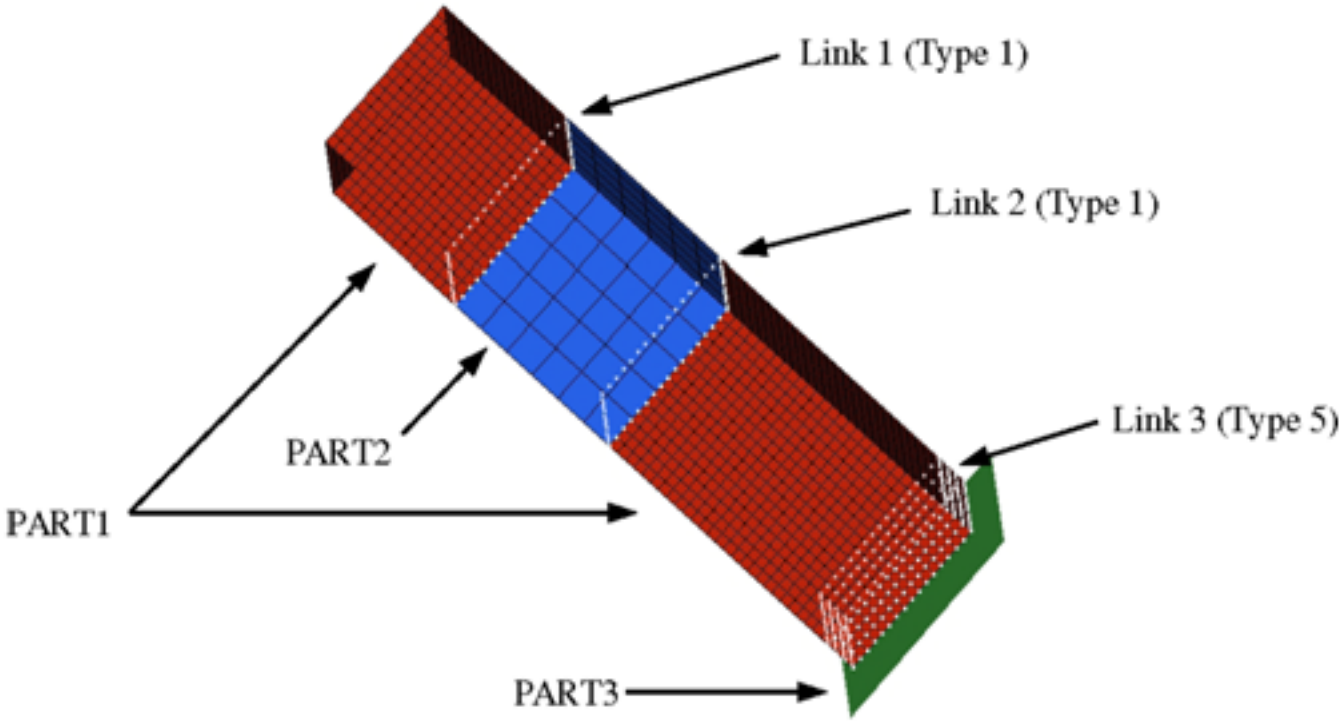


Figure 440: Model

RAD2RAD Input File

```
$=============================================================
$ RAD2RAD R8 INPUT FILE :
$=============================================================
$ 1. PARTS DEFINITION
$=============================================================
/DOMAIN/PART1
1 2 3
```

```
/DOMAIN/PART2
4 5
/DOMAIN/PART3
9
$=================================================================
$ 2. INTERFACES  DEFINITION
$=================================================================
/LINK/TYPE1
PART1 2
PART2 5
/LINK/TYPE1
PART1 1PART2 4
/LINK/TYPE5
PART3 9
PART1 3
$=================================================================
$3 OPTIONS
$=================================================================
/MLTPS/ON
0.1 0.1
/MESHL/MORFIN
/RADIUS/1e-7
/END   $=================================================================
```

Design Optimization

Optimization in Radioss was introduced in version 13.0. It is implemented by invoking the optimization capabilities of OptiStruct and simultaneously using the Radioss solver for analysis.

Optimization in Radioss provides Radioss users an easy way to set up optimization problems, and take advantage of optimization solutions in OptiStruct. The equivalent static load method (ESLM) is used to perform the optimization run.

Implementation

The optimization functionality in Radioss requires an extra input file apart from the usual Starter and Engine input files.

The extra file is an optimization input file termed `<name>.radopt` (the Starter and Engine files are usually named as `<name>_0000.rad` and `<name>_0001.rad`, respectively). The optimization input file (`<name>.radopt`) defines optimization entities such as the optimization objective, optimization constraints, design variables, optimization responses and so on. The typical set of input data required to optimize a model is listed as follows. The various optimization entries in Radioss that can be chosen to define the corresponding input data are also described.

Optimization Objective

The optimization objective defines the overall objective of the optimization run. It is driven by the application requirements for the model in the real world. For example, the overall objective of an optimized automobile vehicle design can be to minimize the mass of the vehicle. The optimization objective can be defined in Radioss by selecting an optimization response and specifying whether the response is minimized or maximized. The `/DESOBJ` data entry is used to define the optimization objective.

Optimization Constraint

The optimization constraint defines the upper and/or lower bound of a specific optimization response. The model is therefore optimized such that the value of the selected response lies within the defined bounds or constraints. For example, for a structure with a minimum mass objective, the stress response at a region of interest can be defined as an optimization constraint. The stress in the selected region can be constrained not to exceed a specific value (upper bound). The `/DCONSTR` data entry is used to define the optimization constraints in Radioss.

Design Variables

Design variables typically correlate the properties of a model to the optimization run. They can be linked to the physical properties (for example, shell thickness) of a system via `/DVPREL1` entries. The design variables vary (within their defined bounds) based on the optimization objective and the optimization constraints. Design variables are defined by `/DESVAR` (for size optimization), `/DTPL` (for topology optimization), `/DSIZE` (for free-size optimization), `/DSHAPE` (for free-shape optimization), and `/DTPG` (for topography optimization).

Refer to the respective data entry documentation for further information regarding their usage in Radioss. If you are an OptiStruct user, you will find that the Radioss optimization input entries are very similar to the OptiStruct optimization entries of the same name (for example, `/DTPL` is similar to the `DTPL` Bulk Data Entry in OptiStruct).

Pass Bulk Data to OptiStruct Input File

Information can be passed from an RADOPT run to the intermediate OptiStruct input file. The `/BULK`, `/BULKFMT`, `/BULKMAT`, and `/BULKPROP` Block Format Entries are available.

Set Environment Variables

The number of cores used by Radioss during the optimization can be defined using the OptiStruct script options `-rnt` and `-rnp` options.
If more advanced Radioss options need to be used, such as single precision or a specific version of Radioss, set the following environment variables.

Linux (bash):

```
export RADIOSS_SCRIPT=$ALTAIR_HOME/altair/scripts/radioss
export RADIOSS_ARGS=" -np 24 -nt -2 -sp -v 2019"
```

Windows:

```
set RADIOSS_SCRIPT=%ALTAIR_HOME%\hwsolvers\scripts\radioss.bat
set RADIOSS_ARGS=" -np 24 -nt -2 -sp -v 2019"
```

The script defined via `RADIOSS_SCRIPT` will be executed by OptiStruct when the Radioss outerloop simulation is done. The `RADIOSS_ARGS` options will be used as arguments to the `RADIOSS_SCRIPT`. The `RADIOSS_SCRIPT` must be a script that runs both the Starter and the Engine.

Run Optimization

The `-radopt` run option is used to run the `<name>.radopt` Radioss optimization input file in OptiStruct. OptiStruct then looks for the Radioss Starter and Engine files within the same working directory.

Use Solver GUI

Set `<name>.radopt` as Input files and use "`-radopt`" as option.

To specify the number of SMP cores used by OptiStruct use the `-nt` option. For Radioss, specify the cores using `-rnt` and `-rnp` which are the same as the `-np` and `-nt` options when running a Radioss simulation.

Using Solver Scripts (Linux machines)	`[optistruct@host~]$ $ALTAIR_HOME/scripts/optistruct <name>.radopt -radopt`
Using OptiStruct Solver Scripts (Windows machines)	`[optistruct@host~]$ $ALTAIR_HOME/hwsolvers/scripts/optistruct.bat <name>.radopt -radopt`

Using Radioss Solver Scripts (Windows machines)	`[optistruct@host~]$ $ALTAIR_HOME/hwsolvers/scripts/radioss.bat <name>.radopt -radopt`
Using Direct Calls to Executable (Linux machines)	`[optistruct@host~]$ $ALTAIR_HOME/hwsolvers/optistruct/bin/linux64/<optistruct_executable> <name>.radopt -radopt`
Using Direct Calls to Executable (Windows machines)	`[optistruct@host~]$ $ALTAIR_HOME/hwsolvers/optistruct/bin/win64/<optistruct_executable> <name>.radopt -radopt`

The Starter and Engine files (`<name>_0000.rad` and `<name>_0001.rad`) should be located in the same directory as `<name>.radopt`.

For MMO (Multi-Model Optimization) which with *MMOCID*=1 in `/DTPL`, then run with the command shown here.

```
[optistruct@host~]$ $ALTAIR_HOME/hwsolvers/optistruct/bin/win64/<optistruct_executable> -mmo <name>.fem
```

For example:

Two Load cases need be considered in optimization. Then create Radioss Optimization file `<name>.radopt` for each load case and create an `.fem` file (in this example is `mmo_topo.fem`) as shown below to define MMO. Run MMO with:

```
[optistruct@host~]$ $ALTAIR_HOME/hwsolvers/optistruct/bin/win64/<optistruct_executable> -mmo -mpi -np 3 -rnt 2  mmo_topo.fem
```

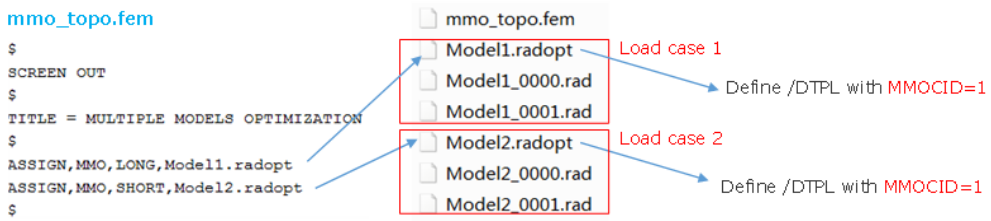


Figure 442:

Optimization Process

Radioss parses all three decks (optimization, Starter and Engine decks) to construct an optimization model. For each optimization iteration, an updated/optimized Starter file

(`<name>_rad_s#_i###_0000.rad`) and Engine file (`<name>_s#_i###_0001.rad`) will be written out, where `###` is the optimization iteration number.

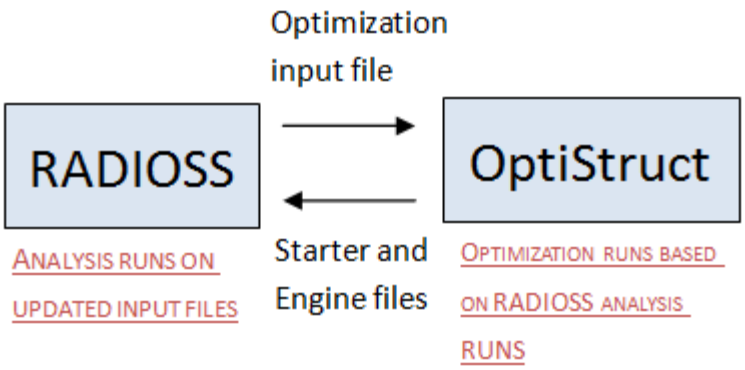


Figure 443: Optimization in Radioss Process Overview

The updated Radioss Starter and Engine files will be solved by Radioss and the results will be communicated to OptiStruct to update the OptiStruct optimization model.

Note: For Solid Topology Optimization using Radioss Optimization, direct density control is used. After Linear static response topology optimization is complete, solid elements having values from 0 to 1 as topology design variables are updated in the finite element model. Each solid design element is directly assigned a filling percentage which has a value equal to the associated topology design variable. This utilizes the Radioss option, `/INIBRI/FILL`, to map the filling percentage with the topology design variable.

Implementation Sequence

The following steps illustrate the sequential process used to implement optimization in Radioss:

1. Pass the optimization file `<name>.radopt` to OptiStruct using the syntax specified in Optimization Process. OptiStruct will automatically search for `<name>_0000.rad` and `<name>_0001.rad` in the same directory.
2. OptiStruct parses the Radioss input files, constructs an equivalent OptiStruct model, and continues with the execution of this OptiStruct model.
3. OptiStruct applies the optimized design variables and writes out the updated/optimized Radioss Starter/Engine files (`<name>_rad_s#_i###_0000.rad` and `<name>_s#_i###_0001.rad`). In the first outer loop, the design variables are not updated yet, and the Starter/Engine files will be identical to the initial user input.
4. Radioss solves the updated Starter/Engine files generated in Step 3.
5. OptiStruct reads the Radioss results obtained in Step 4 and starts the next optimization run (inner loop).
6. If the results from Radioss vary beyond the threshold as compared to the previous outer loop, then the optimization process did not converge.
7. The optimization process is then continued (using the ESLM method) on the updated results from the Radioss analysis and new optimized design variables are generated. Steps 3 to 6 are repeated

to verify if the process converges after the subsequent iteration. If convergence is achieved, go to Step 8.

8. The process ends and optimized results are output.

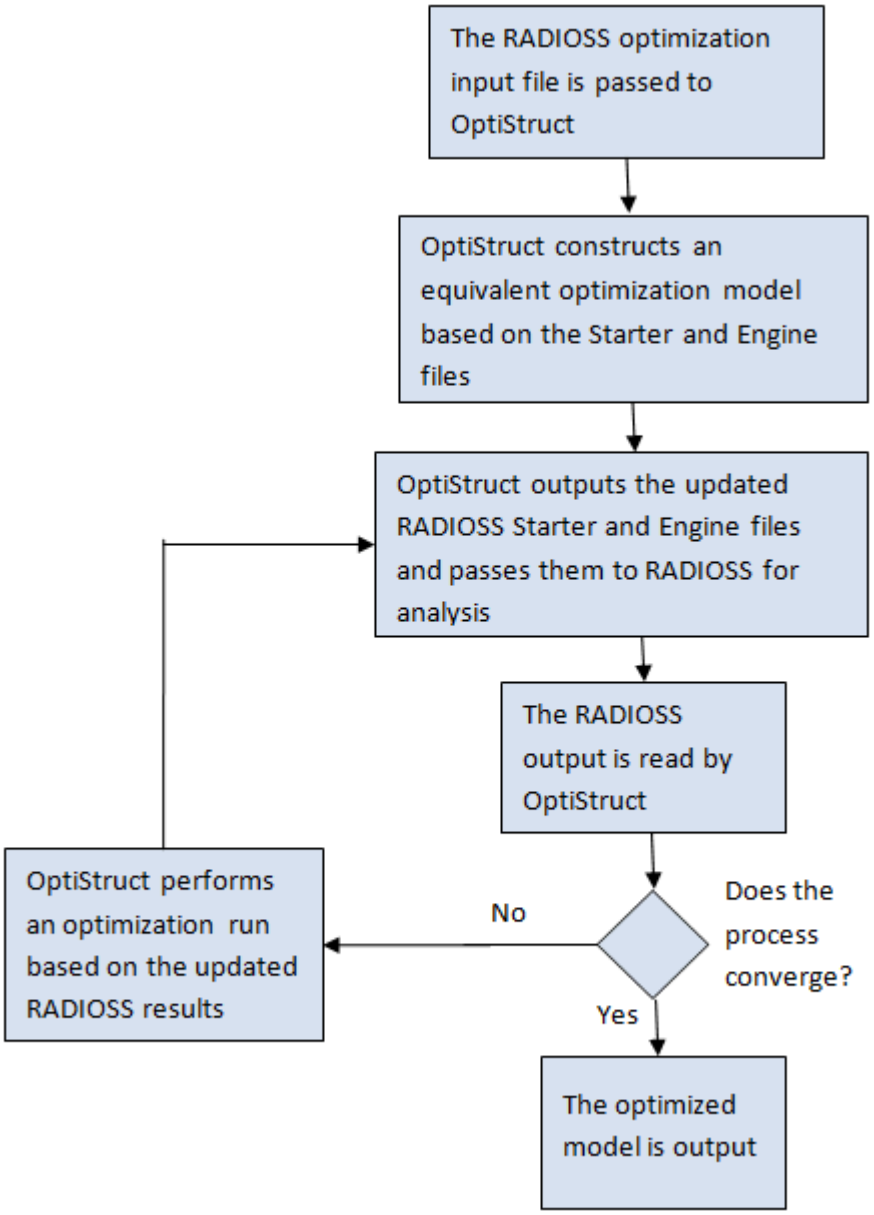


Figure 444: Optimization Sequence in Radioss and OptiStruct

Error Message Database

This section is comprised of error messages in ascending numerical order.

#1 through #299

Error 119

```
ERROR ID :    119
** ERROR IN INTERFACE DEFINITION
DESCRIPTION :
   -- INTERFACE ID: 3
   -- INTERFACE TITLE: contact
   LAGRANGIAN SURFACE IS EMPTY
```

Description:

In 2D analysis (N_{2D3D}=1 or 2 in `/ANALY`), all mesh must be in YZ plane, and must be in Y positive axis and Z positive axis. If mesh is in Y negative axis or Z negative axis, surface is not found for contact, and this error message will be printed.

Suggestions for Resolution:

In pre-processor (HyperMesh or HyperWorks) move all mesh in Y positive axis and Z positive axis.

Error 189

```
MESSAGE ID :           189
** ERROR : BEGIN IS NOT A VALID KEYWORD
MESSAGE ID :           189
** ERROR : MAT IS NOT A VALID KEYWORD
```

Description:

This message usually occurs when the single file input format is used, and no Engine keywords are defined.

Head indicate "`#RADIOSS STARTER`" used for Starter file and "`# RADIOSS`" used for single file input (Starter and Engine in one file). Single file input also need end Engine part with "`/END/ENGINE`".

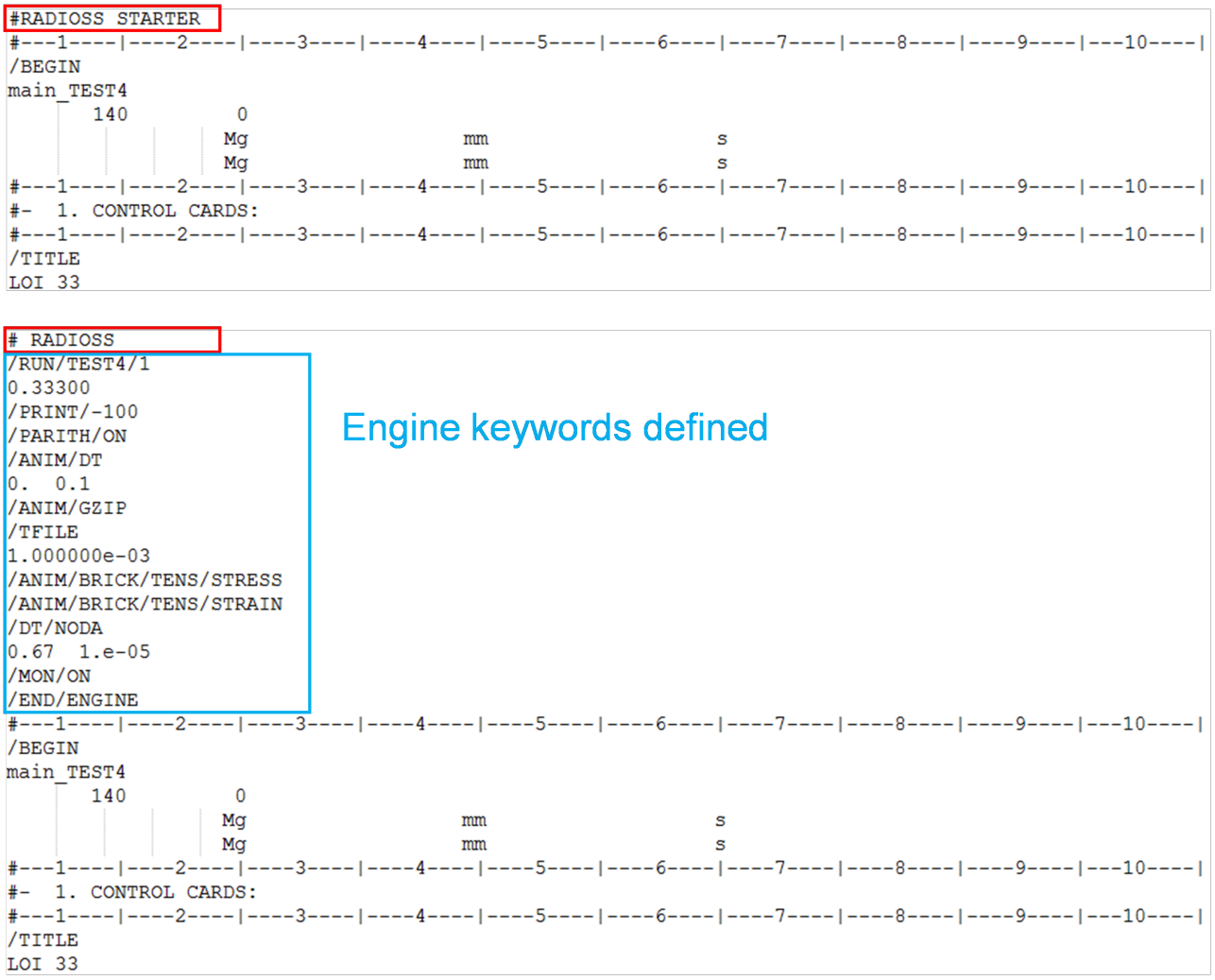


Figure 445:

If head indicates "`# RADIOSS`", but no Engine is defined, then error message ID 189 occurs.

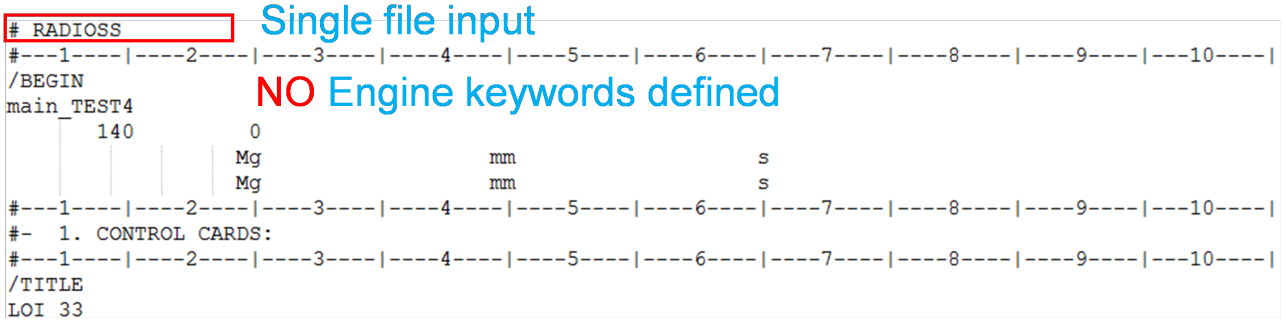


Figure 446:

Suggestion for Resolution:

Either use head "`#RADIOSS STARTER`" or add Engine keywords.

Error 197

```
ERROR ID :           197
** ERROR IN INPUT DATA
DESCRIPTION :
-- PROPERTY ID : 521003304
   -- PROPERTY TITLE : New PROP 521003304
   REFERENCE DIRECTION IS ALMOST NORMAL TO SHELL ID=662193988
```

Description:

The Starter will print this message when the direction defined in the property (reference vector / skew) is almost normal to a shell element.

The solver provides the property ID (in this example, 521003304) and the shell element ID (in this example, 662193988) for which the reference direction is almost normal to the shell element.

The solver projects the vector V (or the Direction X of the skew onto the shell element) along the normal of the shell element (below left figure). If the vector V is normal to the shell element, then the projection of the vector onto the shell element fails (the project of vector V project on shell is one point) and the solver is not able to identify the direction 1 of orthotropy (below right figure) and this error message appear.

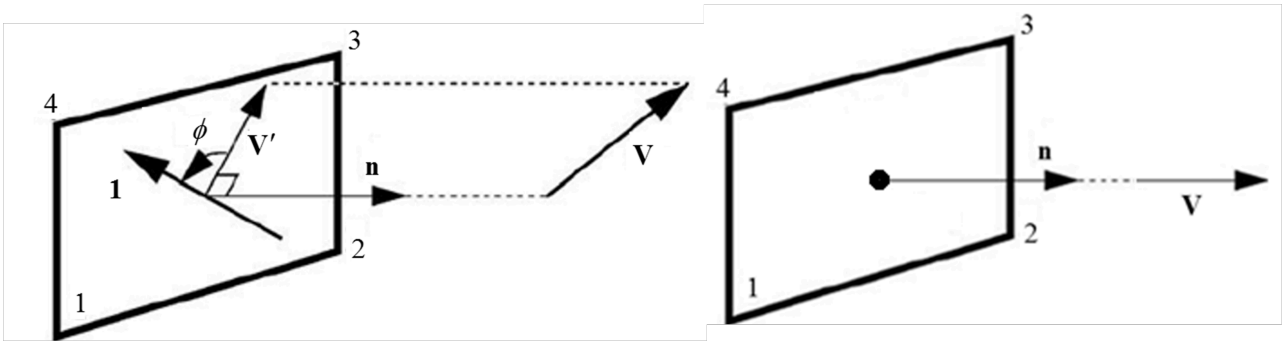


Figure 447:

Suggestion for Resolution:

The direction of the vector in the property reported by the Starter needs to be modified.

> **Note:** Changing the vector direction, changes the direction of orthotropy.

If the material is not really orthotropic (E11=E22 and G12=G23=G31=E11/(2*(1+nu12)), changing the vector will not change the results and an arbitrary vector can be given (VX=VY=VZ=1).

If the material is orthotropic, you need to identify what is the direction of orthotropy and input this direction in the property. If the part has a complex geometry, you may need to split the part in different section and input deck vector.

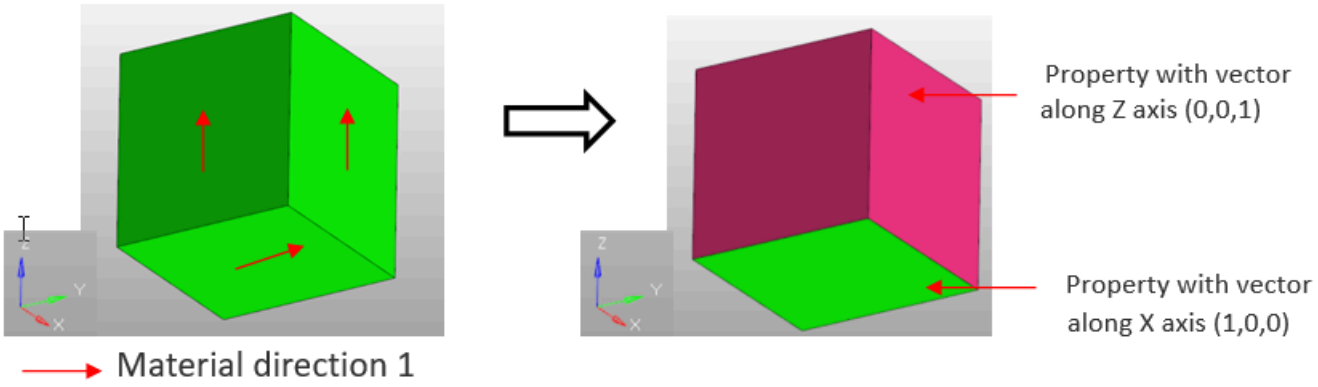


Figure 448:

Error 223

```
ERROR ID: 223
** ERROR IN BEAM DEFINITION

DESCRIPTION :
   IN BEAM ID=1, N1 and N3 HAVE SAME LOCATION

SOLUTION:
   CHANGE N1 OR N3 COORDINATES
```

Description:

Beam elements need an orientation node (N3) to determine the initial position of the local frame. The coordinates of this node (N3) needs to be different than N1 and N2. It also cannot be on the axis defined by N1N2.

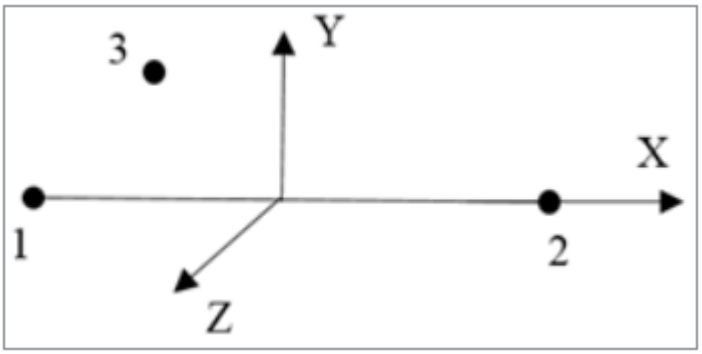


Figure 449:

Suggestion for Resolution:

This issue can be corrected manually:

- If the beam cross-section is circular or square (Iyy=Izz), then the same N3 can be assigned to all the beam elements. The main requirement is to pick a node that is valid for all the selected beams: N3 should not lie on the axis of any of the selected beams.

- If Iyy≠Izz, same the node N3 can still be assigned to multiple beam elements if their local X-axis is similar (defined by N1-N2). If the beam local X-axis is not similar, then pick different N3 for different beam: N3 should not lie on the axis of the beam.

#300 through #699

Error 480

```
ERROR ID :           480
** ERROR IN MATERIAL LAW
DESCRIPTION :
-- MATERIAL ID : 521002880
-- MATERIAL TITLE : Default MAT24 MAT_PIECEWISE_LINEAR_PLASTICITY.2
EPS_T MUST BE STRICTLY LESS THAN EPS_M
```

Description:

In material LAW27 and LAW36, when the failure in tension is defined, there are certain rules that should be followed. In this example the solver is reporting that for material 521002880, $\varepsilon_t > \varepsilon_m$ which is not allowed.

Suggestion for Resolution:

LAW36 Example:

Eps_t < Eps_m < Eps_f

PLAS_TAB

ID	521002880
Title	Default MAT24 MAT_PIECEWISE_LINEAR_PLASTICITY
Local Unit System	None
[RHO_I] Initial density*	7.82E-6
[RHO_0] Reference density	0
[E] Young's modulus*	206.8
[Nu] Poisson's ratio*	.3
[Eps_p_max] Failure plastic strain	.4
[Eps_t] Tensile failure strain	.2
[Eps_m] Tensile failure strain	.24
[C_hard] Hardening coefficient	0
[F_smooth] Strain rate smoothing	☑
[F_cut] Cutoff frequency for strain rate filtering	10
[Eps_f] Maximum tensile failure strain	.28

Figure 450:

LAW27 Example:

EPS_t1 < EPS_m1 < EPS_f1

EPS_t2 < EPS_m2 < EPS_f2

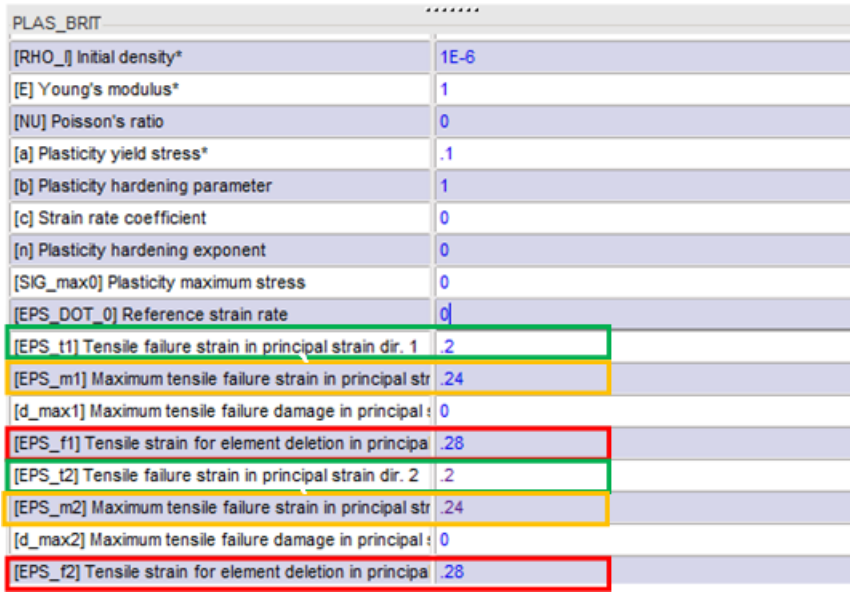


Figure 451:

Error 592

```
ERROR ID :           592
** ERROR IN STARTER INPUT DECK FORMAT
DESCRIPTION :
INPUT VERSION=0
LINE : /VERS/2017
SOLUTION :
/BEGIN CARD IS COMPULSORY FOR STARTER INPUT VERSION >= 50
```

Description:

The solver is expecting a `/BEGIN` card, but instead found a different keyword. Example: `/VERS/2017`

The most common reason for this message to occur are:

- The header indicates separate Starter and Engine files, but the Engine file is present underneath the header.
- `/BEGIN` or `/END` is missing.

Error 611

```
MESSAGE ID :           611
** ERROR IN INITIAL PENETRATION IN INTERFACE
DESCRIPTION :           189
-- INTERFACE ID : 11
-- INTERFACE TITLE : Global interface Gap 0.49 Shells Only
INACTI = 6
IMPOSSIBLE TO CHANGE NODE COORDINATES OF SECONDARY NODE : 823868424
** INITIAL PENETRATION = 0.4900000e+00 IMPOSSIBLE TO CALCULATE NEW COORDINATES OF SECONDARY NODE********
```

Description:

The solver will print this message when it cannot compute the new possible coordinates of a node which is initially penetrating. The solver will provide the following information:

- Node ID for which a new position could not be computed (in this example: `823868424`)

- Interface ID (in this example: `11`)

Suggestions for Resolution:

In HyperCrash or HyperMesh, check the initial penetration for the interface ID for which the error occurs. Sort the initial penetration by minimum residual distance (Res. Dist. Column).

In most cases, the nodes for which the Starter will not be able to compute a new position will be nodes where the residual distance is equal to 0. This means that the nodes are initially on the main segment.

Select these nodes in the list and highlight them in the display (in HyperMesh).

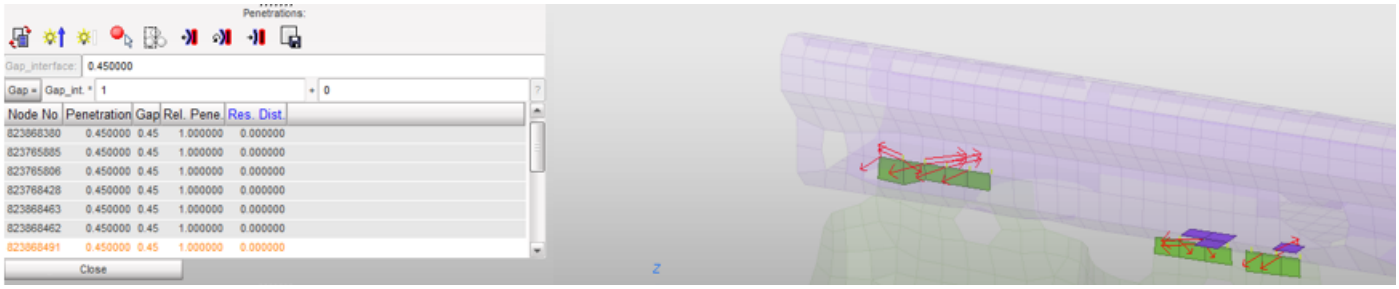


Figure 452:

There are three common modeling errors that can lead to this type of problem.

1. Springs that represent welds or bolts are included in the global contact. They should be removed.

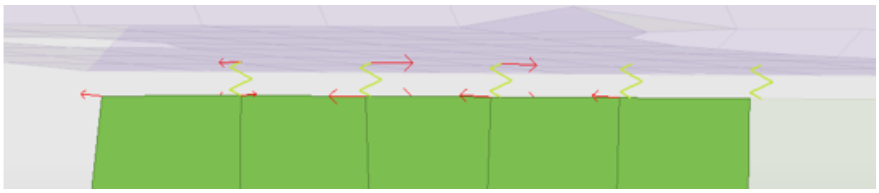


Figure 453:

2. Mesh issue:

   In an area where there are some cracks in the mesh, the nodes should be equivalenced (with HyperMesh).

Figure 454:

   There is a skin on top of the solid mesh and the nodes of the skin/solid are not equivalenced. The nodes should be equivalenced (with HyperMesh).

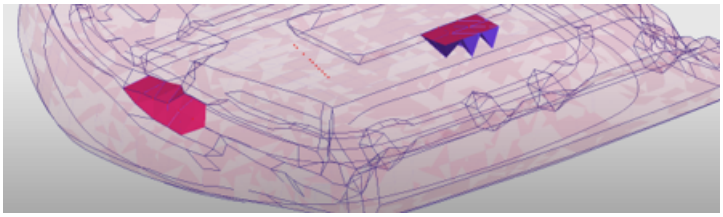


Figure 455:

3. Solid to solid contact with 0 gap. This is not supported with interface TYPE7, TYPE11 or TYPE19. Either move the nodes to create a gap (0.1mm) or use interface TYPE24.

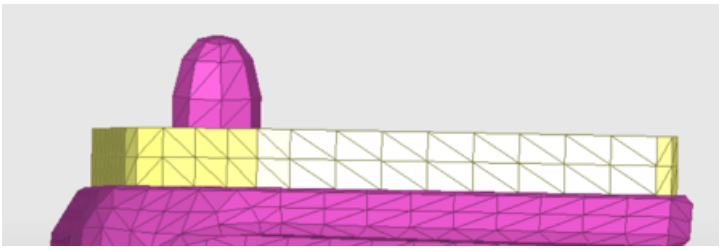


Figure 456:

#700 through #1199

Error 953

```
ERROR ID :           953
** ERROR WHILE READING PARAMETERS
DESCRIPTION :
UNEXPECTED END OF FILE
ERROR ID :           968
** INPUT ERROR
DESCRIPTION :
ERROR WHILE READING STARTER INPUT FILE
```

Description:

This error occurs typically when parameters are present in the input and `/END` is missing at the end of the file.

In most cases, this occurs when the job was submitted before it finished copying into the scratch directory.

Suggestions for Resolution:

Add `/END` at the end of the input file or finish copying all input files.

Error 958

```
ERROR ID :           958
** ERROR DURING PARAMETERS EVALUATION
ERROR ID :           952
** ERROR IN STARTER INPUT DECK FORMAT
DESCRIPTION :
```

```
INPUT VERSION=0
LINE : /PARAMETER/GLOBAL/INTEGER/6620015
```

Description:

This error occurs when the solver cannot evaluate the value of a parameter. This usually happens when an expression is not complete (check the value of the expression in HyperCrash) or there are some parameters referenced in the expression and they are not defined in the model.

Suggestion for Resolution:

The parameter "`Time1`" is equal to "`ActTTF`" + `20`. The parameter "`ActTTF`" is not defined, so the solver cannot evaluate the value of "`Time1`".

```
#RADIOSS STARTER
#-------------------------------------------------
#- RADIOSS DECK WRITTEN BY HyperCrash
#HyperCrash 2017.4 (64 bits)
#Mon Apr 15 17:43:07 2019
#-
#---1----|----2----|----3----|----4----|----5----|
/PARAMETER/GLOBAL/INTEGER/6620015
vent_size_mm
VSize              55
#---1----|----2----|----3----|----4----|----5----|
/PARAMETER/GLOBAL/REAL/6620020
Active Vent leakage
VALeak                        .8
#---1----|----2----|----3----|----4----|----5----|
/PARAMETER/GLOBAL/REAL/6620021
Seam leakage
SLeak                      .0175
#---1----|----2----|----3----|----4----|----5----|
/PARAMETER/GLOBAL/REAL_EXPR/6620022
Active vent leakage curve
Time1     ActTTF+20
#---1----|----2----|----3----|----4----|----5----|
```

Figure 457: Example

Error 1078

```
ERROR ID :   1078
** ERROR: TIED INTERFACE
DESCRIPTION :
   -- INTERFACE ID: 1
   -- INTERFACE TITLE: new Contact
   1 SECONDARY NODE(S) ARE DEACTIVATED FROM INTERFACE
     SECONDARY NODE:  82              NO MAIN SEGMENT FOUND WITHIN  1.0000000000000E-02
```

Description:

The solver will print this message when a node is included in an interface TYPE2 and no main segment is found within the search distance for this node and flag "*Ignore*" is set to 1000.

The solver will provide the interface ID (in this example `1`) and the node ID (in this example `82`) for which the solver did not find a main segment within the search distance (1.0e-2 in this example).

The Model Checker of HyperCrash detects this type of error (Secondary nodes no proj.).

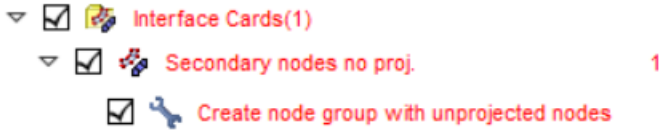


Figure 458:

Suggestions for Resolution:

There are several ways to resolve this issue:

- Use *Ignore* = 1, if those nodes do not need to be tied or remove these nodes from the interface TYPE2.
- Translate the nodes that do not find a projection so that they fall within the search distance.
- Increase the search distance slightly (D_{search}).

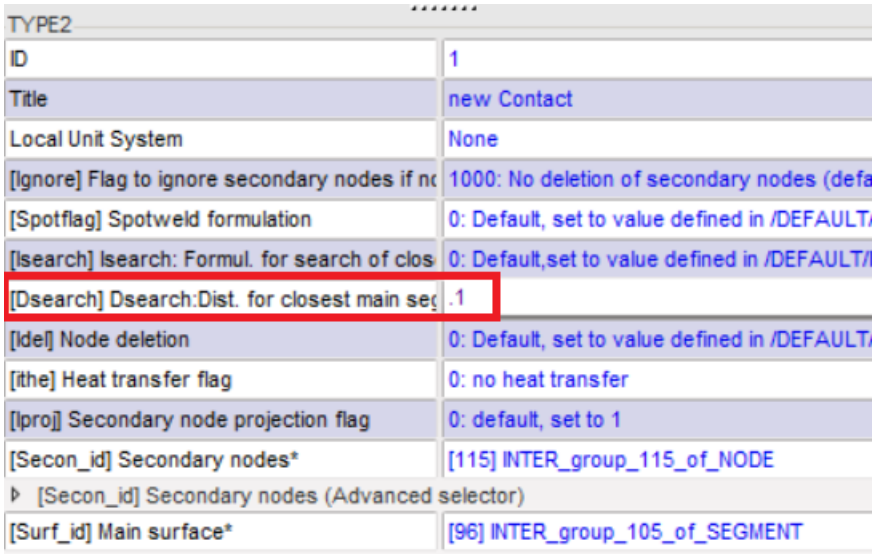


Figure 459:

Error 1104

```
ERROR ID :           1104
** ERROR WHILE BUILDING SURFACE FROM SOLID PARTS
DESCRIPTION :
-- SURFACE ID : 830005865
-- SURFACE TITLE : New SURF 830005865
UNAVALABLE OPTION /SURF/PART/Surf_ID FOR CREATING SURFACE
SOLUTION :
FOR A PART OF SOLIDS
USE /SURF/PART/EXT/Surf_ID or /SURF/PART/ALL/Surf_ID
```

Description:

The solver will print this message when a PART with solid elements is included in the a `/SURF/PART`. This keyword allows to build surfaces only from 2D elements and not 3D.

Suggestions for Resolution:

1. Run the check named **Volumic Parts in Surf** in the HyperCrash Model Checker.

2. Select either **Switch to Surf Ext** or **Switch to Surf All** to automatically switch the keyword to `/SURF/PART/EXT` or `/SURF/PART/ALL`.

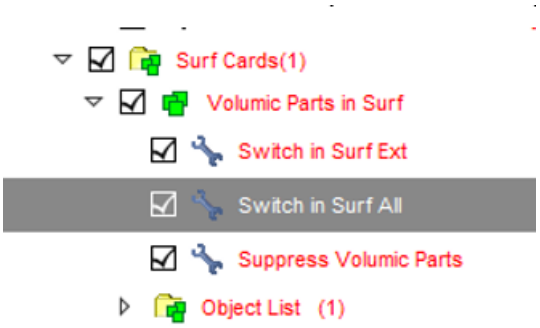


Figure 460:

Error 1130

```
ERROR ID :           1130
** ERROR IN USER INTERFACE
DESCRIPTION :
OPTION : /MAT/USER1
USER INTERFACE OPTION USED, BUT NO USER INTERFACE ROUTINE IS FOUND
```

Description:

The Starter will print this message when it reads a user material (LAW29) and the user library defining the behavior and format of this material is not defined.

Suggestions for Resolution:

Replace user material with any other compatible and supported Radioss material keyword.

Engine Error Messages

MASS OR ENERGY ERROR DUE TO CONTACT

```
**WARNING MINIMUM TIME STEP    0.248E-04 IN INTERFACE       11(DTMIN=  0.2500E-04)
REMOVED SECONDARY NODE 63086879 FROM INTERFACE

MESSAGE ID :         206
**  RUN STOPPED : ENERGY ERROR LIMIT REACHED

***  NODE WITH HIGHEST MASS CHANGE
```

Description:

In several cases, the simulation can fail with ERROR termination because too much mass is added to maintain a constant time step for the contact.

Why does mass need to be added to maintain stability of the contact?

- Time step is constant when `/DT/NODA/CST`, `/DT/NODA/CST2` or `/DT/INTER/CST` are used.
- Interface time step is computed as $\Delta t = \sqrt{\frac{2m}{K}}$

  Stiffness increases in function of the penetration $K = K_0 \frac{Gap^2}{(Gap - p)^2}$.

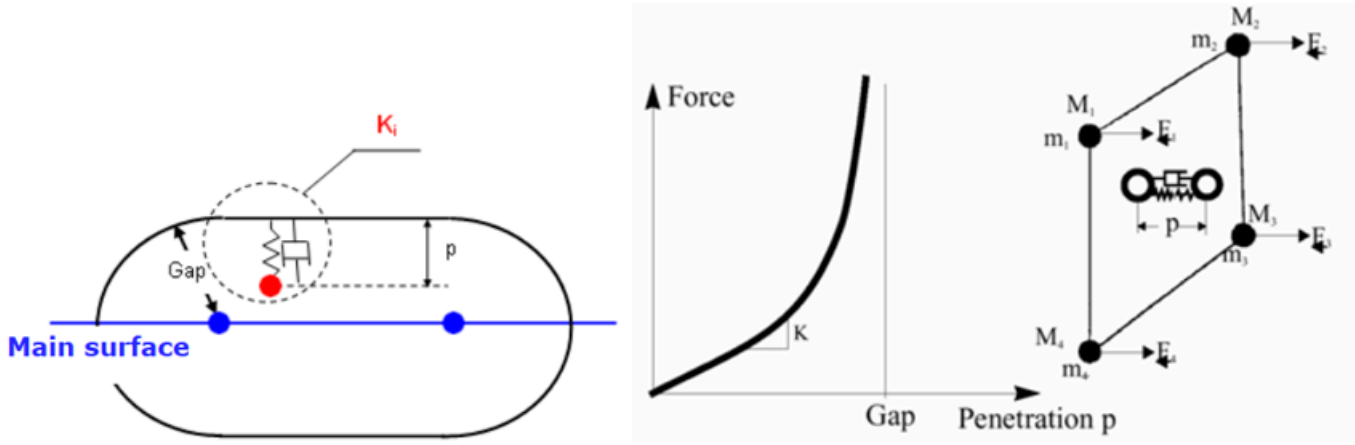


Figure 461:

As the penetration in the contact increases, stiffness increase leads mass to be increased to maintain time step constant.

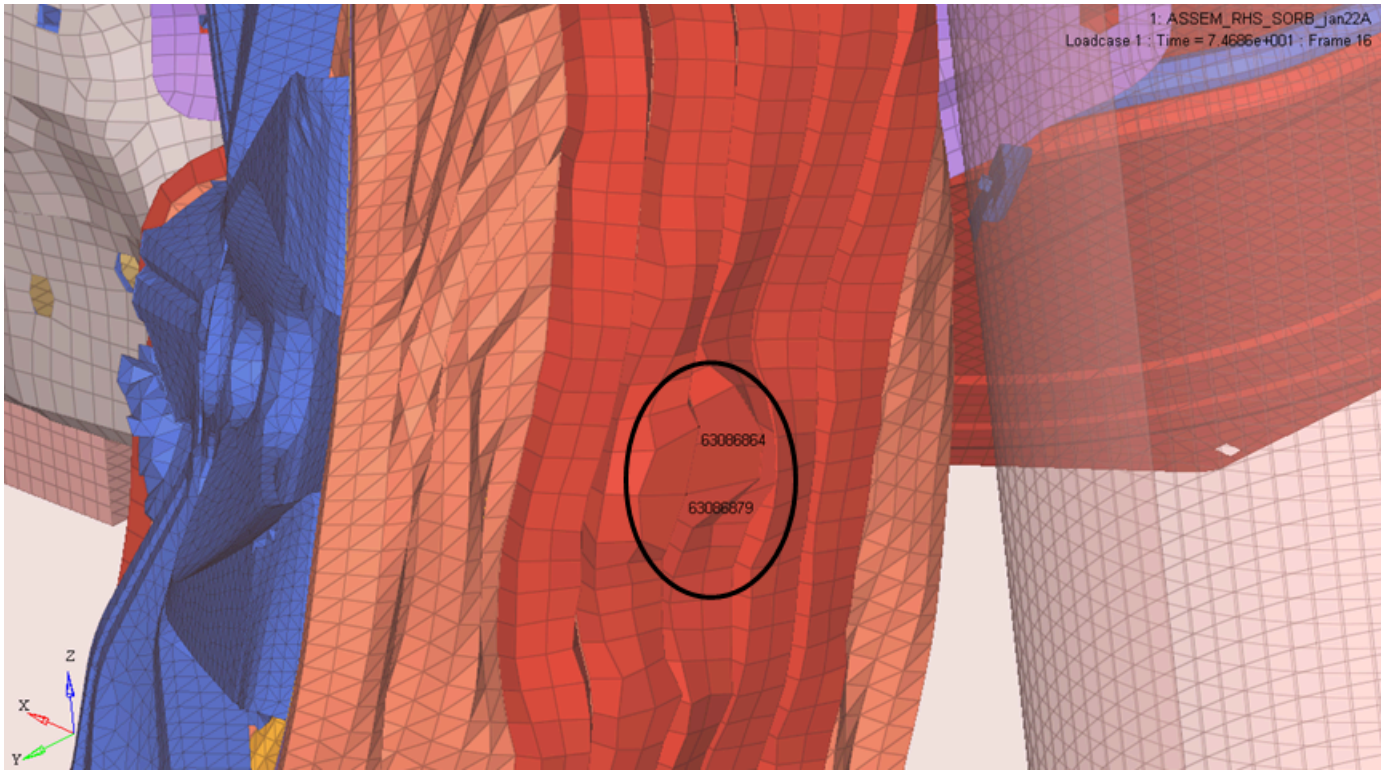


Figure 462:

```
 **WARNING MINIMUM TIME STEP     0.2484e-04     IN  INTERFACE          11(DTMIN=  0.2500e-04)
    REMOVE SECONDARY NODE    63086864  FROM INTERFACE          11
 **WARNING MINIMUM TIME STEP     0.2484e-04     IN  INTERFACE          11(DTMIN=  0.2500e-04)
    REMOVE SECONDARY NODE    63086879  FROM INTERFACE          11

MESSAGE ID :          206

 ** RUN STOPPED : ENERGY ERROR LIMIT REACHED

189137   74.69      0.1943E-04 NODE    63109533  99.9%  0.2295E+06  0.4643E+10   784.8      -4904.      0.1574
     ANIMATION FILE: ASSEM_RHS_SORB_jan22A016 WRITTEN

     NODE    VELOCITY     K-ENER       MASS       MASS0    DM/MASS0
 63086864  7941.      0.1205E+05 0.3821E-03 0.3821E-03  0.000
 63086879  4845.       283.9     0.2418E-04 0.2418E-04  0.000
 63109654  2449.       73.93     0.2466E-04 0.2466E-04  0.000
 63110454  709.5       3.127     0.1242E-04 0.1242E-04  0.000
 15063500  238.9      0.7197     0.2522E-04 0.2522E-04  0.000
 15228459  200.6      0.2494     0.1239E-04 0.1239E-04  0.000
 50559051  199.7      0.4943     0.2478E-04 0.2478E-04  0.000
 15050554  198.2       0.000      0.000      0.000      0.000
 63109653  194.2       0.000      0.000      0.000      0.000
 15063617  190.0      0.4293     0.2379E-04 0.2379E-04  0.000

***  NODES WITH HIGHEST MASS CHANGE

     NODE        MASS       MASS0    DM/MASS0
 63086864 0.3174E-03 0.3425E-04 0.2048E+07
 63086879 0.5536E-03 0.6182E-04 0.1597E+07
 63109654 0.1952E-03 0.2815E-04 0.1192E+06
 63109653 0.4837E-03 0.7405E-04 0.8804E+05
```

Figure 463:

For example, the tire stack up against the rigid barrier. Behind the tire, the brake disk (or caliber) compresses the tire against the barrier. The global contact is not stiff enough and leads to large penetrations.

- Stiffness increased
- Mass is added to maintain time step constant

Suggestions for Resolution:

A typical solution for this problem is to create a local contact with a higher gap and stiffness. Increasing the initial stiffness leads to a smaller time step at the beginning of the penetration, but it will increase the time step, if the penetration is large.

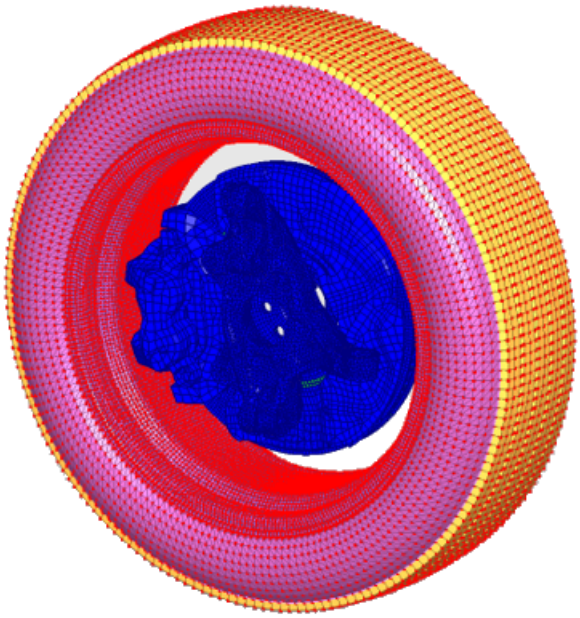


Figure 464: Local contact

Contact between two rigid parts is possible with interface TYPE7 and TYPE11 or TYPE19. The main issue is that it will generate deep penetrations in the contact, which can lead to mass being added to maintain the timestep to the value specified in `/DT/NODA/CST` or `/DT/NODA/CST2`.

There are two solution to resolve this type of issue:

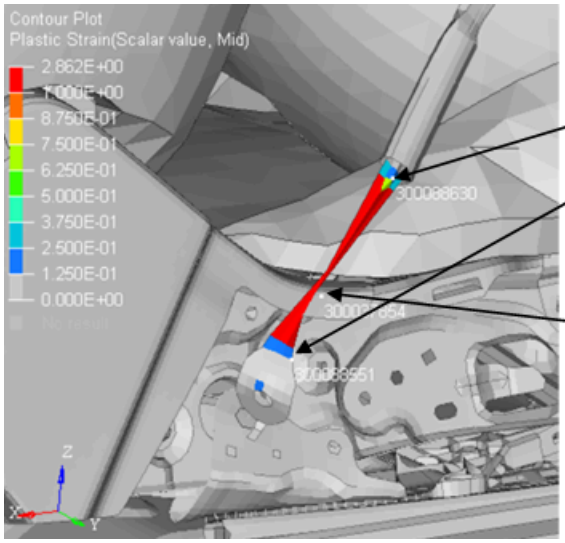
- Switch at least one of the parts to deformable
- Add a local contact with a higher gap and stiffness

MASS OR ENERGY ERROR DUE TO HIGH DISTORTION

```
***  NODES WITH HIGHEST VELOCITY
***  NODES WITH HIGHEST MASS CHANGE
```

Description:

When there is very large plastic deformation, the simulation can become unstable and fail with ERROR termination, due to energy error or mass error.



*** NODES WITH HIGHEST VELOCITY

NODE	VELOCITY	K-ENER	MASS	MASS0	DM/MASS0
300088630	1341.	95.80	0.1066E-03	0.1066E-03	0.000
300088551	1239.	81.06	0.1056E-03	0.1056E-03	0.000
300088555	588.5	44.96	0.2596E-03	0.2596E-03	0.000
300088568	579.4	89.71	0.5344E-03	0.5344E-03	0.000
300088587	568.0	36.55	0.2266E-03	0.2266E-03	0.000
300088602	548.4	44.46	0.2957E-03	0.2957E-03	0.000
300031715	538.2	17.87	0.1234E-03	0.1234E-03	0.000
300088583	515.3	37.71	0.2840E-03	0.2840E-03	0.000
300088560	480.7	26.12	0.2261E-03	0.2261E-03	0.000
300088565	479.1	29.79	0.2596E-03	0.2596E-03	0.000

*** NODES WITH HIGHEST MASS CHANGE

NODE	MASS	MASS0	DM/MASS0
300037854	51.00	0.9468E-04	0.1238E+34
300028235	1.273	0.3351E-05	0.7483E+32
300037853	73.01	0.6557E-03	0.2181E+32
300028969	50.95	0.5672E-03	0.1467E+32
300061210	8.088	0.9314E-04	0.8621E+21
300033776	3.702	0.5025E-04	0.3689E+21
300060334	0.6554	0.9048E-05	0.1401E+21
300060335	0.6554	0.9283E-05	0.6196E+20
300033785	68.23	0.1184E-02	0.5304E+20
300032669	56.07	0.1003E-02	0.3847E+20

Figure 465:

Suggestion for Resolution:

Based on the results, there are different ways to resolve this issue:

- Define proper strain failure
- Material may be incorrect (unit system) or type (plastic instead of steel, yield point is too low…)

62100 18.85 0.2985E-03 SOLID 230440718 -1.3% 0.3484E+5 0.1317E+06 109.7 1523.
RUPTURE OF SOLID ELEMENT NUMBER 230025[illegible]98
DELETE CONNECTION SOLID ELEMENT 110010956, ENERGY= 4.732011155E-05
RUPTURE OF SOLID ELEMENT NUMBER 230024559
** WARNING MINIMUM TIME STEP 0.5975E-05 IN INTERFACE NB 2300040(DTMIN= 0.1000E-04)
SECONARY NODES NB 230373002 230373019
MAIN NODES NB 230373003 230373004
DELET SECONDARY LINE FROM INTERFACE

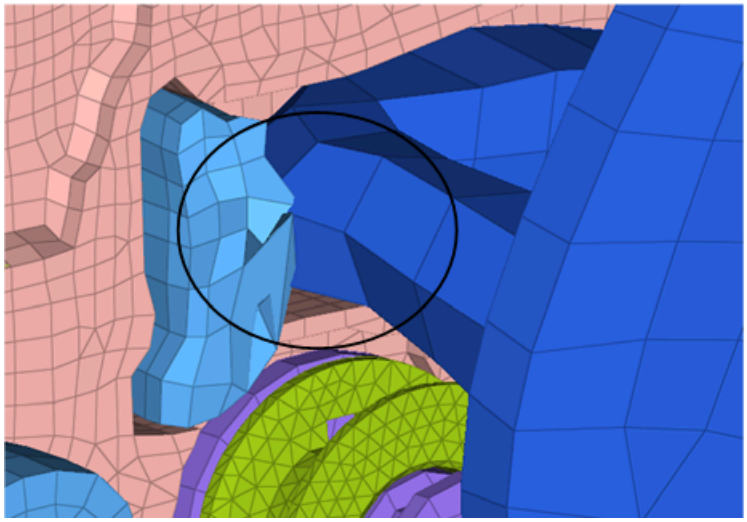


Figure 466:

Initially, there is an intersection that generates the local distortion. To resolve this issue, correct the intersection and define proper strain failure in the material.

RIGID BODY BECOMES UNSTABLE

```
WARNING ID :          542
** WARNING RIGID BODY INERTIA
DESCRIPTION :
-- RIGID BODY ID : 713264
-- RIGID BODY TITLE : NULL
UNPHYSICAL INERTIA :
PRINCIPAL INERTIA 150.0050404753 IS CREATER THAN THE SUM
OF THE 2 OTHER ONES 0.60340391378794E-02 + 0.3645995262784E-02
```

Description:

When the rigid body inertia is not physical, the rigid body can become unstable. When the rigid body inertia is not physical, Radioss will print the warning in the Starter.

If those warnings are not corrected, the following can occur:

- The rigid body becomes unstable and is propelled at a very high speed in the vehicle
- The rigid secondary nodes are still in the global contact and will generate unrealistic deformation
- Numerical issues as the impact velocity is very high (>1500 m/s)

Suggestion for Resoltuion:

Correct the unphysical rigid body inertia.

NEGATIVE INERTIA OR STIFFNESS

```
NEGATIBE INERTIA ON NODE ID=823767293
```

Description:

This issue typically occurs more often in Radioss v12.x than more recent versions (2017.x). It occurs when the node of an interface TYPE2 projects outside of the closest main segment. The error is not related to the height of the secondary node relative to the main segment, but that its projection is outside of the main segment.

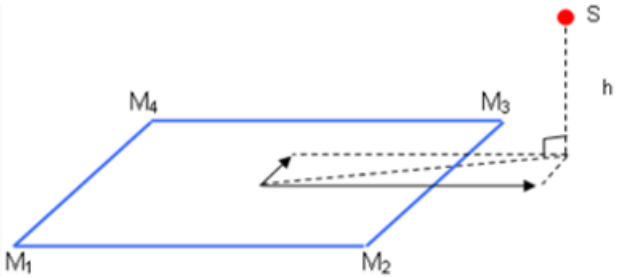


Figure 467:

A warning in the Starter is printed for the nodes that project outside of the main segment.

```
WARNING ID:          345
** WARNING : CHECK TIED INTERFACE
DESCRIPTION :
-- INTERFACE ID : 557392358
-- INTERFACE TITLE : CONNECTION.37
```

```
8238684 2 823767497 823767293 823767296 823767504 -1.279664112565 -0.9934883789504 0.7066172931858
```

Use HyperCrash Model Checker to check “Secondary nodes no proj.” which allows to identify the nodes that project out of the closest main segment.

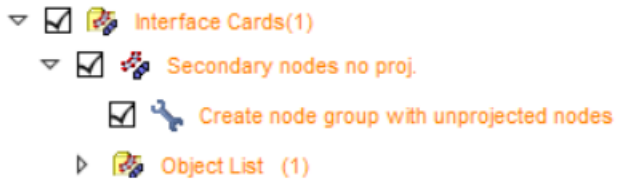


Figure 468:

Suggestion for Resolution:

There are several ways to resolve this issue:

- Remove the nodes that project outside the closest main segment from the interface TYPE2
- Translate the nodes so that they do not project outside the segment anymore

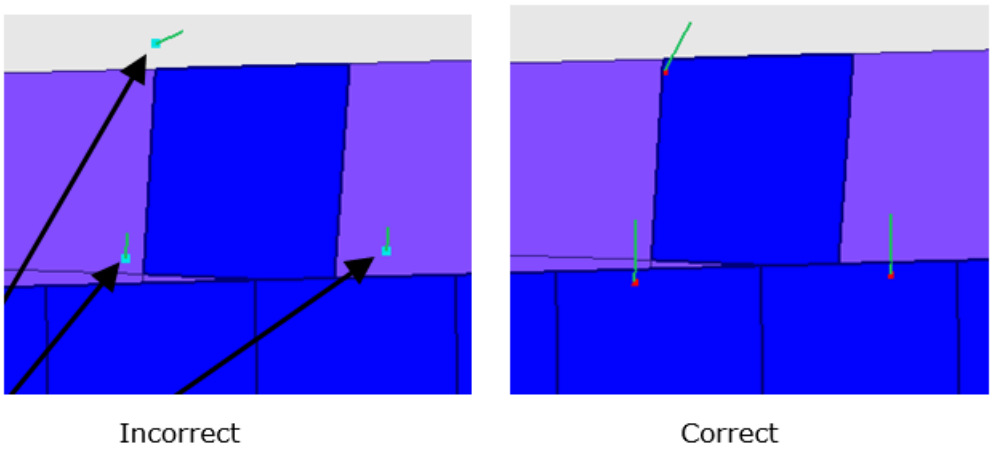


Figure 469:

Warning Message Database

This section is comprised of warning messages in ascending numerical order.

#100 through #699

Warning 194

```
WARNING ID :           194
** WARNING IN HIERARCHY REFERENCE
DESCRIPTION :
-- SURFACE ID : 8300005857
-- INTER_group_830006186_of_PART_TITLE : PART
REFERENCE TO UNEXISTING ID=600000000
```

Description:

The Starter will print this message when an object (node, element, part, subset) is defined in a group of node, element, surface or line but does not exist in the model.

In the example above, the part ID `600000000` is defined in the surface `830005857`. This part does not exist in the model. The objects that are referenced but not defined in the model will be shown in HyperCrash in the Model Browser as “Unresolved IDs”.

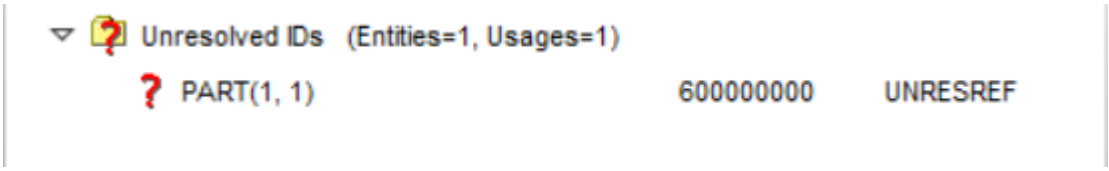


Figure 470:

Suggestions for Resolution:

Either remove the group or select object to redefine the group.

Warning 542

```
WARNING ID :           542
** WARNING RIGID BODY INERTIA
DESCRIPTION :
-- RIGID BODY ID : 136849917
-- RIGID BODY TITLE : MAT_RIGID_1500016_PART_325
UNPHYSICAL INERTIA :
PRINCIPAL INERTIA 249.2162969709 IS GREATER THAN THEN SUM
OF THE 2 OTHER ONES 112.70695453317 + 48.42692228428
```

Description:

The Starter will print this message when the inertia is not physical. In most cases this occurs when:

1. Inertia have been defined but incorrectly in the case of *Icog*=3 and 4.
2. Constant inertia has been defined (50) for Jxx, Jyy, Jzz, Jxy, Jxz and Jyz and *Isphere* is not set to 1.

This warning must be corrected as it could generate problems during the Engine. This can be corrected in HyperCrash.

Suggestion for Resolution:

For the above case 2, a simple way to resolve is with HyperCrash. Inertia column is displayed in the rigid body panel, select all the rigid with Jxx, Jyy, Jzz =50 and set Jxy, Jyz, Jxz = 0.

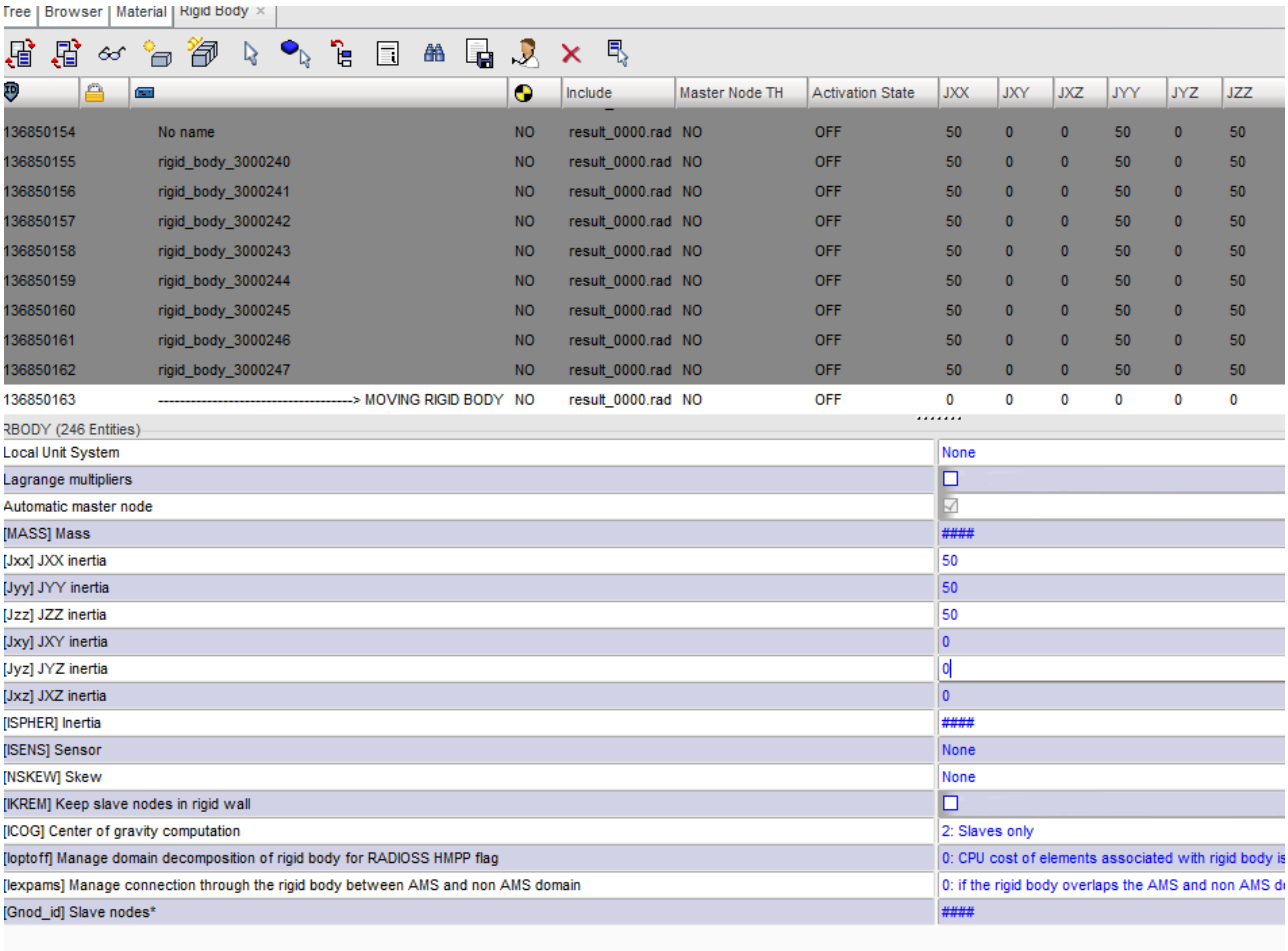


Figure 471:

#700 through #1199

Error 1079

```
WARNING ID :   1079
** WARNING: CHECK TIED INTERFACE
DESCRIPTION :
   -- INTERFACE ID: 1
   -- INTERFACE TITLE: new Contact
   CONCERNING 1 NODE(S)
    SECONDARY NODE:  82               NEAREST SEGMENT :   1                  MAIN NODES:   146        145
        142        143
                S = 1.400000000000  T = 1.160000000000   DIST =  2.9393876913398E-02
```

Description:

This message means that the projection of secondary node ID `82` is outside the main segment, but Radioss treats the projection as moving this secondary node onto the nearest edge of the segment. `S` and `T` are the distance of local 's' and 't' direction.

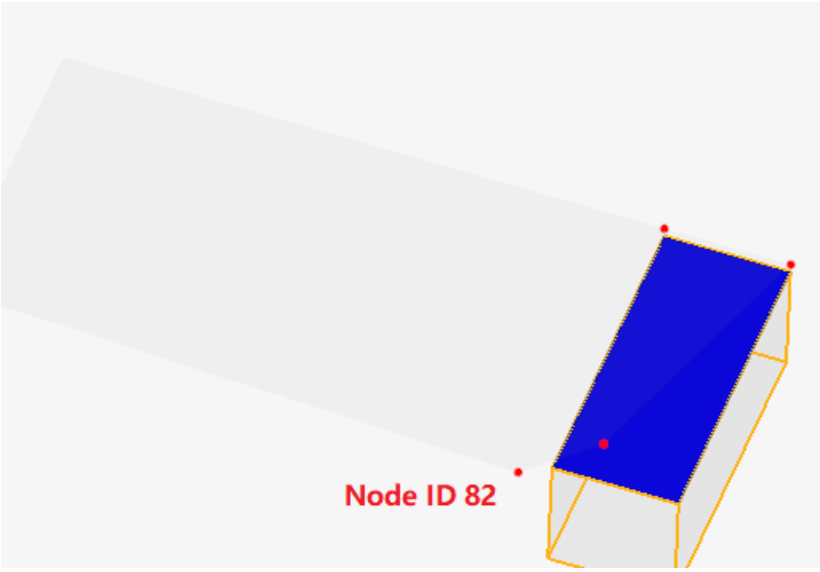


Figure 472:

Index

Special Characters

-check argument (Starter) *21*
-dylib [FILE] argument (Starter) *22*, *27*
-help argument
  Engine *23*
  Starter *19*
-HSTP_READ argument (Starter) *22*
-HSTP_WRITE argument (Starter) *22*
-mds libpath [PATH] argument (Starter) *22*, *27*
-norst argument (Engine) *26*
-notrap argument
  Engine *26*
  Starter *20*
-nspmd [spmd domain number] argument (Starter) *20*
-outfile argument
  Engine *26*
  Starter *21*
-rseed (seed] argument (Starter) *22*
-rxalea (xalea] argument (Starter) *22*
-version argument
  Engine *24*
  Starter *19*

A

ABF files *542*
advanced mass scaling
  .out file examples *70*
  automotive example *59*
  current capabilities and limitations *53*
  metal forming example *70*
  recommended checklist *56*
advanced techniques *32*
airbag model requirements *389*
airbag modeling
  debug deployment *429*
  element time step control *388*
  FVMBAG1 guidelines *389*
  monitored volume *379*
  setup recommendation *426*
  tank test *422*
airbag, fabric material *393*
airbag, finite volume method *399*
airbag, initial air inside *393*
ALE formulation *491*

ALE grid calculation *499*
ALE interfaces *508*
ALE keywords, summary *487*
ALE/CFD materials *501*
animation files *542*
appendix *453*
application field *389*
arbitrary lagrangian-eulerian (ALE) *487*
artificial damping for shock waves *507*
auto-contact gap warning *368*
automatic creation of subdomain *538*

B

bad volume, elminate *421*
balance energy *445*
balance mass *445*
balance momentum *445*
basic relations *453*
bolt modeling for crash *323*
bolt tensioning *374*

C

check units *442*
checkpoint file *35*
classic beam elements (/PROP/BEAM) *93*
command line argument
  Engine *22*
  Starter *18*
common problems, interface TYPE3 and TYPE5 *350*
communication between AcuSolve and Radioss *465*, *479*
communication between domains - single input file format *538*
communication between Radioss and AcuSolve *470*
composite
  failure model *294*
  material *265*
  modeling *279*
  properties *280*
composite material
  LAW12 and LAW14 *266*
  LAW19 and LAW58 *277*
  LAW25 (Tsai-WU and CRASURV) *272*
composite properties
  shell elements *280*
  solid element *292*
computational fluid dynamics (CFD) *487*
computer cluster *29*

concrete materials
/MAT/LAW10 and /MAT/LAW21 *212*
/MAT/LAW24 *216*
/MAT/LAW81 *232*
connection between domains, multiple input file format *546*
contact between domains, multiple input file format *548*
contact between Sol2SPH and other parts *531*
contact interface time step
control *49*
contact interface, implicit structural finite element analysis *459*
contact treatment Lagrange multiplier method *345*
contact treatment penalty method *345*
control file (C-File) *32*
coordinate system
material system *154*
CPU allocation
multiple input file format *550*
single input file format *543*
current capabilities and limitations *53*
current limitations, single input file format *545*
cylindrical joint *336*

D

dashpot - stiffness formulation *116*
data input
multiple input file format *549*
single input file format *541*
dc-fsi using Radioss and AcuSolve *466*
DC-FSI using Radioss and AcuSolve *465*, *479*
debugging guidelines
computation checks *445*
debug model *447*
model checking *444*
model instrumentation *443*
model preparation *433*
unit consistency *438*
design optimization
implementation *555*
process *557*
run optimization *556*
set environment variables *556*
different number of cores *30*
differential gear joint *339*
diffuse necking, material instability *261*
Drücker condition stability check *187*

ductile failure model
/FAIL/BIQUAD *239*
/FAIL/JOHNSON *237*

E

edge to edge impact locking *367*
elastic plastic piecewise linear material - /MAT/LAW36 *206*
elasto-plastic materials *202*
elasto-plastic materials - Johnson-Cook (/MAT/LAW2) *202*
element failure treatment *259*
element formulations, implicit structural finite element analysis *458*
element option
guidelines *91*
element time step
control *47*
elementary boundary conditions *505*
elements solid *82*
elements solid-shell *84*
energy balance *445*
engine file, implicit analysis examples *485*
Engine input
User Guide *515*
error message database *560*, *577*
Euler formulation *491*
exit codes (Starter and Engine) *17*
explicit structural finite element analysis *37*

F

failure models for ductile materials *236*
failure, explicit structural finite element analysis *236*
-input [file] argument
Engine *24*
Starter *19*
filtering, appendix *455*
finite elements *82*
first checks during/after computation *445*
fluid and fluid-structure simulation *487*
fluid-structure interaction analysis with AcuSolve and Radioss *465*, *479*
fluid/structure connection *493*
fluid/structure interaction, ALE/Lagrange interface *512*
force and moment - spring stiffness *96*
force and moment computation, modeling tools *164*
friction *360*
FVM airbag, standalone run *418*
FVMBAG1 airbag
checklist *421*

fabric porosity *406*
housing *411*
internal contacts *409*
stability run *416*

G

gas generator *391*
general recommendations - time step *79*
generation of tangential force *367*
global time step *39*

H

heat exchange *364*
HILL materials - User Guide *208*
hourglass formulations *147*
hyperelastic materials
/MAT/LAW42 *178*
/MAT/LAW62 (VISC_HYP) *182*
/MAT/LAW69 *184*
/MAT/LAW82 *186*
/MAT/LAW88 *189*
/MAT/LAW92 (Arruda-Boyce) *192*
/MAT/LAW92 (Yeoh) *195*
Bergstrom-Boyce *196*
conclusion *191*
Ogden *177*

I

implementation, design optimization *555*
implicit analysis activation *461*
implicit finite element analysis appendix *485*
implicit finite element analysis output messages *481*
implicit finite element analysis troubleshooting *483*
implicit solvers w/parallel version, compatibility *463*
implicit structural finite element analysis *457*
incompatible kinematic conditions *342*
inconsistent stiffness, springs *117*
increase computation speed and maintain accuracy *50*
increase in mass *365*
initial penetrations *355*
interface drawbead (/INTER/TYPE8) *369*
interface edge-to-edge (/INTER/TYPE11) *371*
interface general purpose (/INTER/TYPE7) *353*
interface non-symmetric (/INTER/TYPE5) *348*
interface stiffness *358*

interface symmetric (/INTER/TYPE3) *347*
interface time step control *365*
interface TYPE6 (/INTER/TYPE6) *352*
interfaces explicit finite element *344*
interfaces, ALE *508*

J

Jacobian matrix *144*

K

kinematic conditions penalty method *341*
kinematic conditions, implicit structural finite element analysis *459*
kinematic constraints *326*, *507*

L

Lagrangian formulation *491*
lauch multi-domain analysis
    multi-domain input file format *550*
launch multi-domain analysis
    single input file format *543*
LAW12 and LAW14, composite material *266*
linear solvers *463*
loads, implicit features and compatibility *459*
local system of cutting plane, modeling tools *163*
lode angle, failure *251*

M

mass balance *445*
material compatibility, implicit structure finite element analysis *458*
material instability, diffuse necking *261*
materials, explicit structural fea *172*
mesh recommendations *144*
meshing and folding *389*
methodology
    CAD cleaning *494*
methods to apply kinematic conditions *341*
model setup
    multi-domain *535*
    multiple input file format *546*
    single input file format *537*
model size, multiple cores *30*
modeling guidelines, Sol2SPH *531*
modeling tools, skew and frame *159*
momentum balance *445*

monitored volume
AIRBAG1 type *380*
COMMU1 type *384*
FVMBAG1 type *387*
GAS type *387*
LFLUID type *388*
PRES type *388*
time step *388*
most popular units *442*
mpi and Radioss options coherency *26*
multi-domain
model setup *535*
multi-domain technique *533*
multi-point constraints (/MPC) *339*
multiple cores *29*
multiple engine files *34*
multiple input file format, Radioss *546*

N

new beam elements (/PROP/INT_BEAM) *93*
nodal boundary conditions *504*
nodal time step *38*
nodal time step control *44*
nonlinear analysis with contact *476*
nonlinear implicit analysis notes *476*
nonlinear solvers *464*
nspmd [spmd domain number] argument (Starter) *20*

O

other modeling guidelines, Sol2SPH *531*
output data, single input file format *542*

P

preload
bolt tensioning *374*
define *375*
pulley spring TYPE12 *129*

Q

quadratic surface contact *372*
quasi-compressible option *503*
quasi-static analysis notes *475*
quasi-static simulations - time step *81*

R

rack and pinion joint *339*
RAD2RAD input file example *553*
RAD2RAD output file *542*
RAD2RAD parallelization *552*
reference geometry *390*
rigid body, /RBODY *326*
rigid link, /RLINK *338*
rigid wall, /RWALL *330*
rotational gear type joint *339*
run executables directly *16*
run options
    Altair Simulation script *16*
    Compute Console *11*
    HyperMesh *16*
run Radioss *11*
rupture interface *369*

S

scaling failure strain *256*
section cutting plane, modeling tools *161*
section output, modeling tools *170*
sections (/SECT) *160*
select main and secondary sides, fluid/fluid interface *510*
shell elements
    /PROP/SHELL *86*
single input file format *537*
single precision computation *35*
skew and frame *159*
smooth particle hydrodynamics (SPH)
    cell distribution *518*
    maximum stored number of neighbors *522*
    symmetry conditions *523*
-nthread [smp thread number] argument
    Engine *25*
    Starter *20*
soft part against hard part *366*
Sol2SPH particle behavior *529*
solid elements, /PROP/SOLID *82*
solid to SPH option (Sol2SPH) *528*
solid-shell elements (/PROP/TSHELL) *84*
solution process *27*
space integration *499*
SPEEDUP estimation *543*
sph, maximum stored number of neighbors *522*

spotweld
bolt or adhesive *309*
solid *313*
spring
/PROP/KJOINT *135*
/PROP/KJOINT2 *138*
/PROP/SPR_AXI *144*
/PROP/SPR_BEAM *131*
/PROP/SPR_GENE *126*
/PROP/SPR_PUL *129*
/PROP/SPRING *124*
coordinate system *118*
failure *99*
hardening *106*
stiffness formulation *96*
spring type 25 *144*
spring type 8 *126*
spring type13 *131*
springs, inconsistent stiffness *117*
stress-strain computation options - /PROP *149*
structural dynamic simulations - time step *79*
structural FSI data *465*, *479*
symmetry conditions, SPH *523*

T

tabulated failure model
/FAIL/TAB1 *249*
theoretical speedup, multi-domain technique *534*
thermal analysis, implicit analysis *460*
tied contact, /INTER/TYPE2 *334*
tied interface (/INTER/TYPE10) *373*
time History files *542*
time integration *498*
time step - stiffness formulation *99*
time step output from a model *40*
time step scale factor *77*
time step, explicit structural fea *37*
troubleshooting
ALE/CFD *516*
troubleshooting, implicit FEA *483*
turbulent wall *503*

U

uniform pressure run *416*
unit systems *454*
units, ALE *514*

V

vent holes *404*
venting and fabric porosity activation *408*
verify consistency *441*

W

workstation
    single *29*
    three *30*
    two *30*