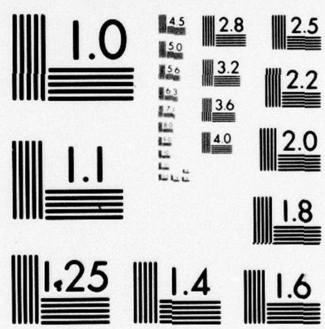
AD-A073 769 AIR FORCE WEAPONS LAB KIRTLAND AFB NM F/G 9/2
AFWL IMPLEMENTATION OF ALTRAN.(U)
FEB 79 C E RHOADES

UNCLASSIFIED AFWL-TR-78-129 SBIE-AD-E200 314 NL

1 OF 1
ADA
073769

END
DATE
FILMED
10-79
DDC



MICROCOPY RESOLUTION TEST CHART

AFWL-TR-78-129

AFWL-TR-78-129

AFWL IMPLEMENTATION OF ALTRAN

Clifford E. Rhoades, Jr.

February 1979

Final Report

Approved for public release; distribution unlimited.

AIR FORCE WEAPONS LABORATORY
Air Force Systems Command
Kirtland Air Force Base, NM 87117

This final report was prepared by the Air Force Weapons Laboratory, Kirtland Air Force Base, New Mexico under Job Order 88091822. Dr. Clifford E. Rhoades, Jr. (DYP) was the Laboratory Project Officer-in-Charge.

This technical report has been reviewed and is approved for publication.

Clifford E. Rhoades, Jr.

CLIFFORD E. RHOADES, JR.
Project Officer

FOR THE COMMANDER

Norman F. Roderick	Thomas W. Ciambrone
NORMAN F. RODERICK	THOMAS W. CIAMBRONE
Lt Colonel, USAF	Colonel, USAF
Chief, Advanced Concepts Branch	Chief, Applied Physics Division

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER AFWL-TR-78-129	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) AFWL IMPLEMENTATION OF ALTRAN		5. TYPE OF REPORT & PERIOD COVERED Final Report 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Clifford E. Rhoades, Jr.		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Weapons Laboratory (DYS) Kirtland Air Force Base, NM 87117		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62601F 88091822
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Weapons Laboratory (DYS) Kirtland Air Force Base, NM 87117		12. REPORT DATE February 1979 13. NUMBER OF PAGES 26
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Symbolic computing Algebraic processing Formula verification		

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

ALTRAN is a system for processing algebra. This report provides a short introduction to the use of ALTRAN and to its implementation at the Air Force Weapons Laboratory. A number of examples are included for reference. A complete description of ALTRAN is contained in the Fourth Edition of the ALTRAN User's Manual (see ref. 1).

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

PREFACE

The ALTRAN algebraic processing system of Bell Laboratories, Inc., is available at the Air Force Weapons Laboratory under a SOFTWARE AGREEMENT and a PATENT LICENSE AGREEMENT between Western Electric Company, Inc., and the United States of America as represented by the Department of Defense.

These agreements grant a personal, nontransferable, and nonexlusive right to use ALTRAN solely for internal business purposes and solely on the worker computers located at the Air Force Weapons Laboratory, Kirtland Air Force Base, New Mexico. These computers are CDC Cyber 176, Serial Numbers 102 and 104, and CDC 6600, Serial Numbers 6 and 1043. In addition, these agreements provide that ALTRAN shall be held in confidence, that it will not be used except as authorized, and that no copies shall be made except those copies which are necessary for use or for safekeeping.

It is a pleasure to acknowledge the assistance of Dr. Wayne Fullerton of Los Alamos Scientific Laboratory (LASL/C-3) in the installation of ALTRAN and in the preparation of this report.

CONTENTS

SECTION I

INTRODUCTION

The ALTRAN algebraic processor has been implemented at the Air Force Weapons Laboratory (AFWL). This makes possible the solution and checking of research involving much algebraic computation. The ALTRAN language is fully described in reference 1. The reader should use this technical report in conjunction with reference 1; comparable sections from reference 1 are cited in parenthesis throughout the report.

ALTRAN is available at a number of computing installations, but few as large as the AFWL. We have a minimum of 10K 60 bit words of workspace, and the CDC Cyber 176 is generally 10 to 30 times faster than the computers on which ALTRAN is usually implemented. These facts coupled with the efficiency of ALTRAN permit truly large algebraic problems to be done.

ALTRAN is about 15 times faster than FORMAC in a PL/I environment, and it is at least 12 times faster than REDUCE. Moreover, ALTRAN uses very little core compared with FORMAC and REDUCE. ALTRAN has already solved in 15 seconds problems which could not be solved with FORMAC or REDUCE because more than a half-million words of storage were required and a prohibitive amount of execution time would have been required.

The ALTRAN system is still under development--a substantial advantage because bugs can be fixed as they are discovered. One feature of ALTRAN has not yet been implemented: interlanguage communication still is not (easily) possible.

The best way to learn ALTRAN is to read the manual carefully, especially sections B, H, and L. Run several simple problems at the same time--a good one to try can be found in the examples section of this report. Turn on all the options (LISTOUT, LISTF) just this once to get an idea of what is going on. After you have run several programs, you will probably find a rereading of section B very helpful.

---

1. Brown, Dr. W. S., ALTRAN User's Manual, 4th ed., Bell Laboratories, 600 Mountain Ave., Murray Hill, NJ 07974.

Note that the restriction of the language to rational functions still includes a very large class of problems and that this restriction does not prohibit the use of special functions (e.g., SIN and COS) if one is willing to treat each function as an independent variable and define special simplification rules via AMOD. Also note that the prohibition of floating point numbers in algebraics can be circumvented by declaring the algebraics to have another independent variable, call it TEN. Then 1.8603E12 could be represented by 18603 * TEN ** 8, and 0.054 as 54/TEN**3. Do not use too many significant figures, or integer overflow may result.

In writing efficient programs there are two good general rules which you can follow. First, try to formulate your problem in a way which minimizes the number of terms in intermediate expressions. Second, try to formulate your problem so that it is easy for ALTRAN to do. The order of preference, with most preferable formulations first, is

1. Polynomials
2. Polynomials with side conditions
3. Rational polynomials
4. Rational polynomials with side conditions

If you publish any research which required ALTRAN as a step, a reprint would be appreciated by ALTRAN authors A. D. Hall (ref. 2) and Dr. W. D. Brown of Bell Laboratories (ref. 1). We hope that your application of ALTRAN along with any difficulties you may have encountered will influence the future development of ALTRAN and other algebraic languages.

---

2. Hall, A. D., "The ALTRAN System for Rational Function Manipulation: A Survey," Comm. ACM, 14, pp 517-521, 1971.

SECTION II

LOCAL CONSIDERATIONS

1. INTRODUCTION (Q.6.1)

The ALTRAN system has been designed and implemented in a way that allows it to be installed on a variety of computers. One of the advantages of this flexibility is that certain characteristics of the implementation can easily be varied to fit particular computers and operating systems. The purpose of this section is to provide a quick reference guide to those features which are peculiar to the ALTRAN installation on the CDC Cyber 176 computers at the AFWL.

2. COMPILATION AND EXECUTION (Q.6.2)

The control cards needed to compile and execute an ALTRAN program are similar to those used to execute a small FORTRAN program--the biggest difference is that two extra control cards are needed in order to set up the ALTRAN run-time library. Another difference is that there are a few extra options for the ALTRAN user. If the ALTRAN program uses default input/output units (Q.6.4), default long precision (Q.6.3 and Q.6.13), and other default parameters, then the deck setup for executing an ALTRAN program is as follows:

```
EDR,T77.   (sample job card)
ACCOUNT(NAME,JOBORDER-DCD,ORG,PHONE)    (sample account card)
ATTACH(OLD,ALTRANLIB,ID=DYSXCER)
LIBRARY(OLD)
MAP(OFF)
COPYCR(INPUT,SYSIN)
ALTRAN(SYSIN)
FTN(A,I=FORTIN,R=0)
RFL,160000.     (optional see below)
LOAD(LGO)
XEQ.
7/8/9 card
           { ALTRAN source procedures
7/8/9 card
           { ALTRAN input data, if any
6/7/8/9 card
```

Many ALTRAN jobs will execute in less than 1 minute and print fewer than 50 pages. Output from WRITE and WRITE(6) statements is sent to OUTPUT, the printer. Diagnostic output, including symbolic snaps (E.4) and run statistics (B.5.13), is also sent to OUTPUT. If the program reads input via READ or READ (5) statements, the input will be taken from INPUT, the card reader.

The remainder of this section is devoted to a detailed explanation of the control cards.

The ATTACH and LIBRARY cards are used to make available the ALTRAN library from which both ALTRAN and the main program, XEQ, for the user's job step execution are loaded. ALTRAN is a FORTRAN program which translates ALTRAN source code to legal FORTRAN subroutines. These subroutines are written on the file FORTIN and the files TAPE10 and TAPE11 are used for intermediate storage. FORTIN is rewound both before and after it is used. The file INPUT is rewound prior to use. The ALTRAN program card is

```
PROGRAM ALTRAN(INPUT,OUTPUT,FORTIN,TAPE10,TAPE11,
TAPE5=INPUT,TAPE6=OUTPUT,TAPE20=FORTIN)
```

If ALTRAN translation time errors are detected, ALTRAN issues a job step abort after the last procedure in the input stream.

Otherwise, the FORTRAN compiler, FTN, is invoked to process the legal FORTRAN subroutines. A listing of the FORTRAN subroutine corresponding to an ALTRAN procedure is produced by FTN if the ALTRAN option LISTF is selected for that ALTRAN procedure. Otherwise, no FORTRAN source listing is made. Only the most enthusiastic paper recyclers could want a FORTRAN source listing. The DECK, NODECK options for producing a punched object deck are not implemented. There is little advantage in attempting to punch such decks. The time required to translate and compile ALTRAN codes is usually trivial, and the source decks usually have fewer cards than the resulting object decks.

Now the ALTRAN object code produced by FTN by default on LGO is ready to be executed. Because the load map produced with the default PART map option requires 10 pages, it is suggested that the map continue to be suppressed. Execute-time error reference the ALTRAN source code, and this is usually entirely sufficient. Of course, if a disastrous error occurs, both the FORTRAN source listing and the load map may have to be obtained.

If input for READ or READ (5) statements is to be obtained from a file other than INPUT, then INPUT should be assigned to that file on the XEQ card. Likewise, if output from WRITE or WRITE (6) statements is to go to the file other than OUTPUT, then OUTPUT should be assigned to that file. The XEQ program card is

```
PROGRAM XEQ(INPUT,OUTPUT,TAPE25,TAPE27,
TAPE5=INPUT,TAPE6=OUTPUT,TAPE26=OUTPUT)
```

Hence, only three files may be used.

In the current implementation of ALTRAN, there is only one run-time parameter which may be set. This is the number of words of work space to be used. The minimum value is 10,000 words. An RFL card can be used to specify any value of central memory allowed by NOS/BE, in which case the work space value is the memory remaining after all routines are loaded. With RFL, 160000., normally about 20,000 decimal words are available.

There are two restrictions in the current implementation. First, the number of words to be used for long precision integers is fixed at 2. Two words allow integers as large as 10^{28} which should be large enough for virtually all algebraic problems. Second, all hardware and software interrupts are automatically recovered. They are intercepted to permit ALTRAN to give a snap dump of the work space. Normally, this is the most useful information, since the work space is a dynamic stack.

3. LIMITS (Q.6.3)

The maximum and minimum precision for long integers is 2 words, and the maximum work space size is approximately 80,000 decimal words on our current NOS/BE. If a logical unit other than 5 or 6 is used, it must be 25.

4. RUN-TIME INPUT AND OUTPUT (Q.6.4)

At run-time, all ALTRAN input and output is carried out by FORTRAN READ and WRITE statements. Because the FORTRAN main program used to drive ALTRAN procedures is in the ALTRAN run-time library and not accessible to the user, all nondefault unit assignments must be made on the XEQ card (and then be processed by the main program). The units with a preassigned function in the run-time system are given below:

Logical unit 5 is used for READ statements with no specified unit or with a null unit number. By default, unit 6 is assigned to OUTPUT.

Logical unit 25 may be used for punched output, because it has a linelength of 80 columns. (Other units have a default linelength of 80 at AFWL.) See section Q.6.14 for another use of unit 25. (Note: For section Q.6.14 use, the linelength must be set to 72 by OPTS calls.) Punched output may be obtained by writing on unit 25 and assigning that unit to PUNCH on the XEQ card or by using a DISPOSE card.

5. REAL UNDERFLOW AND OVERFLOW (Q.6.5)

When a real underflow occurs, zero is assumed and execution continues.

When a real overflow occurs, the run is terminated (Q.6.6).

6. ABNORMAL TERMINATIONS (Q.6.6)

If the execution of an ALTRAN program is terminated because of an error detected by the FORTRAN run-time system or by the operating system, a SNAP and run statistics are automatically given. The first procedure in the SNAP will be the one in control at the time the error was detected. The reason for termination can be found in the dayfile.

7. END OF FILES (Q.6.7 also see B.3.6)

.EOF is not required at the end of translator input.

.EOF is not required at the end of run-time input.

8. LINE LENGTHS (Q.6.8)

The translator reads the first 80 characters from an input line, and prints them in the source listing (D.2). Sequence numbers may be used by preceding them with # in Column 72.

At run-time, only the first 80 characters of each input line are considered meaningful. When output is to be punched or used for later input, be sure that the value of the output line length option (D.4) is no more than 80 characters (the default at AFWL is 80).

The maximum value of the output line option is 136.

9. THE DECK AND LISTF OPTIONS (Q.6.9)

The punched object deck option (D.2) is not effective, and is permanently set to NODECK.

The FORTRAN listing option (D.2) is effective and defaults to NOLISTF for each ALTRAN procedure.

10. The translator reads input from files INPUT (by default) and writes program listings, symbol tables, etc., on file OUTPUT. Diagnostics are also written on file OUTPUT.

File OUTPUT is used for the listing of compiled FORTRAN translations, and the file LGO is used for object decks.

11. CHARACTER SET (Q.6.11)

The local representations of the ALTRAN special characters are the same as the standard representations (B.3.2), except as follows:

ALTRAN character		Local representation Data 100s (key punch)		All others
"	quote	"	(8-4)	≠
:	colon	:	(8-2)	:
;	semicolon	;	(12-8-7)	;
-	hyphen	-	(0-8-5)	↱
#	sharp	#	(0-8-6)	≡
**	exponentiation operator	**	(11-8-4,11-8-4)	**

12. TOKEN LENGTHS (Q.6.12)

In the interests of compatibility and uniformity, the maximum length of an identifier is 6 characters. For procedure names with external scope the maximum is 6 characters.

The maximum length of a character string is 135 characters.

All other tokens are limited to 145 characters in length.

13. PRECISION AND RANGE OF NUMBERS (Q.6.13)

The magnitude of a short integer cannot exceed 10^{14}-1. The magnitude of a long integer cannot exceed 10^{28}-1.

There are approximately 14 significant digits in a short real, but only 13 digits occur in output. Similarly, there are 28 significant digits in a long real, but only 27 occur in output.

The largest value that may be represented by short and long reals is approximately 10^{321}.

For further information about the precision and range of numbers see Section B.5.4 of reference 1.

14. GENERATING FORTRAN STATEMENTS IN ALTRAN (Q.6.14)

One nonstandard feature and one nonstandard convention have been implemented in ALTRAN in order to facilitate the generation of legal FORTRAN statements from ALTRAN programs. The convention requires on to WRITE 72-column output on unit number 25 by changing the line length via the procedure OPTS. That is, the line length of unit 25 must be set to 72 columns for use by FTNOUT (see below). Output written on 25 is otherwise normal ALTRAN output which could later be read from another ALTRAN procedure. There will be blank lines between output lines, no FORTRAN continuation characters, and multiple-precision integers will contain hyphens.

In order to obtain legal FORTRAN statements from output on unit 25, it is necessary to invoke the ALTRAN procedure FTNOUT, a nonstandard feature in ALTRAN. FTNOUT rewinds unit 25, then reads it and puts variable names on the left-hand side of equations. FTNOUT also deletes blank lines, inserts FORTRAN continuation numbers in column 6, and converts multiple precision integers to single precision floating point numbers. Output from FTNOUT is written on unit 27. Most likely it will be DISPOSED to PUNCH.

A very simple example of the use of these features follows this section. The example includes the intermediate output from TAPE25 as well as the output from FTNOUT. Note that a complete FORTRAN routine was written. In many cases, you will probably want to obtain only an equation or two as legal FORTRAN statements. Later you can insert the appropriate FORTRAN statements to make a complete routine. You may also wish to change some of the names of the variables on the left-hand side of the equation

Several notes are in order now.

a. Only 19 continuation cards are permitted by FORTRAN, but FTNOUT will write as many as are needed. If more than 19 are written, you will have to edit the output by hand and perhaps use a temporary variable in order to ensure that all equations have fewer than 19 continuation cards. In order to minimize the probability of such long equations, use 1 or 2 character independent variable names, use the factored option whenever possible, and try to formulate your problem to minimize the number of terms. Sometimes a simple substitution such as X = Y - 1 can greatly reduce the number of terms.

b. Because multiple precision integers are truncated to single precision floating point numbers, you must be certain that the attendant loss of precision will not adversely affect the answers you need. Ordinarily, you can ensure this simply by examining the equation. In complicated cases, however, you may wish to substitute several numbers for the variables in ALTRAN and compare the (exact or double precision) ALTRAN numerical results with the (approximate) FORTRAN results.

c. FTNOUT obtains the name of the left-hand variable from the comment automatically written by ALTRAN immediately preceding the variable's value. Arrays, however, are written as lists; they contain only one name and subexpressions may appear as an equals sign followed by a number. Thus, arrays must be written to unit 25, one element at a time.

SECTION III

EXAMPLES

The following three examples are provided for instruction purposes.

```
ALTRAN VERSION 1 LEVEL 9

          PROCEDURE MAIN  # SIMPLE EXAMPLE OF USE OF FTNOUT
          LONG ALGEBRAIC (X:10,Y:10) F
          ALTRAN FTNOUT
          OPTS(201,72)  # FTNOUT REQUIRES A LINE LENGTH OF 72
          F = EXPAND( (X+2*Y+1000000)**3 )
          WRITE F  # PRINT F
          WRITE (25) "      FUNCTION F(X,Y)",
                     "C EXAMPLE PROG WRITTEN WITH FTNOUT." ,
                     F,
                     "      RETURN",  "      END"
     # WE HAVE WRITTEN A SIMPLE PROGRAM ON UNIT 25,  NOW WE INVOKE FTNOUT TO
     # THIS ALTRAN OUTPUT TO LEGAL FORTRAN.
          FTNOUT
          END
```

```
NAME/EXTNAME         USE  TYPE STRUC PREC CLASS SCOPE  DB   LAY   ADDR


F                    VAR  ALG        L                      L*001
X                    IND  ALG                               L*001
Y                    IND  ALG                               L*001
EXPAND/S9XPND        PROC ALG        L     S     X
FTNOUT               PROC            L     S     X
MAIN                 PROC            L     S     X
OPTS/S9OPTS          PROC INT              S     X
L*001                LAY
C EXAMPLE PROG WRITT CONS CHAR             S
1000000              CONS INT              S
10                   CONS INT              S
201                  CONS INT              S
25                   CONS INT              S
2                    CONS INT              S
3                    CONS INT              S
72                   CONS INT              S
      END            CONS CHAR             S
      FUNCTION F(X,Y CONS CHAR             S
      RETURN         CONS CHAR             S
```

```
# F

      X**3 + 6*X**2*Y + 3000000*X**2 + 12*X*Y**2 + 12000000*X*Y +

      3000000000000*X + 8*Y**3 + 12000000*Y**2 + 6000000000000*Y +

      10000_00000000000000
```

```
*** NORMAL RETURN FROM MAIN PROCEDURE

*** RUN STATISTICS

        .046 SECONDS ELAPSED
          14 DIGITS IN SHORT INTEGERS
          28 DIGITS IN LONG  INTEGERS
       10017 WORDS IN WORKSPACE
         388 MAXIMUM WORDS USED
         149 CURRENT WORDS USED
           0 GARBAGE COLLECTIONS


#        FUNCTION F(X,Y)

# C EXAMPLE PROG WRITTEN WITH FTNOUT.

# F

      X**3 + 6*X**2*Y + 3000000*X**2 + 12*X*Y**2 + 12000000*X*Y +

      3000000000000*X + 8*Y**3 + 12000000*Y**2 + 6000000000000*Y +

      10000_0000000000000

#        RETURN

#        END


      FUNCTION F(X,Y)
C EXAMPLE PROG WRITTEN WITH FTNOUT.
      F =X**3 + 6*X**2*Y + 3000000*X**2 + 12*X*Y**2 + 12000000*X*Y +
     1 3000000000000*X + 8*Y**3 + 12000000*Y**2 + 6000000000000*Y +
     2 1000000000000000E4
      RETURN
      END
```

```
MFY   NOS/BE 1.2    KAFB  009 MFY    03/27/78
FLCM=314000 MXCM=314000 FLEC=1720K  MXEC=0750K

20.52.14.EDROOK7  FROM MX2/IU
20.52.14.IP  00000256 WORDS - FILE INPUT   , DC 04
20.52.14.EDR,T77,P60.
20.52.15.ACCOUNT(RHOADES,********-***,DYP,1851)
20.52.15.SYSBULL(BATCH)
20.52.16.DISPOSE(OUTPUT)
20.52.16.MAP(OFF)

20.52.16.ATTACH(OLD,ALTRANLIB,ID=DYSXCER)
20.52.16.PF CYCLE NO. = 003
20.52.16.LIBRARY(OLD)
20.52.16.COPYCR(INPUT,SYSIN)
20.52.17.ALTRAN(SYSIN)
20.52.23.     STOP
20.52.23.FTN(A,I=FORTIN,R=0)
20.52.25.      .047 CP SECONDS COMPILATION TIME
20.52.25.LOAD,LGO.
20.52.26.XEQ.
20.52.35.  EXIT
20.52.35.REWIND(TAPE25,TAPE27)
20.52.36.COPYSBF(TAPE25,OUTPUT)
20.52.37.COPYSBF(TAPE27,OUTPUT)
20.52.37.OP  00001024 WORDS - FILE OUTPUT  , DC 40
20.52.37.MS     3584  WORDS  (    75264 MAX USED)
20.52.37.CPA       1.172 SEC.          1.172 ADJ.
20.52.37.IO        3.587 SEC.           .896 ADJ.
20.52.37.CM       62.126 KWS.           .118 ADJ.
20.52.37.SS                            3.282
20.52.37.** PRIORITY JOB **
20.52.37.PP       11.050 SEC.      DATE 05/01/78
20.52.37.COST ESTIMATE                 $.55
20.52.37.EJ  END OF JOB, IU
```

ALTRAN VERSION 1 LEVEL 9

```
PROCEDURE MAIN
ALGEBRAIC (X:20) TCHEBY,T
ALTRAN TCHEBY
INTEGER N
OPTS(101,1)
DO N = 0,3
T = TCHEBY(N,X)
WRITE T
DOEND
END
```

NAME/EXTNAME	USE	TYPE	STRUC	PREC	CLASS	SCOPE	DB	LAY	ADDR
TCHEBY	PROC	ALG		L	S	X		L*001	
T	VAR	ALG						L*001	
X	IND	ALG						L*001	
MAIN	PROC			L	S	X			
N	VAR	INT							
OPTS/S9OPTS	PROC	INT			S	X			
L*001	LAY								
0	CONS	INT			S				
101	CONS	INT			S				
1	CONS	INT			S				
20	CONS	INT			S				
3	CONS	INT			S				

ALTRAN VERSION 1 LEVEL 9

```
PROCEDURE TCHEBY(N,X)
ALGEBRAIC TCHEBY,X
STATIC X
INTEGER N
IF(N .EQ. 0) RETURN (1)
IF(N .EQ. 1) RETURN (X)
RETURN ( 2*X*TCHEBY(N-1) - TCHEBY(N-2) )
END
```

NAME/EXTNAME	USE	TYPE	STRUC	PREC	CLASS	SCOPE	DB	LAY	ADDR
N	VAR	INT				B			
TCHEBY	PROC	ALG		L	S	X			
X	VAR	ALG			S	D			
0	CONS	INT			S				
1	CONS	INT			S				
2	CONS	INT			S				

```
# T

      1

# T

      X

# T

      2*X**2 - 1

# T

      X * ( 4*X**2 - 3 )


*** NORMAL RETURN FROM MAIN PROCEDURE

*** RUN STATISTICS

       .050 SECONDS ELAPSED
         14 DIGITS IN SHORT INTEGERS
         28 DIGITS IN LONG  INTEGERS
      10062 WORDS IN WORKSPACE
        395 MAXIMUM WORDS USED
        194 CURRENT WORDS USED
          0 GARBAGE COLLECTIONS
```

```
MFY   NOS/BE 1.2    KAFB  009 MFY    03/27/78
FLCM=314000 MXCM=314000 FLEC=1720K  MXEC=0750K
20.49.41.EDROOK5  FROM MX2/IU
20.49.41.IP  00000320 WORDS - FILE INPUT  , DC 04
20.49.41.EDR,T77,P60.
20.49.42.ACCOUNT(RHOADES,********-***,DYP,1851)
20.49.42.SYSBULL(BATCH)
20.49.43.DISPOSE(OUTPUT)
20.49.43.ATTACH(OLD,ALTRANLIB,ID=DYSXCER)
20.49.43.PF CYCLE NO. = 003
20.49.43.LIBRARY(OLD)
20.49.51.MAP(OFF)
20.49.51.COPYCR(INPUT,SYSIN)
20.49.52.ALTRAN(SYSIN)
20.49.58.    STOP
20.49.58.FTN(A,I=FORTIN,R=0)
20.50.01.       .071 CP SECONDS COMPILATION TIME
20.50.01.LOAD(LGO)
20.50.01.XEQ.
20.50.10.  EXIT
20.50.10.EXIT(S)
20.50.10.OP  00000640 WORDS - FILE OUTPUT , DC 40
20.50.10.MS   3584  WORDS  (    75264 MAX USED)
20.50.10.CPA       1.243 SEC.       1.243 ADJ.
20.50.10.IO        3.758 SEC.        .939 ADJ.
20.50.10.CM       67.549 KWS.        .128 ADJ.
20.50.10.SS                         3.467
20.50.10.** PRIORITY JOB **
20.50.10.PP       11.239 SEC.     DATE 05/01/78
20.50.10.COST ESTIMATE             $.58
20.50.10.EJ  END OF JOB, IU
```

```
ALTRAN VERSION 1 LEVEL 9

          PROCEDURE MAIN  # SIMPLE EXAMPLE OF USE OF FTNOUT
          LONG ALGEBRAIC (X:10,Y:10) F
          ALTRAN FTNOUT
          OPTS(201,72)  # FTNOUT REQUIRES A LINE LENGTH OF 72
          F = EXPAND( (X+2*Y+1000000)**3 )
          WRITE F  # PRINT F
          WRITE (25) "       FUNCTION F(X,Y)",
                     "C EXAMPLE PROG WRITTEN WITH FTNOUT." ,
                     F,
                     "       RETURN",  "       END"
     # WE HAVE WRITTEN A SIMPLE PROGRAM ON UNIT 25,  NOW WE INVOKE FTNOUT TO
     # THIS ALTRAN OUTPUT TO LEGAL FORTRAN.
          FTNOUT
          END

NAME/EXTNAME         USE  TYPE STRUC PREC CLASS SCOPE  DB    LAY   ADDR


F                    VAR  ALG        L                      L*001
X                    IND  ALG                               L*001
Y                    IND  ALG                               L*001
EXPAND/S9XPND        PROC ALG        L         S     X
FTNOUT               PROC            L         S     X
MAIN                 PROC            L         S     X
OPTS/S9OPTS          PROC INT                  S     X
L*001                LAY
C EXAMPLE PROG WRITT CONS CHAR                 S
1000000              CONS INT                  S
10                   CONS INT                  S
201                  CONS INT                  S
25                   CONS INT                  S
2                    CONS INT                  S
3                    CONS INT                  S
72                   CONS INT                  S
      END            CONS CHAR                 S
      FUNCTION F(X,Y CONS CHAR                 S
      RETURN         CONS CHAR                 S


# F

      X**3 + 6*X**2*Y + 3000000*X**2 + 12*X*Y**2 + 12000000*X*Y +

      300000000000*X + 8*Y**3 + 12000000*Y**2 + 600000000000*Y +

      10000_00000000000000
```

```
*** NORMAL RETURN FROM MAIN PROCEDURE

*** RUN STATISTICS

       .046 SECONDS ELAPSED
         14 DIGITS IN SHORT INTEGERS
         28 DIGITS IN LONG  INTEGERS
      10017 WORDS IN WORKSPACE
        388 MAXIMUM WORDS USED
        149 CURRENT WORDS USED
          0 GARBAGE COLLECTIONS


#     FUNCTION F(X,Y)

# C EXAMPLE PROG WRITTEN WITH FTNOUT.

# F

      X**3 + 6*X**2*Y + 3000000*X**2 + 12*X*Y**2 + 12000000*X*Y +

      3000000000000*X + 8*Y**3 + 12000000*Y**2 + 6000000000000*Y +

      10000_00000000000000

#     RETURN

#     END
```

ALTRAN VERSION 1 LEVEL 9

```
PROCEDURE MAIN
ALGEBRAIC (X:20) TCHEBY,T
ALTRAN TCHEBY
INTEGER N
OPTS(101,1)
DO N = 0,3
T = TCHEBY(N,X)
WRITE T
DOEND
END
```

NAME/EXTNAME	USE	TYPE	STRUC	PREC	CLASS	SCOPE	DB	LAY	ADDR
TCHEBY	PROC	ALG		L	S	X		L*001	
T	VAR	ALG						L*001	
X	IND	ALG						L*001	
MAIN	PROC			L	S	X			
N	VAR	INT							
OPTS/S9OPTS	PROC	INT			S	X			
L*001	LAY								
0	CONS	INT			S				
101	CONS	INT			S				
1	CONS	INT			S				
20	CONS	INT			S				
3	CONS	INT			S				

ALTRAN VERSION 1 LEVEL 9

```
PROCEDURE TCHEBY(N,X)
ALGEBRAIC TCHEBY,X
STATIC X
INTEGER N
IF(N .EQ. 0) RETURN (1)
IF(N .EQ. 1) RETURN (X)
RETURN ( 2*X*TCHEBY(N-1) - TCHEBY(N-2) )
END
```

NAME/EXTNAME	USE	TYPE	STRUC	PREC	CLASS	SCOPE	DB	LAY	ADDR
N	VAR	INT				D			
TCHEBY	PROC	ALG		L	S	X			
X	VAR	ALG			S	D			
0	CONS	INT			S				
1	CONS	INT			S				
2	CONS	INT			S				

```
# T

        1

# T

        X

# T

        2*X**2 - 1

# T

        X * ( 4*X**2 - 3 )


*** NORMAL RETURN FROM MAIN PROCEDURE

*** RUN STATISTICS

        .048 SECONDS ELAPSED
          14 DIGITS IN SHORT INTEGERS
          28 DIGITS IN LONG  INTEGERS
       17614 WORDS IN WORKSPACE
         395 MAXIMUM WORDS USED
         194 CURRENT WORDS USED
           0 GARBAGE COLLECTIONS
```

```
  MFY   NOS/BE 1.2    KAFB  009 MFY    03/27/78
  FLCM=314000 MXCM=314000 FLEC=1720K  MXEC=0750K

20.48.26.EDR00K4  FROM MX2/IU
20.48.27.IP  00000320 WORDS - FILE INPUT  , DC 04
20.48.27.EDR,T77,P60.
20.48.27.ACCOUNT(RHOADES,********-***,DYP,1851)
20.48.27.SYSBULL(BATCH)
20.48.28.DISPOSE(OUTPUT)
20.48.28.ATTACH(OLD,ALTRANLIB,ID=DYSXCER)
20.48.28.PF CYCLE NO. = 003
20.48.29.LIBRARY(OLD)
20.48.29.MAP(OFF)
20.48.29.COPYCR(INPUT,SYSIN)
20.48.29.ALTRAN(SYSIN)
20.48.37.    STOP
20.48.38.FTN(A,I=FORTIN,R=0)
20.48.41.      .074 CP SECONDS COMPILATION TIME
20.48.41.RFL,140000.     GETS ADDITIONAL WORKSPAC
20.48.41.E -NOT NEEDED FOR THIS EXAMPLE
20.48.41.LOAD(LGO)
20.48.41.XEQ.
20.48.52.  EXIT
20.48.52.EXIT(S)
20.48.52.OP  00000640 WORDS - FILE OUTPUT , DC 40
20.48.52.MS    3584  WORDS  (   75264 MAX USED)
20.48.52.CPA       1.190 SEC.         1.190 ADJ.
20.48.52.IO        3.693 SEC.          .923 ADJ.
20.48.52.CM       88.634 KWS.          .168 ADJ.
20.48.52.SS                           3.424
20.48.52.** PRIORITY JOB **
20.48.52.PP       10.571 SEC.     DATE 05/01/78
20.48.52.COST ESTIMATE                $.57
20.48.52.EJ  END OF JOB, IU
```