

AD-A073 769

AIR FORCE WEAPONS LAB KIRTLAND AFB NM
AFWL IMPLEMENTATION OF ALTRAN.(U)
FEB 79 C E RHOADES

F/G 9/2

UNCLASSIFIED

AFWL-TR-78-129

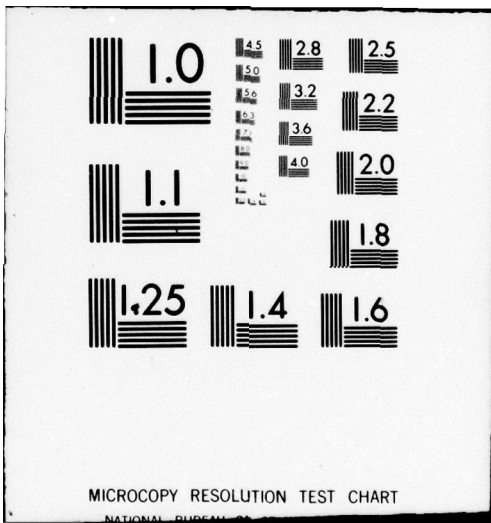
SBIE-AD-E200 314

NL

| OF |
ADA
073769



END
DATE
FILMED
10-79
DDC



MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS-1963-A

AFWL-TR-78-129

DDC
RECEIVED
SEP 13 1979

②

AFWL-TR-78-129

ADE800314

LEVEL III

AD A 073 769

AFWL IMPLEMENTATION OF ALTRAN

Clifford E. Rhoades, Jr.

February 1979

Final Report

Approved for public release; distribution unlimited.



DDC FILE COPY

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

79 08 20 050

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.

When US Government drawings, specifications, or other data are used for any purpose other than a definitely related Government procurement operation, the Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise, as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report has been authored by an employee of the United States Government. Accordingly, the United States Government retains a nonexclusive, royalty-free license to publish or reproduce the material contained herein, or allow others to do so, for the United States Government purposes.

This report has been reviewed by the Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

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

Clifford E. Rhoades, Jr.

CLIFFORD E. RHOADES, JR.
Project Officer

Norman F. Roderick

NORMAN F. RODERICK
Lt Colonel, USAF
Chief, Advanced Concepts Branch

FOR THE COMMANDER

Thomas W. Ciambrone

THOMAS W. CIAMBRONE
Colonel, USAF
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). A		

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

<p>1. REPORT NUMBER</p> <p>AFM-78-150</p>	<p>2. TITLE AND SUBTITLE</p> <p>REAL IMPLEMENTATION OF ALTIM</p>
<p>3. PERFORMING ORGANIZATION NUMBER</p> <p>Final Report</p>	<p>4. AUTHOR(s)</p> <p>Clayton E. Woodard, Jr.</p>
<p>5. CONTRACT OR GRANT NUMBER(s)</p>	<p>6. PERFORMING ORGANIZATION NAME AND ADDRESS</p> <p>Air Force Weapons Laboratory (AFWL) Kirtland Air Force Base, NM 87117</p>
<p>7. AUTHORING ORGANIZATION NAME AND ADDRESS</p> <p>AFWL</p>	<p>8. PERFORMING ORGANIZATION NUMBER</p>
<p>9. PERFORMING ORGANIZATION REPORT NUMBER</p> <p>AFWL-78-150</p>	<p>10. CONTROLLING ORGANIZATION NAME AND ADDRESS</p> <p>Air Force Weapons Laboratory (AFWL) Kirtland Air Force Base, NM 87117</p>
<p>11. NUMBER OF PAGES</p> <p>UNCLASSIFIED</p>	<p>12. DISTRIBUTION STATEMENT (When Data Entered)</p> <p>Approved for public release; distribution unlimited.</p>
<p>13. SECURITY CLASSIFICATION OF THIS REPORT</p> <p>UNCLASSIFIED</p>	<p>14. SECURITY CLASSIFICATION OF THIS ABSTRACT</p> <p>UNCLASSIFIED</p>
<p>15. SECURITY CLASSIFICATION OF THIS ABSTRACT</p> <p>UNCLASSIFIED</p>	<p>16. DISTRIBUTION STATEMENT (When Data Entered)</p> <p>Approved for public release; distribution unlimited.</p>
<p>17. DISTRIBUTION STATEMENT (When Data Entered)</p> <p>Approved for public release; distribution unlimited.</p>	<p>18. DISTRIBUTION STATEMENT (When Data Entered)</p> <p>Approved for public release; distribution unlimited.</p>
<p>19. DISTRIBUTION STATEMENT (When Data Entered)</p> <p>Approved for public release; distribution unlimited.</p>	<p>20. DISTRIBUTION STATEMENT (When Data Entered)</p> <p>Approved for public release; distribution unlimited.</p>
<p>21. DISTRIBUTION STATEMENT (When Data Entered)</p> <p>Approved for public release; distribution unlimited.</p>	<p>22. DISTRIBUTION STATEMENT (When Data Entered)</p> <p>Approved for public release; distribution unlimited.</p>
<p>23. DISTRIBUTION STATEMENT (When Data Entered)</p> <p>Approved for public release; distribution unlimited.</p>	<p>24. DISTRIBUTION STATEMENT (When Data Entered)</p> <p>Approved for public release; distribution unlimited.</p>
<p>25. DISTRIBUTION STATEMENT (When Data Entered)</p> <p>Approved for public release; distribution unlimited.</p>	<p>26. DISTRIBUTION STATEMENT (When Data Entered)</p> <p>Approved for public release; distribution unlimited.</p>
<p>27. DISTRIBUTION STATEMENT (When Data Entered)</p> <p>Approved for public release; distribution unlimited.</p>	<p>28. DISTRIBUTION STATEMENT (When Data Entered)</p> <p>Approved for public release; distribution unlimited.</p>
<p>29. DISTRIBUTION STATEMENT (When Data Entered)</p> <p>Approved for public release; distribution unlimited.</p>	<p>30. DISTRIBUTION STATEMENT (When Data Entered)</p> <p>Approved for public release; distribution unlimited.</p>
<p>31. DISTRIBUTION STATEMENT (When Data Entered)</p> <p>Approved for public release; distribution unlimited.</p>	<p>32. DISTRIBUTION STATEMENT (When Data Entered)</p> <p>Approved for public release; distribution unlimited.</p>
<p>33. DISTRIBUTION STATEMENT (When Data Entered)</p> <p>Approved for public release; distribution unlimited.</p>	<p>34. DISTRIBUTION STATEMENT (When Data Entered)</p> <p>Approved for public release; distribution unlimited.</p>

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 nonexclusive 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.

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DDC TAB	<input type="checkbox"/>
Unannounced Justification	<input type="checkbox"/>
By _____	
Distribution/	
Availability Codes	
Dist	Avail and/or special
A	

CONTENTS

<u>Section</u>		<u>Page</u>
I	INTRODUCTION	5
II	LOCAL CONSIDERATIONS	7
III	EXAMPLES	15

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^{2^8} 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 XEO 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 $10^{20}-1$.

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 one 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.

ALTRAV VERSION 1 LEVEL 2

SECTION III

EXAMPLES

The following three examples are provided for instruction purposes.

```

PROGRAMME WITH A
FORM A SPECIAL
ALTRAV FORM
WRITE 1 1 PAINT 1
WRITE 2 2 PAINT 2
WRITE 3 3 PAINT 3
WRITE 4 4 PAINT 4
WRITE 5 5 PAINT 5
WRITE 6 6 PAINT 6
WRITE 7 7 PAINT 7
WRITE 8 8 PAINT 8
WRITE 9 9 PAINT 9
WRITE 10 10 PAINT 10
WRITE 11 11 PAINT 11
WRITE 12 12 PAINT 12
WRITE 13 13 PAINT 13
WRITE 14 14 PAINT 14
WRITE 15 15 PAINT 15
WRITE 16 16 PAINT 16
WRITE 17 17 PAINT 17
WRITE 18 18 PAINT 18
WRITE 19 19 PAINT 19
WRITE 20 20 PAINT 20
WRITE 21 21 PAINT 21
WRITE 22 22 PAINT 22
WRITE 23 23 PAINT 23
WRITE 24 24 PAINT 24
WRITE 25 25 PAINT 25
WRITE 26 26 PAINT 26
WRITE 27 27 PAINT 27
WRITE 28 28 PAINT 28
WRITE 29 29 PAINT 29
WRITE 30 30 PAINT 30
WRITE 31 31 PAINT 31
WRITE 32 32 PAINT 32
WRITE 33 33 PAINT 33
WRITE 34 34 PAINT 34
WRITE 35 35 PAINT 35
WRITE 36 36 PAINT 36
WRITE 37 37 PAINT 37
WRITE 38 38 PAINT 38
WRITE 39 39 PAINT 39
WRITE 40 40 PAINT 40
WRITE 41 41 PAINT 41
WRITE 42 42 PAINT 42
WRITE 43 43 PAINT 43
WRITE 44 44 PAINT 44
WRITE 45 45 PAINT 45
WRITE 46 46 PAINT 46
WRITE 47 47 PAINT 47
WRITE 48 48 PAINT 48
WRITE 49 49 PAINT 49
WRITE 50 50 PAINT 50
WRITE 51 51 PAINT 51
WRITE 52 52 PAINT 52
WRITE 53 53 PAINT 53
WRITE 54 54 PAINT 54
WRITE 55 55 PAINT 55
WRITE 56 56 PAINT 56
WRITE 57 57 PAINT 57
WRITE 58 58 PAINT 58
WRITE 59 59 PAINT 59
WRITE 60 60 PAINT 60
WRITE 61 61 PAINT 61
WRITE 62 62 PAINT 62
WRITE 63 63 PAINT 63
WRITE 64 64 PAINT 64
WRITE 65 65 PAINT 65
WRITE 66 66 PAINT 66
WRITE 67 67 PAINT 67
WRITE 68 68 PAINT 68
WRITE 69 69 PAINT 69
WRITE 70 70 PAINT 70
WRITE 71 71 PAINT 71
WRITE 72 72 PAINT 72
WRITE 73 73 PAINT 73
WRITE 74 74 PAINT 74
WRITE 75 75 PAINT 75
WRITE 76 76 PAINT 76
WRITE 77 77 PAINT 77
WRITE 78 78 PAINT 78
WRITE 79 79 PAINT 79
WRITE 80 80 PAINT 80
WRITE 81 81 PAINT 81
WRITE 82 82 PAINT 82
WRITE 83 83 PAINT 83
WRITE 84 84 PAINT 84
WRITE 85 85 PAINT 85
WRITE 86 86 PAINT 86
WRITE 87 87 PAINT 87
WRITE 88 88 PAINT 88
WRITE 89 89 PAINT 89
WRITE 90 90 PAINT 90
WRITE 91 91 PAINT 91
WRITE 92 92 PAINT 92
WRITE 93 93 PAINT 93
WRITE 94 94 PAINT 94
WRITE 95 95 PAINT 95
WRITE 96 96 PAINT 96
WRITE 97 97 PAINT 97
WRITE 98 98 PAINT 98
WRITE 99 99 PAINT 99
WRITE 100 100 PAINT 100

```

LINE	TEXT	TYPE	CLASS	SCORE	DE	JAY	ABD
1	VAR ALB						
2	VAR ALB						
3	VAR ALB						
4	VAR ALB						
5	VAR ALB						
6	VAR ALB						
7	VAR ALB						
8	VAR ALB						
9	VAR ALB						
10	VAR ALB						
11	VAR ALB						
12	VAR ALB						
13	VAR ALB						
14	VAR ALB						
15	VAR ALB						
16	VAR ALB						
17	VAR ALB						
18	VAR ALB						
19	VAR ALB						
20	VAR ALB						
21	VAR ALB						
22	VAR ALB						
23	VAR ALB						
24	VAR ALB						
25	VAR ALB						
26	VAR ALB						
27	VAR ALB						
28	VAR ALB						
29	VAR ALB						
30	VAR ALB						
31	VAR ALB						
32	VAR ALB						
33	VAR ALB						
34	VAR ALB						
35	VAR ALB						
36	VAR ALB						
37	VAR ALB						
38	VAR ALB						
39	VAR ALB						
40	VAR ALB						
41	VAR ALB						
42	VAR ALB						
43	VAR ALB						
44	VAR ALB						
45	VAR ALB						
46	VAR ALB						
47	VAR ALB						
48	VAR ALB						
49	VAR ALB						
50	VAR ALB						
51	VAR ALB						
52	VAR ALB						
53	VAR ALB						
54	VAR ALB						
55	VAR ALB						
56	VAR ALB						
57	VAR ALB						
58	VAR ALB						
59	VAR ALB						
60	VAR ALB						
61	VAR ALB						
62	VAR ALB						
63	VAR ALB						
64	VAR ALB						
65	VAR ALB						
66	VAR ALB						
67	VAR ALB						
68	VAR ALB						
69	VAR ALB						
70	VAR ALB						
71	VAR ALB						
72	VAR ALB						
73	VAR ALB						
74	VAR ALB						
75	VAR ALB						
76	VAR ALB						
77	VAR ALB						
78	VAR ALB						
79	VAR ALB						
80	VAR ALB						
81	VAR ALB						
82	VAR ALB						
83	VAR ALB						
84	VAR ALB						
85	VAR ALB						
86	VAR ALB						
87	VAR ALB						
88	VAR ALB						
89	VAR ALB						
90	VAR ALB						
91	VAR ALB						
92	VAR ALB						
93	VAR ALB						
94	VAR ALB						
95	VAR ALB						
96	VAR ALB						
97	VAR ALB						
98	VAR ALB						
99	VAR ALB						
100	VAR ALB						

ALTRAN VERSION 1 LEVEL 9

```

1      PROCEDURE MAIN # SIMPLE EXAMPLE OF USE OF FTNOUT
2      LONG ALGEBRAIC (X:10,Y:10) F
3      ALTRAN FTNOUT
4      OPTS(201,72) # FTNOUT REQUIRES A LINE LENGTH OF 72
5      F = EXPAND( (X+2*Y+1000000)**3 )
6      WRITE F # PRINT F
7      WRITE (25) "      FUNCTION F(X,Y)",
8                "C EXAMPLE PROG WRITTEN WITH FTNOUT." ,
9                F,
10             "      RETURN", "      END"
11     # WE HAVE WRITTEN A SIMPLE PROGRAM ON UNIT 25, NOW WE INVOKE FTNOUT TO
12     # THIS ALTRAN OUTPUT TO LEGAL FORTRAN.
13     FTNOUT
14     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 +
3000000000000000*X + 8*Y**3 + 12000000*Y**2 + 6000000000000*Y +
10000_0000000000000000

```

*** NORMAL RETURN FROM MAIN PROCEDURE

*** RUN STATISTICS

.046 SECONDS ELAPSED
 14 DIGITS IN SHORT INTEGERS
 20 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

$X^{**3} + 6*X^{**2}*Y + 3000000*X^{**2} + 12*X*Y^{**2} + 12000000*X*Y +$
 $30000000000000*X + 8*Y^{**3} + 12000000*Y^{**2} + 60000000000000*Y +$
 10000_00000000000000

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 $30000000000000*X + 8*Y^{**3} + 12000000*Y^{**2} + 60000000000000*Y +$
 2 $10000000000000000E4$
 RETURN
 END

MFY NOS/BE 1.2 KAFB 009 MFY 03/27/78
 FLCH=314000 NXCH=314000 FLEC=1720K NXEC=0750K

20.52.14.EDR00K7 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,SYISIN)
 20.52.17.ALTRAN(SYISIN)
 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,L60.
 20.52.26.XEQ.
 20.52.35. EXIT
 20.52.35.REWIND(TAPE25,TAPE27)
 20.52.36.COPYSDF(TAPE25,OUTPUT)
 20.52.37.COPYSDF(TAPE27,OUTPUT)
 20.52.37.OP 00001024 WORDS - FILE OUTPUT , DC 40
 20.52.37.NS 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.CH 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

```

1      PROCEDURE MAIN
2      ALGEBRAIC (X;20) TCHEBY,T
3      ALTRAN TCHEBY
4      INTEGER N
5      OPTS(101,1)
6      DO N = 0,3
7      T = TCHEBY(N,X)
8      WRITE T
9      DOEND
10     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

```

1      PROCEDURE TCHEBY(N,X)
2      ALGEBRAIC TCHEBY,X
3      STATIC X
4      INTEGER N
5      IF(N .EQ. 0) RETURN (1)
6      IF(N .EQ. 1) RETURN (X)
7      RETURN ( 2*X*TCHEBY(N-1) - TCHEBY(N-2) )
8      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				

ALPHAN VERSION 1 LEVEL 9

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

ALPHAN VERSION 1 LEVEL 9

PROCEDURE TCHBYIN, 11
 ALPHANIC TCHBYIN, 11
 STATIC F
 INTERM A
 WITH EQ. 03 WITHIN (1)
 WITH EQ. 13 RETURN (2)
 RETURN (2)*TCHBYIN(1) - TCHBYIN(1)
 END

NAMELISTNAME USE TYPE STAGE FREE CLASS SCOPE RB LAY ADDR

NAMELISTNAME	USE	TYPE	STAGE	FREE	CLASS	SCOPE	RB	LAY	ADDR
MAIN	1	INT							
PROC	2	ALG							
INT	3	INT							
DATA	4	INT							
LONG	5	INT							
LONG	6	INT							
LONG	7	INT							
LONG	8	INT							

MFY NOS/BE 1.2 KAFB 009 MFY 03/27/78
 FLCN=314000 MXCN=314000 FLEC=1720K MXEC=0750K

20.49.41.EDROOKS 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,SYISIN)
 20.49.52.ALTRAN(SYISIN)
 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(L60)
 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.CH 67.549 KUS. .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

```

1      PROCEDURE MAIN # SIMPLE EXAMPLE OF USE OF FTNOUT
2      LONG ALGEBRAIC (X:10,Y:10) F
3      ALTRAN FTNOUT
4      OPTS(201,72) # FTNOUT REQUIRES A LINE LENGTH OF 72
5      F = EXPAND( (X+2*Y+1000000)**3 )
6      WRITE F # PRINT F
7      WRITE (25) "      FUNCTION F(X,Y)",
8                "C EXAMPLE PROG WRITTEN WITH FTNOUT." ,
9                F,
10             "      RETURN", "      END"
11     # WE HAVE WRITTEN A SIMPLE PROGRAM ON UNIT 25, NOW WE INVOKE FTNOUT TO
12     # THIS ALTRAN OUTPUT TO LEGAL FORTRAN.
13     FTNOUT
14     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

$$\begin{aligned}
 & X**3 + 6*X**2*Y + 3000000*X**2 + 12*X*Y**2 + 12000000*X*Y + \\
 & 30000000000000*X + 8*Y**3 + 12000000*Y**2 + 6000000000000*Y + \\
 & 10000_00000000000000
 \end{aligned}$$

*** 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

$X^{**3} + 6*X^{**2}*Y + 3000000*X^{**2} + 12*X*Y^{**2} + 12000000*X*Y +$
 $30000000000000*X + 8*Y^{**3} + 12000000*Y^{**2} + 6000000000000*Y +$
 10000_00000000000000

RETURN

END

ALTRAN VERSION 1 LEVEL 9

```

1      PROCEDURE MAIN
2      ALGEBRAIC (X:20) TCHEBY,T
3      ALTRAN TCHEBY
4      INTEGER N
5      OPTS(101,1)
6      DO N = 0,3
7      T = TCHEBY(N,X)
8      WRITE T
9      DOEND
10     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

```

1      PROCEDURE TCHEBY(N,X)
2      ALGEBRAIC TCHEBY,X
3      STATIC X
4      INTEGER N
5      IF(N .EQ. 0) RETURN (1)
6      IF(N .EQ. 1) RETURN (X)
7      RETURN ( 2*X*TCHEBY(N-1) - TCHEBY(N-2) )
8      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.EDROOK4 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,SYISIN)
 20.48.29.ALTRAN(SYISIN)
 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(L60)
 20.48.41.XEQ.
 20.48.52. EXIT
 20.48.52.EXIT(S)
 20.48.52.DP 00000640 WORDS - FILE OUTPUT , DC 40
 20.48.52.HS 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.CH 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