PUFFT USER'S GUIDE

Michael D. Shapiro

Purdue University Computing Center

February 1971

Table of Contents

INTRODUCTION

PUFFT is a very fast translator for the FORTRAN IV language described in the IBM document C28-6390, Version 13 FORTRAN IV language. It is a "one-pass compile-and-go" system which reads a FORTRAN IV program and compiles it into machine language written directly into the IBM 7094 computer memory.

The technique used in PUFFT necessitates introducing a few minor differences from the IBM FORTRAN IV. Most of the differences should not concern the average user at all. The most important difference concerns the ordering of certain statements. For example, in programs using DIMENSION, COMMON, and EQUIVALENCE statements, all the DIMENSION statements must appear before all the COMMON statements, which in turn must appear before all the EQUIVALENCE statements. The section of this manual on sequencing rules contains full details. PUFFT detects any violation of the sequencing rules and prints an error message on the program listing.

PUFFT accepts only FORTRAN IV source decks. It neither produces nor accepts binary decks.

PROGRAM SIZE

PUFFT is designed to handle short to medium length programs. It is difficult to give exact size limitations, but a good rule of thumb is that PUFFT jobs should contain no more than about 1500 source language statements and should use no more than 3000 words of storage for arrays. If a program exceeds the storage limitation of PUFFT, a diagnostic message is written, and execution is deleted.

SEQUENCING RULES

A. The following statements, if they appear in a program, must appear before any executable statements and in the order indicated:

1. Type and DIMENSION.

   Type includes COMPLEX, DOUBLE PRECISION, INTEGER, REAL, and LOGICAL. There is no ordering requirement among the type and DIMENSION statements.

2. COMMON.

   Labelled and blank COMMON may be intermixed according to FORTRAN IV rules.

3. EQUIVALENCE.

4. DATA.

   This refers to the DATA statement as defined in the FORTRAN IV manual.

5. Arithmetic statement function definitions.

B. The statements ENTRY, RETURN, EXTERNAL, and FORMAT are included among the executable statements for the purposes of these sequencing rules.

C. Only one of the statements FUNCTION, SUBROUTINE, and BLOCK DATA may appear in any one subprogram. Such a statement must be the first statement in the subprogram.

D. The statement END must be the last statement of each subprogram. It is translated as a RETURN in subroutines or functions and as a STOP in the main program.

E. A BLOCK DATA subprogram that enters data into a block of labelled COMMON must appear ahead of any other subprogram in which that block of labelled COMMON is declared.

F. A NAMELIST statement, if it is used, must appear among the executable statements of a program, and must precede any use of the name being defined.

INPUT/OUTPUT

PUFFT uses the standard tape assignments of Version 13 FORTRAN IV. Logical unit 5 is the system input tape and unit 6 is the system output tape. No other units are available.

READ n, L and PRINT n, L use the system input and output units and are the preferred statement forms:

READ n, L is equivalent to READ(5,n)L
PRINT n, L is equivalent to WRITE(6,n)L

PUFFT accepts quote marks around Hollerith fields in FORMAT statements. During compilation the field is translated into a standard Hollerith field. For example:

'BCD DATA' is translated to: 008HBCD DATA

The quote mark appears on most keypunches as either a quote mark (') or at sign (@). It is the 4-8 punch combination. Quote marks are not permitted in formats read in at execution time or created with DATA statements.

SYSTEM FUNCTIONS AND SUBROUTINES

The PUFFT system includes all the standard FORTRAN IV subroutines and functions mentioned in Chapter 5 of the Version 13 FORTRAN IV manual, except ERF, GAMMA and ALGAMA. PUFFT includes the built-in functions of figure 8 of that manual, the mathematical subroutines of figure 9 of that manual, and subroutines such as DUMP and PDUMP mentioned elsewhere in that chapter. DUMP and PDUMP in particular are especially useful for program debugging.

PUFFT contains the three machine dependent functions AND, OR, and COMPL, as well as corresponding integer functions IAND, IOR, and ICOMPL, documented in Appendix H of the IBM manual.

Two subroutines provide date and time information:

```
CALL DATE(A)
```

returns to A the current date (month, day, year) as a BCD word:

```
MMDDYY
```

```
CALL CLOCK(A)
```

returns to A the current time (hours, minutes, seconds) as a BCD word:

```
HHMMSS
```

For example, December 11, 1970 at 7:46:24 pm is:

```
121170   194624
```

Note that 24 hour time is used.

Another PUFFT system function is RECDIS, a random number generator. It is designed to supply pseudo-random floating point numbers with a rectangular distribution in the interval (0,1). Successive numbers in the sequence are supplied as the values of the function RECDIS(X). X is a dummy variable, and any variable name may be used in its place. Thus the statement A = RECDIS(X) places the next random number in A. The statement sequence:

```
  DO 5 I=1, 1000
5 A(I) = RECDIS(X)
```

generates a table of 1000 random numbers in the array A.

The random numbers are generated by multiplying a number called RANDOM by the constant 5**15 (30,517,578,125). The result is a new value of RANDOM which is then used as the multiplier in generating the next random number. The system initially sets RANDOM to the same value as the multiplying constant (5**15), but the user may select a different value of the initial multiplier by the statement:

```
CALL STORNM(integer)
```

where "integer" is any positive odd integer less than 2**35, and greater than one. The current value of the multiplier RANDOM may be retrieved by the statement:

```
CALL GETNM(L)
```

which places into L the value of RANDOM that will next be used as a multiplier.

SPECIAL FEATURES OF PUFFT

A. Subscript Checking

A standard option of PUFFT compilation is the generation of code to check the values taken on by subscripts during execution of the object program. Each place a subscripted array reference is made in the source program, code is generated to check the values of the subscripts against the dimensions declared for the array. If a subscript is outside its declared range, a diagnostic is printed and execution is terminated.

Actually, only if the final address computed for an array element lies outside the bounds of the array is an error detected. Thus a single subscript in a multiply subscripted array may be greater than its declared maximum without generating an error. The inclusion of the subscript checking feature adds one cell of object code and causes several extra instructions to be executed for each subscripted array reference in the program. For most programs, however, the checking adds a valuable debugging aid.

B. Address Checking

Several other types of address checking are also normally made by the PUFFT system. These provide checking on such items as incorrect number of arguments on function and subroutine calls, errors in specifying dummy routines, and attempts to store data over code.

C. Resident Subprogram Library

A library of numerical analysis and other routines is maintained with the PUFFT system. These routines are taken from SHARE, from the Purdue FORTRAN subroutine library, from the library of the Nuclear Division of Union Carbide Corporation at Oak Ridge, Tennessee, and from several other sources. A list of available routines (LO-PUFTLIB) and their descriptions may be obtained at the Document Station.

No special action is required to use these subroutines and functions. After compiling the user's source deck, PUFFT references the library for any undefined subprogram name. Any time it finds one, the message:

```
**O** name COMPILED FROM LIBRARY
```

is written, giving the name of the routine. No source listing or map is produced for these routines, although the object code listing, if requested, will include their object code.

The library is fluid, and other FORTRAN IV routines, such as special routines required for class assignments, may be included by arrangement with the Computing Center Librarian.

D. Hollerith Constants

As indicated in the section on input/output, Hollerith strings enclosed in quote marks (4-8 punch) may be used in format statements. Hollerith constants may also be enclosed in quote marks in the other places they are valid: DATA statement values and parameters in subroutine CALL statements:

'THIS IS'

is the same as

7HTHIS IS

E. ANSI Standard FORTRAN Compatibility

The PUFFT compiler meets or exceeds the American National Standards Institute (ANSI) Fortran Standard (X3.9-1966), with the following exceptions:

1. The G-specification in input/output format lists is not supported.

2. The real and double precision constant forms:

   nnnnEee   nnnnDee

   where nnnn is a string of digits, possibly preceded by a sign, and ee is any valid exponent, are not supported. PUFFT requires that there be a decimal point somewhere before the E or D.

Variations from the ANSI FORTRAN Standard within the rules of IBM 7094 FORTRAN IV, as modified by this guide, are not indicated.

COMPILATION AND EXECUTION OUTPUT

A PUFFT run normally produces a listing of the source language program and of all diagnostic messages produced during the run. As discussed in the section on control cards, the source listing may be totally or partially suppressed, but the diagnostic mess-ges may never be suppressed.

PUFFT provides many diagnostic messages in response to source language errors, and some in response to run time errors. A severity code is associated with each source language diagnostic message. Messages with severity code 0 are comments to the programmer and do not normally mean an error has occurred. Messages with severity codes 1 and 2 are warnings, and messages with severity code 3 indicate fatal errors that result in deletion or termination of execution.

To the right of each line of the source language listing is an octal number that corresponds to an address in the object program listing when the LIST option is invoked. For those source program language statements for which code is generated, this address is the starting address of that code. Because of the compiling technique, the first executable statement following a DATA statement normally has an incorrect address. The address of the first DATA statement in the program is the correct address for this statement.

2. MAP

This option causes the name and absolute address of all variables and arrays to be listed after each subprogram, and the names and starting addresses of all subprogram and common blocks to be listed after the last subprogram. The option is invoked by the LIST and DUMP options.

3. LIST

This option produces a full listing of the object code and causes the MAP option to be invoked. The listing has mnemonic and octal instructions and octal addresses.

The correspondence between source statements and object code produced is discussed in the section on PUFFT output. The use of this option increases processing time significantly, and it should be used only when needed.

4. DUMP=Oct1,Oct2,C

This option causes an area of computer memory to be dumped if the program is terminated for any reason other than a STOP, CALL EXIT, or CALL DUMP statement being executed or an end-of-file being read. The area of memory dumped is between the octal locations "Oct1" and "Oct2" inclusive, and the mode of the dump is determined from "C" as follows:

C=0	Octal
C=1	Floating point
C=2	Integer
C=3	Octal with mnemonic instructions

The dump always produces full lines that start with a cell whose octal address ends with "0". The DUMP option also causes the MAP option to be invoked.

5. TABLE=N

This option, used with an integer N from one through six, permits the size of the PUFFT name table to be altered to 2**(N+7) words. If the option is not present, the table size is 4096 (N=5).

An increase in the size of the name table may permit PUFFT to run a program that otherwise gets the name-table-overflow diagnostic. It also increases the amount of space available for blank common storage. An increase in the table size, however, decreases the room available for the compiled program. Conversely, a decrease in the size of the name table may permit PUFFT to run a program that otherwise runs out of space.

6. NOCHK

This option eliminates execution time checking of subscripts and other addresses discussed under special features of PUFFT.

C. The $OPTIONS Card

If the first form of the PUFFT card (without options) is used, a $OPTIONS card may be used to specify options, if desired. The format of the card is:

```
Col. 1
     $OPTIONS( options )
```

The list of options is the same as indicated for the PUFFT card.

If options are specified on the PUFFT card, the system generates a $PUFFT card. Only one $PUFFT or $OPTIONS card is permitted for each job.

D. The $LIST and $UNLIST Cards

These control cards permit the user to control listing of the source program. The $UNLIST card suppresses listing of the source program until the next $LIST card is encountered. Error messages will not be suppressed, nor do these cards in any way affect the listing of the object program produced by the LIST option on the PUFFT card. Unless the $UNLIST card is used, the entire source program is listed. These cards may appear anywhere in the input program deck.

E. Other $-Control Cards

The PUFFT system considers any card with a $ in column 1 to be a control card. This is for compatibility with other IBM 7094 systems. All control cards other than those recognized are ignored by PUFFT, with a warning message.

DECK STRUCTURE

Decks to be run under the PUFFT system are set up in the following order:

1. The job card.
2. The PUFFT card.
3. 7/8/9 end-of-record (EOR) card. This card has 7, 8, and 9 rows multipunched in column one.

(4) The $OPTIONS card, if used.

5. The FORTRAN IV program, following PUFFT rules.

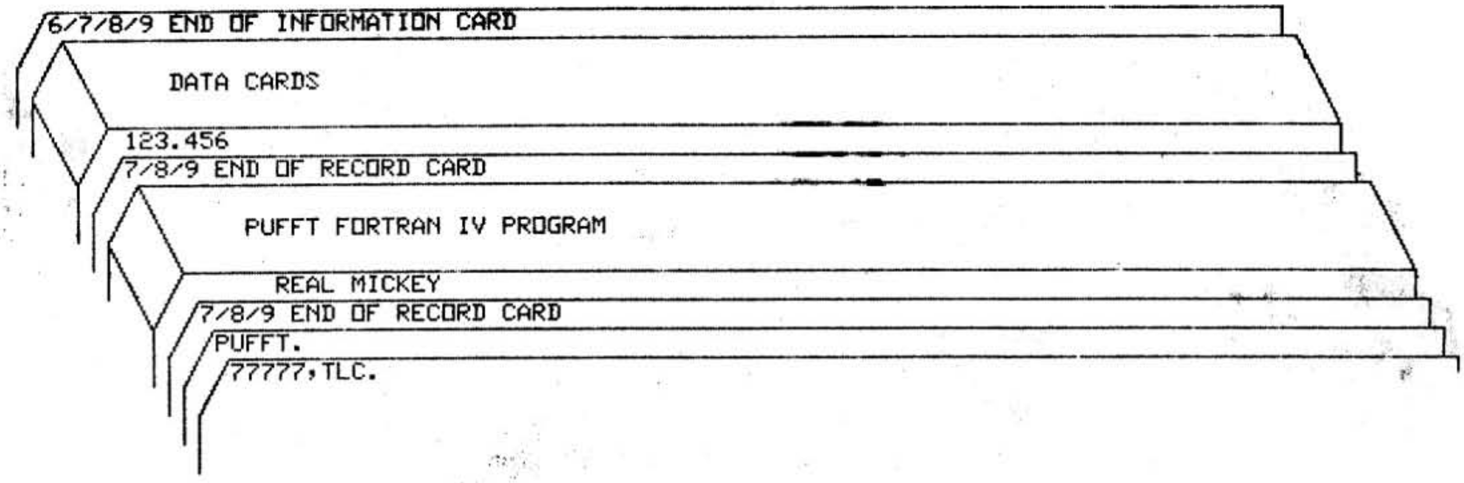
(6) Another 7/8/9 EOR card if there are any data cards.

(7) Data cards, if any.

8. 6/7/8/9 end-of-information (EOI) card. This card has 6, 7, 8, and 9 rows multipunched in column one. It must be the last card of the deck.

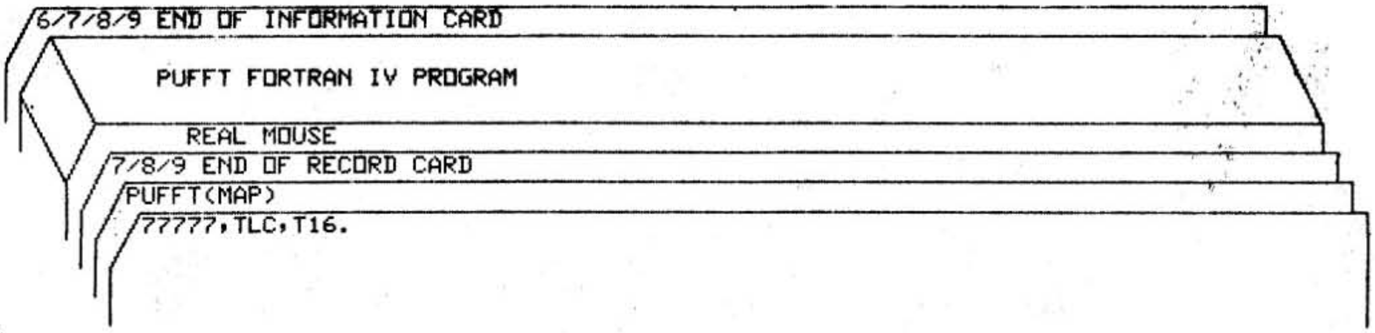
A number in parentheses indicates an optional item. All others are required.

Example 1. Standard PUFFT job using default time limit and standard options:



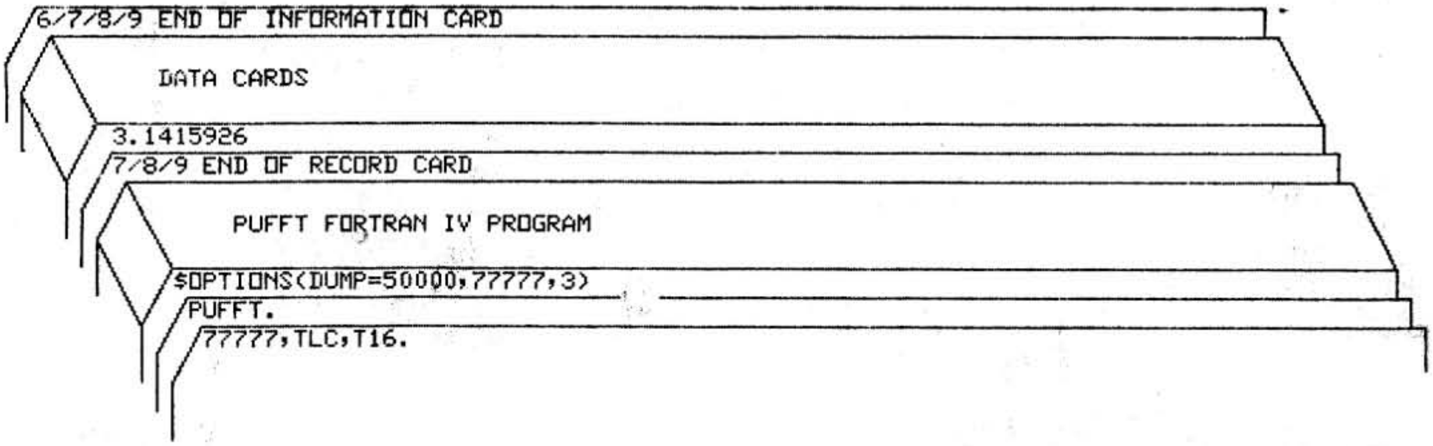
```
6/7/8/9 END OF INFORMATION CARD
      DATA CARDS
123.456
7/8/9 END OF RECORD CARD
      PUFFT FORTRAN IV PROGRAM
      REAL MICKEY
7/8/9 END OF RECORD CARD
PUFFT.
77777,TLC.
```

Example 2. Using a 16 second time limit and the MAP option, without data cards:



```
6/7/8/9 END OF INFORMATION CARD
      PUFFT FORTRAN IV PROGRAM
      REAL MOUSE
7/8/9 END OF RECORD CARD
PUFFT(MAP)
77777,TLC,T16.
```

Example 3. Using a 16 second time limit and the DUMP option specified on a $OPTIONS card:



```
6/7/8/9 END OF INFORMATION CARD
      DATA CARDS
3.1415926
7/8/9 END OF RECORD CARD
      PUFFT FORTRAN IV PROGRAM
$OPTIONS(DUMP=50000,77777,3)
PUFFT.
77777,TLC,T16.
```