



**AUSTRALIAN ATOMIC ENERGY COMMISSION  
RESEARCH ESTABLISHMENT  
LUCAS HEIGHTS**

**AAEC FACILITIES TO AID IN THE DEVELOPMENT  
OF FORTRAN PROGRAMS**

by

**S. G. JOHNSON**

**P. L. SANGER**

**June 1974**

**ISBN 0 642 99637 7**



AUSTRALIAN ATOMIC ENERGY COMMISSION

RESEARCH ESTABLISHMENT

LUCAS HEIGHTS

AAEC FACILITIES TO AID IN THE DEVELOPMENT  
OF FORTRAN PROGRAMS

by

S.G. JOHNSON

P.L. SANGER

ABSTRACT

Two subroutines, DBGON and DBGOFF, designed to give users more control over the debug options supplied with the IBM360 FORTRAN IV [G] compiler are described. Examples of the use of these subroutines in the debugging of a FORTRAN program are presented.

Macros developed to simulate FORTRAN I/O in Assembler programs are also described, and their value to programmers converting FORTRAN programs into Assembler language routines is demonstrated.

**National Library of Australia card number and ISBN 0 642 99637 7**

**The following descriptors have been selected from the INIS Thesaurus to describe the subject content of this report for information retrieval purposes. For further details please refer to IAEA-INIS-12 (INIS: Manual for Indexing) and IAEA-INIS-13 (INIS: Thesaurus) published in Vienna by the International Atomic Energy Agency.**

**COMPUTER CODES; FORTRAN; IBM COMPUTERS; PROGRAMMING**

## CONTENTS

	Page
1. INTRODUCTION	1
2. CONTROL OF FORTRAN DEBUG OPTIONS	1
2.1 General Discussion	1
2.2 Monitoring Calls to IHCDEBUG	2
2.3 Example of DBGON, DBGOFF Usage	3
3. FORTRAN I/O IN ASSEMBLER PROGRAMS	4
3.1 General Discussion	4
3.2 Macro Expansions	6
3.3 Macro Extensions	7
3.4 JCL Requirements	8
3.5 Example of Macro Usage	9
4. CONCLUSIONS	10
5. REFERENCES	10

Table 1	IHCDEBUG Displacement Table
Table 2	IHCECOMH Displacement Table
Table 3	FORTRAN I/O Parameter Lists
Table 4	Example of Use of Macros

Appendix A	Assembly Listing of the AEDBUG Routine
Appendix B	Using the Debug INIT Option
Appendix C	Using the DBGON and DBGOFF Subroutines
Appendix D	Using the DBGON and DBGOFF Subroutines in Debug Packets
Appendix E	Using the DISPLAY Statement
Appendix F	Macro Listings
Appendix G	Using the Macros in an Assembler Language Subroutine



## 1. INTRODUCTION

The debug facility available with the IBM360 FORTRAN IV [G] compiler allows the user to locate errors in a FORTRAN program (IBM 1971). The facility consists of debug packets that are included with the FORTRAN source statements; these indicate the desired debugging operations for a single main program or subprogram.

Unfortunately, when simple options such as DEBUG INIT or DEBUG SUBCHK that apply to the whole single program unit are used, they often produce far more output than expected by the user and waste large amounts of computer time. The fact that the debug output is not formatted also adds to this problem. Use of the DISPLAY statement in debug packets can significantly reduce the debug output. However, this often requires the user to insert additional statement numbers into his program and the problem of selecting unique statement numbers arises here, especially for large single program units. The debug packets also allow statement number tracing to be controlled, but provide no way of restricting the SUBCHK option. The DEBON and DBGOFF subroutines, discussed in Section 2, give the user control over all of the debug options within a single program unit, and allow him to reduce the amount of debug output produced and the computer time used by even the simplest of debug options.

When a FORTRAN programmer codes part of his program in IBM360 Assembler language, he often faces difficulties in debugging his Assembler language routine. One of the problems commonly encountered arises when the programmer wishes to compare the floating point numbers produced by his Assembler routine with the E-, D- or F-format output produced by his FORTRAN program. To cope with this type of problem, a set of macros was developed to simulate FORTRAN I/O in Assembler programs (Johnson 1972) and these, plus extensions to cope with arguments passed via subroutine CALLs, are discussed in Section 3.

## 2. CONTROL OF FORTRAN DEBUG OPTIONS

### 2.1 General Discussion

The IBM360 FORTRAN IV [G] debug facility consists of a DEBUG specification statement, an AT debug packet identification statement, and three executable statements that are used to indicate the desired debugging operations for a single program unit in source language. These statements result in code being generated to produce non-standard calls to IHCDEBUG, the FORTRAN object-time debug facility support routine, followed by the appropriate parameter list. Entry into the IHCDEBUG occurs via the DEBUG# transfer table, which

consists of unconditional branches to various routines within IHDEBUG; this transfer is shown in Table 1. The linkage sequence used is:

```
L      15,=V(DEBUG#)
CNOP   0,4
BAL    14,N(15)
```

where N is the displacement from DEBUG#.

The TRACE ON, TRACE OFF and DISPLAY statements cause code to be generated to produce debug output for part of a single program unit. The DEBUG statement options INIT and SUBCHK, however, cause code to be generated throughout the whole program unit to produce debug output or to carry out subscript checking. In the case of the INIT option, three possible calls to IHDEBUG can be generated depending on whether an assignment occurs for a scalar variable, an array element or to a full array by means of an input statement or a subprogram call (IBM 1968).

## 2.2 Monitoring Calls to IHDEBUG

To control the effect of the debug options, an Assembler routine, AEDBUG, was written to monitor all calls to IHDEBUG. This routine was given the entry point DEBUG#, while the entry point of IHDEBUG was changed to DEBUG. FORTRAN calls to the subroutines DBGON and DBOFF, which are also part of the AEDBUG routine, are used to indicate which of the debug calls are passed through to IHDEBUG or are nullified by immediately returning to the calling FORTRAN program.

Calls to the DBGON and DBOFF subroutines take the form:

```
CALL {DBGON
      DBOFF} ('option', ....., 'option')
```

where option may be any of the following:

```
TRACE
SUBTRACE
SUBCHK
DISPLAY
INIT
INITSCLR - indicates changes only in scalar variables.
INITARIT - indicates changes only in array elements.
INITARAY - indicates changes only in entire arrays.
ALL      - implies all of the above options.
```

By inserting appropriate calls to DBGON and DBOFF in the FORTRAN program, the user can control any of the normal debug options with the extra advantage

of being able to control the printing out of changes in scalar variables, array elements or entire arrays using the INITSLR, INITARIT or INITARAY options. An Assembly listing of the AEDBUG routine is shown in Appendix A.

### 2.3 Example of DBGON, DBGOFF Usage

To show how the DBGON and DBGOFF subroutines might be used to debug a FORTRAN program, a general non-linear regression subroutine, NLRP, was written with a corresponding main line and second subroutine, FUN, set up to solve a particular three-parameter problem. To check the parameter shifts that were calculated for each of five least squares cycles, the DEBUG INIT statement was added to the end of the NLRP subroutine in the standard way as shown in Appendix B and this resulted in 1,600 lines of printer output being produced.

However, by inserting CALL DBGOFF('INIT') as statement 14, CALL DBGON('INITARIT') as statement 23, CALL DBGOFF('INITARIT') as statement 69, and by including the card INCLUDE SYSLIB(AEDBUG) as linkage editor input, the output was reduced to 600 lines. As shown in Appendix C, only changes in array elements used in the calculation of the parameter shifts were printed out and this provided enough information for a thorough step-by-step check of the parameter shifts calculated each cycle. Alternatively, the subroutine calls can be set up as debug packets as shown in Appendix D but this required the insertion of CONTINUE statements with statement numbers 990, 991, 992 as statements 14, 23 and 69 respectively.

A substantial reduction in debug output can also be obtained by using the DISPLAY statement. However, this statement cannot be used to list individual array elements and thus is better placed at the end of a program loop to show changes in array elements within a program loop. By adding debug packets to the end of the subroutine NLRP and inserting CONTINUE statements with statement numbers 990, 991, 992 as statements 30, 57 and 67 respectively, as shown in Appendix E, the resultant output is reduced to 100 lines. The output produced in this case was quite adequate for the latter stages of testing the subroutine since it showed the changes in array variables at the end of the different loops required to calculate the parameter shifts. One interesting problem encountered with use of the DISPLAY statement occurred when the array P was referred to in a debug packet. If the DISPLAY statement appears in a subroutine, the item name may not be a dummy argument of the subroutine statement - this occurs because the DISPLAY statement makes use of the namelist option (IBM 1971).

### 3. FORTRAN I/O IN ASSEMBLER PROGRAMS

#### 3.1 General Discussion

When a FORTRAN programmer codes part of his program in IBM360 Assembler language, he often wants to debug the Assembler routine by comparing his results with the E-, D-, F- or I-format output produced by his FORTRAN program. To cope with this problem, a set of macros was developed to simulate FORTRAN I/O in Assembler programs (Johnson 1972) and, by inserting these macros at various points within the Assembler routine, it is possible, for example, to generate output similar to that produced by the FORTRAN DISPLAY debug options. Once the Assembler routine has been completely debugged, the programmer can then simply remove the macros and reassemble the program to produce the required routine.

The macros provided were designed to follow the syntax of the corresponding FORTRAN statements, as closely as possible, within the framework of the basic macro format:

NAME	OPERATION	OPERAND
a symbol or blank	mnemonic operation code	0-200 operands separated by commas

The contents of each field and the significance of the operands for each of the initial macro definitions are described below.

#### (a) AREAD

label	AREAD	(unit, format, end, err), list
-------	-------	--------------------------------

where:

- label - is optional and is a symbol used to identify the statement. It must consist of eight or fewer characters, begin in the first column, and contain no blanks.
- unit - is an unsigned integer constant or an F-type variable name representing the data set reference number.
- format - is optional and is the 'statement number' of the AFORMAT statement describing the format of the record(s) being read. It must consist of five or less digits and/or letters.
- end - is optional and is the label of the statement to which transfer is made upon encountering the end of the data set. It corresponds to the FORTRAN END=... option.

`err` - is optional and is the label of the statement to which transfer is made upon encountering an error condition in reading the data. It corresponds to the FORTRAN `ERR=...` option.

Unlike the FORTRAN convention the 'end' subparameter, if present, must be the third item and the 'err' subparameter, if present, must be the fourth item inside the parentheses. Thus, when either the 'format' or 'end' subparameter is omitted and some following subparameter is present, a comma must be coded to replace the absent subparameter. For example, the following would preserve the positions of the subparameters: `(unit,,end)`, `(unit,,,err)`.

`list` - is optional and is an I/O list. The components of the list are separated by commas and are of two types, (a) symbols and (b) array symbols. An array symbol has the form `(name,n)`, where `name` is a symbol defining the start of the array and `n` is an unsigned integer equal to the number of array elements. The array symbol is used to indicate `n` consecutive locations starting at `name`. As in FORTRAN, the elements of an array have the same length and type.

(b) AWRITE

label	AWRITE	(unit, format), list
-------	--------	----------------------

where `label`, `unit`, `format` and `list` are the same as for (a).

(c) AFORMAT

format	AFORMAT	'(C <sub>1</sub> , C <sub>2</sub> , ..., C <sub>n</sub> )'
--------	---------	--

where:

`format` - is a 'statement number' consisting of five or less digits or letters in any combination, starting in the first column and containing no blanks, and

`C1, C2, ..., Cn` - are FORTRAN format codes (IBM 1971).

Each single apostrophe or ampersand appearing within a format code must be represented as a pair. For example the FORTRAN statement

```
19 FORMAT (' AMPERSAND=&')
```

would be translated as

```
19 AFORMAT>('' AMPERSAND=&&'')
```

Alternatively, the format may be coded as

```
FMT19 DC C>('' AMPERSAND=&&'')
```

this being the statement generated by the macro extension (see Section 3.2).

Note that, because the AFORMAT instruction statement generates a DC statement and the return from IHCECOMH (see Section 3.2) is normally via register 14 to the statement following the AREAD or AWRITE instruction statement, an AFORMAT instruction statement must not immediately follow an AREAD or AWRITE instruction statement.

(d) ABACKSP, AREWIND, AENDFILE

label	ABACKSP	unit
label	AREWIND	unit
label	AENDFILE	unit

where label and unit are as previously defined.

### 3.2 Macro Expansions

The object codes for the statements generated by the AREAD, AWRITE, ABACKSP, AREWIND and AENDFILE macro instruction statements are the same as those produced by the FORTRAN compilers (G and H) for the READ, WRITE, BACKSPACE, REWIND and END FILE statements respectively. Each sequence of code is a non-standard call to IHCECOMH, the FORTRAN object-time input/output processor, followed by a list of parameters. If the Extended Error Message facility has not been specified, IHCECOMH is replaced by IHCFOMH. Entry into IHCECOMH is via the IBCOM# transfer table, which consists of unconditional branches to various routines within IHCECOMH. This table has the format given in Table 2.

The linkage sequence is then

```
CNOP  0,4
L      15,=V(IBCOM#)
BAL    14,N(15)
```

where N is the displacement from IBCOM#.

The contents of the parameter lists also vary with the routine as shown in Table 3. List items generated by the AREAD and AWRITE macros must specify the length, type and address of each operand. To provide this information, the operand was assumed to be defined using a DS or DC Assembler statement and the type attribute of a macro instruction operand used to determine the item type. The length of the item was taken to be 4, 8, 4 or 2, depending on whether the item type was E, D, F or H respectively. This,

of course, means that the list items have to be defined very carefully so that the correct information is generated by the macro expansions. Array list items must also specify the number of elements in the array; for this reason, operands of the form (name,n), where n is the number of array elements, are used for array items.

The AFORMAT macro instruction statement generates a DC instruction statement whose operand field is formed by preceding the operand of the macro instruction statement with the letter C, and whose name field is formed by concatenating the 'statement number' in the name field of the macro instruction statement with the letters FMT. It is the programmer's responsibility to ensure that the symbolic name thus generated (FMTformat) is unique.

Table 4 shows a sample routine coded in FORTRAN and the same routine coded in Assembler language with all I/O operations performed by macros.

### 3.3 Macro Extensions

When the above macros were first used as a debugging aid in testing the conversion of a FORTRAN subroutine to Assembler language, the problem of handling arguments passed to the subroutine was encountered. These arguments are defined in the calling program and it is, therefore, not possible to use the type attribute to determine item type and length as discussed in Section 3.2. However, the subroutine can save the values or addresses of passed arguments and two new macros, SREAD and SWRITE, can be used to handle this case.

The parameters generated for simple list items set up an S-type address constant for the operand; this means that the values of these items must be saved in the subroutine if they are to be accessed by the macros. The first character of the macro operand is then used to indicate the item type by making use of the FORTRAN variable naming conventions - if the first character is a letter in the range I-N then the item is taken to be INTEGER\*4 (type = F, length = 4); otherwise, it is taken to be REAL\*4 (type = E, length = 4). A § character preceding the item name is used to indicate a variable of non-standard length, so that:

NUMB indicates an INTEGER\*4 variable,  
 §NUMB indicates an INTEGER\*2 variable (type = H, length = 2),  
 ANSWER indicates a REAL\*4 variable,  
 and §ANSWER indicates a REAL\*8 variable (type = D, length = 8).

Parameters generated for array list items include a fullword address constant and it is therefore sufficient just to save the address of the array

item to be able to use the macros. In this case, a # character should precede the array item name to indicate that the operand contains the address, rather than the value, of the first array element to be referenced. The FORTRAN-like naming convention is again used to indicate the array item type and length, so that:

- (#NUMB,3) - indicates an array item of 3 elements, where each element is INTEGER\*4 and NUMB is a full word containing the address of the first element to be referenced;
- (#%NUMB,3) - as above, but each element is INTEGER\*2;
- (#ANSWER,6) - indicates an array item of 6 elements, where each element is REAL\*4 and ANSWER is a full word containing the address of the first element to be referenced;
- (#%ANSWER,6) - as above, but each element is REAL\*8.

The macros SREAD and SWRITE were set up to handle list items of this form and a third macro SFORMAT, which is identical to AFORMAT, was provided for completeness. All of the macros were set up to save registers 14 and 15 before the first entry to IHCECOMH, and to restore them after the last call so as to minimise the effect of inserting the macros throughout the Assembler program. Care was also taken to ensure that the macros could be used in multiple CSECT assemblies - the saving and restoring of registers 14 and 15 made this task a little complicated. A listing of all macros is given in Appendix F.

### 3.4 JCL Requirements

At the AAEC, the macros are members of AAE.MACLIB which is concatenated with the system macro library SYSL.MACLIB in the catalogued procedures ASMFCLG, ASMFCL and ASMFC. Several additional JCL (Job Control Language) cards must be included in the deck when these macros are used. The first of these is a LKED.SYSLIB DD card giving the linkage editor access to the FORTRAN system library, SYSL.FORTLIB, which contains the modules used during the I/O processing, *i.e.*

```
//LKED.SYSLIB DD DSN=SYSL.FORTLIB,DISP=SHR
```

There must also be present appropriate DD cards to describe any data set reference number used in the program.

**EXAMPLE:** The JCL required to assemble, link edit and execute an assembler language program with data set reference numbers 1, 2, 3 and 4, defined as

the card reader, the card punch, the printer and a scratch tape respectively, is:

```
//EXAMPLE EXEC ASMFCLG
//ASM.SYSIN DD *
.
.
    program
.
.
/*
//LKED.SYSLIB DD DSN=SYS1.FORTLIB,DISP=SHR
//GO.FT04F001 DD UNIT=(280,,DEFER),DISP=OLD,VOL=SER=SCRATCH,
//
    DCB=(RECFM=FB,LRECL=133,BLKSIZE=13300),LABEL=(1,NL)
//GO.FT03F001 DD SYSOUT=A
//GO.FT02F001 DD SYSOUT=B
//GO.FT01F001 DD *
.
.
    input data cards
.
/*
```

### 3.5 Example of Macro Usage

To see how the above macros can be used to debug Assembler programs, part of the FORTRAN NLRP subroutine discussed in Section 2.3 was written in Assembler language as the NLRPS subroutine. The NLRPS subroutine was set up to accept information from the NLRP subroutine and then to calculate and return the values of the parameter shifts for each least squares cycle. The Assembler coding was written to be as efficient as possible, using tight BXLE loops and optimising the information saved in the registers. The SWRITE and SFORMAT macros were inserted into the program so that they would produce debug output to be compared with the earlier FORTRAN calculations but, at the same time, have no effect on the Assembler logic.

The output of such a run is shown in Appendix G, where the two pages of debug output is similar to that produced by using the DISPLAY statement in FORTRAN debug packets. It was easy to use the macros to print out the values of variables or entire arrays at the end of a program loop, but it was not possible to print out changes in the values of individual array elements

without changing the logic of the Assembler routine.

#### 4. CONCLUSIONS

The DBGON and DBGOFF subroutines give the FORTRAN programmer greater control over the debug options provided by the IBM360 FORTRAN IV [G] compiler. They are easy to use and result in a reduction in the amount of unnecessary debug output produced and the amount of computer time used by even the simplest of debug options.

Macros developed to simulate FORTRAN I/O in Assembler programs provide a powerful debugging tool for the Assembler programmer. They are set up to be used in Assembler programs designed to be main programs or subroutines for a FORTRAN application package without interfering with the efficiency of the Assembler coding. Use of the macros to print results in FORTRAN format is much better than using Assembler language subroutines, and they give the Assembler programmer almost full FORTRAN I/O capability.

#### 5. REFERENCES

- IBM (1968) Program Logic  
 IBM System 360 Operating System FORTRAN IV [G] Compiler  
 Program Logic Manual  
 Program Number 360S-FO-520  
 (File No. S360-25(OS) Form Y28-6638-1).
- IBM (1969) System Reference Library  
 IBM System 360 Operating System Assembler Language  
 (File No. S360-21, Form C28-6514-6).
- IBM (1970) Program Logic  
 IBM System 360 Operating System FORTRAN IV [H] Compiler  
 Program Logic Manual  
 Program Number 360S-FO-500  
 (File No. S360-25(OS) Order No. GY28-6642-4).
- IBM (1971) System Reference Library  
 IBM System 360 and System 370 FORTRAN IV Language  
 (File No. S360-25 Order No. GC28-6515-8).
- Johnson, S.G. (1972) - Macros to Simulate FORTRAN I/O in Assembler Programs.  
 AAEC/E248.

TABLE 1  
IHCDEBUG DISPLACEMENT TABLE

Displacement From DEBUG#	Function of Routine
0	TRACE OPTION
+ 4	SUBTRACE ENTRY
+ 8	SUBTRACE RETURN
+12	UNIT OPTION
+16	INIT SCALAR VARIABLE
+20	INIT ARRAY ITEM
+24	INIT FULL ARRAY (READ AND SUBPROGRAM CALL)
+28	SUBCHK
+32	TRACE ON
+36	TRACE OFF
+40	DISPLAY
+44	BEFORE FIRST CALL TO IBCOM FOR I/O OPERATION
+48	AFTER LAST CALL TO IBCOM FOR I/O OPERATION

TABLE 2  
IHCECOMH DISPLACEMENT TABLE

DISPLACEMENT FROM IBCOM#	FUNCTION OF ROUTINE
0	main entry, formatted READ
+4	main entry, formatted WRITE
+8	second list item, formatted
+12	second list array, formatted
+16	final entry, end of I/O list, formatted
+20	main entry, unformatted READ
+24	main entry, unformatted WRITE
+28	second list item, unformatted
+32	second list array, unformatted
+36	final entry, end of I/O list, unformatted
+40	backspace tape
+44	rewind tape
+48	write tapemark

TABLE 3  
FORTRAN I/O PARAMETER LISTS

<p>main entry for AREAD and AWRITE, both formatted and unformatted</p>	<p>XL.4'PI',XL.4'UI',AL3(unit) AL1(01), AL3(FMTformat) ...formatted only AL4(EOFi) ...optional AL4(ERRj) ...optional</p> <p>where</p> <p>PI - =0 if neither end nor err is specified       =1 if end only is specified       =2 if err only is specified       =3 if both end and err are specified,</p> <p>UI - =0 if unit is an integer constant       =1 if unit is a variable name of type F, and ERRj and EOFi are the names of address constants generated by the macro which point to the users error and end-of-file exits. The uniqueness of such names is guaranteed by the four digit integer constants i and j, provided that the programmer himself does not create a duplicate label</p>
<p>entry for list item, both formatted and unformatted</p> <p>entry for list array, both formatted and unformatted</p>	<p>AL1(L'item),XL.4'T',XL.4'O',SL2(item)</p> <p>AL1(0),AL3(array) AL1(L'array),XL.4'T',AL.20(E)</p> <p>where</p> <p>L - is the length in bytes of the item or the array element in the I/O list,</p> <p>T - =7 for an E type item or array element       =6       D       =5       F       =4       H       =7 for any other type not already           mentioned</p> <p>and E is the number of elements in the array</p>
<p>final list entry, both formatted and unformatted</p>	<p>no parameters</p>
<p>ABACKSPACE, AREWIND and END FILE</p>	<p>XL1'UI',AL3(unit) where UI is defined as above</p>

TABLE 4  
EXAMPLE OF USE OF MACROS

PORTMAN	ASSEMBLER (I/O performed by macros)
<pre> * * * INTCER#2 A DIMENSION POINTS (250) IN = 1 1 READ (14, 19, END=10, ERR=15) A, (POINTS(1), I=1, 100) * * * WRITE (3, 29) A, TIME, EFFCY GO TO 1 10 WRITE (3, 49) * * * CALL EXIT 15 WRITE (3, 39) A * * * CALL EXIT </pre>	<pre> * * * LOOP AREAD (14, 19, END=10, LOERR), A, (POINTS, 100) * * * AWRITE (3, 007), A, TIME, EFFCY B LOOP ENDIO AWRITE (1, END) * * * B EXIT LOERR AWRITE (3, ERR), A * * * B EXIT IN DC F'1' TIME DS E EFFCY DS E POINTS DS 250Z A DS H 19 AFORMAT '(5X, I3/5 (E10.3, 5X))' OUT AFORMAT '(1H1, 6BATCH, I3, 8H RESULTS/1X, 10HMEAN DELAY TIME, 8 E14.7/1X, 10HEFFICIENCY, 6X, E14.7)' ERR AFORMAT '(1H1, 20HINPUT ERROR IN BATCH NUMBER, I3/1X, 8 20H*****)' END AFORMAT '(1H1, 20HINPUT DATA EXHAUSTED)' * * * END </pre>



APPENDIX A

ASSEMBLY LISTING OF THE AEDBUG ROUTINE

EXTERNAL SYMBOL DICTIONARY

```

SYMBOL   TYPE ID  ADDR  LENGTH LD ID
AEDBUG   SD  01  000000 000254
DEBUG    ER  02
DEBUG#   LD      000000      01
DBGOFF   LD      0000DC      01
DBGON    LD      0000F6      01
    
```

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	
000000				1	AEDBUG CSECT	
				2	EXTRN DEBUG	
				3	ENTRY DEBUG#	
				4	ENTRY DBGOFF	
				5	ENTRY DBGON	
				6	USING *.15	
000000				7	DEBUG# 0 TRACE	INTERCEPT
000004	47F0 F034		00034	8	0 SUBTREN	INTERCEPT
000008	47F0 F054		00054	9	0 SUBTREN	INTERCEPT
00000C	47F0 F064		00064	10	0 UNIT	
000010	47F0 F06C		0006C	11	0 INITSCLR	INTERCEPT
000014	47F0 F07C		0007C	12	0 INITARIT	INTERCEPT
000018	47F0 F08C		0008C	13	0 INITARAY	INTERCEPT
00001C	47F0 F09C		0009C	14	0 SUBCHK	
000020	47F0 F0AC		000AC	15	0 TRACEON	
000024	47F0 F0B4		000B4	16	0 TRACEOFF	
000028	47F0 F0BC		000BC	17	0 DISPLAY	INTERCEPT
00002C	47F0 F0CC		000CC	18	0 STARTIO	
000030	47F0 F0D4		000D4	19	0 ENDIO	
000034	9180 F21A	0021A		20	TRACE TH TFLAG,X'80'	TEST FOR ONES
000038	4780 E004		00004	21	BZ 4(0,14)	RETURN IF ZEROS
00003C	58F0 F250		00250	22	L 15,=V(DEBUG)	CONTINUE
000040	47F0 F000		00000	23	0 0(0,15)	IF BUG ON
000044	9140 F21A	0021A		24	SUBTREN TH TFLAG,X'40'	TEST FOR ONES
000048	4780 E000		00000	25	BZ 0(0,14)	RETURN IF ZEROS
00004C	58F0 F250		00250	26	L 15,=V(DEBUG)	CONTINUE
000050	47F0 F004		00004	27	0 4(0,15)	IF BUG ON
000054	9140 F21A	0021A		28	SUBTREN TH TFLAG,X'40'	TEST FOR ONES
000058	4780 E000		00000	29	BZ 0(0,14)	RETURN IF ZEROS
00005C	58F0 F250		00250	30	L 15,=V(DEBUG)	CONTINUE
000060	47F0 F008		00008	31	0 8(0,15)	IF BUG ON
000064	58F0 F250		00250	32	UNIT L 15,=V(DEBUG)	GO TO
000068	47F0 F00C		0000C	33	0 12(0,15)	IHCDEBUG
00006C	9120 F21A	0021A		34	INITSCLR TH TFLAG,X'20'	TEST FLAG FOR ONES
000070	4780 E00C		0000C	35	BZ 12(0,14)	RETURN IF ZEROS
000074	58F0 F250		00250	36	L 15,=V(DEBUG)	CONTINUE
000078	47F0 F010		00010	37	0 16(0,15)	IF BUG ON
00007C	9110 F21A	0021A		38	INITARIT TH TFLAG,X'10'	TEST FOR ONES
000080	4780 E010		00010	39	BZ 16(0,14)	RETURN IF ZEROS
000084	58F0 F250		00250	40	L 15,=V(DEBUG)	CONTINUE
000088	47F0 F014		00014	41	0 20(0,15)	IF BUG ON
00008C	9108 F21A	0021A		42	INITARAY TH TFLAG,X'08'	TEST FOR ONES
000090	4780 E014		00014	43	BZ 20(0,14)	RETURN IF ZEROS
000094	58F0 F250		00250	44	L 15,=V(DEBUG)	CONTINUE
000098	47F0 F018		00018	45	0 24(0,15)	IF BUG ON
00009C	9104 F21A	0021A		46	SUBCHK TH TFLAG,X'04'	TEST FOR ONES
0000A0	4780 E018		00018	47	BZ 16(0,14)	RETURN IF ZEROS
0000A4	58F0 F250		00250	48	L 15,=V(DEBUG)	CONTINUE
0000A8	47F0 F01C		0001C	49	0 28(0,15)	IF BUG ON
0000AC	58F0 F250		00250	50	TRACEON L 15,=V(DEBUG)	GO TO
0000B0	47F0 F020		00020	51	0 32(0,15)	IHCDEBUG
0000B4	58F0 F250		00250	52	TRACEOFF L 15,=V(DEBUG)	GO TO
0000B8	47F0 F024		00024	53	0 36(0,15)	IHCDEBUG
0000BC	9102 F21A	0021A		54	DISPLAY TH TFLAG,X'02'	TEST FOR ONES
0000C0	4780 E008		00008	55	BZ 8(0,14)	RETURN IF ZEROS

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	
0000C4	58F0 F250		00250	56	L 15,=V(DEBUG)	CONTINUE
0000C8	47F0 F028		00028	57	B 40(0,15)	IF BUG ON
0000CC	58F0 F250		00250	58	L 15,=V(DEBUG)	GO TO
0000D0	47F0 F02C		0002C	59	B 44(0,15)	IHDEBUG
0000D4	58F0 F250		00250	60	L 15,=V(DEBUG)	GO TO
0000D8	47F0 F030		00030	61	B 48(0,15)	IHDEBUG
0000DC	47FF 000C		0000C	62	USING *15	BRANCH AROUND
0000E0	07			63	BC 15,12(15)	CONSTANTS IN
0000E1	C4C2C7D6C6C640			64	DC X'7'	CALLING SEQUENCE
0000E8	90EC D00C		0000C	65	DC CL7'DBCOFF'	SAVE REGISTERS
0000EC	05C0			66	STM 14,12,12(13)	ESTABLISH BASE REGISTER
				67	BALR 12,0	
				68	DROP 15	
0000EE				69	USING *12	USE 12 AS BASE
0000EE	41A0 C096		00184	70	LA 10,FLAGOFF	BRANCH AROUND
0000F2	47F0 C01E		0010C	71	B COMMON	CONSTANTS IN
0000F6				72	USING *15	CALLING SEQUENCE
0000F6	47FF 000A		0000A	73	BC 15,10(15)	SAVE REGISTERS
0000FA	05			74	DC X'5'	ESTABLISH BASE REGISTER
0000FB	C4C2C7D6D5			75	DC CL5'DBGON'	
000100	90EC D00C		0000C	76	STM 14,12,12(13)	
000104	58C0 F152		00248	77	L 12,BASEAD	
				78	DROP 15	
0000EE				79	USING BASE,12	USE 12 AS BASE
000108	41A0 C0DE		001CC	80	LA 10,FLAGON	LOAD REG 8 WITH ADDRESS
00010C	5800 1000		00000	81	L 8,0(0,1)	WORD INCREMENT REGISTER
000110	4170 0000		00000	82	LA 7,0(0,0)	TEST FOR TRACE
000114	D504 8000	C120	00218	83	CLC 0(5,8),TCE	BRANCH TO TURN TRACE OFF OR ON
00011A	4780 A000		00000	84	BE 0(0,10)	TEST FOR SUBTRACE
00011E	D507 8000	C132	00220	85	CLC 0(8,8),SUB	BRANCH TO TURN SUBTRACE OFF/ON
000124	4780 A008		00008	86	BE 0(0,10)	TEST FOR INIT
000128	D503 8000	C13A	00228	87	CLC 0(4,8),INT	BRANCH IF NOT FOUND
00012E	4770 C066		00154	88	BNE SKIP	TEST FOR SCLR
000132	D503 8004	C13E	0022C	89	CLC 4(4,8),SCR	BRANCH IF FOUND
000138	4780 A010		00010	90	BE 16(0,10)	TEST FOR ARIT
00013C	D503 8004	C142	00230	91	CLC 4(4,8),ART	BRANCH IF FOUND
000142	4780 A018		00018	92	BE 24(0,10)	TEST FOR ARAY
000146	D503 8004	C146	00234	93	CLC 4(4,8),ARY	BRANCH IF FOUND
00014C	4780 A020		00020	94	BE 32(0,10)	BRANCH TO TURN INIT OFF OR ON
000150	47F0 A028		00028	95	B 40(0,10)	TEST FOR SUBCHK
000154	D505 8000	C14A	00238	96	CLC 0(6,8),CHK	BRANCH TO TURN SUBCHK OFF OR ON
00015A	4780 A030		00030	97	BE 48(0,10)	TEST FOR DISPLAY
00015E	D506 8000	C150	0023E	98	CLC 0(7,8),DIS	BRANCH TO TURN DISPLAY OFF OR ON
000164	4780 A038		00038	99	BE 56(0,10)	TEST FOR ALL
000168	D502 8000	C157	00245	100	CLC 0(3,8),ALL	BRANCH TO TURN ALL OFF OR ON
00016E	4780 A040		00040	101	BE 64(0,10)	TEST FOR LAST ARGUMENT
000172	1288			102	LTR 8,8	BRANCH TO END
000174	4740 C122		00210	103	BM FINISH	LOAD REGISTER 8 WITH
000178	4170 7004		00004	104	LA 7,4(0,7)	ADDRESS OF NEXT ARGUMENT
00017C	5887 1000		00000	105	L 8,0(7,1)	TEST NEXT WORD IN LIST
000180	47F0 C026		00114	106	B LOOP	TURN TRACE OFF
000184	947F C12C	0021A	00172	107	FLAGOFF NI TFLAG,X'7F'	RETURN TO LOOP
000188	47F0 C084		00172	108	B NEXT	TURN SUBTRACE OFF
00018C	948F C12C	0021A	00172	109	NI TFLAG,X'8F'	RETURN TO LOOP
000190	47F0 C084		00172	110	B NEXT	

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT	
000194	94DF C12C	0021A		111	TFLAG,X'DF'	TURN INITSCLR OFF
000198	47F0 C084	00172		112	NEXT	RETURN TO LOOP
00019C	94EF C12C	0021A		113	TFLAG,X'EF'	TURN INITARIT OFF
0001A0	47F0 C084	00172		114	NEXT	RETURN TO LOOP
0001A4	94F7 C12C	0021A		115	TFLAG,X'F7'	TURN INITARAY OFF
0001A8	47F0 C084	00172		116	NEXT	RETURN TO LOOP
0001AC	94C7 C12C	0021A		117	TFLAG,X'C7'	TURN INIT OFF
0001B0	47F0 C084	00172		118	NEXT	RETURN TO LOOP
0001B4	94FB C12C	0021A		119	TFLAG,X'FB'	TURN SUBCHK OFF
0001B8	47F0 C084	00172		120	NEXT	RETURN TO LOOP
0001BC	94FD C12C	0021A		121	TFLAG,X'FD'	TURN DISPLAY OFF
0001C0	47F0 C084	00172		122	NEXT	RETURN TO LOOP
0001C4	9400 C12C	0021A		123	TFLAG,X'00'	TURN ALL
0001C8	47F0 C122	00210		124	FINISH	OFF AND BRANCH TO END
0001CC	9680 C12C	0021A		125	TFLAG,X'80'	TURN TRACE ON
0001D0	47F0 C084	00172		126	NEXT	RETURN TO LOOP
0001D4	9640 C12C	0021A		127	TFLAG,X'40'	TURN SUBTRACE ON
0001D8	47F0 C084	00172		128	NEXT	RETURN TO LOOP
0001DC	9620 C12C	0021A		129	TFLAG,X'20'	TURN INITSCLR ON
0001E0	47F0 C084	00172		130	NEXT	RETURN TO LOOP
0001E4	9610 C12C	0021A		131	TFLAG,X'10'	TURN INITARIT ON
0001E8	47F0 C084	00172		132	NEXT	RETURN TO LOOP
0001EC	9600 C12C	0021A		133	TFLAG,X'00'	TURN INITARAY ON
0001F0	47F0 C084	00172		134	NEXT	RETURN TO LOOP
0001F4	9638 C12C	0021A		135	TFLAG,X'38'	TURN INIT ON
0001F8	47F0 C084	00172		136	NEXT	RETURN TO LOOP
0001FC	9604 C12C	0021A		137	TFLAG,X'04'	TURN SUBCHK ON
000200	47F0 C084	00172		138	NEXT	RETURN TO LOOP
000204	9602 C12C	0021A		139	TFLAG,X'02'	TURN DISPLAY ON
000208	47F0 C084	00172		140	NEXT	RETURN TO LOOP
00020C	96FF C12C	0021A		141	TFLAG,X'FF'	TURN ALL ON
000210	982C D01C	0001C		142	FINISH	RESTORE REGISTERS
000214	92FF D00C	0000C		143	2,12,28(13)	INDICATE RETURN
000218	07FE			144	12(13),X'FF'	RETURN
00021A	FF			145	15,14	
00021B	E3D9C1C3C5			146	X'FF'	
000220	E2E4C2E3D9C1C3C5			147	CL5'TRACE'	
000228	C9D5C9E3			148	CL8'SUBTRACE'	
00022C	E2C3D3D9			149	CL4'INIT'	
000230	C1D9C9E3			150	CL4'SCLR'	
000234	C1D9C1E8			151	CL4'ARIT'	
000238	E2E4C2C3C8D2			152	CL4'ARY'	
00023E	C4C9E2D7D3C1E8			153	CL6'SUBCHK'	
000245	C103D3			154	CL7'DISPLAY'	
000248	000000EE			155	CL3'ALL'	
				156	A(BASE)	
000250	00000000			157	=V(DEBUG)	

RELOCATION DICTIONARY

POS.ID REL.ID FLAGS ADDRESS  
 01 01 0C 000248  
 01 02 1C 000250

CROSS-REFERENCE

SYMBOL	LEN	VALUE	DEFN	REFERENCES
ADEBUG	00001	000000	00001	
ALL	00003	000245	00154	0100
ART	00004	000230	00150	0091
ARY	00004	000234	00151	0093
BASE	00004	0000EE	00070	0079 0155
BASEAD	00004	000248	00155	0077
CHK	00006	000238	00152	0096
COMMON	00004	00010C	00081	0071
DBGOFF	00004	0000DC	00063	0004
DRGON	00004	0000F6	00073	0005
DEBUG	00001	000000	00002	
DEBUG#	00004	000000	00007	0003
DIS	00007	00023E	00153	0098
DISPLAY	00004	00008C	00054	0017
ENDIO	00004	0000D4	00060	0019
FINISH	00004	000210	00142	0103 0124
FLAGOFF	00004	000184	00107	0070
FLAGON	00004	0001CC	00125	0000
INITARAY	00004	00008C	00042	0013
INITARIT	00004	00007C	00038	0012
INITSCLR	00004	00006C	00034	0011
INT	00004	000228	00148	0087
LOOP	00006	000114	00083	0106
NEXT	00002	000172	00102	0108 0110 0112 0114 0116 0118 0120 0122 0126 0128 0130 0132 0134 0136 0138
SCR	00004	00022C	00149	0140
SKIP	00006	000154	00096	0089
STARTIO	00004	0000CC	00058	0088
SUB	00008	000220	00147	0018 0085
SUBCHK	00004	00009C	00046	0014
SUBTREN	00004	000044	00024	0008
SUBTRES	00004	000054	00028	0009
TCE	00005	00021R	00146	0083
TFLAG	00001	00021A	00145	0020 0024 0028 0034 0038 0042 0046 0054 0107 0109 0111 0113 0115 0117 0119
TRACE	00004	000034	00020	0121 0123 0125 0127 0129 0131 0133 0135 0137 0139 0141
TRACEOFF	00004	0000B4	00052	0007
TRACEON	00004	0000AC	00050	0016 0015
UNIT	00004	000064	00032	0010

NO STATEMENTS FLAGGED IN THIS ASSEMBLY  
 \*STATISTICS\* SOURCE RECORDS (SYSIN) = 156  
 \*OPTIONS IN EFFECT\* LIST, NODECK, LOAD, NORENT, XREF, NOTEST, ALGN, OS, NOTERM, LINECNT = 55  
 220 PRINTED LINES

\*\*\*\*\*

APPENDIX B  
USING THE DEBUG INIT OPTION

```
FORTRAN IV G LEVEL 20          MAIN          DATE = 73292          19/41/56

0001          DIMENSION FO(11),FC(11),A(11,3),P(3)
0002          COMMON XHV(11),HL,XLD,DL
0003          1000 FORMAT(F5.3,F5.1)
0004          1001 FORMAT(1H0,'ID=',F5.3,' HL=',F5.1)
0005          1002 FORMAT(15,3E10.3)
0006          1003 FORMAT(2F10.1)
0007          N=3
0008          NOBS=11
0009          READ(1,1000)XID,HL
0010          DL=XID/HL
0011          XLD=HL/XID
0012          WRITE(3,1001)XID,HL
0013          READ(1,1002)NCYC,P(1),P(2),P(3)
0014          READ(1,1003)(FO(I),XHV(I),I=1,NOBS)
0015          CALL NLRP(FO,FC,A,P,N,NOBS,NCYC)
0016          STOP
0017          END
```

```
FORTRAN IV G LEVEL 20          FUN          DATE = 73292          19/41/56

0001          SUBROUTINE FUN(FO,FC,A,P,N,NOBS,SIG2)
0002          DIMENSION FO(NOBS),FC(NOBS),A(NOBS,N),P(N)
0003          COMMON XHV(11),HL,XLD,DL
0004          SIG2=0.0
0005          DO 10 I=1,NOBS
0006          G=XHV(I)
0007          A1=HL*G
0008          B=DL*G*G
0009          C=P(1)*XLD+P(2)*A1+P(3)*B
0010          FC(I)=G/C
0011          D=-FC(I)/C
0012          A(I,1)=XLD*D
0013          A(I,2)=A1*D
0014          A(I,3)=B*D
0015          T=FO(I)-FC(I)
0016          10 SIG2=SIG2+T*T
0017          RETURN
0018          END
```

FORTRAN IV G LEVEL 20

NLRP

DATE = 73292

19/41/56

```

0001      SUBROUTINE NLRP(FO,FC,A,P,N,NOBS,NCYC)
0002      DIMENSION FO(NOBS),FC(NOBS),A(NOBS,N),P(N),D(3,3),G(3)
0003      1000 FORMAT(1H0,'CYCLE (FO-FC)**2 PARAMETERS')
0004      1001 FORMAT(1H ,13,2X,E12.6,' P(1)=',E12.6)
0005      1002 FORMAT(1H ,19X,'P(',I1,')=',E12.6)
0006      1003 FORMAT(1H )
0007      1004 FORMAT(1H ,P(',I1,')=',E13.6,' E.S.D.=',E13.6)
0008      1005 FORMAT(1H0,'CORRELATION MATRIX (TRIANGULAR PART)')
0009      1006 FORMAT(1H ,E13.6)
0010      1007 FORMAT(1H , 'ROW',I2)
0011      1008 FORMAT(1H0.6X,'FO',12X,'FC',10X,'FO-FC')
0012      1009 FORMAT(1H ,E13.6,2X,E13.6,1X,E13.6)
0013      1010 FORMAT(1H0,'SINGULAR MATRIX')
0014      WRITE(3,1000)
0015      NC=0
0016      1 CALL FUN(FO,FC,A,P,N,NOBS,SIG2)
0017      WRITE(3,1001)NC,SIG2,P(1)
0018      DO 2 I=2,N
0019      2 WRITE(3,1002)I,P(I)
0020      NC=NC+1
0021      IF(NC.GT.NCYC) GO TO 20

C
C      FORM A**T*A AND A**T*C
C
0022      DO 4 I=1,N
0023      DO 3 J=I,N
0024      D(I,J)=0.0
0025      DO 3 K=1,NOBS
0026      3 D(I,J)=D(I,J)+A(K,I)*A(K,J)
0027      G(I)=0.0
0028      DO 4 K=1,NOBS
0029      4 G(I)=G(I)+A(K,I)*(FO(K)-FC(K))

C
C      CALCULATION OF S(I,J) FOR UPPER TRIANGULAR MATRIX
C
0030      D(1,1)=SQRT(D(1,1))
0031      DO 5 J=2,N
0032      5 D(1,J)=D(1,J)/D(1,1)
0033      I=2
0034      6 K=I-1
0035      X=D(I,I)
0036      DO 7 L=1,K
0037      7 X=X-D(L,I)*D(L,I)
0038      IF(X.LE.0.0) GO TO 100
0039      D(I,I)=SQRT(X)
0040      M=I+1
0041      IF(M.GT.N) GO TO 10
0042      DO 9 J=M,N
0043      X=D(I,J)
0044      DO 8 L=1,K
0045      8 X=X-D(L,I)*D(L,J)
0046      9 D(I,J)=X/D(I,I)
0047      I=I+1

```

FORTRAN IV G LEVEL 20

NLRP

DATE \* 73292

19/41/56

```

0048      GO TO 6
          C
          C      CALCULATION OF K(I)
          C
0049      10 G(I)=G(I)/D(I,I)
0050      DO 12 I=2,N
0051      K=I-1
0052      X=G(I)
0053      DO 11 L=1,K
0054      11 X=X-D(L,I)*G(L)
0055      12 G(I)=X/D(I,I)
          C
          C      CALCULATION OF SOLUTIONS WHICH OVERWRITE ORIGINAL K(I)
          C
0056      G(N)=G(N)/D(N,N)
0057      M=N-1
0058      DO 14 J=1,M
0059      I=N-J
0060      X=G(I)
0061      K=I+1
0062      DO 13 L=K,N
0063      13 X=X-D(I,L)*G(L)
0064      14 G(I)=X/D(I,I)
          C
          C      APPLY SHIFTS FOR NEXT CYCLE
          C
0065      DO 15 I=1,N
0066      15 P(I)=P(I)+G(I)
0067      GO TO 1
          C
          C      END OF REFINEMENT
          C
0068      20 IF(NCYC.EQ.0) GO TO 40
          C
          C      CALCULATE R(I,J) WHICH OVERWRITE S(I,J)
          C
0069      D(N,N)=1.0/D(N,N)
0070      M1=N-1
0071      DO 22 K=1,M1
0072      I=N-K
0073      D(I,I)=1.0/D(I,I)
0074      M2=I+1
0075      M3=M2+N
0076      DO 22 H=M2,N
0077      J=M3-H
0078      X=0.0
0079      DO 21 L=M2,J
0080      21 X=X-D(I,L)*D(L,J)
0081      22 D(I,J)=X*D(I,I)
          C
          C      CALCULATION OF INVERSE WHICH OVERWRITES D(I,J)
          C
0082      DO 24 I=1,N

```

FORTRAN IV G LEVEL 20

NLRP

DATE = 73292

19/41/56

```

0083      DO 24 J=I,N
0084      X=0.0
0085      DO 23 L=J,N
0086      23 X=X+D(I,L)*D(J,L)
0087      24 D(I,J)=X
      C
      C      PRINT PARAMETERS AND ESTIMATED STANDARD DEVIATIONS
      C
0088      WRITE(3,1003)
0089      SIG2=SIG2/(NOBS-N)
0090      DO 25 I=1,N
0091      G(I)=SQRT(SIG2*D(I,I))
0092      25 WRITE(3,1004)I,P(I),G(I)
0093      WRITE(3,1005)
0094      DO 26 I=1,N
0095      WRITE(3,1007)I
0096      DO 26 J=I,N
0097      X=D(I,J)*SIG2/(G(I)*G(J))
0098      26 WRITE(3,1006)X
0099      40 WRITE(3,1003)
0100      WRITE(3,1008)
0101      DO 41 I=1,NOBS
0102      X=FO(I)-FC(I)
0103      41 WRITE(3,1009)FO(I),FC(I),X
0104      WRITE(3,1003)
0105      RETURN
0106      100 WRITE(3,1010)
0107      STOP
0108      *-----> DEBUG INIT
0109      END

```

APPENDIX C

USING THE DBGON AND DBGOFF SUBROUTINES

FORTRAN IV G LEVEL 20

NLRP

DATE = 73292

19/44/57

```

0001      SUBROUTINE NLRP(FO,FC,A,P,N,NOBS,NCYC)
0002      DIMENSION FO(NOBS),FC(NOBS),A(NOBS,N),P(N),D(3,3),G(3)
0003      1000 FORMAT(1H0,'CYCLE (FO-FC)*2 PARAMETERS')
0004      1001 FORMAT(1H ,13,2X,E12.6,' P(1)=' ,E12.6)
0005      1002 FORMAT(1H ,19X,'P(' ,I1,')=' ,E12.6)
0006      1003 FORMAT(1H )
0007      1004 FORMAT(1H , 'P(' ,I1,')=' ,E13.6,' E.S.D.=' ,E13.6)
0008      1005 FORMAT(1H0,'CORRELATION MATRIX (TRIANGULAR PART)')
0009      1006 FORMAT(1H ,E13.6)
0010      1007 FORMAT(1H , 'ROW' ,I2)
0011      1008 FORMAT(1H0,6X,'FO',12X,'FC',10X,'FO-FC')
0012      1009 FORMAT(1H ,E13.6,2X,E13.6,1X,E13.6)
0013      1010 FORMAT(1H0,'SINGULAR MATRIX')
0014      * → CALL DBGOFF('INIT')
0015      WRITE(3,1000)
0016      NC=0
0017      1 CALL FUN(FO,FC,A,P,N,NOBS,SIG2)
0018      WRITE(3,1001)NC,SIG2,P(1)
0019      DO 2 I=2,N
0020      2 WRITE(3,1002)I,P(I)
0021      NC=NC+1
0022      IF(NC.GT.NCYC) GO TO 20
0023      * → CALL DBGON('INITARIT')
      C
      C      FORM A**T*A AND A**T*C
      C
0024      DO 4 I=1,N
0025      DO 3 J=I,N
0026      D(I,J)=0.0
0027      DO 3 K=1,NOBS
0028      3 D(I,J)=D(I,J)+A(K,I)*A(K,J)
0029      G(I)=0.0
0030      DO 4 K=1,NOBS
0031      4 G(I)=G(I)+A(K,I)*(FO(K)-FC(K))
      C
      C      CALCULATION OF S(I,J) FOR UPPER TRIANGULAR MATRIX
      C
0032      D(1,1)=SQRT(D(1,1))
0033      DO 5 J=2,N
0034      5 D(1,J)=D(1,J)/D(1,1)
0035      I=2
0036      6 K=I-1
0037      X=D(I,I)
0038      DO 7 L=1,K
0039      7 X=X-D(L,I)*D(L,I)
0040      IF(X.LE.0.0) GO TO 100
0041      D(I,I)=SQRT(X)
0042      M=I+1
0043      IF(M.GT.N) GO TO 10
0044      DO 9 J=M,N
0045      X=D(I,J)
0046      DO 8 L=1,K
0047      8 X=X-D(L,I)*D(L,J)

```

FORTRAN IV G LEVEL 20

NLRP

DATE = 73292

19/44/57

```

0048          9 D(I,J)=X/D(I,I)
0049          I=I+1
0050          GO TO 6

      C
      C          CALCULATION OF K(I)
      C
0051          10 G(1)=G(1)/D(1,1)
0052          DO 12 I=2,N
0053             K=I-1
0054             X=G(I)
0055             DO 11 L=1,K
0056                11 X=X-D(L,I)*G(L)
0057                12 G(I)=X/D(I,I)

      C
      C          CALCULATION OF SOLUTIONS WHICH OVERWRITE ORIGINAL K(I)
      C
0058          G(N)=G(N)/D(N,N)
0059          M=N-1
0060          DO 14 J=1,M
0061             I=N-J
0062             X=G(I)
0063             K=I+1
0064             DO 13 L=K,N
0065                13 X=X-D(I,L)*G(L)
0066                14 G(I)=X/D(I,I)

      C
      C          APPLY SHIFTS FOR NEXT CYCLE
      C
0067          DO 15 I=1,N
0068             15 P(I)=P(I)+G(I)
0069          CALL DBGOFF('INITARIT')
0070          GO TO 1

      C
      C          END OF REFINEMENT
      C
0071          20 IF(NCYC.EQ.0) GO TO 40

      C
      C          CALCULATE R(I,J) WHICH OVERWRITE S(I,J)
      C
0072          D(N,N)=1.0/D(N,N)
0073          M1=N-1
0074          DO 22 K=1,M1
0075             I=N-K
0076             D(1,I)=1.0/D(1,I)
0077             M2=I+1
0078             M3=M2+N
0079             DO 22 M=M2,N
0080                J=M3-M
0081                X=0.0
0082                DO 21 L=M2,J
0083                   21 X=X-D(I,L)*D(L,J)
0084                   22 D(I,J)=X*D(I,I)

```

FORTRAN IV G LEVEL 20

NLRP

DATE = 73292

19/44/57

C CALCULATION OF INVERSE WHICH OVERWRITES D(I,J)

C

```

0085 DO 24 I=1,N
0086 DO 24 J=1,N
0087 X=0.0
0088 DO 23 L=J,N
0089 23 X=X+D(I,L)*D(J,L)
0090 24 D(I,J)=X


```

C

C PRINT PARAMETERS AND ESTIMATED STANDARD DEVIATIONS

C

```

0091 WRITE(3,1003)
0092 SIG2=SIG2/(NOBS-N)
0093 DO 25 I=1,N
0094 G(I)=SQRT(SIG2*D(I,I))
0095 25 WRITE(3,1004)I,P(I),G(I)
0096 WRITE(3,1005)
0097 DO 26 I=1,N
0098 WRITE(3,1007)I
0099 DO 26 J=1,N
0100 X=D(I,J)*SIG2/(G(I)*G(J))
0101 26 WRITE(3,1006)X
0102 40 WRITE(3,1003)
0103 WRITE(3,1008)
0104 DO 41 I=1,NOBS
0105 X=FO(I)-FC(I)
0106 41 WRITE(3,1009)FO(I),FC(I),X
0107 WRITE(3,1003)
0108 RETURN
0109 100 WRITE(3,1010)
0110 STOP
0111  → DEBUG INIT
0112 END

```

FORTRAN IV G LEVEL 20

NLRP

DATE = 73292

19/44/57

```
*OPTIONS IN EFFECT* ID,EBCDIC,SOURCE,NOLIST,NODECK,LOAD,NOMAP
*OPTIONS IN EFFECT* NAME = NLRP , LINECNT = 55
*STATISTICS* SOURCE STATEMENTS = 112,PROGRAM SIZE = 5878
*STATISTICS* NO DIAGNOSTICS GENERATED

*STATISTICS* NO DIAGNOSTICS THIS STEP 0
```

ID=0.633 HL= 51.0

N = 3

NOBS = 11

NCYC = 5

CYCLE (FO-FC)\*\*2 PARAMETERS

```
0 0.186622E+09 P(1)=0.450000E-01
P(2)=0.400000E-06
P(3)=0.400000E-09
```

```
D(1) = 0.0
D(1) = 0.6841798E+11
D(1) = 0.1349480E+12
D(1) = 0.1957997E+12
D(1) = 0.2499333E+12
D(1) = 0.2974586E+12
D(1) = 0.3417398E+12
D(1) = 0.3681495E+12
D(1) = 0.3844487E+12
D(1) = 0.3914156E+12
D(1) = 0.3948546E+12
D(1) = 0.3958975E+12
D(4) = 0.0
D(4) = 0.6496284E+16
D(4) = 0.1491798E+17
D(4) = 0.2454983E+17
D(4) = 0.3482980E+17
D(4) = 0.4535645E+17
D(4) = 0.5589745E+17
D(4) = 0.6534386E+17
D(4) = 0.7318818E+17
D(4) = 0.7889721E+17
D(4) = 0.8136255E+17
D(4) = 0.8285459E+17
D(7) = 0.0
D(7) = 0.2371480E+18
D(7) = 0.6478616E+18
D(7) = 0.1233833E+19
D(7) = 0.1983577E+19
D(7) = 0.2888292E+19
D(7) = 0.3844859E+19
D(7) = 0.5143656E+19
D(7) = 0.6566670E+19
D(7) = 0.7938780E+19
D(7) = 0.9138734E+19
D(7) = 0.9951483E+19
G(1) = 0.0
G(1) = -0.5528154E+09
G(1) = -0.1485663E+10
G(1) = -0.2467159E+10
G(1) = -0.3582838E+10
G(1) = -0.4531681E+10
G(1) = -0.5447789E+10
G(1) = -0.6181179E+10
```

G(1) = -0.6784082E+10  
G(1) = -0.7197331E+10  
G(1) = -0.7473046E+10  
G(1) = -0.7528292E+10  
D(5) = 0.0  
D(5) = 0.6168223E+21  
D(5) = 0.1683009E+22  
D(5) = 0.3207123E+22  
D(5) = 0.5159288E+22  
D(5) = 0.7491643E+22  
D(5) = 0.1000049E+23  
D(5) = 0.1337866E+23  
D(5) = 0.1707992E+23  
D(5) = 0.2064058E+23  
D(5) = 0.2374906E+23  
D(5) = 0.2580361E+23  
D(8) = 0.0  
D(8) = 0.2251721E+23  
D(8) = 0.7441227E+23  
D(8) = 0.1671423E+24  
D(8) = 0.3096707E+24  
D(8) = 0.5083378E+24  
D(8) = 0.7379127E+24  
D(8) = 0.1202421E+25  
D(8) = 0.1880700E+25  
D(8) = 0.2862099E+25  
D(8) = 0.3993933E+25  
D(8) = 0.5167965E+25  
G(2) = 0.0  
G(2) = -0.5241385E+14  
G(2) = -0.1706136E+15  
G(2) = -0.3259354E+15  
G(2) = -0.5226105E+15  
G(2) = -0.7505329E+15  
G(2) = -0.9685742E+15  
G(2) = -0.1230897E+16  
G(2) = -0.1518270E+16  
G(2) = -0.1813866E+16  
G(2) = -0.2075658E+16  
G(2) = -0.2154694E+16  
D(9) = 0.0  
D(9) = 0.8219946E+24  
D(9) = 0.3347909E+25  
D(9) = 0.8989786E+25  
D(9) = 0.1939583E+26  
D(9) = 0.3631802E+26  
D(9) = 0.5732556E+26  
D(9) = 0.1211968E+27  
D(9) = 0.2454954E+27  
D(9) = 0.5153858E+27  
D(9) = 0.9285639E+27  
D(9) = 0.1574295E+28  
G(3) = 0.0  
G(3) = -0.1913377E+16  
G(3) = -0.7666572E+16  
G(3) = -0.1711665E+17  
G(3) = -0.3147599E+17  
G(3) = -0.5089014E+17  
G(3) = -0.7084228E+17  
G(3) = -0.1069125E+18  
G(3) = -0.1595752E+18  
G(3) = -0.2408658E+18  
G(3) = -0.3364332E+18  
G(3) = -0.3799040E+18  
D(1) = 629203.9  
D(4) = 0.1316816E+12

D(7) = 0.1581586E+14  
 D(5) = 0.9243145E+11  
 D(8) = 0.3337940E+14  
 D(9) = 0.1449032E+14  
 G(1) = -11964.79  
 G(2) = -6265.734  
 G(3) = 1275.048  
 G(3) = 0.8799304E-10  
 G(2) = -0.9956443E-07  
 G(1) = -0.3904460E-03  
 P(1) = 0.4460955E-01  
 P(2) = 0.3004355E-06  
 P(3) = 0.4879928E-09  
 1 0.762321E+07 P(1)=0.446096E-01  
 P(2)=0.300436E-06  
 P(3)=0.487993E-09

D(1) = 0.0  
 D(1) = 0.1105624E+12  
 D(1) = 0.2250562E+12  
 D(1) = 0.3344803E+12  
 D(1) = 0.4347897E+12  
 D(1) = 0.5246533E+12  
 D(1) = 0.6090522E+12  
 D(1) = 0.6602382E+12  
 D(1) = 0.6912156E+12  
 D(1) = 0.7035314E+12  
 D(1) = 0.7091210E+12  
 D(1) = 0.7105703E+12  
 D(4) = 0.0  
 D(4) = 0.1049790E+17  
 D(4) = 0.2499282E+17  
 D(4) = 0.4230918E+17  
 D(4) = 0.6135795E+17  
 D(4) = 0.8126724E+17  
 D(4) = 0.1013548E+18  
 D(4) = 0.1196612E+18  
 D(4) = 0.1344266E+18  
 D(4) = 0.1432359E+18  
 D(4) = 0.1405432E+18  
 D(4) = 0.1506166E+18  
 D(7) = 0.0  
 D(7) = 0.3832276E+18  
 D(7) = 0.1088747E+19  
 D(7) = 0.2142308E+19  
 D(7) = 0.3533065E+19  
 D(7) = 0.5220912E+19  
 D(7) = 0.7067055E+19  
 D(7) = 0.9584241E+19  
 D(7) = 0.1229008E+20  
 D(7) = 0.1471271E+20  
 D(7) = 0.1665016E+20  
 D(7) = 0.1779056E+20  
 G(1) = 0.0  
 G(1) = 0.2336891E+09  
 G(1) = 0.2536317E+09  
 G(1) = 0.3032934E+09  
 G(1) = 0.5277148E+09  
 G(1) = 0.7059812E+09  
 G(1) = 0.1033957E+10  
 G(1) = 0.1386830E+10  
 G(1) = 0.1614917E+10  
 G(1) = 0.1645910E+10  
 G(1) = 0.1613349E+10  
 G(1) = 0.1646457E+10  
 D(5) = 0.0  
 D(5) = 0.9967758E+21

D(5) = 0.2831833E+22  
 D(5) = 0.5572146E+22  
 D(5) = 0.9189505E+22  
 D(5) = 0.1360041E+23  
 D(5) = 0.1838142E+23  
 D(5) = 0.2492861E+23  
 D(5) = 0.3196650E+23  
 D(5) = 0.3826777E+23  
 D(5) = 0.4330706E+23  
 D(5) = 0.4627323E+23  
 D(8) = 0.0  
 D(8) = 0.3638745E+23  
 D(8) = 0.1257061E+24  
 D(8) = 0.2924322E+24  
 D(8) = 0.5565370E+24  
 D(8) = 0.9322520E+24  
 D(8) = 0.1369745E+25  
 D(8) = 0.2270002E+25  
 D(8) = 0.3559736E+25  
 D(8) = 0.5292622E+25  
 D(8) = 0.7132226E+25  
 D(8) = 0.8763657E+25  
 G(2) = 0.0  
 G(2) = 0.2218877E+14  
 G(2) = 0.2471351E+14  
 G(2) = 0.4523250E+14  
 G(2) = 0.7265815E+14  
 G(2) = 0.1121531E+15  
 G(2) = 0.1902138E+15  
 G(2) = 0.3164170E+15  
 G(2) = 0.4251341E+15  
 G(2) = 0.4473034E+15  
 G(2) = 0.4163863E+15  
 G(2) = 0.4637502E+15  
 D(9) = 0.0  
 D(9) = 0.1328329E+25  
 D(9) = 0.5675790E+25  
 D(9) = 0.1581973E+26  
 D(9) = 0.3510211E+26  
 D(9) = 0.6710501E+26  
 D(9) = 0.1071303E+27  
 D(9) = 0.2309265E+27  
 D(9) = 0.4672702E+27  
 D(9) = 0.9438316E+27  
 D(9) = 0.1615383E+28  
 D(9) = 0.2512689E+28  
 G(3) = 0.0  
 G(3) = 0.8100045E+15  
 G(3) = 0.9328924E+15  
 G(3) = 0.2181308E+16  
 G(3) = 0.4183664E+16  
 G(3) = 0.7547707E+16  
 G(3) = 0.1469004E+17  
 G(3) = 0.3204416E+17  
 G(3) = 0.5196726E+17  
 G(3) = 0.5806400E+17  
 G(3) = 0.4677771E+17  
 G(3) = 0.7282842E+17  
 D(1) = 842953.3  
 D(4) = 0.1786772E+12  
 D(7) = 0.2110503E+14  
 D(5) = 0.1197819E+12  
 D(8) = 0.4168133E+14  
 D(9) = 0.1816405E+14  
 G(1) = 1953.201  
 G(2) = 958.0562

G(3) = -458.4326  
 G(3) = -0.2523845E-10  
 G(2) = 0.1678074E-07  
 G(1) = -0.6079539E-03  
 P(1) = 0.4400159E-01  
 P(2) = 0.3172162E-06  
 P(3) = 0.4627543E-09  
 2 0.269235E+07 P(1)=0.440016E-01  
 P(2)=0.317216E-06  
 P(3)=0.462754E-09

D(1) = 0.0  
 D(1) = 0.1054112E+12  
 D(1) = 0.2129003E+12  
 D(1) = 0.3145432E+12  
 D(1) = 0.4070122E+12  
 D(1) = 0.4894240E+12  
 D(1) = 0.5666726E+12  
 D(1) = 0.6133014E+12  
 D(1) = 0.6416404E+12  
 D(1) = 0.6531041E+12  
 D(1) = 0.6584084E+12  
 D(1) = 0.6598341E+12  
 D(4) = 0.0  
 D(4) = 0.1000879E+17  
 D(4) = 0.2361793E+17  
 D(4) = 0.3970166E+17  
 D(4) = 0.5726151E+17  
 D(4) = 0.7551982E+17  
 D(4) = 0.9390550E+17  
 D(4) = 0.1105821E+18  
 D(4) = 0.1240899E+18  
 D(4) = 0.1322896E+18  
 D(4) = 0.1373260E+18  
 D(4) = 0.1393656E+18  
 D(7) = 0.0  
 D(7) = 0.3653724E+18  
 D(7) = 0.1027777E+19  
 D(7) = 0.2006343E+19  
 D(7) = 0.3208393E+19  
 D(7) = 0.4843616E+19  
 D(7) = 0.6526030E+19  
 D(7) = 0.8819107E+19  
 D(7) = 0.1129440E+20  
 D(7) = 0.1354948E+20  
 D(7) = 0.1538806E+20  
 D(7) = 0.1650984E+20  
 G(1) = 0.0  
 G(1) = 0.1323960E+09  
 G(1) = 3095776.  
 G(1) = -0.5876800E+08  
 G(1) = -0.1304328E+09  
 G(1) = -0.1814996E+09  
 G(1) = -0.9172838E+08  
 G(1) = 0.4688134E+08  
 G(1) = 0.1147705E+09  
 G(1) = 0.6962240E+08  
 G(1) = 2554736.  
 G(1) = 0.3032432E+08  
 D(5) = 0.0  
 D(5) = 0.9503350E+21  
 D(5) = 0.2673251E+22  
 D(5) = 0.5218501E+22  
 D(5) = 0.8553115E+22  
 D(5) = 0.1259825E+23  
 D(5) = 0.1697427E+23  
 D(5) = 0.2293850E+23

D(5) = 0.2937695E+23  
 D(5) = 0.3524219E+23  
 D(5) = 0.4002433E+23  
 D(5) = 0.4294208E+23  
 D(8) = 0.0  
 D(8) = 0.3469211E+23  
 D(8) = 0.1185525E+24  
 D(8) = 0.2734105E+24  
 D(8) = 0.5168720E+24  
 D(8) = 0.8614317E+24  
 D(8) = 0.1261859E+25  
 D(8) = 0.2081965E+25  
 D(8) = 0.3261850E+25  
 D(8) = 0.4874824E+25  
 D(8) = 0.6620554E+25  
 D(8) = 0.8225350E+25  
 G(2) = 0.0  
 G(2) = 0.1257100E+14  
 G(2) = -0.3798406E+13  
 G(2) = -0.1358836E+14  
 G(2) = -0.2719749E+14  
 G(2) = -0.3851133E+14  
 G(2) = -0.1714506E+14  
 G(2) = 0.3242798E+14  
 G(2) = 0.6478725E+14  
 G(2) = 0.3249329E+14  
 G(2) = -0.3118749E+14  
 G(2) = 0.8539116E+13  
 D(9) = 0.0  
 D(9) = 0.1266439E+25  
 D(9) = 0.5348224E+25  
 D(9) = 0.1477009E+26  
 D(9) = 0.3254530E+26  
 D(9) = 0.6189443E+26  
 D(9) = 0.9853611E+26  
 D(9) = 0.2113032E+27  
 D(9) = 0.4275244E+27  
 D(9) = 0.8711016E+27  
 D(9) = 0.1508383E+28  
 D(9) = 0.2391039E+28  
 G(3) = 0.0  
 G(3) = 0.4589060E+15  
 G(3) = -0.3378507E+15  
 G(3) = -0.9334907E+15  
 G(3) = -0.1927099E+16  
 G(3) = -0.2890797E+16  
 G(3) = -0.9356490E+15  
 G(3) = 0.5880789E+16  
 G(3) = 0.1181082E+17  
 G(3) = 0.2929790E+16  
 G(3) = -0.2031699E+17  
 G(3) = 0.1533093E+16  
 D(1) = 812301.8  
 D(4) = 0.1715687E+12  
 D(7) = 0.2032474E+14  
 D(5) = 0.1162166E+12  
 D(8) = 0.4077097E+14  
 D(9) = 0.1776719E+14  
 G(1) = 37.33134  
 G(2) = 18.36423  
 G(3) = 1.441758  
 G(3) = 0.8114722E-13  
 G(2) = 0.1295493E-09  
 G(1) = 0.1656459E-04  
 P(1) = 0.4401816E-01  
 P(2) = 0.3173458E-06

P(3) = 0.4628353E-09  
 3 0.269060E+07 P(1)=0.448182E-01  
 P(2)=0.317346E-06  
 P(3)=0.462835E-09

D(1) = 0.0  
 D(1) = 0.1052490E+12  
 D(1) = 0.2125810E+12  
 D(1) = 0.3140602E+12  
 D(1) = 0.4063879E+12  
 D(1) = 0.4886746E+12  
 D(1) = 0.5658060E+12  
 D(1) = 0.6123658E+12  
 D(1) = 0.6406637E+12  
 D(1) = 0.6521114E+12  
 D(1) = 0.6574087E+12  
 D(1) = 0.6588326E+12  
 D(4) = 0.0  
 D(4) = 0.9993401E+16  
 D(4) = 0.2358163E+17  
 D(4) = 0.3964071E+17  
 D(4) = 0.5717375E+17  
 D(4) = 0.7540430E+17  
 D(4) = 0.9376216E+17  
 D(4) = 0.1104140E+18  
 D(4) = 0.1239022E+18  
 D(4) = 0.1320906E+18  
 D(4) = 0.1371204E+18  
 D(4) = 0.1391574E+18  
 D(7) = 0.0  
 D(7) = 0.3648107E+18  
 D(7) = 0.1026198E+19  
 D(7) = 0.2003264E+19  
 D(7) = 0.3283357E+19  
 D(7) = 0.4836216E+19  
 D(7) = 0.6516000E+19  
 D(7) = 0.8805757E+19  
 D(7) = 0.1127755E+20  
 D(7) = 0.1352942E+20  
 D(7) = 0.1536557E+20  
 D(7) = 0.1648596E+20  
 G(1) = 0.0  
 G(1) = 0.1292261E+09  
 G(1) = -3565744.  
 G(1) = -0.6922352E+08  
 G(1) = -0.1447397E+09  
 G(1) = -0.1996179E+09  
 G(1) = -0.1137074E+09  
 G(1) = 0.2168515E+08  
 G(1) = 0.8710696E+08  
 G(1) = 0.4055611E+08  
 G(1) = -0.2735530E+08  
 G(1) = 18640.00  
 N(5) = 0.0  
 D(5) = 0.9488735E+21  
 D(5) = 0.2669143E+22  
 D(5) = 0.5210489E+22  
 D(5) = 0.8540009E+22  
 D(5) = 0.1257900E+23  
 D(5) = 0.1694832E+23  
 D(5) = 0.2290377E+23  
 D(5) = 0.2933289E+23  
 D(5) = 0.3519002E+23  
 D(5) = 0.3996503E+23  
 D(5) = 0.4287999E+23  
 D(8) = 0.0  
 D(8) = 0.3463877E+23

D(8) = 0.1183704E+24  
D(8) = 0.2729911E+24  
D(8) = 0.5168808E+24  
D(8) = 0.8601169E+24  
D(8) = 0.1259937E+25  
D(8) = 0.2078827E+25  
D(8) = 0.3257003E+25  
D(8) = 0.4867748E+25  
D(8) = 0.6611166E+25  
D(8) = 0.8213985E+25  
G(2) = 0.0  
G(2) = 0.1227002E+14  
G(2) = -0.4541429E+13  
G(2) = -0.1493177E+14  
G(2) = -0.2927230E+14  
G(2) = -0.4143055E+14  
G(2) = -0.2098316E+14  
G(2) = 0.2743925E+14  
G(2) = 0.5862248E+14  
G(2) = 0.2532514E+14  
G(2) = -0.3915678E+14  
G(2) = 0.3825205E+10  
D(9) = 0.0  
D(9) = 0.1264493E+25  
D(9) = 0.5340010E+25  
D(9) = 0.1474744E+26  
D(9) = 0.3249549E+26  
D(9) = 0.6180002E+26  
D(9) = 0.9838613E+26  
D(9) = 0.2109861E+27  
D(9) = 0.4268940E+27  
D(9) = 0.8698581E+27  
D(9) = 0.1506296E+28  
D(9) = 0.2387864E+28  
G(3) = 0.0  
G(3) = 0.4479186E+15  
G(3) = -0.3703540E+15  
G(3) = -0.1002523E+16  
G(3) = -0.2049530E+16  
G(3) = -0.3085153E+16  
G(3) = -0.1214090E+16  
G(3) = 0.5444129E+16  
G(3) = 0.1115864E+17  
G(3) = 0.2001674E+16  
G(3) = -0.2153756E+17  
G(3) = 0.1274066E+13  
D(1) = 811605.1  
D(4) = 0.1714425E+12  
D(7) = 0.2031078E+14  
D(5) = 0.1161355E+12  
D(8) = 0.4074424E+14  
D(9) = 0.1775511E+14  
G(1) = 0.2296457E-01  
G(2) = -0.9635133E-03  
G(3) = 0.4480257E-01  
G(3) = 0.2527868E-14  
G(2) = -0.8951570E-12  
G(1) = 0.1541108E-06  
P(1) = 0.4401831E-01  
P(2) = 0.3173449E-06  
P(3) = 0.4628378E-09  
4 0.269061E+07 P(1)=0.440183E-01  
P(2)=0.317345E-06  
P(3)=0.462838E-09  
D(1) = 0.0  
D(1) = 0.1052407E+12

D(1) = 0.2125802E+12  
D(1) = 0.3140593E+12  
D(1) = 0.4063869E+12  
D(1) = 0.4886736E+12  
D(1) = 0.5658051E+12  
D(1) = 0.6123648E+12  
D(1) = 0.6406628E+12  
D(1) = 0.6521105E+12  
D(1) = 0.6574078E+12  
D(1) = 0.6588317E+12  
D(4) = 0.0  
D(4) = 0.9993362E+16  
D(4) = 0.2358154E+17  
D(4) = 0.3964060E+17  
D(4) = 0.5717362E+17  
D(4) = 0.7540423E+17  
D(4) = 0.9376209E+17  
D(4) = 0.1104139E+18  
D(4) = 0.1239021E+18  
D(4) = 0.1320905E+18  
D(4) = 0.1371203E+18  
D(4) = 0.1391573E+18  
D(7) = 0.0  
D(7) = 0.3648093E+18  
D(7) = 0.1026195E+19  
D(7) = 0.2003259E+19  
D(7) = 0.3203351E+19  
D(7) = 0.4836210E+19  
D(7) = 0.6516074E+19  
D(7) = 0.8805753E+19  
D(7) = 0.1127754E+20  
D(7) = 0.1352942E+20  
D(7) = 0.1536556E+20  
D(7) = 0.1648595E+20  
G(1) = 0.0  
G(1) = 0.1292182E+09  
G(1) = -3581040.  
G(1) = -0.6924250E+08  
G(1) = -0.1447599E+09  
G(1) = -0.1996380E+09  
G(1) = -0.1137253E+09  
G(1) = 0.2166896E+08  
G(1) = 0.8709077E+08  
G(1) = 0.4053992E+08  
G(1) = -0.2737229E+08  
G(1) = 560.0000  
D(5) = 0.0  
D(5) = 0.9488699E+21  
D(5) = 0.2669134E+22  
D(5) = 0.5210480E+22  
D(5) = 0.8540000E+22  
D(5) = 0.1257899E+23  
D(5) = 0.1694832E+23  
D(5) = 0.2290377E+23  
D(5) = 0.2933290E+23  
D(5) = 0.3519003E+23  
D(5) = 0.3996583E+23  
D(5) = 0.4287997E+23  
D(8) = 0.0  
D(8) = 0.3463864E+23  
D(8) = 0.1183699E+24  
D(8) = 0.2729903E+24  
D(8) = 0.5160799E+24  
D(8) = 0.8601160E+24  
D(8) = 0.1259937E+25  
D(8) = 0.2078028E+25

D(8) = 0.3257004E+25  
 D(8) = 0.4867749E+25  
 D(8) = 0.6611163E+25  
 D(8) = 0.8213975E+25  
 G(2) = 0.0  
 G(2) = 0.1226927E+14  
 G(2) = -0.4543117E+13  
 G(2) = -0.1493404E+14  
 G(2) = -0.2927478E+14  
 G(2) = -0.4143303E+14  
 G(2) = -0.2098513E+14  
 G(2) = 0.2743793E+14  
 G(2) = 0.5862115E+14  
 G(2) = 0.2532382E+14  
 G(2) = -0.3915884E+14  
 G(2) = 0.2013266E+09  
 D(9) = 0.0  
 D(9) = 0.1264489E+25  
 D(9) = 0.5339992E+25  
 D(9) = 0.1474740E+26  
 D(9) = 0.3249545E+26  
 D(9) = 0.6179999E+26  
 D(9) = 0.9838613E+26  
 D(9) = 0.2109861E+27  
 D(9) = 0.4260940E+27  
 D(9) = 0.8698581E+27  
 D(9) = 0.1506295E+28  
 D(9) = 0.2387860E+28  
 G(3) = 0.0  
 G(3) = 0.4478915E+15  
 G(3) = -0.3704270E+15  
 G(3) = -0.1002631E+16  
 G(3) = -0.2049654E+16  
 G(3) = -0.3085278E+16  
 G(3) = -0.1214166E+16  
 G(3) = 0.5444142E+16  
 G(3) = 0.1115865E+17  
 G(3) = 0.2001687E+16  
 G(3) = -0.2153782E+17  
 G(3) = 0.9448928E+11  
 D(1) = 811684.5  
 D(4) = 0.1714426E+12  
 D(7) = 0.2031078E+14  
 D(5) = 0.1161354E+12  
 D(8) = 0.4074421E+14  
 D(9) = 0.1775504E+14  
 G(1) = 0.6899231E-03  
 G(2) = 0.7150655E-03  
 G(3) = 0.2891663E-02  
 G(3) = 0.1628643E-15  
 G(2) = -0.5098109E-13  
 G(1) = 0.7542766E-08  
 P(1) = 0.4401832E-01  
 P(2) = 0.3173448E-06  
 P(3) = 0.4628378E-09  
 5 0.269062E+07 P(1)=0.440183E-01  
 P(2)=0.317345E-06  
 P(3)=0.462838E-09  
  
 P(1)= 0.440183E-01 E.S.D.= 0.204765E-02  
 P(2)= 0.317345E-06 E.S.D.= 0.125001E-07  
 P(3)= 0.462838E-09 E.S.D.= 0.326632E-10

CORRELATION MATRIX (TRIANGULAR PART)

ROW 1  
0.100000E+01

-0.923484E+00  
0.782893E+00  
ROW 2  
0.100000E+01  
-0.916739E+00  
ROW 3  
0.100000E+01

FO	FC	FO-FC
0.241780E+05	0.245763E+05	-0.398301E+03
0.289230E+05	0.285176E+05	0.405352E+03
0.316460E+05	0.314399E+05	0.206121E+03
0.338850E+05	0.336365E+05	0.248531E+03
0.354920E+05	0.353007E+05	0.191309E+03
0.356920E+05	0.360013E+05	-0.309344E+03
0.382720E+05	0.388995E+05	-0.627473E+03
0.392620E+05	0.396509E+05	-0.388906E+03
0.391730E+05	0.387379E+05	0.435070E+03
0.377440E+05	0.368109E+05	0.933082E+03
0.318090E+05	0.325344E+05	-0.725406E+03

.....

APPENDIX D

USING THE DBGN AND DBGOFF SUBROUTINES IN DEBUG PACKETS

```

FORTRAN IV G LEVEL  20                NLRP                DATE = 73292                18/27/55

0001          SUBROUTINE NLRP(FO,FC,A,P,N,NOBS,NCYC)
0002          DIMENSION FO(NOBS),FC(NOBS),A(NOBS,N),P(N),D(3,3),G(3)
0003          1000  FORMAT(1H0,'CYCLE (FO-FC)**2  PARAMETERS')
0004          1001  FORMAT(1H ,13,2X,E12.6,' P(1)=',E12.6)
0005          1002  FORMAT(1H ,19X,'P(',I1,')=',E12.6)
0006          1003  FORMAT(1H )
0007          1004  FORMAT(1H , 'P(',I1,')=',E13.6,' E.S.D.=',E13.6)
0008          1005  FORMAT(1H0,'CORRELATION MATRIX (TRIANGULAR PART)')
0009          1006  FORMAT(1H ,E13.6)
0010          1007  FORMAT(1H , 'ROW',I2)
0011          1008  FORMAT(1H0,6X,'FO',12X,'FC',10X,'FO-FC')
0012          1009  FORMAT(1H ,E13.6,2X,E13.6,1X,E13.6)
0013          1010  FORMAT(1H0,'SINGULAR MATRIX')
0014  ⊕ → 990  CONTINUE
0015          WRITE(3,1000)
0016          NC=0
0017          1  CALL FUN(FO,FC,A,P,N,NOBS,SIG2)
0018          WRITE(3,1001)NC,SIG2,P(1)
0019          DO 2 I=2,N
0020          2  WRITE(3,1002)I,P(I)
0021          NC=NC+1
0022          IF(NC.GT.NCYC) GO TO 20
0023  ⊕ → 991  CONTINUE
          C
          C  FORM A**T*A AND A**T*C
          C
0024          DO 4 I=1,N
0025          DO 3 J=I,N
0026          D(I,J)=0.0
0027          DO 3 K=1,NOBS
0028          3  D(I,J)=D(I,J)+A(K,I)*A(K,J)
0029          G(I)=0.0
0030          DO 4 K=1,NOBS
0031          4  G(I)=G(I)+A(K,I)*(FO(K)-FC(K))
          C
          C  CALCULATION OF S(I,J) FOR UPPER TRIANGULAR MATRIX
          C
0032          D(1,1)=SQRT(D(1,1))
0033          DO 5 J=2,N
0034          5  D(1,J)=D(1,J)/D(1,1)
0035          I=2
0036          6  K=I-1
0037          X=D(I,I)
0038          DO 7 L=1,K
0039          7  X=X-D(L,I)*D(L,I)
0040          IF(X.LE.0.0) GO TO 100
0041          D(I,I)=SQRT(X)
0042          M=I+1
0043          IF(M.GT.N) GO TO 10
0044          DO 9 J=M,N
0045          X=D(I,J)
0046          DO 8 L=1,K
0047          8  X=X-D(L,I)*D(L,J)

```

FORTRAN IV G LEVEL 20

NLRP

DATE = 73292

18/27/55

```

0048      9 D(I,J)=X/D(I,I)
0049      I=I+1
0050      GO TO 6
      C
      C      CALCULATION OF K(I)
      C
0051      10 G(1)=G(1)/D(1,1)
0052      DO 12 I=2,N
0053      K=I-1
0054      X=G(I)
0055      DO 11 L=1,K
0056      11 X=X-D(L,I)*G(L)
0057      12 G(I)=X/D(I,I)
      C
      C      CALCULATION OF SOLUTIONS WHICH OVERWRITE ORIGINAL K(I)
      C
0058      G(N)=G(N)/D(N,N)
0059      M=N-1
0060      DO 14 J=1,M
0061      I=N-J
0062      X=G(I)
0063      K=I+1
0064      DO 13 L=K,N
0065      13 X=X-D(I,L)*G(L)
0066      14 G(I)=X/D(I,I)
      C
      C      APPLY SHIFTS FOR NEXT CYCLE
      C
0067      DO 15 I=1,N
0068      15 P(I)=P(I)+G(I)
0069      992 CONTINUE
0070      GO TO 1
      C
      C      END OF REFINEMENT
      C
0071      20 IF(NCYC.EQ.0) GO TO 40
      C
      C      CALCULATE R(I,J) WHICH OVERWRITE S(I,J)
      C
0072      D(N,N)=1.0/D(N,N)
0073      M1=N-1
0074      DO 22 K=1,M1
0075      I=N-K
0076      D(I,I)=1.0/D(I,I)
0077      M2=I+1
0078      M3=M2+N
0079      DO 22 M=M2,N
0080      J=M3-M
0081      X=0.0
0082      DO 21 L=M2,J
0083      21 X=X-D(I,L)*D(L,J)
0084      22 O(I,J)=X*D(I,I)
      C

```

FORTRAN IV G LEVEL 20

NLRP

DATE = 73292

18/27/55

```

      C      CALCULATION OF INVERSE WHICH OVERWRITES D(I,J)
      C
0085      DO 24 I=1,N
0086      DO 24 J=1,N
0087      X=0.0
0088      DO 23 L=J,N
0089      23 X=X+D(I,L)*D(J,L)
0090      24 D(I,J)=X

      C      PRINT PARAMETERS AND ESTIMATED STANDARD DEVIATIONS
      C
0091      WRITE(3,1003)
0092      SIG2=SIG2/(NOBS-N)
0093      DO 25 I=1,N
0094      G(I)=SQRT(SIG2*D(I,I))
0095      25 WRITE(3,1004)I,P(I),G(I)
0096      WRITE(3,1005)
0097      DO 26 I=1,N
0098      WRITE(3,1007)I
0099      DO 26 J=1,N
0100      X=D(I,J)*SIG2/(G(I)*G(J))
0101      26 WRITE(3,1006)X
0102      40 WRITE(3,1003)
0103      WRITE(3,1008)
0104      DO 41 I=1,NOBS
0105      X=FO(I)-FC(I)
0106      41 WRITE(3,1009)FO(I),FC(I),X
0107      WRITE(3,1003)
0108      RETURN
0109      100 WRITE(3,1010)
0110      STOP
0111      DEBUG INIT
0112      AT 990
0113      CALL DBGOFF('INIT')
0114      AT 991
0115      CALL DBGON('INITARIT')
0116      AT 992
0117      CALL DBGOFF('INITARIT')
0118      END

```





APPENDIX E  
USING THE DISPLAY STATEMENT

FORTRAN IV G LEVEL 20

NLRP

DATE = 73292

19/48/11

```

0001      SUBROUTINE NLRP(FO,FC,A,P,N,NOBS,NCYC)
0002      DIMENSION FO(NOBS),FC(NOBS),A(NOBS,N),P(N),D(3,3),G(3)
0003      1000 FORMAT(1H0,'CYCLE (FO-FC)**2 PARAMETERS')
0004      1001 FORMAT(1H ,I3,2X,E12.6,' P(1)=',E12.6)
0005      1002 FORMAT(1H ,19X,'P(',I1,')=',E12.6)
0006      1003 FORMAT(1H )
0007      1004 FORMAT(1H , 'P(',I1,')=',E13.6,' E.S.D.=',E13.6)
0008      1005 FORMAT(1H0,'CORRELATION MATRIX (TRIANGULAR PART)')
0009      1006 FORMAT(1H ,E13.6)
0010      1007 FORMAT(1H , 'ROW',I2)
0011      1008 FORMAT(1H0,6X,'FO',12X,'FC',10X,'FO-FC')
0012      1009 FORMAT(1H ,E13.6,2X,E13.6,1X,E13.6)
0013      1010 FORMAT(1H0,'SINGULAR MATRIX')
0014      WRITE(3,1000)
0015      NC=0
0016      1 CALL FUN(FO,FC,A,P,N,NOBS,SIG2)
0017      WRITE(3,1001)NC,SIG2,P(1)
0018      DO 2 I=2,N
0019      2 WRITE(3,1002)I,P(I)
0020      NC=NC+1
0021      IF(NC.GT.NCYC) GO TO 20
      C
      C      FORM A**T*A AND A**T*C
      C
0022      DO 4 I=1,N
0023      DO 3 J=I,N
0024      D(I,J)=0.0
0025      DO 3 K=1,NOBS
0026      3 D(I,J)=D(I,J)+A(K,I)*A(K,J)
0027      G(I)=0.0
0028      DO 4 K=1,NOBS
0029      4 G(I)=G(I)+A(K,I)*(FO(K)-FC(K))
0030      990 CONTINUE
      C
      C      CALCULATION OF S(I,J) FOR UPPER TRIANGULAR MATRIX
      C
0031      D(1,1)=SQRT(D(1,1))
0032      DO 5 J=2,N
0033      5 D(1,J)=D(1,J)/D(1,1)
0034      I=2
0035      6 K=I-1
0036      X=D(I,I)
0037      DO 7 L=1,K
0038      7 X=X-D(L,I)*D(L,I)
0039      IF(X.LE.0.0) GO TO 100
0040      D(I,I)=SQRT(X)
0041      M=I+1
0042      IF(M.GT.N) GO TO 10
0043      DO 9 J=M,N
0044      X=D(I,J)
0045      DO 8 L=1,K
0046      8 X=X-D(L,I)*D(L,J)
0047      9 D(I,J)=X/D(I,I)

```

FORTRAN IV G LEVEL 20

NLRP

DATE = 73292

18/27/55

```

0048      9 D(I,J)=X/D(I,I)
0049      I=I+1
0050      GO TO 6

      C
      C      CALCULATION OF K(I)
      C

0051      10 G(I)=G(I)/D(I,I)
0052      DO 12 I=2,N
0053      K=I-1
0054      X=G(I)
0055      DO 11 L=1,K
0056      11 X=X-D(L,I)*G(L)
0057      12 G(I)=X/D(I,I)

      C
      C      CALCULATION OF SOLUTIONS WHICH OVERWRITE ORIGINAL K(I)
      C

0058      G(N)=G(N)/D(N,N)
0059      M=N-1
0060      DO 14 J=1,M
0061      I=N-J
0062      X=G(I)
0063      K=I+1
0064      DO 13 L=K,N
0065      13 X=X-D(I,L)*G(L)
0066      14 G(I)=X/D(I,I)

      C
      C      APPLY SHIFTS FOR NEXT CYCLE
      C

0067      DO 15 I=1,N
0068      15 P(I)=P(I)+G(I)
0069      992 CONTINUE
0070      GO TO 1

      C
      C      END OF REFINEMENT
      C

0071      20 IF(NCYC.EQ.0) GO TO 40

      C
      C      CALCULATE R(I,J) WHICH OVERWRITE S(I,J)
      C

0072      D(N,N)=1.0/D(N,N)
0073      M1=N-1
0074      DO 22 K=1,M1
0075      I=N-K
0076      D(I,I)=1.0/D(I,I)
0077      M2=I+1
0078      M3=M2+N
0079      DO 22 M=M2,N
0080      J=M3-M
0081      X=0.0
0082      DO 21 L=M2,J
0083      21 X=X-D(I,L)*D(L,J)
0084      22 U(I,J)=X*D(I,I)

      C

```

FORTRAN IV G LEVEL 20

NLRP

DATE = 73292

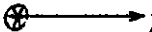
19/48/11

```

C          CALCULATION OF INVERSE WHICH OVERWRITES D(I,J)
C
0085      DO 24 I=1,N
0086      DO 24 J=1,N
0087      X=0.0
0088      DO 23 L=J,N
0089      23 X=X+D(I,L)*D(J,L)
0090      24 D(I,J)=X

C
C          PRINT PARAMETERS AND ESTIMATED STANDARD DEVIATIONS
C
0091      WRITE(3,1003)
0092      SIG2=SIG2/(NOBS-N)
0093      DO 25 I=1,N
0094      G(I)=SQRT(SIG2*D(I,I))
0095      25 WRITE(3,1004)I,P(I),G(I)
0096      WRITE(3,1005)
0097      DO 26 I=1,N
0098      WRITE(3,1007)I
0099      DO 26 J=I,N
0100      X=D(I,J)*SIG2/(G(I)*G(J))
0101      26 WRITE(3,1006)X
0102      40 WRITE(3,1003)
0103      WRITE(3,1008)
0104      DO 41 I=1,NOBS
0105      X=FO(I)-FC(I)
0106      41 WRITE(3,1009)FO(I),FC(I),X
0107      WRITE(3,1003)
0108      RETURN
0109      100 WRITE(3,1010)
0110      STOP
0111      DEBUG
0112      AT 990
0113      DISPLAY D,G
0114      AT 10
0115      DISPLAY D
0116      AT 991
0117      DISPLAY G
0118      AT 992
0119      DISPLAY G
0120      END

```



\*OPTIONS IN EFFECT\* ID,EBCDIC,SOURCE,NOLIST,NODECK,LOAD,NOMAP  
 \*OPTIONS IN EFFECT\* NAME = NLRP , LINECNT = 55  
 \*STATISTICS\* SOURCE STATEMENTS = 120,PROGRAM SIZE = 4386  
 \*STATISTICS\* NO DIAGNOSTICS GENERATED

\*STATISTICS\* NO DIAGNOSTICS THIS STEP 0

ID=0.633 HL= 51.0

CYCLE (FO-FC)\*\*2 PARAMETERS

0 0.186622E+09 P(1)=0.450000E-01  
 P(2)=0.400000E-06  
 P(3)=0.400000E-09

0B8G00#  
 D= 0.39589754E+12, 0.0 , 0.0 , 0.82854592E+17, 0.25883615E+23, 0.0  
 0.15742950E+28,G=-0.75282924E+10,-0.21546938E+16,-0.37990401E+18

0B8G01#  
 D= 629203.88 , 0.0 , 0.0 , 0.13168162E+12, 0.92431450E+11, 0.0  
 0.14490321E+14

0B8G02#  
 G= -11964.789 , -6265.7344 , 1275.0476

0B8G03#  
 G=-0.39044605E-03,-0.99564431E-07, 0.07993043E-10

1 0.762321E+07 P(1)=0.446096E-01  
 P(2)=0.380436E-06  
 P(3)=0.487993E-09

0B8G00#  
 D= 0.171057028E+12, 0.0 , 0.0 , 0.15061660E+18, 0.46273234E+23, 0.0  
 0.25126886E+28,G= 0.10464571E+10, 0.46375017E+15, 0.72828420E+17

0B8G01#  
 D= 842953.31 , 0.0 , 0.0 , 0.17067722E+12, 0.11978185E+12, 0.0  
 0.18164054E+14

0B8G02#  
 G= 1953.2007 , 958.05615 , -458.43262

0B8G03#  
 G=-0.60795387E-03, 0.16780739E-07,-0.25238450E-10

2 0.269235E+07 P(1)=0.440016E-01  
 P(2)=0.317216E-06  
 P(3)=0.462754E-09

0B8G00#  
 D= 0.65983408E+12, 0.0 , 0.0 , 0.13936557E+18, 0.42942084E+23, 0.0  
 0.23910392E+28,G= 30324320. , 0.85391164E+13, 0.15330929E+16

0B8G01#  
 D= 812301.75 , 0.0 , 0.0 , 0.17156866E+12, 0.11621656E+12, 0.0  
 0.17767109E+14

0B8G02#  
 G= 37.331345 , 18.364227 , 1.4417582

0B8G03#  
 G= 0.16564591E-04, 0.12954932E-09, 0.81147220E-13

0.99514027E+19, 0.516  
 0.15815863E+14, 0.333  
 0.15061660E+18, 0.46273234E+23, 0.0  
 0.17790558E+20, 0.876  
 0.21105033E+14, 0.416  
 0.16509837E+20, 0.822  
 0.20324742E+14, 0.407

```

&END
3 0.269060E+07 P(1)=0.440182E-01
P(2)=0.317346E-06
P(3)=0.462835E-09
&D8G00#
D= 0.95883262E+12, 0.0 , 0.0 , 0.13915735E+18, 0.42879988E+23, 0.0 , 0.16485964E+20, 0.821
0.23878643E+28,G= 18640.000 , 0.38252052E+10, 0.12240657E+13
&END
&D8G01#
D= 811685.06 , 0.0 , 0.0 , 0.17144250E+12, 0.11613549E+12, 0.0 , 0.20310783E+14, 0.407
0.17755110E+14
&END
&D8G02#
G= 0.22964571E-01,-0.96351327E-03, 0.44882566E-01
&END
&D8G03#
G= 0.15411081E-06,-0.89515700E-12, 0.25278676E-14
&END
4 0.269061E+07 P(1)=0.440183E-01
P(2)=0.317345E-06
P(3)=0.462838E-09
&D8G00#
D= 0.65883170E+12, 0.0 , 0.0 , 0.13915728E+18, 0.42879975E+23, 0.0 , 0.16485953E+20, 0.821
0.23878596E+28,G= 560.00000 , 0.201322659E+09, 0.94489281E+11
&END
&D8G01#
D= 811684.50 , 0.0 , 0.0 , 0.17144257E+12, 0.11613536E+12, 0.0 , 0.20310783E+14, 0.407
0.17755042E+14
&END
&D8G02#
G= 0.68992306E-03, 0.71506551E-03, 0.28916628E-02
&END
&D8G03#
G= 0.75427664E-08,-0.50981094E-13, 0.16286431E-15
&END
5 0.269062E+07 P(1)=0.440183E-01
P(2)=0.317345E-06
P(3)=0.462838E-09
P(1)= 0.440183E-01 E.S.D.= 0.204765E-02
P(2)= 0.317345E-06 E.S.D.= 0.125001E-07
P(3)= 0.462838E-09 E.S.D.= 0.326632E-10
CORRELATION MATRIX (TRIANGULAR PART)
ROW 1
0.100000E+01
-0.923484E+00
0.782893E+00
ROW 2
0.100000E+01
-0.916739E+00
ROW 3
0.100000E+01
0.325344E+05 -0.725406E+03
0.241780E+05 -0.398301E+03
0.282330E+05 0.405352E+03
0.316460E+05 0.314399E+05 0.206121E+03
0.338850E+05 0.336365E+05 0.248531E+03
0.354920E+05 0.353007E+05 0.191309F+03
0.360013E+05 -0.30344E+03
0.382720E+05 0.388995E+05 -0.627473E+03
0.392620E+05 0.396509E+05 -0.388906E+03
0.391730E+05 0.387379F+05 0.435078E+03
0.377440E+05 0.368109F+05 0.933082E+03
0.318090E+05 0.325344E+05 -0.725406E+03

```

8

9

10

APPENDIX F  
MACRO LISTINGS

```

MACRO
&HERE ARFAD &LIST1,&LIST2
.*
.* *****
.*
.* &CSN TABLE OF CSECT NAMES
.* &LBN TABLE OF SAVE AREA NAMES
.* &D EQUALS 0 - READ, 4 - WRITE, 8 - SECOND CALL, 12 - ARRAY,
.*      16 - LAST CALL TO IBCOM
.* &D1 EQUALS 0 - FORMATTED, 20 - UNFORMATTED
.* &DD DISPLACEMENT IN IBCOM
.* &UI UNIT TYPE 0 - INTEGER, 1 - VARIABLE NAME
.* &PI END=, ERR= OPTIONS 0 - NEITHER, 1 - END, 2 - ERR, 3 - BOTH
.* &CHAR FIRST CHARACTER OF OPERAND
.* &K COUNTER
.* &T SYMBOL TYPE CODE
.* &L OPERAND LENGTH IN BYTES
.* &I LOOP COUNTER FOR TABLES
.*
.* *****
.*
GBLC &CSN(256),&LBN(256)
GBLA &D
LCLA &D1
LCLA &DD
LCLA &UI
LCLA &PI
LCLC &CHAR
LCLA &K
LCLA &T
LCLA &L
LCLA &I
&K SETA 1
.* CHECK FOR OPERANDS
AIF (K'&LIST1 EQ 0).MOD9
CNOP 0,4
.MOD01 ANOP
&I SETA &I+1
AIF ('&SYSECT' EQ '&CSN(&I)').MOD02
AIF ('&CSN(&I)' NE '').MOD01
&CSN(&I) SETC '&SYSECT'
&LBN(&I) SETC 'IHB'.'&SYSNOX'
&HERE B ++12
&LBN(&I) DC 2A(0)
STM 14,15,&LBN(&I)
AGO .MOD06
.MOD02 ANOP
&HERE STM 14,15,&LBN(&I)
.MOD06 ANOP
L 15,=V(IBCOP#)
AIF (80 EQ 4).WT02
AIF (K'&LIST1(4) EQ 0).WT01
&PI SETA &PI+2
B ++8
ERR&SYSNOX DC A(&LIST1(4))
.WT01 ANOP
AIF (K'&LIST1(3) EQ 0).WT02
&PI SETA &PI+1
B ++8
EOF&SYSNOX DC A(&LIST1(3))

```

```

.WT02      ANOP
           AIF (K'&LIST1(2) NE 0).WT03
&D1        SETA 20
.WT03      ANOP
&CHAR      SETC '&LIST1(1)''(1,1)
           AIF ('&CHAR' GE '0' AND '&CHAR' LE '9').WT6
&UI        SETA 1
           AGO .WT6
.WT1       ANOP
&D         SETA 8
&L         SETA L'&SYSLIST(&K,1)
           AIF (&D1 EQ 20).WT5
           AIF (T'&SYSLIST(&K) EQ 'E').WT4
           AIF (T'&SYSLIST(&K) EQ 'D').WT3
           AIF (T'&SYSLIST(&K) EQ 'F').WT2
           AIF (T'&SYSLIST(&K) NE 'M').WT4
&T         SETA 4
           AGO .WT5
.WT2       ANOP
&T         SETA 5
           AGO .WT5
.WT3       ANOP
&T         SETA 6
           AGO .WT5
.WT4       ANOP
&T         SETA 7
.WT5       AIF (N'&SYSLIST(&K) EQ 1).WT6
&D         SETA &D+4
.WT6       ANOP
&DD        SETA &D+&D1
           BAL 14,&DD.(15)
           AIF (&D EQ 8).WT9
           AIF (&D EQ 12).WT8
           DC XL.4'&PI',XL.4'&UI',AL3(&LIST1(1))
           AIF (&D1 EQ 20).WT7
           DC AL1(01),AL3(FMT&LIST1(2))
.WT7       AIF (&PI EQ 0).WT10
           AIF (&PI EQ 2).WT75
           DC AL4(EOF&SYSNDX)
           AIF (&PI EQ 1).WT10
.WT75      DC AL4(ERR&SYSNDX)
           AGO .WT10
.WT8       DC AL1(0),AL3(&SYSLIST(&K,1))
           DC AL1(&L),XL.4'&T',AL.20(&SYSLIST(&K,2))
           AGO .WT10
.WT9       DC AL1(&L),XL.4'&T',XL.4'0',SL2(&SYSLIST(&K))
.WT10      ANOP
&K         SETA &K+1
.* LOOP THROUGH OPERANDS IN LIST
           AIF (K'&SYSLIST(&K) GT 0).WT1
.* LAST CALL TO IBCOM
&DD        SETA 16+&D1
           BAL 14,&DD.(15)
&D         SETA 0
           LM 14,15,&LBN(&I)
           MEXIT
.H009      MNOTE '*** WARNING - MACRO INSTRUCTION HAS NO OPERANDS ***'
&D         SETA 0
           MEND

```

```

MACRO
&HERE AWRITE &A1,&A2,&A3,&A4,&A5,&A6,&A7,&A8,&A9,&A,&B1,&B2,&B3,&J9,X
      &B4,&B5,&B6,&B7,&B8,&B9,&B,&C1,&C2,&C3,&C4,&C5,&C6,&C7,&X
      C8,&C9,&C,&D1,&D2,&D3,&D4,&D5,&D6,&D7,&D8,&D9,&K,&E1,&E2X
      ,&E3,&E4,&E5,&E6,&E7,&E8,&E9,&E,&F1,&F2,&F3,&F4,&F5,&F6,X
      &F7,&F8,&F9,&F,&G1,&G2,&G3,&G4,&G5,&G6,&G7,&G8,&G9,&G,&HX
      1,&H2,&H3,&H4,&H5,&H6,&H7,&H8,&H9,&H,&I1,&I2,&I3,&I4,&I5X
      ,&I6,&I7,&I8,&I9,&I,&J1,&J2,&J3,&J4,&J5,&J6,&J7,&J8,&J
      GBLA &D
&D SETA 4
&HERE AREAD &A1,&A2,&A3,&A4,&A5,&A6,&A7,&A8,&A9,&A,&B1,&B2,&B3,&J9,X
      &B4,&B5,&B6,&B7,&B8,&B9,&B,&C1,&C2,&C3,&C4,&C5,&C6,&C7,&X
      C8,&C9,&C,&D1,&D2,&D3,&D4,&D5,&D6,&D7,&D8,&D9,&K,&E1,&E2X
      ,&E3,&E4,&E5,&E6,&E7,&E8,&E9,&E,&F1,&F2,&F3,&F4,&F5,&F6,X
      &F7,&F8,&F9,&F,&G1,&G2,&G3,&G4,&G5,&G6,&G7,&G8,&G9,&G,&HX
      1,&H2,&H3,&H4,&H5,&H6,&H7,&H8,&H9,&H,&I1,&I2,&I3,&I4,&I5X
      ,&I6,&I7,&I8,&I9,&I,&J1,&J2,&J3,&J4,&J5,&J6,&J7,&J8,&J
MEND

```

```

MACRO
&N AFORMAT &LIST
FMT&N DC C&LIST
MEND

```

```

MACRO
&HERE ABACKSP &A
.*
.* *****
.*
.* *CSN TABLE OF CSECT NAMES
.* *LBN TABLE OF SAVE AREA NAMES
.* *D EQUALS 0 - BA&KSPACE, =4 - REWIND, =8 - ENDFILE.
.* *DD DISPLACEMENT IN IBCOM
.* *I LOOP COUNTER FOR TABLES
.* *CHAR FIRST CHARACTER OF OPERAND
.*
.* *****
.*
      GBLC &CSN(256),&LBN(256)
      GBLA &D
      LCLA &DD
      LCLA &I
      LCLC &CHAR
.* CHECK FOR OPERANDS
      AIF (K'&A EQ 0).MOD9
      CNOP 0,4
&DD SETA &D+40
.MOD01 ANOP
&I SETA &I+1
      AIF ('&SYSECT' EQ '&CSN(&I)').MOD02
      AIF ('&CSN(&I)' NE '').MOD01
&CSN(&I) SETC '&SYSECT'
&LBN(&I) SETC 'IHB','&SYSNOX'
&HERE B **12
&LBN(&I) DC 2A(0)
      STM 14,15,&LBN(&I)

```

```

      AGO      .MOD06
.MOD02 ANOP
&HERE STM     14,15,&LBN(&I)
.MOD06 ANOP
      L       15,=V(I@COM#)
      BAL     14,&DD:(15)
&CHAR SETC    '&A'(1,1)
      AIF     ('&CHAR' GT '0' AND '&CHAR' LE '9').BS1
      DC      XL1'1',AL3(&A)
      AGO     .BS2
      .BS1   DC      XL1'0',AL3(&A)
      .BS2   ANOP
&D      SETA   0
      &D     LM      14,15,&LBN(&I)
      MEXIT
.MOD9   MNOTE  '*** WARNING - MACRO INSTRUCTION HAS NO OPERANDS ***'
&D      SETA   0
      &D     MEND

```

```

&HERE   MACRO
      AENDFILE      &A
      GBLA  &D
&D      SETA  8
&HERE   ABACKSP    &A
      MEND

```

```

&HERE   MACRO
      AREWIND      &A
      GBLA  &D
&D      SETA  4
&HERE   ABACKSP    &A
      MEND

```



```

.WT01 ANOP
      AIF (K'&LIST1(3) EQ 0).WT02
&PI SETA &PI+1
      B **8
EOF&SYSNDX DC A(&LIST1(3))
.WT02 ANOP
      AIF (K'&LIST1(2) NE 0).WT03
&D1 SETA 20
.WT03 ANOP
&CHAR SETC '&LIST1(1)''(1,1)
      AIF ('&CHAR' GE '0' AND '&CHAR' LE '9').WT6
&U1 SETA 1
      AGO .WT6
.WT1 ANOP
&D SETA 8
&ICCHAR SETA 1
&ADDR SETA 0
&LENGTH SETA 0
&CHAR SETC ('&D EQ 20).WT5
      SETC '&SYSLIST(&K,1)''(1,1)
      AIF ('&CHAR' NE '#').MOD1
&ADDR SETA 1
&ICCHAR SETA &ICCHAR+1
&CHAR SETC '&SYSLIST(&K,1)''(2,1)
.MOD1 ANOP
      AIF ('&CHAR' NE 'S').MOD2
&LENGTH SETA 1
&ICCHAR SETA &ICCHAR+1
&CHAR SETC '&SYSLIST(&K,1)''(&ICCHAR,1)
.MOD2 ANOP
      AIF ('&CHAR' EQ 'I' OR ('&CHAR' GE 'J' AND '&CHAR' LE 'N')).X
      MOD3
&T SETA 7-&LENGTH
&L SETA 4+4*&LENGTH
      AGO .WT5
.MOD3 ANOP
&T SETA 5-&LENGTH
&L SETA 4-2*&LENGTH
.WT5 AIF (N'&SYSLIST(&K) EQ 1 AND &ADDR EQ 0).WT6
&D SETA &D+4
      AIF (&ADDR EQ 1).MOD6
.WT6 ANOP
&D SETA &D+&D1
      BAL 14,&DD.(15)
      AIF (&D EQ 8).WT9
      AIF (&D EQ 12).WT8
      DC XL.4'&PI',XL.4'&UI',AL3(&LIST1(1))
      AIF (&D1 EQ 20).WT7
      DC AL1(01),AL3(FMT&LIST1(2))
.WT7 AIF (&PI EQ 0).WT10
      AIF (&PI EQ 2).WT75
      DC AL4(EOF&SYSNDX)
      AIF (&PI EQ 1).WT10
.WT75 DC AL4(ERR&SYSNDX)
      AGO .WT10
.MOD6 ANOP
&D SETA &D+&D1
&CHAR SETC '&SYSLIST(&K,1)''(&ICCHAR,8)
      MVC **11(3),&CHAR+1
      BAL 14,&DD.(15)

```

```

DC      AL1(0),AL3(0)
AIF     (N'&SYSLIST(&K) GT 1).WT85
DC      AL1(&L),XL.4'&T',AL.20(1)
AGO     .WT10
.WT8    DC      AL1(0),AL3(&SYSLIST(&K,1))
.WT85   DC      AL1(&L),XL.4'&T',AL.20(&SYSLIST(&K,2))
AGO     .WT10
.WT9    DC      AL1(&L),XL.4'&T',XL.4'0',SL2(&SYSLIST(&K))
.WT10   ANOP
&K      SETA   &K+1
.* LOOP THROUGH OPERANDS IN LIST
AIF     (K'&SYSLIST(&K) GT 0).WT1
.* LAST CALL TO IBCOM
&DD     SETA   16+&D1
        BAL   14,&DD.(15)
&D      SETA   0
        LM   14,15,&LBN(&I)
        MEXIT
.MOD9   MNOTE  '*** WARNING - MACRO INSTRUCTION HAS NO OPERANDS ***'
&D      SETA   0
        MEND

```

```

MACRO
&HERE  SHWRITE &A1,&A2,&A3,&A4,&A5,&A6,&A7,&A8,&A9,&A,&B1,&B2,&B3,&J9,X
        &B4,&B5,&B6,&B7,&B8,&B9,&B,&C1,&C2,&C3,&C4,&C5,&C6,&C7,&X
        C8,&C9,&C,&D1,&D2,&D3,&D4,&D5,&D6,&D7,&D8,&D9,&K,&E1,&E2X
        ,&E3,&E4,&E5,&E6,&E7,&E8,&E9,&E,&F1,&F2,&F3,&F4,&F5,&F6,X
        &F7,&F8,&F9,&F,&G1,&G2,&G3,&G4,&G5,&G6,&G7,&G8,&G9,&G,&HX
        1,&H2,&H3,&H4,&H5,&H6,&H7,&H8,&H9,&H,&I1,&I2,&I3,&I4,&I5X
        ,&I6,&I7,&I8,&I9,&I,&J1,&J2,&J3,&J4,&J5,&J6,&J7,&J8,&J
        GBLA  &D
&D      SETA   4
&HERE  SREAD  &A1,&A2,&A3,&A4,&A5,&A6,&A7,&A8,&A9,&A,&B1,&B2,&B3,&J9,X
        &B4,&B5,&B6,&B7,&B8,&B9,&B,&C1,&C2,&C3,&C4,&C5,&C6,&C7,&X
        C8,&C9,&C,&D1,&D2,&D3,&D4,&D5,&D6,&D7,&D8,&D9,&K,&E1,&E2X
        ,&E3,&E4,&E5,&E6,&E7,&E8,&E9,&E,&F1,&F2,&F3,&F4,&F5,&F6,X
        &F7,&F8,&F9,&F,&G1,&G2,&G3,&G4,&G5,&G6,&G7,&G8,&G9,&G,&HX
        1,&H2,&H3,&H4,&H5,&H6,&H7,&H8,&H9,&H,&I1,&I2,&I3,&I4,&I5X
        ,&I6,&I7,&I8,&I9,&I,&J1,&J2,&J3,&J4,&J5,&J6,&J7,&J8,&J
MEND

```

```

MACRO
&N     SFORMAT &LIST
FMT&N DC      C&LIST
MEND

```



APPENDIX G

USING THE MACROS IN AN ASSEMBLER LANGUAGE SUBROUTINE

EXTERNAL SYMBOL DICTIONARY

SYMBOL    TYPE ID    ADDR    LENGTH LD ID  
 NLRPS    SD 01 000000 000440  
 IBCOM#   ER 02  
 SORT    ER 03

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT
000000				1 *	
000000	47F0 F054		00054	2 **	SUBROUTINE NLRPS(FD,FC,A,P,D,G,N,NOBS)
000004	05D5D3D9D7E24040			3 **	DIMENSION FD(NOBS),FC(NOBS),A(NOBS,N),P(N),D(N,N),G(N)
00000C				4 *	
000054	90EC D00C		0000C	5 NLRPS	CSECT
000058	18ED			6	B 04(0,15)
00005A	41D0 F00C			7	DC AL1(5),CL7,NLRPS,
00005E	50D0 E008			8	DS 18F
000062	50E0 D004			9	STM 14,12,12(13)
000066	9829 1000		00000	10	LR 14,13
				11	LA 13,12(0,15)
				12	ST 13,8(0,14)
				13	ST 14,4(0,13)
				14	USING NLRPS+12,13
				15	LM 2,9,0(1)

BRANCH AROUND PROGRAM ID  
 AND SAVE AREA  
 SAVE REGISTERS  
 LOAD BASE REGISTER 13  
 LOAD ADDRESSES OF ARGUMENTS

```

16 **
17 ***
18 **
19 *
20 *
21 *
22 *
23 *
24 *
25 *
26 *
27 *
28 *
29 *
30 *
31 *

```

FORM A\*\*T+A AND A\*\*T+C

```

DO 1 I=1,N
DO 2 J=1,N
D(I,J)=0.0
DO 3 K=1,NOBS
D(I,J)=D(I,J)+A(K,I)*A(K,J)
3 CONTINUE
2 CONTINUE
G(I)=0.0
DO 4 K=1,NOBS
G(I)=G(I)+A(K,I)*(FO(K)-FC(K))
4 CONTINUE
1 CONTINUE

```

00006A	9027	D344	STM	2,7	ADRF0	SAVE ARRAY ADDRESSES
00006E	50F0	8000	L	15,0	(0,8)	R15=N
000072	181F	0002	LR	1,15		R1=N
000074	8910	0002	SLL	1,2	(0)	R1=4N
000078	1801	0004	LR	8,1		R10=4
00007A	41A0	0004	LA	10,4	(0,0)	R1=4*(N-1)
00007E	1B1A		SR	1,10		R15=4*N*(N-1)
000080	1CE1		MR	14,1		R12=4*N*(N-1)
000082	18CF		LR	12,15		R14=NOBS
000084	50E0	9000	L	14,0	(0,9)	R1=4*NOBS*(N-1)
000088	1C0E		MR	0,14		R14=4*NOBS
00008A	89E0	0002	SLL	14,2	(0)	R11=A(A(1,2))
00008E	4184	E000	LA	11,0	(4,14)	R11=A(A(1,2))
000092	1B8A		SR	11,10		R15=A(A(NOBS,1))
000094	41F8	1000	LA	15,0	(11,1)	R15=A(A(NOBS,N))
000098	4196	C000	LA	9,0	(6,12)	R9=A(D(1,N))
00009C	7860	D35C	LE	6,ZERO		ZERO FPT REGISTER 6
0000A0	1814		LR	1,4		R1=A(A(1,J=I))
0000A2	1836		LR	3,6		R3=A(D(1,I))
0000A4	1854		LR	5,4		R5=A(1,I)
0000A6	3806		LER	0,6		INITIALISE SUM
0000A8	7820	5000	LE	2,0	(0,5)	LOAD A(K,I)
0000AC	7C20	1000	ME	2,0	(0,1)	MULTIPLY BY A(K,J)
0000B0	3A02		AER	0,2		INCREASE SUM

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT
000082	4110 1004		00004	56	LA 1,4(0,1)
000086	875A D09C		000A8	57	BXLE 5,10,NLRP3
00008A	7000 3000		00000	58	STE 0,0(0,3)
00008E	8738 D098		000A4	59	BXLE 3,8,NLRP2
0000C2	4168 6004		00004	60	LA 6,4(8,6)
0000C6	3006			61	LER 0,6
0000C8	1854			62	LR 5,4
0000CA	1012			63	LR 1,2
0000CC	7020 1000		00000	64	NLRP4
0000D0	7020 1000		00000	65	SE 2,0(14,1)
0000D4	7020 5000		00000	66	ME 2,0(0,5)
0000D8	3A02			67	AER 0,2
0000DA	4110 1004		00004	68	LA 1,4(0,1)
0000DE	875A D0C0		000CC	69	BXLE 5,10,NLRP4
0000E2	7000 7000		00000	70	STE 0,0(0,7)
0000E6	4170 7004		00004	71	LA 7,4(0,7)
0000EA	1A4E			72	AR 4,14
0000EC	4190 9004		00004	73	LA 9,4(0,9)
0000F0	878E 0094		000A0	74	BXLE 11,14,NLRP1
0000F4	47F0 D0F4		00100	77+	SHRUTE (3,1),(#ADDR,9)
0000F8	0000000000000000			78+	CNOP 0,4
000100	90EF D0EC		000F8	79+	B **12
000104	58F0 D42C		00438	80+	DC 2A(0)
000108	45EF 0004		00004	81+	STM 14,15,IHB0002
00010C	01000003			82+	L 15,V(IBCOM#)
000110	01000374			83+	BAL 14,4(15)
000114	D202 D113	0355 0011F	00361	84+	DC XL,4*0',XL,4*0',AL3(3)
00011A	45EF 000C		0000C	85+	MVC **11(3),ADRG+1
00011E	00000000			86+	BAL 14,12(15)
000122	04700009			87+	DC AL1(0),AL3(0)
000126	45EF 0010		00010	88+	DC AL1(4),XL,4*7',AL,20(9)
00012A	98EF D0EC		000F8	89+	LM 14,15,IHB0002
00012E	0700			90	SHRUTE (3,2),(#ADRG,3)
000130	90EF D0EC		000F8	91+	CNOP 0,4
000134	58F0 D42C		00438	92+	STM 14,15,IHB0002
000138	45EF 0004		00004	93+	L 15,V(IBCOM#)
00013C	00000003			94+	BAL 14,4(15)
000140	01000404			95+	DC XL,4*0',XL,4*0',AL3(3)
000144	D202 D143	0359 0014F	00365	96+	DC AL1(0),AL3(FMT2)
00014A	45EF 000C		0000C	97+	MVC **11(3),ADRG+1
00014E	00000000			98+	BAL 14,12(15)
000152	04700003			99+	DC AL1(0),AL3(0)
000156	45EF 0010		00010	100+	DC AL1(4),XL,4*7',AL,20(3)
00015A	98EF D0EC		000F8	101+	BAL 14,16(15)
				102+	LM 14,15,IHB0002
				103 **	
				104 ***	
				105 **	
				106 *	D(1,1)=SORT(0(1,1))
				107 *	D0 5 J=2,N
				108 *	D(1,J)=D(1,J)/D(1,1)
				109 *	5 CONTINUE
				110 *	

CALCULATION OF S(I,J) FOR UPPER TRIANGULAR MATRIX

F01MAY72 10/19/73

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT
00015E	9200 D354	00360		111	MVI ADRD,X*00'
000162	4110 D354	00360		112	LA 1,ADRD
000166	58F0 0430	0043C		113	L 15.=V(SORT)
00016A	05EF			114	BALR 14,15
00016C	5820 D354	00360		115	L 2,ADRD
000170	7000 2000	00000		116	STE 0,0(0,2)
000174	419C 2000	00000		117	LA 9,0(12,2)
000178	4138 2000	00000		118	LA 3,0(8,2)
00017C	7820 3000	00000		119	LE 2,0(0,3)
000180	3020	00000		120	DER 2,0
000182	7020 3000	00000		121	STE 2,0(0,3)
000186	8738 0170	0017C		122	BXLE 3,8,NLRP5
				123 *	
				124 *	I=2
				125 *	K=I-1
				126 *	X=D(I,I)
				127 *	DO 7 L=1,K
				128 *	X=X-D(L,I)*D(L,I)
				129 *	7 CONTINUE
				130 *	O(I,I)=SORT(X)
				131 *	M=I+1
				132 *	IF(CH.GT.N) GO TO 10
				133 *	DO 9 J=M,N
				134 *	X=D(I,J)
				135 *	DO 8 L=1,K
				136 *	X=X-D(L,I)*D(L,J)
				137 *	8 CONTINUE
				138 *	D(I,J)=X/D(I,I)
				139 *	9 CONTINUE
				140 *	I=I+1
				141 *	GO TO 6
				142 *	
00018A	4138 2000	00000		143	LA 3,0(0,2)
00018E	1868	00000		144	LR 6,8
000190	4178 9000	00000		145	LA 7,0(8,9)
000194	187A	00000		146	SR 7,10
000196	4188 2000	00000		147	NLRP6
00019A	4120 8004	00004		148	LA 11,0(8,2)
00019E	4190 9004	00004		149	LA 2,4(0,11)
0001A2	7800 2000	00000		150	LE 9,4(0,9)
0001A6	1843	00000		151	LE 0,0(0,2)
0001A8	7820 4000	00000		152	NLRP7
0001AC	3C22	00000		153	LE 2,0(0,4)
0001AE	3B02	001A8		154	MER 2,2
0001B0	874A D19C	00370		155	SER 0,2
0001B4	7000 D364	0036C		156	BXLE 4,10,NLRP7
0001B8	4110 D360	0043C		157	STE 0,ARG
0001BC	58F0 D430	00000		158	LA 1,PARML
0001C0	05EF	00000		159	L 15.=V(SORT)
0001C2	7000 2000	00000		160	BALR 14,15
0001C6	1852	00000		161	STE 0,0(0,2)
0001C8	8656 D1F0	001FC		162	LR 5,2
0001CC	4118 3000	00000		163	BXH 5,6,NLRP10
0001D0	7820 5000	00000		164	LA 1,0(8,3)
0001D4	18E1	00000		165	LE 2,0(0,5)
					LR 1,4,1

INDICATE END OF PARMLIST  
 POINT TO 1 WORD PARMLIST  
 LOAD ADDR IHCSORT  
 FORM SORT(O(1,1))  
 R2=A(D(1,1))  
 SET D(1,1)=SORT(D(1,1))  
 R9=A(D(1,N))  
 R3=A(D(1,2))  
 LOAD D(1,J)  
 FORM D(1,J)/D(1,1)  
 STORE NEW D(1,J)  
 FORM A(D(1,J+1)) AND LOOP

R3=A(D(1,2))  
 R6=4N  
 R7=A(D(N,N))+4  
 R7=A(D(N,N))  
 R11=A(D(1-1,1))  
 R2=A(D(1,1))  
 R9=A(D(1,N))  
 LOAD O(1,1)  
 R4=A(D(1,1))  
 LOAD D(L,1)  
 FORM D(L,1)\*D(L,1)  
 FORM X-D(L,1)\*O(L,1)  
 FORM A(D(L+1,1)) AND LOOP  
 STORE X  
 LOAD ADDR OF PARMLIST  
 LOAD ADDR OF IHCSORT  
 FORM SORT(X)  
 O(1,1)=SORT(X)  
 R5=A(O(1,1))  
 FORM A(D(1,1+1)) & SEE IF END  
 R1=A(D(1,1+1))  
 LOAD O(1,J)  
 R14=A(D(1,J))

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT
000106	1843			166	LR 4,3
000108	7840 4000	00000		167	LE 4,0(0,4)
00010C	7C40 E000	00000		168	ME 4,0(0,14)
0001E0	3B24			169	SER 2,4
0001E2	41E0 E004	00004		170	LA 14,4(0,14)
0001E6	874A 01CC	00108		171	BXLE 4,10,NLRP9
0001EA	3D20			172	DER 2,0
0001EC	7020 5000	00000		173	STE 2,0(0,5)
0001F0	1A18			174	AR 1,8
0001F2	8758 01C4	00100		175	BXLE 5,8,NLRP8
0001F6	1A38			176	AR 3,8
0001F8	47F0 018A	00196		177	B NLRP6
0001FC	5820 0354	00360		178 **	
000200	90EF 00EC			179 ***	CALCULATION OF K(I)
000204	58F0 042C			180 **	
000208	45EF 0004			181 *	10 G(I)=G(I)/D(I,1)
00020C	00000003			182 *	DO 12 I=2,N
000210	01000374			183 *	K=I-1
000214	0202 0213			184 *	X=G(I)
00021A	45EF 000C			185 *	DO 11 L=1,K
00021E	00000000			186 *	X=X-D(L,I)*G(L)
000222	04700009			187 *	11 CONTINUE
000226	45EF 0010			188 *	G(I)=X/D(I,1)
00022A	98EF 00EC			189 *	12 CONTINUE
00022E	5830 0358			190 *	
000232	7800 3000			191	NLRP10 L 2,ADRD
000236	7000 2000			192	SWRITE (3,1),(#ADRD,9)
00023A	7000 3000			193+	CNOP 0,4
00023E	4148 2000			194+	STM 14,15,IH0002
000242	18B4			195+	L 15,=Y(I,BCOM#)
000244	4150 3004			196+	BAL 14,4(15)
000248	4160 6004			197+	DC XL,4'0',XL,4'0',AL3(3)
00024C	41E0 8004			198+	DC AL1(0),AL3(FMT1)
000250	7800 5000			199+	MVC **11(3),ADRD+1
000254	1813			200+	BAL 14,12(15)
000256	18F4			201+	DC AL1(0),AL3(0)
000258	7820 F000			202+	DC AL1(4),XL,4'7',AL,20(9)
00025C	7C20 1000			203+	BAL 14,16(15)
000260	3802			204+	LH 14,15,IH0002
000262	4110 1004			205	L 3,ADRG
				206	LE 0,0(0,3)
				207	DE 0,0(0,2)
				208	STE 0,0(0,3)
				209	LA 4,0(8,2)
				210	LR 11,4
				211	LA 5,4(0,3)
				212	LA 6,4(0,6)
				213	LA 14,4(0,11)
				214	LE 0,0(0,5)
				215	LR 1,3
				216	LR 15,4
				217	LE 2,0(0,15)
				218	ME 2,0(0,1)
				219	SER 0,2
				220	LA 1,4(0,1)

R4=A(D(1,1))  
 LOAD D(L,1)  
 FORM D(L,I)\*D(L,J)  
 FORM X-D(L,I)\*D(L,J)  
 R14=A(D(L+1,J))  
 FORM A(D(L+1,I)) AND LOOP  
 FORM X/D(I,1)  
 STORE NEW D(I,J)  
 R1=A(D(1,J+1))  
 FORM A(D(I,J+1)) AND LOOP  
 R3=A(D(1,I+1))  
 LOOP AGAIN

R2=A(D(1,1))

R3=A(G(1))  
 LOAD G(1)  
 FORM G(1)/D(1,1)  
 STORE NEW G(1)  
 R4=A(D(1,2))  
 R11=A(D(1,2))  
 R5=A(G(2))  
 R6=4N+4  
 R14=A(D(1,1))  
 LOAD G(1)  
 R1=A(G(1))  
 R15=A(D(1,1))  
 LOAD D(L,1)  
 FORM D(L,I)\*G(L)  
 FORM X-D(L,I)\*G(L)  
 R1=A(G(L+1))

F01MAY72 10/19/73

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT
000266	87FA D24C		00258	221	BXLE 15,10,NLRP12
00026A	7000 E000		00000	222	DE 0,0(0,14)
00026E	7000 5000		00000	223	STE 0,0(0,5)
000272	4150 5004		00004	224	LA 5,4(0,5)
000276	1A48			225	AR 4,8
000278	8786 D240		0024C	226	BXLE 11,6,NLRP11
00027C				227	SHRTR (3,2),(#ADRG,3)
00027C				228	GNOP 0,4
00027C	90EF 00EC		000F8	229	STM 14,15,IH80002
000280	58F0 D42C		00438	230	L 15,V(IBC0M#)
000284	45EF 0004		00004	231	BAL 14,4(15)
000288	00000003			232	DC XL,4'0',XL,4'0',AL3(3)
00028C	01000404			233	DC AL1(01),AL3(FMT2)
000290	0202 D28F	D359	00298	234	HVC *+11(3),ADRG+1
000296	45EF 000C		0000C	235	BAL 14,12(15)
00029A	00000000			236	DC AL1(0),AL3(0)
00029E	04700003			237	DC AL1(4),XL,4'7',AL,20(3)
0002A2	45EF 0010		00010	238	BAL 14,16(15)
0002A6	98EF 00EC		000F8	239	LM 14,15,IH80002
240	**			240	**
241	**			241	**
242	**			242	**
243	*			243	*
244	*			244	*
245	*			245	*
246	*			246	*
247	*			247	*
248	*			248	*
249	*			249	*
250	*			250	*
251	*			251	*
252	*			252	*
253	*			253	*
254	*			254	*
255				255	
0002AA	185A			256	SR 5,10
0002AC	7000 E000		00000	257	DE 0,0(0,14)
0002B0	1885			258	LR 11,5
0002B2	7000 5000		00000	259	STE 0,0(0,5)
0002B6	112A			260	LNR 2,10
0002B8	183A			261	SR 3,10
0002BA	185A			262	SR 5,10
0002BC	187A			263	SR 7,10
0002BE	1897			264	LR 9,7
0002C0	7800 5000		00000	265	LE 0,0(0,5)
0002C4	1847			266	LR 4,7
0002C6	4110 5004		00004	267	LA 1,4(0,5)
0002CA	7820 4000		00000	268	LE 2,0(0,4)
0002CE	7C20 1000		00000	269	ME 2,0(0,1)
0002D2	3802			270	SER 0,2
0002D4	4110 1004		00004	271	LA 1,4(0,1)
0002D8	8748 D28E		002CA	272	BXLE 4,8,NLRP14
0002DC	18E6			273	SR 14,6
0002DE	7000 E000		00000	274	DE 0,0(0,14)
0002E2	7000 5000		00000	275	STE 0,0(0,5)
0002E6	189A			276	SR 9,10

CALCULATION OF SOLUTIONS WHICH OVERWRITE ORIGINAL K(I)

FORM A(D(L+1,I)) AND LOOP  
 FORM X/D(I,I)  
 STORE NEW G(I)  
 R5=A(G(I+1))  
 R4=A(D(I,I+1))  
 FORM A(D(I,I+1)) AND LOOP

R5=A(G(N))  
 G(N)=G(N)/D(N,N)  
 R11=A(G(N))  
 STORE NEW G(N)  
 R2=-4  
 R3=A(G(1))-4  
 R5=A(G(N-1))  
 R7=A(D(N-1,N))  
 R9=A(D(N-1,N))  
 LOAD G(N-J)  
 R4=A(D(N-J,N-J+1))  
 R1=A(G(N-J+1))  
 LOAD D(N-J,L)  
 FORM D(N-J,L)\*G(L)  
 FORM X-D(N-J,L)\*G(L)  
 R1=A(G(L+1))  
 FORM A(D(N-J,L+1)) AND LOOP  
 R14=A(D(N-J,N-J))  
 FORM X/D(N-J,N-J)  
 STORE NEW G(N-J)  
 R9=A(D(N-J-1,N))

LOC	OBJECT CODE	ADDR1	ADDR2	STMT	SOURCE STATEMENT
0002E8	1876			276	SR 7,6
0002EA	8652 D2B4	002C00		277	BXH 5,2,NLRP13
0002EE	0700			278	SHRITE (3,2),(#ADRG,3)
0002F0	90EF D0EC	000F8		279+	CNOP 0,4
0002F4	58F0 D42C	00438		280+	STM 14,15,1H80002
0002F8	45EF 0004	00004		281+	L 15,=V(18COM#)
0002FC	0000003			282+	BAL 14,4(15)
000300	01000404			283+	DC XL,4,0',XL,4,0',AL3(3)
000304	D202 D303	D359 0030F		284+	AL1(0),AL3(FMT2)
00030A	45EF 000C	0000C		285+	MVC ++11(3),ADRG+1
00030E	00000000			286+	BAL 14,12(15)
000312	04700003			287+	DC AL1(0),AL3(0)
000316	45EF 0010	00010		288+	DC AL1(4),XL,4,7',AL,20(3)
00031A	98EF D0EC	000F8		289+	BAL 14,16(15)
				290+	LM 14,15,1H80002
				291 **	
				292 ***	APPLY SHIFTS FOR NEXT CYCLE
				293 **	
				294 *	DO 15 I=1,N
				295 *	P(I)=P(I)+G(I)
				296 *	15 CONTINUE
				297 *	
00031E	5820 D350	0035C		298	L 2,ADRP
000322	4150 5004	00004		299	LA 5,4(0,5)
000326	7800 2000	00000		300	LE 0,0(0,2)
00032A	7A00 5000	00000		301	AE 0,0(0,5)
00032E	7000 2000	00000		302	STE 0,0(0,2)
000332	4120 2004	00004		303	LA 2,4(0,2)
000336	875A D31A	00326		304	0XLE 5,10,NLRP15
				305 **	
				306 ***	RETURN
				307 **	
00033A	41F0 0000	00000		308	LA 15,0(0,0)
00033E	5800 D004	00004		309	L 13,4(0,13)
000342	980C 0014	00014		310	LM 0,12,20(13)
000346	58E0 000C	0000C		311	L 14,12(0,13)
00034A	92FF D00C	0000C		312	MVI 12(13),255
00034E	07FE			313	BR 14
000350				314	ADRFO DS F
000354				315	ADRFC DS F
000358				316	ADRA DS F
00035C				317	ADRP DS F
000360				318	ADRD DS F
000364				319	ADRG DS F
000368	00000000			320	ZERO DC E'0,0E0'
00036C	80000370			321	PARNL DC X'00',AL3(ARG)
000370				322	ARG DS F
				323	1
				324	FMT1
				325	2
000374	40F1E76B70C440F1			325 2	SFORMAT '(1X,'G(1)',,E14.7,' G(2)',,E14.7,' G(3)',,E14.7X '/1X,' D(4)',,E14.7,' D(5)',,E14.7,' D(6)',,E14.7/1X X,' D(7)',,E14.7,' D(8)',,E14.7,' D(9)',,E14.7) C'(1X,'D(1)',,E14.7,' D(2)',,E14.7,' D(3)',,E14.7/X 1X,' D(4)',,E14.7,' D(5)',,E14.7,' D(6)',,E14.7/1XX ,,' D(7)',,E14.7,' D(8)',,E14.7,' D(9)',,E14.7) ,,' D(1)',,E14.7,' G(2)',,E14.7,' G(3)',,E14.7 )',

R7=A(D(N-J-1,N-J))  
FORM A(G(N-J-1)) AND LOOP

R2=A(P(1))  
R5=A(G(1))  
LOAD P(I)  
FORM P(I)+G(I)  
STORE NEW P(I)  
R2=A(P(I+1))  
FORM A(G(I+1)) AND LOOP

SOURCE STATEMENT

```

LOC  OBJECT CODE  ADDR1 ADDR2  STMT  SOURCE STATEMENT
000404 40F1E7687DC740F1  326+FMT2  DC  C'(1X,'G(1)='',E14.7,' G(2)='',E14.7,' G(3)='',E14.7X
          +
          327  END
          328  =V(IBC0M#)
          329  =V(SDRT)
    
```

RELOCATION DICTIONARY

POS.ID	REL.ID	FLAGS	ADDRESS
01	01	08	000111
01	01	08	000141
01	01	08	000211
01	01	08	000280
01	01	08	000301
01	01	08	000360
01	02	1C	000438
01	03	1C	00043C

CROSS-REFERENCE

10/19/73

SYMBOL	LEN	VALUE	DEFN	REFERENCES
ADRA	0004	000358	00316	
ADRD	0004	000360	00318	0004 0111 0112 0115 0191 0199
ADRFC	0004	000354	00315	
ADRFO	0004	000350	00314	0032
ADRC	0004	000364	00319	0097 0205 0234 0285
ADRP	0004	00035C	00317	0098
ARG	0004	000370	00322	0321
FMT1	0014	000374	00324	0083 0198
FMT2	0049	000404	00326	0096 0233 0284
IHB0002	0004	0000F8	00078	0079 0089 0092 0102 0194 0204 0229 0239 0280 0290
NLRPS	0001	000000	00005	0014
NLRP1	0002	0000A0	00049	0074
NLRP10	0004	0001FC	00191	0162
NLRP11	0004	00024C	00213	0226
NLRP12	0004	000258	00217	0221
NLRP13	0004	0002C0	00264	0277
NLRP14	0004	0002CA	00267	0271
NLRP15	0004	000326	00300	0304
NLRP2	0002	0000A4	00051	0059
NLRP3	0004	0000A8	00053	0057
NLRP4	0004	0000CC	00064	0069
NLRP5	0004	00017C	00119	0122
NLRP6	0004	000196	00147	0177
NLRP7	0004	0001A8	00152	0155
NLRP8	0004	000100	00164	0175
NLRP9	0004	0001D8	00167	0171
PARML	0001	00036C	00321	0157
ZERO	0004	000360	00320	0048

NO STATEMENTS FLAGGED IN THIS ASSEMBLY  
 \*STATISTICS\* SOURCE RECORDS (SYSIN) = 266 SOURCE RECORDS (SYSLIB) = 156  
 \*OPTIONS IN EFFECT\* LIST, NODECK, LOAD, MORENT, XREF, NOTEST, ALGN, OS, NOTERM, LINECNT = 55  
 399 PRINTED LINES

\*\*\*\*\*

```

FORTRAN IV G LEVEL 20          MAIN          DATE = 73292          19/55/10
0001  DIMENSION FO(11),FC(11),A(11,3),P(3)
0002  COMMON XMV(11),HL,XLD,DL
0003  FORMAT(F5.3,F5.1)
0004  1000 FORMAT(1H0,'ID=',F5.3,' HL=',F5.1)
0005  1001 FORMAT(1H0,'ID=',F5.3,' HL=',F5.1)
0006  1002 FORMAT(15,3E10.3)
0007  1003 FORMAT(2F10.1)
0008  N=3
0009  NOBS=11
0010  READ(1,1000)XID,HL
0011  DL=XID/HL
0012  XLD=HL/XID
0013  WRITE(3,1001)XID,HL
0014  READ(1,1002)NCYC,P(1),P(2),P(3)
0015  READ(1,1003)(FO(I),XMV(I),I=1,NOBS)
0016  CALL NLRP(FO,FC,A,P,N,NOBS,NCYC)
0017  STOP
0018  END

```

```

FORTRAN IV G LEVEL 20          FUN          DATE = 73292          19/55/10
0001  SUBROUTINE FUN(FO,FC,A,P,N,NOBS,SIG2)
0002  DIMENSION FO(NOBS),FC(NOBS),A(NOBS,N),P(N)
0003  COMMON XMV(11),HL,XLD,DL
0004  SIG2=0.0
0005  DO 10 I=1,NOBS
0006  G=XMV(I)
0007  A1=HL*G
0008  B=DL*G*G
0009  C=P(1)*XLD+P(2)*A1+P(3)*B
0010  FC(I)=G/C
0011  D=-FC(I)/C
0012  A(I,1)=XLD*D
0013  A(I,2)=A1*D
0014  A(I,3)=B*D
0015  T=FO(I)-FC(I)
0016  10 SIG2=SIG2+T*T
0017  RETURN
0018  END

```



## C PRINT PARAMETERS AND ESTIMATED STANDARD DEVIATIONS

```
0044 WRITE(3,1003)
0045 SIG2=SIG2/(NOBS-N)
0046 DO 25 I=1,N
0047 G(I)=SQRT(SIG2*D(I,I))
0048 WRITE(3,1004)I,P(I),G(I)
0049 WRITE(3,1005)
0050 DO 26 I=1,N
0051 WRITE(3,1007)I
0052 DO 26 J=1,N
0053 X=D(I,J)*SIG2/(G(I)*G(J))
0054 WRITE(3,1006)X
0055 WRITE(3,1003)
0056 WRITE(3,1008)
0057 DO 41 I=1,NOBS
0058 X=FO(I)-FC(I)
0059 WRITE(3,1009)FO(I),FC(I),X
0060 RETURN
0061 WRITE(3,1010)
0062 STOP
0063 END
0064
```

FORTRAN IV G LEVEL 20

NLRP

DATE = 73292

19/55/10

```

*OPTIONS IN EFFECT* ID,EBCDIC,SOURCE,NOLIST,NODECK,LOAD,NOMAP
*OPTIONS IN EFFECT* NAME = NLRP , LINECNT = 55
*STATISTICS* SOURCE STATEMENTS = 64,PROGRAM SIZE = 2534
*STATISTICS* NO DIAGNOSTICS GENERATED

*STATISTICS* NO DIAGNOSTICS THIS STEP 0

```

ID=0.633 HL= 51.0

CYCLE (FO-FC)\*\*2 PARAMETERS

```

0 0.186622E+09 P(1)=0.450000E-01
P(2)=0.400000E-06
P(3)=0.400000E-09
D(1)= 0.3958975E+12 D(2)=-0.9382968E+18 D(3)= 0.1778772E+57
D(4)= 0.8285459E+17 D(5)= 0.2588361E+23 D(6)=-0.5206460E+19
D(7)= 0.9951403E+19 D(8)= 0.5167965E+25 D(9)= 0.1574295E+28
G(1)=-0.7528292E+10 G(2)=-0.2154694E+16 G(3)=-0.3799040E+18
D(1)= 0.6292039E+06 D(2)=-0.9382968E+18 D(3)= 0.1778772E+57
D(4)= 0.1316816E+12 D(5)= 0.9243145E+11 D(6)=-0.5206460E+19
D(7)= 0.1581586E+14 D(8)= 0.3337940E+14 D(9)= 0.1449032E+14
G(1)=-0.1196479E+05 G(2)=-0.6265734E+04 G(3)= 0.1275048E+04
G(1)=-0.3904460E-03 G(2)=-0.9956443E-07 G(3)= 0.8799304E-10
1 0.762321E+07 P(1)=0.446096E-01
P(2)=0.300436E-06
P(3)=0.487993E-09
D(1)= 0.7105703E+12 D(2)=-0.9382968E+18 D(3)= 0.1778772E+57
D(4)= 0.1506166E+18 D(5)= 0.4627323E+23 D(6)=-0.5206460E+19
D(7)= 0.1779056E+20 D(8)= 0.8763657E+25 D(9)= 0.2512689E+28
G(1)= 0.1646457E+10 G(2)= 0.4637502E+15 G(3)= 0.7202842E+17
D(1)= 0.8429533E+06 D(2)=-0.9382968E+18 D(3)= 0.1778772E+57
D(4)= 0.1786772E+12 D(5)= 0.1197819E+12 D(6)=-0.5206460E+19
D(7)= 0.2110503E+14 D(8)= 0.4168133E+14 D(9)= 0.1816405E+14
G(1)= 0.1953201E+04 G(2)= 0.9580562E+03 G(3)=-0.4584326E+03
G(1)=-0.6079539E-03 G(2)= 0.1678074E-07 G(3)=-0.2523845E-10
2 0.269235E+07 P(1)=0.440016E-01
P(2)=0.317216E-06
P(3)=0.462754E-09
D(1)= 0.6598341E+12 D(2)=-0.9382968E+18 D(3)= 0.1778772E+57
D(4)= 0.1393656E+18 D(5)= 0.4294200E+23 D(6)=-0.5206460E+19
D(7)= 0.1650984E+20 D(8)= 0.8225350E+25 D(9)= 0.2391039E+28
G(1)= 0.3032432E+08 G(2)= 0.8539116E+13 G(3)= 0.1533093E+16
D(1)= 0.8123018E+06 D(2)=-0.9382968E+18 D(3)= 0.1778772E+57
D(4)= 0.1715687E+12 D(5)= 0.1162166E+12 D(6)=-0.5206460E+19
D(7)= 0.2032474E+14 D(8)= 0.4077097E+14 D(9)= 0.1776719E+14
G(1)= 0.3733134E+02 G(2)= 0.1836423E+02 G(3)= 0.1441758E+01
G(1)= 0.1656459E-04 G(2)= 0.1295493E-09 G(3)= 0.8114722E-13
3 0.269060E+07 P(1)=0.440182E-01
P(2)=0.317346E-06
P(3)=0.462835E-09
D(1)= 0.6588326E+12 D(2)=-0.9382968E+18 D(3)= 0.1778772E+57
D(4)= 0.1391574E+18 D(5)= 0.4287999E+23 D(6)=-0.5206460E+19
D(7)= 0.1648596E+20 D(8)= 0.8213985E+25 D(9)= 0.2387864E+28
G(1)= 0.1864000E+05 G(2)= 0.3825205E+10 G(3)= 0.1224066E+13
D(1)= 0.8116851E+06 D(2)=-0.9382968E+18 D(3)= 0.1778772E+57
D(4)= 0.1714425E+12 D(5)= 0.1161355E+12 D(6)=-0.5206460E+19
D(7)= 0.2031078E+14 D(8)= 0.4074424E+14 D(9)= 0.1775511E+14
G(1)= 0.2296457E-01 G(2)=-0.9635133E-03 G(3)= 0.4488257E-01
G(1)= 0.1541100E-06 G(2)=-0.8951570E-12 G(3)= 0.2527868E-14
4 0.269061E+07 P(1)=0.440183E-01
P(2)=0.317345E-06

```

P(3)=0.462838E-09  
 D(1)= 0.6588317E+12 D(2)=-0.9382968E+18 D(3)= 0.1778772E+57  
 D(4)= 0.1391573E+18 D(5)= 0.4287997E+23 D(6)=-0.5286460E+19  
 D(7)= 0.1648595E+20 D(8)= 0.8213975E+25 D(9)= 0.2387860E+28  
 G(1)= 0.5680000E+03 G(2)= 0.2813266E+09 G(3)= 0.9448928E+11  
 O(1)= 0.8116845E+06 D(2)=-0.9382968E+18 D(3)= 0.1778772E+57  
 D(4)= 0.1714426E+12 D(5)= 0.1161354E+12 D(6)=-0.5286460E+19  
 D(7)= 0.2031878E+14 D(8)= 0.4074421E+14 D(9)= 0.1775504E+14  
 G(1)\* 0.6899231E-03 G(2)= 0.7150655E-03 G(3)= 0.2891663E-02  
 G(1)= 0.7542766E-08 G(2)=-0.5098109E-13 G(3)= 0.1628643E-15  
 5 0.269962E+07 P(1)=0.448183E-01  
 P(2)=0.317345E-06  
 P(3)=0.462838E-09

P(1)= 0.448183E-01 E.S.D.= 0.284765E-02  
 P(2)= 0.317345E-06 E.S.D.= 0.125001E-07  
 P(3)= 0.462838E-09 E.S.D.= 0.326632E-10

CORRELATION MATRIX (TRIANGULAR PART)

ROW 1  
 0.100000E+01  
 -0.923484E+00  
 0.782893E+00  
 ROW 2  
 0.100000E+01  
 -0.916739E+00  
 ROW 3  
 0.100000E+01

	FC	F0-FC
0.241788E+05	0.245763E+05	-0.398301E+03
0.289238E+05	0.285176E+05	0.485352E+03
0.316468E+05	0.314399E+05	0.206121E+03
0.338858E+05	0.336365E+05	0.248531E+03
0.354920E+05	0.353007E+05	0.191389E+03
0.356920E+05	0.360813E+05	-0.309344E+03
0.382780E+05	0.388995E+05	-0.627473E+03
0.392620E+05	0.396509E+05	-0.388906E+03
0.391730E+05	0.387379E+05	0.435078E+03
0.377440E+05	0.368109E+05	0.933882E+03
0.318890E+05	0.325344E+05	-0.725486E+03

\*\*\*\*\*