CLM-P689

CLM-P689

UKAEA

Preprint

THE OLYMPUS FORTRAN GENERATOR

K. V. ROBERTS
M. H. HUGHES

CULHAM LABORATORY
Abingdon Oxfordshire

1982

CLM-P 689

THE OLYMPUS FORTRAN GENERATOR

M H Hughes and K V Roberts

Culham Laboratory
Abingdon, Oxford, OX14 3DB, UK
(EURATOM/UKAEA Fusion Association)

Abstract

This article describes the structure and use of the GENSIS generator program. This is a facility designed to automatically construct documentation modules, COMMON blocks and a number of standard OLYMPUS subprograms. COMMON variables and arrays can be reorganised simply by updating a Master Index and running GENSIS. Regenerated modules are automatically incorporated in the Standard Program File.

(Submitted for publication in Computer Physics Communications)

November 1982

PROGRAM SUMMARY

Title of Program: GENSIS

Catalogue number: ACEB

Program obtainable from: CPC Program Library, Queen's University of Belfast, N.Ireland (see application form in this issue)

Computer: PRIME 750. Installation: Culham Laboratory. The program can be used on any computer equipped with the OLYMPUS System, and IBM support routines are provided.

Operating System: PRIMOS

Programming Language: ANSI Fortran 66

High Speed Storage required: 3 PRIME segments of 64K words each (Minimum size program)

No. of bits in a word: 32

Peripherals used: disc

Number of lines in combined program and test deck: 7500

CPC Library programs used (alternatives):

Catalogue No.	*Title*	*Ref. in CPC*
ABUF	OLYMPUS (ICL 4/470)	7(1974) 245
ABUJ	OLYMPUS (IBM 370/165)	9(1975) 51
ABUK	OLYMPUS (CDC 6500)	10(1975) 167

Keywords OLYMPUS, Fortran, Documentation, Automatic Code Generation, Word-processing, Text formatting

Name of problem

The GENSIS generator is used to construct documentation modules, COMMON blocks and other standard components of an OLYMPUS program, using as input a free-format master index file, MINDEX. This enables the OLYMPUS conventions to be maintained throughout the development phase with very little work on the part of the programmer. It can also be used in the reorganisation of existing non-OLYMPUS Fortran programs.

Method of solution

The MINDEX input file is read in A-format, and combined with a default file on channel NOLYMP that contains standard installation-dependent and OLYMPUS information. Strings are manipulated in both packed and unpacked form, and a system-independent ASCII collating sequence is used for the alphanumeric ordering of identifiers in indexes and COMMON blocks.

Restrictions on the complexity of the problem

GENSIS is written in OLYMPUS form in ANSI Fortran 66 and should run on any type of computer system provided that the OLYMPUS system is installed and that suitable character packing and unpacking routines are available. Versions of these routines are provided for the PRIME and IBM computers. The table sizes can be extended if required.

Typical running time

7.5 seconds/100 lines of input on the PRIME 750.

LONG WRITEUP

1. INTRODUCTION

Because OLYMPUS programs [1,2] have a well-defined standard structure (ref. [3], denoted in the following by II), it follows that many 'house-keeping' sections of a new program can be constructed automatically by the computer itself, thus relieving the programmer of tedious, unnecessary work. The OLYMPUS FORTRAN GENERATOR has been designed for this purpose and the way in which it is used is illustrated in Fig.1.

The programmer is simply asked to establish a Master Index File, MINDEX which contains enough information about the program and the COMMON data structure to allow the GENSIS generator to construct those sections of the Standard Program File (SPF - see II) that depend on this information. MINDEX itself can be in a convenient relatively-free format suitable for typing on-line.

2. FILE CONSTRUCTED BY GENSIS

The published version of GENSIS constructs the following files:

1. Documentation	TITLE	Title page
	INDSUB	Index of subprograms
	INDCOM	Index of COMMON blocks
	INDVAR	Alphabetic index of COMMON variables
	INDBLK	Index of COMMON variables by block
2. COMMON blocks		
3. Fortran routines	0.5	RECORD
	1.1	LABRUN
	1.2	CLEAR
	1.4	DATA
	5.2	CLIST
	5.3	ARRAYS
4. Fortran templates		
	1.3	PRESET
	1.5	AUXVAL

according to the format defined in II.

3. ADVANTAGES

During the development of the program one can add, remove or reorganize COMMON variables and arrays, or change array dimensions, simply by updating MINDEX and then running GENSIS to regenerate the appropriate modules which will then be incorporated automatically in the SPF (Standard Program File). The templates for PRESET and AUXVAL are used only during the initial construction and on the Culham PRIME system it is arranged that these subroutines are not afterwards overwritten by subsequent runs of GENSIS.

In principle it is now possible to exchange and publish OLYMPUS programs in a more concise form than hitherto, since MINDEX is shorter and more convenient than those sections of the program that it generates. However for the time being the full SPF format is provided (II, Table 1).

4. USING GENSIS

The user prepares a file MINDEX and then, on the Culham PRIME, types the command

GENSIS MINDEX

when the relevant modules will be created.

GENSIS recognizes a number of keywords and phrases (in upper-case letters) in MINDEX, and these are shown in Table 1. It analyses MINDEX in two stages 1 & 2 as indicated in the table, of which only Stage 2 is mandatory. Thus the only essential key phrase is

INDEX OF COMMON BLOCKS

and this must follow those of Stage 1 if they are present. An example of a short MINDEX file is shown in Fig.5. Here and in Table 1 only the underlined characters are checked for.

GENSIS does not require the user to choose any particular columns for input but entries should not extend beyond col.72. In general, one or more blank characters delimit fields while completely blank lines terminate sections of the MINDEX file. It is not in general necessary for the subprograms and variables to be in any particular order. The generator will perform an alpha-numeric sort according to the ASCII collating sequence and the order in the various output modules will be that preferred by the OLYMPUS methodology.

4.1. Stage 1

The order of the keywords or phrases in Stage 1 is arbitrary but that shown in Table 1 and discussed below is a logical one.

4.1.1 Title Page

The first group of 7 keywords in Table 1 is used to construct the title page, examples of which are shown in II, Figs.2a and 2b.

PROGRAM

A name of not more than 12 characters including blanks, typed on the same line.

TITLE

50 characters maximum, typed on the same line.

AUTHORS

50 characters maximum, typed on the same line.

ADDRESS

The default file OLYIND, of which an example is provided in the GENSIS package, contains a standard address appropriate for the local installation. If a different address is supplied, it will overwrite the default and should contain lines of 30 characters maximum each,starting on the same line as the keyword and terminated by a blank line.

FINAL

The default file OLYIND contains two alternative standard versions of disclaimer information, corresponding to a preliminary or final published status for the code, of which the Culham examples are shown in II, Figs. 2a and 2b. The normal choice is the preliminary version, but the keyword FINAL causes the version 2b for a published code to be selected.

DISCLAIMER

If an explicit disclaimer is provided it will overwrite either OLYIND version. It should contain lines of 60 characters maximum, starting on the same line as the keyword, and terminated by a blank line. The first non-blank character will be printed in col.7 , so that if indentation is required a dummy character should be inserted in this position and edited out afterwards.

VERSION

50 characters maximum, typed on the same line. This will be used in all generated modules.

4.1.2 Index of Subprograms

The next group of 2 keywords, is used to construct the INDSUB module. The key phrase

INDEX OF SUBPROGRAMS

should be followed by a series of 1-line free-format entries, one for each subprogram included in the program deck, in the form:

<decimal classification> <entryname(<parameters>)> <subprogram label>

Typical examples might be:

```
2.9  PSTAT       Process executable statement
3.10 INOUTI(1)   Input or output subroutine index
```

Blanks are used as delimiters, and the number of formal parameters (if any) in brackets follows the entryname. The label defines the purpose and should not exceed 44 characters. The order is arbitrary because the entries will be sorted alphanumerically according to their decimal classifications using the ASCII collating sequence, and appropriate subheadings will be automatically generated.

The key phrase

ADDITIONAL SUBPROGRAMS

if present, should be followed by a second series of one-line entries in a different, shorter format omitting the decimal classification:

<entryname (<parameters>)> <subprogram label>

again with blank as a delimiter, and the label not exceeding 44 characters. A typical example might be

CONTRA(10) GHOST contour routine (see CLM-R177)

This facility is intended for those subprograms that are referenced in the program but are not provided in the deck itself. They may for example include OLYMPUS routines, mathematical or graphical library routines, system routines and so on. The decimal classification is omitted since it may have no meaning in this case, and the order of the entries is not changed by sorting.

4.1.3 Array Dimensions

The key phrase

ARRAY DIMENSIONS

is followed by a sequence of entries, of which a typical example might be

```
MAXDIM = 100, NX
```

In the Index of Common Variables, analysed in Stage 2, arrays can have either numerical or symbolic values. In the latter case the first parameter, 100,is the explicit numerical size to be inserted in COMMON blocks while the second,NX , is the symbolic size to be inserted in subroutine <5.3> ARRAYS, e.g.

```
CALL RARRAY(8HEFIELD , EFIELD, NX)
```

This enables a run to be made with less than the maximum array size, restric ting the diagnostic printing to only those array elements that are actually used in the run.

4.2. Stage 2

The key phrase

INDEX OF COMMON VARIABLES

is mandatory and must follow any key word or phrase in Stage 1. This key line is followed by sequence of COMMON block indexes, each of which is preceded by a sub-heading line with S as the first non-blank character, in the form:

```
S  <decimal identification>  <block subheading>
```

with the subheading not exceeding 44 characters, a typical example being:

```
S  2.1  Tokamak parameters
```

This must be followed by a second name line in the form:

```
COMMON/<block name>/< optional module name> /<optional VERSION>
```

for example:

```
COMMON/COMTOK/
```

or

```
COMMON/COMTOK/COMTKA/VERSION 1* MHH 1/Oct/82 Culham
```

If the module name is omitted it is taken to be the same as the block name. Subsequently the individual entries for each variable or array should appear in the format:

```
<type>   <identifier(<dimensions>)  <purpose>
```

with blank as a separator, and the purpose not exceeding 44 characters. The variable type comprises one or two characters with C for complex, D for double

precision, R for real, I for integer, L for logical, A for array.

Examples are:

```
R  RIPMAX     **Maximum ripple amplitude
RA Q(MAXMSH)   Safety factor (defined at half points)
R  AMASSI      *Ion mass (AMU)
I  NSPLIT       Maximum number of splitting surfaces
RA CRAD(6,4,MAXIMP) Polynomial coefficient (radiation)
```

A single asterisk * causes the variable name to be included in NAMELIST NEWRUN in subroutine <1.4>DATA, while a double asterisk causes it to be included also in NAMELIST RESET.

The COMMON block indexes must be in decimal order, each preceded by its subheading line and name line, and terminated by a blank line. Blank COMMON is conventionally block [C9.0] and the name line should take a form such as:

```
COMMON/ /COMBLA/
```

where COMBLA is the module name.

Within each block the entries may be in any order and are sorted into the preferred order by GENSIS, i.e. C, D, R, I, L with alphanumeric ordering within each type.

The last key phrase is:

```
ADDITIONAL COMMON BLOCKS
```

and the intention is that this should be followed by details of all the variable and array names that are referenced in the SPF but are not included in the COMMON blocks that it contains. Each block is preceded by a sub-heading line, which takes the shortened form:

```
S  <block subheading>
```

with the subheading not exceeding 44 characters, and by a name line. The last three characters of the block name will be used in the place of the decimal identification when constructing the index INDVAR.

5. STRUCTURE OF THE GENERATOR PROGRAM

The GENSIS package has the OLYMPUS Standard Program File (SPF) form shown in paper II, Table 1 and the program runs under the control of subprogram <0.3> COTROL of the OLYMPUS control and utility package [1 2]. This package also provides the COMMON blocks [C1.1]COMBAS and [C1.9]COMDDP and a number of utility subroutines.

Although not a physics program, GENSIS has the OLYMPUS structure and consists of 3 main parts whose functions are as follows:

1. Prologue	Initialize data structure and read Stage 1 input
2. Calculation	Read and process Stage 2 input
3. Output	Generate and output the modules

The class 3 routines <3.2> GTITLE and <3.18> INOUTI that generate the title page and subroutine index respectively are called directly from the Prologue, as shown in Figs. 2 and 3. All others are called by <3.1> OUTPUT, as shown in Fig.4. The run only has a single step, after which it terminates.

The CPC Library program deck contains detailed indexes and other documentation, together with the MINDEX master index module to enable any necessary modifications to GENSIS to be made using the program itself. However, many extensions and adaptations can often be made by using the OLYMPUS EXPERT facility described in Appendix 2 of the previous paper ([4], referred to as III), and this method is to be preferred since it allows the SPF to remain unaltered.

Utility routines are listed in Table 2, but their calls are not shown in Figs.2-4. Those in Class Z are part of the SPF and are system-independent. Routines that are system-dependent or may need to be modified are contained in a SUPPORT package that is placed after the end of the SPF. The table also indicates the OLYMPUS routines that are explicitly called.

Note that in order to conform to the rules of ANSI Fortran 66 [5] a dummy main program is provided whose purpose is to reference all the COMMON blocks and so ensure that all COMMON variables and arrays remain defined throughout the run. This replaces the main program of the OLYMPUS package [2], and calls a new OLYMPUS subprogram <0.0> MASTER which for convenience is also provided in the SUPPORT. If extra COMMON blocks are added they should also be included in the main program.

The SUPPORT package contains a version of <0.2> MODIFY which sets the appropriate channel numbers for the PRIME, and dummies which suppress OLYMPUS messages. It also contains a specimen version of file OLYIND. Subroutine AREAD uses a dialect Fortran statement which may need to be replaced for use on other types of computer (although it is now part of ANSI Fortran 77).

Subroutines PACK and UNPACK are written in assembly language and versions are provided for the PRIME and IBM computers. PACK is used to pack a sequence of small integers, contained in an array, into successive bytes starting from a given base address, while UNPACK is used to expand a string of bytes into an array of small integers.

Most string manipulations are carried out using the 'hardware' character representation of the computer, in either packed or unpacked form. Although this is system-dependent, (two widely-used representations being ASCII and EBCDIC), for most purposes it is unnecessary for GENSIS to know which representation is being used. The collating sequence of blank, numeric digits and upper-case alphabetic letters is important, however, because it affects the order in which GENSIS places variable and array identifiers in the automatically -generated COMMON blocks and indexes, and this should be the same on all computer systems. The sequence adopted is:

(blank), (digits 0-9), (letters A-Z)

which is the same as that of ASCII, although the numerical values used are different and the order of the special characters is not the same.

An 'internal' character set is constructed in §1.1 of <1.3> PRESET. The array NHARD (1-MAXCHR) contains the hardware code, mapped against the internal representation, and the array NINTRN (1-256) contains the internal code, mapped against the hardware representation. These arrays can be used to convert from

one unpacked representation to another using subroutine <Z19> COLATE. Conversion from the packed hardware representation to the packed internal representation is made using <Z20> ASCII which calls in sequence UNPACK, COLATE, PACK. Currently MAXCHR = 47 corresponding to 10 digits, 26 letters, and 11 special characters including blank. The hardware code is stored in COMMON blocks [C6.1] COMCHR and [C6.3] COMSPC, while the internal code is stored in blocks [C6.2] COMINT and [C6.4] COMSPA.

5.1. Prologue

Subroutine <1.3> PRESET sets up character codes and strings, defines parameters and buffer sizes, and clears buffers to blank.

Subroutine <1.4> DATA reads and analyses Stage 1 of the MINDEX input file, searching for key words and phrases and packing the subsequent information into the appropriate arrays. It terminates when a phrase beginning INDEX OF COMMON is found.

Subroutine <1.5> AUXVAL establishes the actual or symbolic dimensions to be used in generating the module C553 ARRAYS.

The construction of the TITLE and INDSUB modules is controlled by <1.8> START, using a combination of the Stage 1 input data read by subroutine DATA with standard and default data contained in a file OLYIND on channel NOLYMP. This enables installation-dependent as well as OLYMPUS information to be automatically included in the documentation modules.

5.2 Calculation

Subroutine <2.1> STEPON reads and processes the blocks of COMMON data, calling the main routines shown in Fig.3. Each block should contain a subheading processed by <2.3> HEAD, a COMMON name processed by <2.2> COMNAM, and a list of variables and arrays processed by <2.4>STORE.

5.3. Output

Subroutine <3.1> OUTPUT controls the generation and output of the remaining modules, calling the main routines shown in Fig.4. The working of these routines should be clear from the comments in the program listing.

6. DATA STRUCTURE

In addition to the standard OLYMPUS COMMON blocks [C1.1] COMBAS and [C1.9] COMDDP, there are 12 labelled COMMON blocks listed in Table 3. These are contained in Section 2 of the SPF, and are referenced in Section 3 by C/ INSERT statements. In order to compile GENSIS the necessary insertions must be made either automatically or by hand.

Block [C4.1] COMCHN contains chaining variables and arrays that are used in the alphabetic sorting and merging of identifiers in indexes and COMMON blocks. Block [C4.2] COMVAR deals with the variable and array index, while block [C4.3] COMBYT contains variables that define the numbers of characters in various strings or fields.

Block [C5.1] COMIO contains a work area, buffers, together with size and channel information. The hardware and internal character representations are contained in blocks [C6.1] to [C6.4] as already explained in §5, while [C6.5] COMCOL is concerned with conversion from one to the other.

Block [C6.6] COMKEY contains a number of keywords and buffers, [C6.8] COMCNT contains counters and logical variables concerned with the files under construction, and [C6.9] COMARR contains information about arrays.

7. TEST RUN

Section 4 of the SPF contains the MINDEX input file for a test run, shown in Fig.5, parts of the output being reproduced in Fig.6.

REFERENCES

[1] An Introduction to the OLYMPUS System, K.V.Roberts, Comput. Phys. Commun. 7 (1974) 237.

[2] OLYMPUS - A Standard Control and Utility Package for Initial Value Fortran Programs, J.P.Christiansen and K.V.Roberts Comput. Phys. Commun. 7 (1974) 245.

[3] OLYMPUS Conventions, M.H.Hughes and K.V.Roberts, CLM-P685, submitted for publication in Comput. Phys. Commun.

[4] The OLYMPUS FORTRAN COMPOSITOR, M.H.Hughes and K.V.Roberts, CLM- P690, submitted for publication in Comput. Phys. Commun.

[5] USA Standard Fortran, USA X3.9 - 1966, (USA Standards Institute, New York, March 1966).

TABLE 1

Key Words and Phrases

Stage 1

PROGRAM	(12 characters maximum)
TITLE	(50 characters maximum)
AUTHORS	(50 characters maximum)
ADDRESS	(30 characters maximum/line)
FINAL DISCLAIMER	(60 characters maximum/line)
VERSION	(50 characters maximum)

INDEX OF SUBPROGRAMS

ADDITIONAL SUBPROGRAMS

ARRAY DIMENSIONS

Stage 2

INDEX OF COMMON BLOCKS

ADDITIONAL COMMON BLOCKS

Notes

Keywords must be in upper case
Only underlined characters are searched for
The starting column is arbitrary and blanks are ignored
Sections of information are terminated by a blank line

Table 2. Subsidiary Routines

GENSIS Program			
Z1	SAME(6)	Compare character strings	L
Z2	FETCH(1)	Fetch element from free list	
Z3	ADFREE(1)	Return element to free list	
Z4	NFIND(4)	Find specified character	I
Z5	CREAD(4)	Fetch card image	
Z6	CWRITE(2)	Output card image	
Z7	NCOLUM(4)	Locate first or last non-blank character	I
Z8	SIZE(4)	Convert symbolic dimensions	
Z9	COUNT(2)	Count number of array elements	
Z10	CONVRT(3)	Convert unpacked characters to integer or vice-versa	
Z11	ORDER	Sort variables into alphabetic order	
Z12	MERGE(4)	Merge chains	
Z13	CDRIL	Construct *CDRIL* order	
Z14	BEFORE(2)	Compare order of character strings	L
Z15	ADDRES	Fetch address and disclaimer	
Z16	BLANK(3)	Clear array to blank	
Z17	VALID(3)	Check for valid Fortran identifier	L
Z18	ASCEND(2)	Check for ascending order	
Z19	COLATE(4)	Change collating sequence	
Z20	ASCII(3)	Convert to internal collating sequence	
Z21	HDWARE(3)	Convert to hardware collating sequence	
SUPPORT Package			
	(Main)	Defines all COMMON blocks	
0.0	MASTER	OLYMPUS master subprogram	
0.2	MODIFY	Modify basic data if required	P
1.6	INITAL	Dummy to suppress OLYMPUS message	
4.1	TESEND	Dummy to suppress OLYMPUS message	
4.2	ENDRUN	Dummy to suppress OLYMPUS message	
A1	AREAD(4)	Read card image	SIP
A2	APUNCH(2)	Output card image	SIP
A3I	PACK(4)	Pack character string	AI
A3P	PACK(4)	Pack character string	AP
A4I	UNPACK(4)	Unpack character string	AI
A4P	UNPACK(4)	Unpack character string	AP

0.1	BASIC	Initialize basic control data
0.3	COTROL	Control the run
0.4	EXPERT(3)	Modify standard operation of program
U1	MESAGE(1)	Print 48-character message on output channel
U3	BLINES(1)	Insert blank lines on output channel
U15	RESETI(3)	Reset integer array to specified value
U16	RESETH(3)	Reset Hollerith array to specified value
U17	RESETL(3)	Reset logical array to specified value
U24	COPYI(5)	Copy one integer array into another

Notes

The number of arguments is in brackets
A = Assembler Language
I = IBM
L = Logical function
P = PRIME
S = System dependent Fortran

Table 3. COMMON Blocks

1. General OLYMPUS Data

C1.1	COMBAS	Basic system parameters
C1.2	COMDDP	Development and diagnostic parameters

4. Housekeeping

C4.1	COMCHN	Chaining variables
C4.2	COMVAR	Variable index
C4.3	COMBYT	Displacements

5. I/O and Diagnostics

C5.1	COMIO	I/O buffers

6. Text manipulation

C6.1	COMCHR	Character code (hardware)
C6.2	COMINT	Character code (internal)
C6.3	COMSPC	Special characters (hardware)
C6.4	COMSPA	Special characters (internal)
C6.5	COMCOL	Collating sequence
C6.6	COMKEY	Character strings
C6.8	COMCNT	Counters
C6.9	COMARR	Arrays

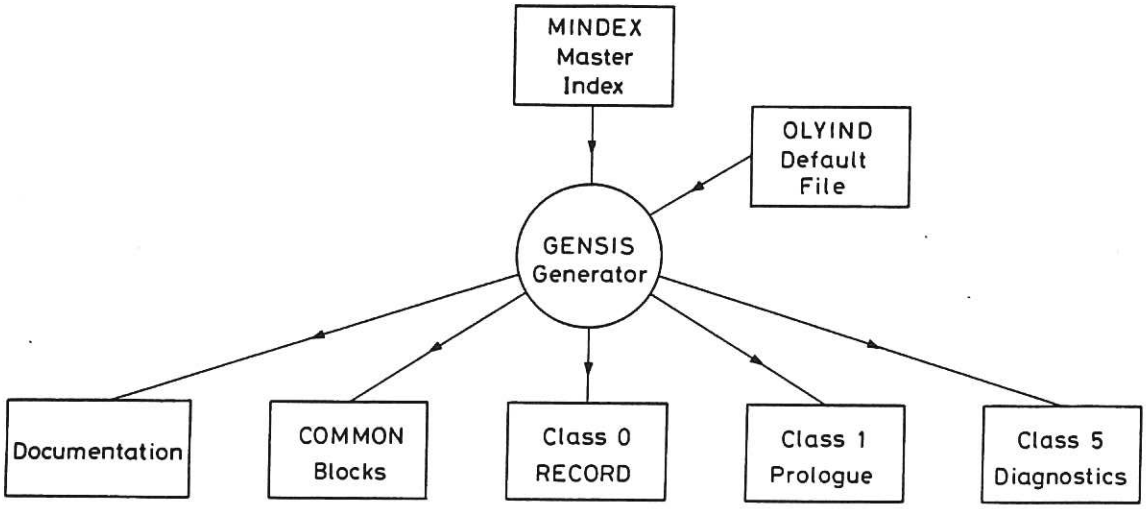

Fig.1 Automatic Program Generation.

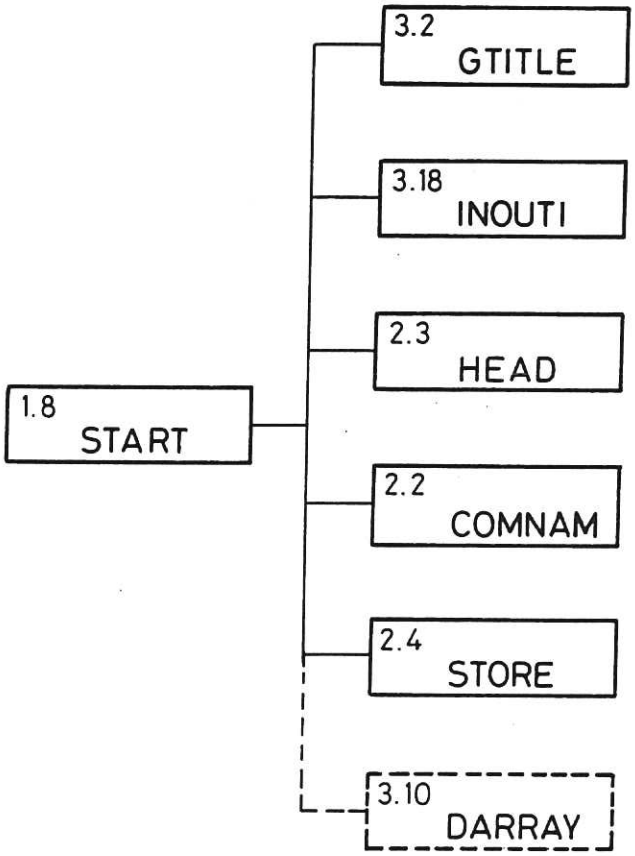

Fig.2 Block diagram of Stage 1.

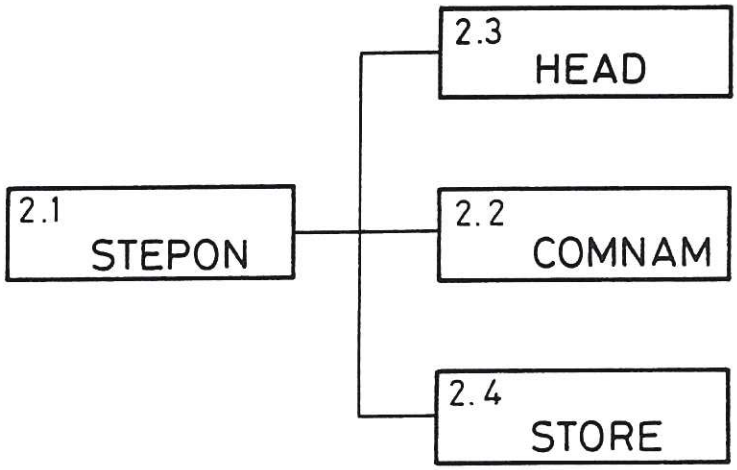

Fig.3 Block diagram of Stage 2 input.

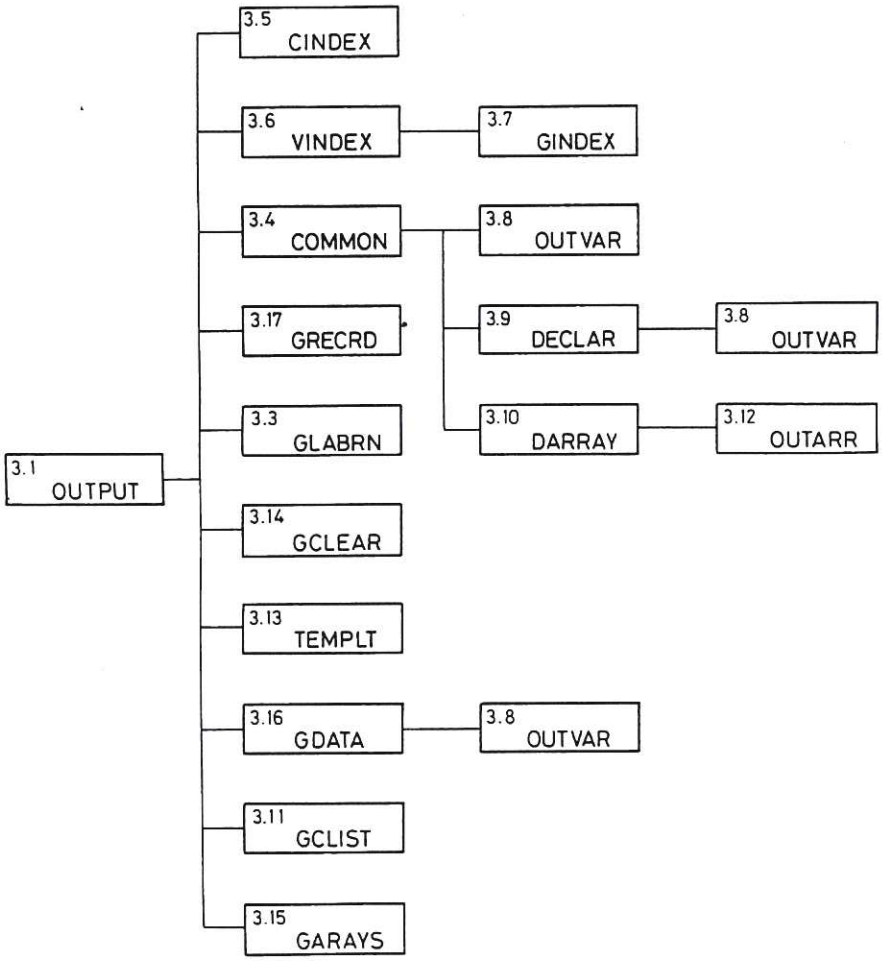

Fig.4 Block diagram of Stage 2 output.

```
PROGRAM HERMES

TITLE  1-D tokamak transport code

AUTHOR M H Hughes

VERSION 1*   MHH  1/Feb/79   Culham

FINAL

ARRAY DIMENSIONS
MATSIZ=5544
MAXU=312
MAXEQN=6
MAXMSH=52,NMESH2
NEQ=6

INDEX OF SUBPROGRAMS
1.1 LABRUN Label the run
1.2 CLEAR Clear variables and arrays
1.3 PRESET Set default values
1.4 DATA Data specific to run
1.5 AUXVAL Set auxiliary values
1.6 INITAL Define physical initial conditions
1.8 START Start or restart the run
2.1 STEPON Advance one timestep
2.2 COEFFS(3) Calculate transport coefficients
3.1 OUTPUT(1) Control the output
3.2 OPRINT(1) Diagnostic printing
A.5 GAUSS(3) Tridiagonal solver

ADDITIONAL SUBPROGRAMS
MESAGE(1) Print 48-character message
DARRAY(3) Print double precision array

INDEX OF COMMON VARIABLES

S 2.1 Tokamak parameters
COMMON/COMTOK/
R RMAJOR *Major radius
R RAWALL *Wall radius
R RALIM *Limiter radius
R BZ *Toroidal field
R CURENT **Tokomak current
I NCOIL **Number of toroidal field colis
R RIPMAX **Maximum ripple amplitude

S 2.2 Plasma parameters
COMMON/COMPLS/
RA PE(MAXMSH) Electron pressure
RA PI(MAXMSH) Ion pressure
RA PEOLD(MAXMSH) Old value of *PE*
RA PIOLD(MAXMSH) Old value of *PI*
RA TE(MAXMSH) Electron temperature
RA TEOLD(MAXMSH) Previous value of Te
RA TI(MAXMSH) Ion temperature

S 3.3 Tridiagonal solver
COMMON/COMTRI/
DA BMATRX(MATSIZ) Block matrix
DA U(MAXU) Solution vector
DA OLDU(MAXU) Old values of *U*
DA RHS(MAXU) R.H.S. of equations
I MAXU Size of solution vector
I MATSIZ Size of block matrix
I NMATSZ (3*NMESH-2)*NEQ*NEQ
R THETA *Implicitness parameter
RA WEIGHT(MAXMSH) Interpolation weights
DA SCALEF(MAXEQN) Scale factors
DA CHI(MAXEQN,MAXMSH) Flux coefficient

S 4.1 Administrative variables
COMMON/COMADM/FILADM/  VERSION 2*  MHH   1/Oct/82   Culham
I NGAS **Frequency for calculating warm neutrals
L NLALPH **.TRUE. if alpha heating
I NIMP **Number of impurity species
I NHYD *Number of hydrogen species

S 9.1 Work areas for SETBLK
COMMON/ /COMBLN/
DA SUMA(MAXEQN) Sum of coefficients at j+1/2
DA DIFFA(MAXEQN) Difference of coefficients at j+1/2
DA SUMB(MAXEQN) Sum of coefficients at j-1/2
DA DIFFB(MAXEQN) Difference of coefficients at j-1/2

ADDITIONAL COMMON BLOCKS

S Basic system parameters
COMMON/COMBAS/
I NSTEP Current step number
I NRUN Maximum number of steps
```

Fig.5 Test Run Input.

```
C/ MODULE INDSUB
C-------------------------------------------------------------------
CL                          INDEX OF SUBPROGRAMS
C
C VERSION 1*   MHH  1/Feb/79   Culham
C
CL                                PROLOGUE            CLASS 1
C  LABRUN          Label the run                                    1.1
C  CLEAR           Clear variables and arrays                       1.2
C  PRESET          Set default values                               1.3
C  DATA            Data specific to run                             1.4
C  AUXVAL          Set auxiliary values                             1.5
C  INITAL          Define physical initial conditions               1.6
C  START           Start or restart the run                         1.8
C
CL                                CALCULATION         CLASS 2
C  STEPON          Advance one timestep                             2.1
C  COEFFS(3)       Calculate transport coefficients                 2.2
C
CL                                OUTPUT              CLASS 3
C  OUTPUT(1)       Control the output                               3.1
C  DPRINT(1)       Diagnostic printing                              3.2
C
CL                                UTILITIES           CLASS A
C  GAUSS (3)       Tridiagonal solver                               A.5
C
CL                          ADDITIONAL SUBPROGRAMS
C  MESAGE(1)       Print 48-character message
C  DARRAY(3)       Print double precision array
C
C/ MODULE INDCOM
C-------------------------------------------------------------------
CL                          INDEX OF COMMON BLOCKS
C
C VERSION 1*   MHH  1/Feb/79   Culham
C
CL                   1.       GENERAL OLYMPUS DATA
C  COMBAS          Basic system parameters                          C1.1
C  COMDDP          Development and diagnostic parameters            C1.9
C
CL                   2.       PHYSICAL PROBLEM
C  COMTOK          Tokamak parameters                               C2.1
C  COMPLS          Plasma parameters                                C2.2
C
CL                   3.       NUMERICAL SCHEME
C  COMTRI          Tridiagonal solver                               C3.3
C
CL                   4.       HOUSEKEEPING
C  COMADM          Administrative variables                         C4.1
C
CL                   9.       BLANK COMMON
C                   Work areas for SETBLK                           C9.1
C
CL                          ADDITIONAL COMMON BLOCKS
C  COMBAS          Basic system parameters
C
C/ MODULE INDVAR
C-------------------------------------------------------------------
CL                          ALPHABETIC INDEX OF COMMON VARIABLES
C
C VERSION 1*   MHH  1/Feb/79   Culham
C
C  BMATRX(MATSIZ)  Block matrix                                   DA 3.3
C  BZ              *Toroidal field                                R  2.1
C  CHI(MAXEQN,MAXMSH)
C                  Flux coefficient                              DA 3.3
C  CURENT          **Tokomak current                              R  2.1
C  DIFFA(MAXEQN)   Difference of coefficients at j+1/2           DA 9.1
C  DIFFB(MAXEQN)   Difference of coefficients at j-1/2           DA 9.1
C  MATSIZ          Size of block matrix                          I  3.3
C  MAXU            Size of solution vector                       I  3.3
C  NCOIL           **Number of toroidal field coils              I  2.1
C  NGAS            **Frequency for calculating warm neutrals     I  4.1
C  NHYD            *Number of hydrogen species                   I  4.1
C  NIMP            **Number of impurity species                  I  4.1
C  NLALPH          **.TRUE. if alpha heating                     L  4.1
C  NMATSZ          (3*NMESH-2)*NEQ*NEQ                           I  3.3
C  NRUN            Maximum number of steps                       I  BAS
C  NSTEP           Current step number                           I  BAS
C  OLDU(MAXU)      Old values of *U*                             DA 3.3
C  PE(MAXMSH)      Electron pressure                             RA 2.2
C  PEOLD(MAXMSH)   Old value of *PE*                            RA 2.2
C  PI(MAXMSH)      Ion pressure                                  RA 2.2
C  PIOLD(MAXMSH)   Old value of *PI*                            RA 2.2
C  RALIM           *Limiter radius                               R  2.1
C  RAWALL          *Wall radius                                  R  2.1
C  RHS(MAXU)       R.H.S. of equations                           DA 3.3
C  RIPMAX          **Maximum ripple amplitude                    R  2.1
C  RMAJOR          *Major radius                                 R  2.1
C  SCALEF(MAXEQN)  Scale factors                                 DA 3.3
C  SUMA(MAXEQN)    Sum of coefficients at j+1/2                  DA 9.1
C  SUMB(MAXEQN)    Sum of coefficients at j-1/2                  DA 9.1
```

Fig.6 Test Run Output.

```
C/ COMMON
C/ MODULE COMTOK
C-----------------------------------------------------------------------
CL                  C2.1     Tokamak parameters
C VERSION 1*   MHH  1/Feb/79   Culham
       COMMON/COMTOK/
     R   BZ     ,   CURENT,   RALIM ,   RAWALL,   RIPMAX,   RMAJOR,
     I   NCOIL
C/ MODULE COMPLS
C-----------------------------------------------------------------------
CL                  C2.2     Plasma parameters
C VERSION 1*   MHH  1/Feb/79   Culham
       COMMON/COMPLS/
     R   PE     ,   PEOLD ,   PI     ,   PIOLD ,   TE     ,   TEOLD ,
     R   TI
       DIMENSION
     R   PE(52),   PEOLD(52),          PI(52),   PIOLD(52),
     R   TE(52),   TEOLD(52),          TI(52)
C/ MODULE COMTRI
C-----------------------------------------------------------------------
CL                  C3.3     Tridiagonal solver
C VERSION 1*   MHH  1/Feb/79   Culham
       COMMON/COMTRI/
     D   BMATRX,   CHI    ,   OLDU  ,   RHS    ,   SCALEF,   U      ,
     R   THETA ,   WEIGHT,
     I   MATSIZ,   MAXU   ,   NMATSZ
       DOUBLE PRECISION
     D   BMATRX,   CHI    ,   OLDU  ,   RHS    ,   SCALEF,   U
       DIMENSION
     D   BMATRX(5544),       CHI(6,52),          OLDU(312),
     D   RHS(312), SCALEF(6),          U(312),
     R   WEIGHT(52)
C/ MODULE FILADM
C-----------------------------------------------------------------------
CL                  C4.1     Administrative variables
C VERSION 2*  MHH   1/Oct/82   Culham
       COMMON/COMADM/
     I   NGAS   ,   NHYD   ,   NIMP   ,
     L   NLALPH
       LOGICAL
     L   NLALPH
C/ MODULE COMBLN
C-----------------------------------------------------------------------
CL                  C9.1     Work areas for SETBLK
C VERSION 1*   MHH  1/Feb/79   Culham
       COMMON
     D   DIFFA ,   DIFFB ,   SUMA  ,   SUMB
       DOUBLE PRECISION
     D   DIFFA ,   DIFFB ,   SUMA  ,   SUMB
       DIMENSION
     D   DIFFA(6), DIFFB(6), SUMA(6),  SUMB(6)
```

Fig.6(iii) Contd.

```
C/ MODULE INDBLK
C-----------------------------------------------------------------------
CL                          INDEX OF COMMON VARIABLES
C
C VERSION 1*   MHH  1/Feb/79   Culham
C
CL                  C2.1     Tokamak parameters
C VERSION 1*   MHH  1/Feb/79   Culham
C      COMMON/COMTOK/
C
C  BZ               *Toroidal field                                  R  2.1
C  CURENT           **Tokomak current                                R  2.1
C  RALIM            *Limiter radius                                  R  2.1
C  RAWALL           *Wall radius                                     R  2.1
C  RIPMAX           **Maximum ripple amplitude                       R  2.1
C  RMAJOR           *Major radius                                    R  2.1
C  NCOIL            **Number of toroidal field coils                 I  2.1
C
CL                  C2.2     Plasma parameters
C VERSION 1*   MHH  1/Feb/79   Culham
C      COMMON/COMPLS/
C
C  PE(MAXMSH)       Electron pressure                                RA 2.2
C  PEOLD(MAXMSH)    Old value of *PE*                                RA 2.2
C  PI(MAXMSH)       Ion pressure                                     RA 2.2
C  PIOLD(MAXMSH)    Old value of *PI*                                RA 2.2
C  TE(MAXMSH)       Electron temperature                             RA 2.2
C  TEOLD(MAXMSH)    Previous value of Te                             RA 2.2
C  TI(MAXMSH)       Ion temperature                                  RA 2.2
C
CL                  C3.3     Tridiagonal solver
C VERSION 1*   MHH  1/Feb/79   Culham
C      COMMON/COMTRI/
C
C  BMATRX(MATSIZ)   Block matrix                                     DA 3.3
C  CHI(MAXEQN,MAXMSH)
C                   Flux coefficient                                 DA 3.3
C  OLDU(MAXU)       Old values of *U*                                DA 3.3
C  RHS(MAXU)        R.H.S. of equations                              DA 3.3
C  SCALEF(MAXEQN)   Scale factors                                    DA 3.3
C  U(MAXU)          Solution vector                                  DA 3.3
C  THETA            *Implicitness parameter                          R  3.3
C  WEIGHT(MAXMSH)   Interpolation weights                            RA 3.3
C  MATSIZ           Size of block matrix                             I  3.3
C  MAXU             Size of solution vector                          I  3.3
C  NMATSZ           (3*NMESH-2)*NEQ*NEQ                              I  3.3
C
CL                  C4.1     Administrative variables
C VERSION 2*  MHH   1/Oct/82   Culham
C      COMMON/COMADM/
C
C  NGAS             **Frequency for calculating warm neutrals        I  4.1
C  NHYD             *Number of hydrogen species                      I  4.1
C  NIMP             **Number of impurity species                     I  4.1
C  NLALPH           **.TRUE. if alpha heating                        L  4.1
C
CL                  C9.1     Work areas for SETBLK
C VERSION 1*   MHH  1/Feb/79   Culham
C      COMMON
C
C  DIFFA(MAXEQN)    Difference of coefficients at j+1/2              DA 9.1
C  DIFFB(MAXEQN)    Difference of coefficients at j-1/2              DA 9.1
C  SUMA(MAXEQN)     Sum of coefficients at j+1/2                     DA 9.1
C  SUMB(MAXEQN)     Sum of coefficients at j-1/2                     DA 9.1
```

Fig.6(ii) Contd.