



LUND UNIVERSITY

Simnon - An Interactive Simulation Program

Implementation

Elmqvist, Hilding

1978

Document Version:

Publisher's PDF, also known as Version of record

[Link to publication](#)

Citation for published version (APA):

Elmqvist, H. (1978). *Simnon - An Interactive Simulation Program: Implementation*. (Technical Reports TFRT-7148). Department of Automatic Control, Lund Institute of Technology (LTH).

Total number of authors:

1

General rights

Unless other specific re-use rights are stated the following general rights apply:

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Read more about Creative commons licenses: <https://creativecommons.org/licenses/>

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

SIMNON - AN INTERACTIVE SIMULATION PROGRAM -
IMPLEMENTATION

HILDING ELMQVIST

Department of Automatic Control
Lund Institute of Technology
August 1978

Dokumentutgivare
Lund Institute of Technology
Handläggare Dept of Automatic Control
Hårding Elmqvist
Författare
Hårding Elmqvist

Dokumentnamn
REPORT LUTFD2/(TFRT-7148)/1-068/(1978)
Utgivningsdatum
Aug 1978
Dokumentbeteckning
06T6
Ärendebeteckning
06T6

10T4

Dokumenttitel och undertitel

SIEMON-An Interactive Simulation Program - IMPLEMENTATION

Referat (sammandrag)

Simmon is a command driven interactive program written in Fortran for simulation of systems governed by ordinary differential equations and difference equations.

This report briefly describes the implementation of Simmon. Description of the data-base, subroutine summary and subroutine trees are included.

Referat skrivet av

Author

Förslag till ytterligare nyckelord

44T0

Klassifikationssystem och -klass(er)

50T0

Indextermer (ange källa)

52T0

Omfång

68 pages

Övriga bibliografiska uppgifter

56T2

Språk

English

Sekretessuppgifter

60T0

ISSN

60T4

ISBN

60T6

Dokumentet kan erhållas från

Department of Automatic Control
Lund Institute of Technology
P O Box 725, S-220 07 LUND 7, SWEDEN

Mottagarens uppgifter

62T4

Pris

66T0

DOKUMENTTABLAD enligt SIS 62 10 12

SIS-
DB 1

Blankett LU 11:25 1976-07

SIMNON

An Interactive Simulation Program

IMPLEMENTATION

H. Elmqvist

Contents:

1. Introduction	4
2. The structure of Simnon	5
3. The command SYST	7
4. Installation dependent subroutines	20
5. Acknowledgements	25
6. References	26

Appendices

1. Subroutine summary	27
2. Cross reference (called subroutines)	33
3. Cross reference (calling subroutines)	37
4. Subroutines with entry points	42
5. Subroutine trees	43
6. Common blocks referenced in the subroutines	64
7. Subroutines referencing the common blocks	67

1. INTRODUCTION

Simnon is a command driven interactive program written in Fortran for simulation of systems governed by ordinary differential equations and difference equations.

The first version of Simnon was developed in the beginning of 1972 (Elmqvist (1972,1973)). The work on defining the current version started in July 1973 and the implementation began in October 1973. A preliminary version including the main facilities was ready in February 1974. After September 1974 there were only minor revisions of Simnon. The User's Manual for Simnon (Elmqvist (1975)) was available in April 1975. A paper about Simnon was presented in June 1977 (Elmqvist (1977)). Sorting of the equations was implemented in February 1978.

This report briefly describes the implementation of Simnon. It is intended to be useful when making new installations of Simnon. The appendices contain a subroutine summary, cross references and subroutine trees.

2. THE STRUCTURE OF SIMNON

The program Simnon consists of about 175 Fortran subroutines or about 200000 Fortran statements. Simnon is modularized as shown in Fig. 1.

There is as a base a set of character handling and file handling routines.

The module Intrac handles the command input, Macro-facility and the execution of the special commands included in the language Intrac. When a Simnon-command has been given, Intrac delivers the index of the command which is used to branch to different command decoding and action routines. Intrac splits up the command line into items such as real numbers, integer numbers, identifiers and delimiters. These items are stored in a COMMON-block. The command decoding routines uses special logical functions to fetch information from the COMMON-block.

Since the commands can be issued in almost arbitrary order there must be a well-defined data base. Part of that data base is described in chapter 3.

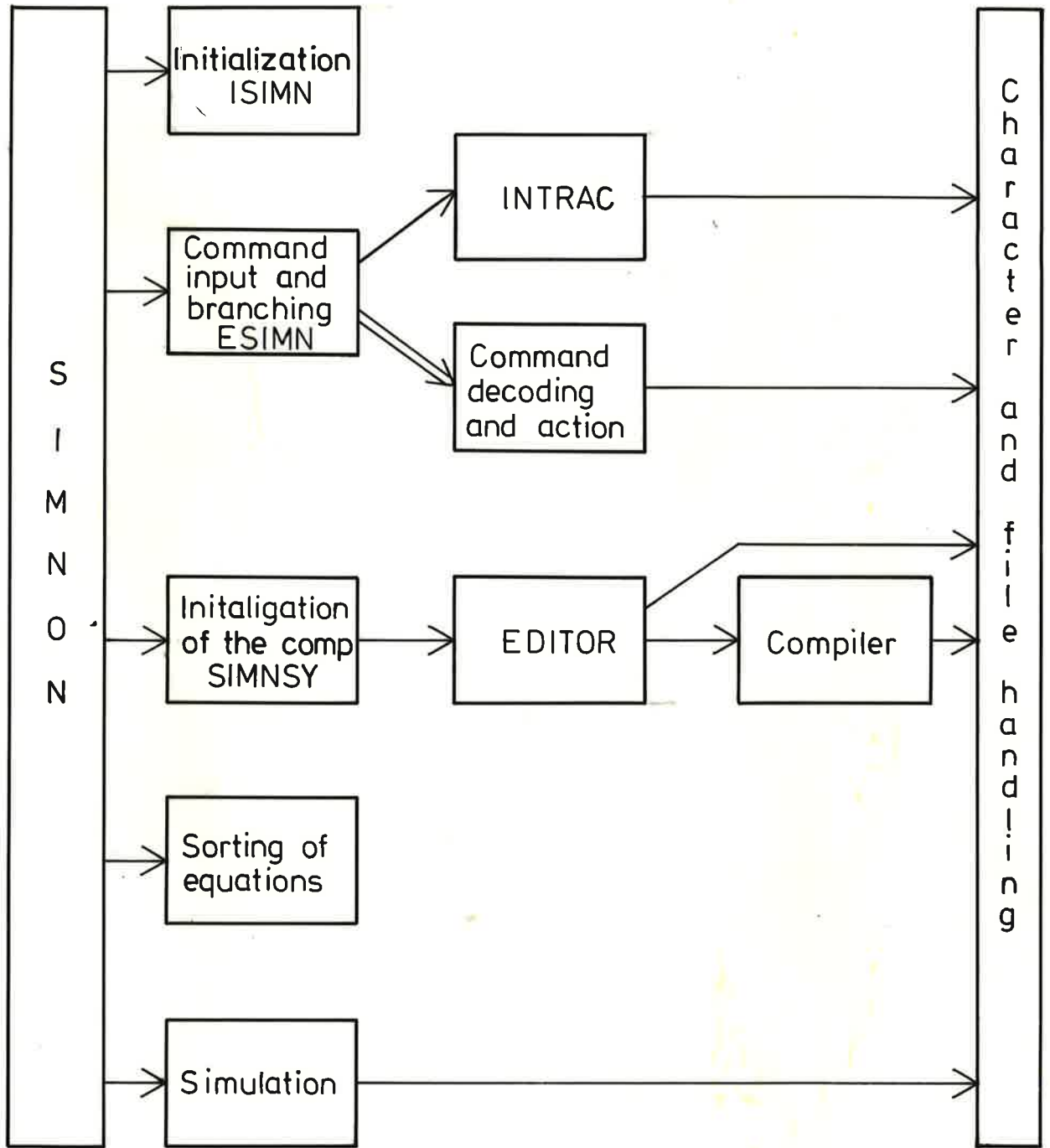


Fig 1. The structure of SIMNON.

3. THE COMMAND SYST

The part of the data base, which is fundamental to the behavior of the compiler, is described below.

Information about the active systems is stored in:

```
COMMON /SYSINF/ NASYST,ASYSTS(2, ),IVARS( ,2),INFSYS( ),
        IFSYS( )
```

NASYST- Number of active systems

ASYSTS- System identifiers for active systems

IVARS - Defines the position of the variable table
for each system in VARTAB

INFSYS- System type for each system

1: Connecting

2: Continuous

3: Discrete

4: Continuous (FORTRAN)

5: Discrete (FORTRAN)

IFSYS - ISYST number of Fortran system

The variable table is:

```
COMMON /VARTB1/ VARS(2, )
```

```
COMMON /VARTB2/ IPNTS( )
```

```
COMMON /VARTB3/ ITYPES( )
```

VARS - Identifier table

IPNTS - Address table

ITYPES- Type table

1: TIME

2: STATE

3: INPUT

4: OUTPUT

5: INIT

6: DER

7: NEW

8: TSAMP
 9: PAR
 10: VAR
 11: INPUT to dynamics section

Information about the currently compiled system is in:

```
COMMON /CMPVAR/ MODE,IASYST,ISYTYP,IERR,IVAR1,IVAR2,
          IVAL1,IVAL2,L,NODE
```

MODE - Compiler mode

1: System heading

2:

3: Declarations

4:

5: Initial-section

6: Output-section

7: Dynamics-section

8: Connect-section

9: End

IASYST- Index in SYSINF for actual system

ISYTYP- System type

1: Connecting

2: Continuous

3: Discrete

IERR - Error flag

IVAR1 - Index for lower bound in variable table

IVAR2 - Index for upper bound in variable table

IVAL1 - Pointer in the value table

IVAL2 - Pointer in the literal table

L - Pointer in the pseudocode area

NODE - Current node

The code produced by the compiler is stored in:

```
COMMON /PSCODE/ IPSEUD( )
```

The values of the variables and the literals are stored in:

COMMON /VALUES/ VALUE()

The decoding subroutine for the command SYST (COSYST) puts the system identifiers and file names in ASYSTS. The subroutine EXTSYS then calls SYSTS with IPART=1 and different values of ISYST to determine the meaning of each identifier in the command. This information is stored in INFSYS. The ISYST-number for each Fortran system is stored in IFSYS.

The subroutine SYSTS is called with IPART=2 and the different values of ISYST to allow the Fortran-systems to declare variables. The subroutines STATE, DER, etc. stores information in the variable table: identifier, address and variable type. The Fortran declarations are checked by the subroutine CHEDEC.

The editor is active during the compilation of the Simnon systems. Each time the editor has accepted a line, and is going to save it on mass storage, the subroutine EXTSUB is called. EXTSUB calls the compiler (COMP). If an error is discovered in the line the editor is forced to edit mode.

The editor is called, for each file, from the subroutine SIMNSY. The state of the compiler is saved before calling the editor in order to be able to restart after a TOP-command has been given.

The structure of the compiler is shown in fig. 2 (compare the subroutine tree). The output from the compiler is a pseudocode which is interpreted and executed by the subroutine CALCUL during the simulations. It is possible to include a machine dependant last stage of the compiler, which produces machine instructions from the pseudocode. A gain in execution time will also be achieved if the

subroutine CALCUL is implemented in assembler.

The compiler is working in one of seven modes indicated by the variable MODE. The mode depends on what language elements have already been received. When e.g. MODE=6 (OUTPUT-section) the next language element could be an assignment statement, constant definition, DYNAMICS or END.

The state of the compiler (MODE, IVAR2, IVAL1, IVAL2, L) is saved each time the subroutine COMP is called. It is then possible to restart the compiler if an error is detected in the line.

The subroutines LEFTP and CODE decodes an assignment statement and stores the code in an integer vector ICODE in a format shown in table 1. Identifiers are stored in the vectors VARIBS and SYSIDS and literals are stored in the vector RLITTS.

The code is then checked for errors and unitary operators are recognized by the subroutine CHECK.

Reverse polish notation for the assignment statement is produced from ICODE by the subroutine REPONO and stored in the integer vector IRPN in a format shown in table 2.

The simulation module must know where the states and the derivatives etc. are located. This information is collected by the subroutine COLECT when all systems are compiled and stored in the COMMON-blocks:

```
COMMON /PNTS1/ NXC,NXD,KTSAMP( ),KX( )
COMMON /PNTS2/ KDX( )
COMMON /PNTS3/ KXI( )
```

```
NXC - Number of states in continuous systems
NXD - Number of states in discrete systems
```

KTSAMP- pointers to TSAMP-variables
KX - Pointers to state variables
KDX - Pointers to DER- and NEW-variables
KXI - Pointers to INIT-variables

This subroutine also groups the discrete states to simplify the updating of the discrete states. This information is stored in:

```
COMMON /NXPNT/ NXP( ,2)
```

All equations of the systems are sorted by a call to the subroutine EQORD. Four different entry points to the pseudo code are produced and stored in the COMMON block ENTRYS:

```
COMMON /ENTRYS/ NTRINT,NTRDER,NTRCON,NTRSMP
```

NTRINT- Entry point for initial computations
NTRDER- Entry point for computations of derivatives
NTRCON- Entry point for other continuous equations
NTRSMP- Entry point for sampling

The pseudo code is organized as linked lists of nodes. A node can contain a single equation or a section. All initial sections and nodes which are independent of time and states are put into the list for initial equations. After that the list for computation of derivatives is constructed by preorder traversals of the equation trees starting at equations assigning derivatives. All remaining equations originating from continuous and connecting systems are then sorted and put into the third list. The list of sorted discrete equations are then constructed.

If an output file name is given in the SYST command, the sorted equations are output in infix notation.

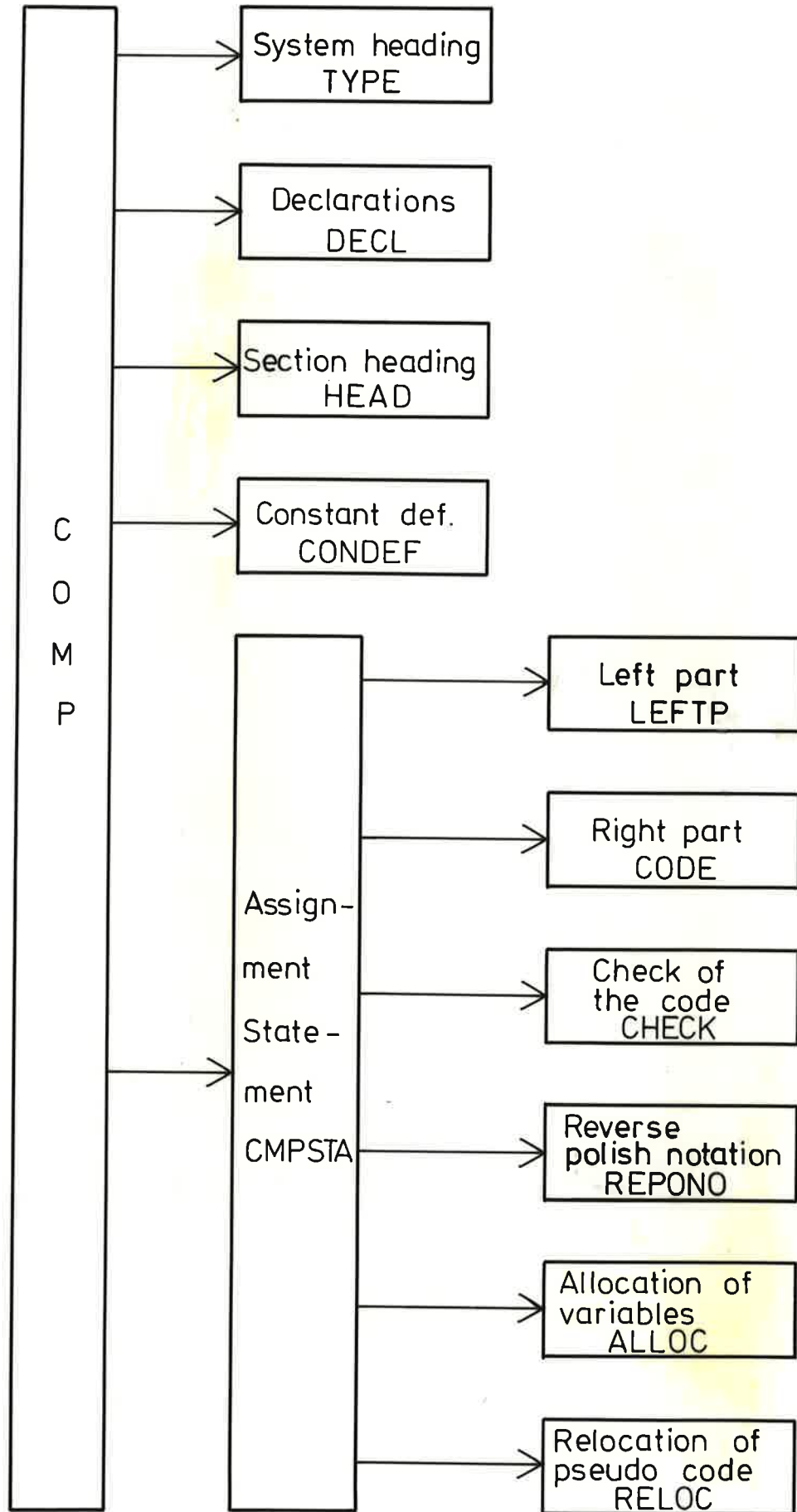


Fig 2. The structure of the compiler.

Table 1. Codes for an assignment statement.

Symbols	Codes
OR	1
AND	2
NOT	3
<	4
>	5
+	6
-	7
*	8
/	9
&	11
IF	12
THEN	13
ELSE	14
(15
(after function	16
)	17
,	18
end of line	19
<procedure identifier>	20
	function number
	expected number of arguments
<variable>	21
	index in variable table
<unsigned number>	21
	-(index in literal table)

Table 2. Operation set

Operation	Mnemonic	Code
logical or	OR	1
logical and	AND	2
logical not	NOT	3
test less than	TLT	4
test greater than	TGT	5
add	ADD	6
subtract	SUB	7
multiply	MUL	8
divide	DIV	9
negate	NEG	10
raise	RAI	11
jump if false	JMPF	12
		number of steps
jump	JMP	13
		number of steps
stack	FETCH	14
		pointer
unstack	DEPOS	15
		pointer
apply function	FUNC	16
		function number
jump to subroutine	JMS	17
		index in PSCODE
return from subroutine	RET	18
stop the interpretation	STOP	19
call FORTRAN system	CALL	20
		ISYST
		IPART
skip if not sampling	SCOND	21
		system number
no operation	NOP	22

Example

Consider the tank system controlled by a discrete PID controller listed below. The sorted equations as output by the SYST command are listed on the next page followed by a printout of some of the database including the pseudo code in symbolic format.

```

CONTINUOUS SYSTEM TANK
STATE H
DER DH
INPUT Q
DH=(Q-QOUT)/A1
QOUT=A2*SQRT(2*G*MAX(H,0))
G:9.81
A1:10
A2:0.01
END

```

```

DISCRETE SYSTEM DPID
INPUT YREF Y
OUTPUT U
STATE I X
NEW NI NX
TIME T
TSAMP TS
E=YREF-Y
P=G*E
D=-TD*(Y-X)/DT
U=P+I+D
NI=I+E/TI*DT
NX=Y
TS=T+DT
G:1
TI:1E10
TD:0
DT:1
END

```

```

CONNECTING SYSTEM REGTA
TIME T
HREF=IF T<25 OR T>300 THEN 5 ELSE 0.5
YREF[DPID]=HREF
Y[DPID]=H[TANK]
QQ=U[DPID]+Q0
Q0:1
Q[TANK]=IF QQ<0 THEN 0 ELSE IF QQ>QMAX THEN QMAX ELSE QQ
QMAX:5
END

```

SORTED INITIAL EQUATIONS

SORTED DERIVATIVE EQUATIONS

```

TANK      QOUT = A2*SQRT(2*G*MAX(H,0))
REGTA     Y[DPID] = H[TANK]
          HREF = IF T<25 OR T>300 THEN 5 ELSE 0.5
          YREF[DPID] = HREF
DPID      E = YREF - Y
          P = G*E
          D = (-TD)*(Y - X)/DT
          U = P + I + D
REGTA     QQ = U[DPID] + Q0
          Q[TANK] = IF QQ<0 THEN 0 ELSE IF QQ>QMAX THEN QMAX
          ELSE QQ
TANK      DH = (Q - QOUT)/A1

```

SORTED CONTINUOUS EQUATIONS

SORTED DISCRETE EQUATIONS

```

DPID      NI = I + E/TI*DT
          NX = Y
          TS = T + DT

```

DATA BASE

=====

SYSINF:

ASYSYS	IVARS	INFSYS	IFSYS
TANK	1	8	0
DPID	9	26	0
REGTA	27	31	0

VARTAB:

VARS	IPNTS	ITYPES
H	28391	2
H	28392	-5
DH	28393	6
Q	28394	3
QOUT	28395	10
A1	28396	9
A2	28397	9
G	28398	9
YREF	28399	3
Y	28400	3
U	28401	4
I	28402	2
I	28403	-5
X	28404	2
X	28405	-5
NI	28406	7
NX	28407	7
T	28390	1

TS	28408	8
E	28409	10
P	28410	10
G	28411	9
D	28412	10
TD	28413	9
DT	28414	9
TI	28415	9
T	28390	1
HREF	28416	10
QQ	28417	10
Q0	28418	9
QMAX	28419	9

PNTS:

```

-----
KX:          28391    28402    28404
KDX:          28393
              28406    28407
KXI:          28392    28403    28405
KTSAMP:           0    28408         0
PSCODE:
-----

```

ORIGINAL EQUATIONS

SORTED INITIAL EQUATIONS

SORTED DERIVATIVE EQUATIONS

```

37  NODE
TANK  QOUT = A2*SQRT(2*G*MAX(H,0))
43  FETCH      A2
45  FETCH      2.00
47  FETCH      G
49  MUL
50  FETCH      H
52  FETCH      .000
54  FUNC      MAX
56  MUL
57  FUNC      SQRT
59  MUL
60  DEPOS      QOUT

217  NODE
REGTA  Y[DPID] = H[TANK]
223  FETCH      H          [TANK  ]
225  DEPOS      Y          [DPID  ]

180  NODE
      HREF = IF T<25 OR T>300 THEN 5 ELSE 0.5
186  FETCH      T
188  FETCH      25.0
190  TLT
191  FETCH      T
193  FETCH      300.
195  TGT
196  OR
197  JMPF      5
199  FETCH      5.00

```

```

201  JMP          3
203  FETCH        .500
205  DEPOS        HREF

207  NODE
      YREF [DPID] = HREF
213  FETCH        HREF
215  DEPOS        YREF          [DPID  ]

62   NODE
DPID  E = YREF - Y
68   SCOND        2
70   FETCH        YREF
72   FETCH        Y
74   SUB
75   DEPOS        E

77   NODE
      P = G * E
83   SCOND        2
85   FETCH        G
87   FETCH        E
89   MUL
90   DEPOS        P

92   NODE
      D = (-TD) * (Y - X) / DT
98   SCOND        2
100  FETCH        TD
102  NEG
103  FETCH        Y
105  FETCH        X
107  SUB
108  MUL
109  FETCH        DT
111  DIV
112  DEPOS        D

114  NODE
      U = P + I + D
120  SCOND        2
122  FETCH        P
124  FETCH        I
126  ADD
127  FETCH        D
129  ADD
130  DEPOS        U

227  NODE
REGTA  QQ = U [DPID] + Q0
233  FETCH        U          [DPID  ]
235  FETCH        Q0
237  ADD
238  DEPOS        QQ

240  NODE
      Q [TANK] = IF QQ < 0 THEN 0 ELSE IF QQ > QMAX THEN QMAX
      ELSE QQ
246  FETCH        QQ
248  FETCH        .000

```

250	TLT		
251	JMPF	5	
253	FETCH	.000	
255	JMP	14	
257	FETCH	QQ	
259	FETCH	QMAX	
261	TGT		
262	JMPF	5	
264	FETCH	QMAX	
266	JMP	3	
268	FETCH	QQ	
270	DEPOS	Q	[TANK]

21	NODE		
TANK	DH = (Q - QOUT)/A1		
27	FETCH	Q	
29	FETCH	QOUT	
31	SUB		
32	FETCH	A1	
34	DIV		
35	DEPOS	DH	

SORTED CONTINUOUS EQUATIONS

SORTED DISCRETE EQUATIONS

132	NODE		
DPID	NI = I + E/TI*DT		
138	SCOND	2	
140	FETCH	I	
142	FETCH	E	
144	FETCH	TI	
146	DIV		
147	FETCH	DT	
149	MUL		
150	ADD		
151	DEPOS	NI	

153	NODE		
	NX = Y		
159	SCOND	2	
161	FETCH	Y	
163	DEPOS	NX	

165	NODE		
	TS = T + DT		
171	SCOND	2	
173	FETCH	T	
175	FETCH	DT	
177	ADD		
178	DEPOS	TS	

4. INSTALLATION DEPENDENT SUBROUTINES

Manipulation of real variables using addresses

The communication between systems described as FORTRAN subroutines and the SIMNON program is done in a nonstandard way. The variables associated with a system is local to the subroutine. It is 'declared' by calling a set of subroutines (STATE, DER etc.) which calls an implementation dependent subroutine IADDR to compute the address of the variable. The addresses are stored in a vector.

The communication is performed by two subroutines FETCH and DEPOS which gets resp. assigns a real variable at a specified address.

Return of the address of a real variable:

```
INTEGER FUNCTION IADDR(VAR)
```

VAR - Real variable

IADDR - Returned address

Return of the value of a real variable stored at a specified address:

```
REAL FUNCTION FETCH(IADDR)
```

IADDR - Address

FETCH - Returned value

Assign a value to a real variable stored at a specified address:

```
SUBROUTINE DEPOS(IADDR,VAL)
```

```
IADDR - Address
```

```
VAL   - Value
```

Interrupting SIMNON

It is recommended to allow the user to interrupt a simulation or the execution of a macro by some action on the terminal. For that purpose Simnon calls two subroutines. The subroutine IBREAK is called before starting a simulation and before starting the execution of a macro. The subroutine BREAK is called each time the simulation module has accepted a new point on the solution and each time a new command is read from a macro.

Initialization of the break facility:

```
SUBROUTINE IBREAK
```

Test for a break action:

```
SUBROUTINE BREAK(IBR)
```

```
IBR   - Returned .TRUE. if break is wanted else .FALSE.
```

Character handling

The following required routines for character handling are described in Essebo (1977).

GAC Gets a character from a character string (buffer).

PAC Puts a character in a character string.

IFAC Gets a character from a string and
FAC returns its type.

HSTORV Assigns a character buffer.
HSTOR2

LCOMPV Compares two character strings.
LCOMP2

Terminal handling

The following routines for terminal handling are described in Essebo (1977).

ERDIS Erases display.

PLCURS Positions cursor at specified line.

TPOS Positions cursor at specified line and column.

TWRITE Writes a character string.

TREAD Reads a character string.

File handling

The required routines for file handling are described in Essebo (1977).

FCHECK Checks if a file exists.

FENTER Opens a file for output.

FSEEK Opens a file for input.

FCLOSE Closes a file.

RDREAL Reads real numbers from a file.

WRREAL Writes real numbers.

RDINT Reads integer numbers.

WRINT Writes integer numbers.

LFINAM Decodes a file name from a command.

Plot routines

The description of required plot routines can be found in Schonthal (1977b).

PLDEV Selects plotting device.

PLMOVE Draws line and moves cursor.

PLSYMB Writes character strings.

PLRSET Terminates plotting temporarily.

Auxiliary routines

IMACON Returns integer machine dependent constants.
Described in Elmqvist et al (1976).

RFLOAT Converts integer to real number.

IFIXR Converts real to integer number.

Logical devices

The FORTRAN logical device numbers are installation dependent. They are for that reason stored in a COMMON block in the following way.

```
COMMON /DEVICE/ LKB,LTP,LLP,LDIS,LTO,LPLOT,LXXX,
                LDK1,LDK2,LDK3,LDK4
```

LKB - Input from keyboard
 LTP - Output to printer
 LLP - Output to line printer
 LDIS - Alphanumeric output to display
 LTO - Output in connection with keyboard
 LPLOT- Graphic output to display
 LXXX - Unused
 LDK1 - Input and output to mass storage.
 LDK2 At most four files are
 LDK3 manipulated at the same time.
 LDK4 Compare 'file handling'.

The COMMON block DEVICE should be initialized with a BLOCK DATA routine. If no graphical output facility is available LPLOT should be set to zero. Normally LTO is equal to either LTP or LDIS.

5. ACKNOWLEDGEMENTS

The author wants to thank all who have participated in the implementation of SIMNON.

Intrac has been implemented by Staffan Selander and Thomas Schonthal. The basic work on the editor was done by Johan Wieslander. Tommy Essebo has implemented some of the commands, redesigned the terminal handling and implemented sorting of the equations. He also made a new stage in the compiler which produces machine instructions for Univac-1108 from the pseudo code.

The standard system for optimization OPTA was implemented by Torkel Glad. The basic work on the system with time delays, DELAY, was done by Jesper Ooppelstrup.

The appendices were generated by a program written by Tommy Essebo.

The author also wants to thank Johan Wieslander, Staffan Selander and Tommy Essebo for many stimulating discussions.

The work has been supported by the Swedish Institute of Applied Mathematics (ITM) and the Swedish Board for Technical Development (STU).

6. REFERENCES

- Elmqvist H. (1972): Simnon - Ett interaktivt simuleringsprogram for olinjara system (Simnon - An Interactive Simulation Program for Nonlinear Systems). Master thesis. RE-113, Dept. of Aut. Contr., LTH (in swedish).
- Elmqvist H. (1973): Simnon - User's Guide. TFRT-3106, Dept. of Aut. Contr., LTH.
- Elmqvist H. (1975): Simnon - An Interactive Simulation Program for Nonlinear Systems - User's Manual. TFRT-3091, Dept. of Aut. Contr., LTH.
- Elmqvist H. (1977): Simnon - An Interactive Simulation Program for Nonlinear Systems. Proc. Simulation '77, Montreux, June 1977.
- Elmqvist, Tysso, Wieslander (1976): Programming and Documentation Rules for Subroutine Libraries - Designed for the SCL. NORDFORSK, The Scandinavian Council for Applied Research.
- Essebo T. (1977): Character and String Handling in Intrac - File Handling in Program Packages. Dept. of Aut. Contr., LTH, Nov 1977.
- Schonthal T. (1977a): Implementation Procedure for Intrac. Dept. of Aut. Contr., LTH, May 1977.
- Schonthal T. (1977b): Implementation Procedure, Plot Routines. Dept. of Aut. Contr., LTH, May 1977.

APPENDICES

1. Subroutine summary

The subroutine summaries for character handling and for INTRAC can be found in Schonthal (1977a). Schonthal (1977b) contains a subroutine summary for the plot routines.

The editor

EDITOR Main subroutine for the editor.

OPEN Opens files.
CLOSEF Closes files.

RCURL Reads a new current line from mass storage.
WCURL Writes the current line on mass storage.

DISPL Writes current line on display.
CROSS Crosses over current line.
ERAS Erases the display.

COMND Reads commands for the editor.

NARG Fetches integer argument from command.

FINLOC Performs the FIND and LOCATE commands.

REINAP Performs the RETYPE, INSERT and APPEND commands.

CHANGE Performs the CHANGE command.

ONESTR Gets one string from a command.

TWOSTR Gets two strings from a command.

COMPAR Compares two strings.

EMESS Writes messages for the editor.

CURDIS Gets display flag and status of current line.

EXTSUB Subroutine called by the editor before saving current line on mass storage. Calls the compiler during the SYST command.

The compiler

COMP Master subroutine for the compiler.

TYPE Handles system heading.

DECL Handles declarations.

HEAD Handles section headings and END statement.

ACTHD Acts according to section headings and END statement.

UNASS Writes messages about unassigned variables.

RESW Scans actual line for certain reserved words.

CONDEF Handles constant definition.

CMPSTA Master subroutine for the compilation of assignment statements.

LEFTP Handles left part of assignment statement.

CODE Decodes expressions.

BOOLW Tests for certain reserved words in expressions.

FUNC Tests for function identifier.

VARIAB Gets variable identifiers.

LITT Gets numerical literals.

OPER Tests for operators.

CHECK Checks the coded expression.

REPONO Produces reverse polish notation for the assignment statement.

LIFO Handles a stack used by REPONO.
PUSH
POP

ALLOC Allocates variables.

SEARCH Searches the variable table.

ALLERR Tests for illegal variable references.

RELOC Relocates and modifies the pseudo code.

INSTR Places an instruction in the pseudo code area.

DEFVAR Allocates variables and literals.
DEFLIT

ERROR Writes error messages for the compiler.

WERROR Writes the different error messages.
ICOMNT Tests for line terminator or comment.
RCHAR Reads the next character from the line buffer.

Simnon

SIMNON Main program.
ISIMN Initialization routine.
ESIMN Command input and branching.
WMESS Writes error messages.
COSYST Decodes the command SYST.
EXTSYS Includes FORTRAN systems.
ALL Initializes the allocation routines.
ALLWRK Reinitializes the allocation routine IWORK.
ALLER1 Tests for allocation error.
IALL Allocates a variable area in the allocation vector.
IWORK Allocates a work area in the allocation vector.
CHEDEC Checks the declarations of FORTRAN systems.
SIMNSY Handles the compilation of systems described in the simulation language. Calls the editor which calls subroutine COMP.
COLLECT Collects pointers to STATE and DER variables etc.
EQORD Sorts the equations.
PREORD Performs preorder traversal of equation trees.
INLIST Tests if two nodes are in the same list.
NODEIN Inserts a node in a list.
NODMOV Moves a node from one list to another.
NXTINS Searches the pseudo code for a specified instruction.
PREQND Prints the equations in a node in infix notation.

INFIX Prints an equation in infix notation.

NEWNOD Creates new node in the syntax tree for an expression.

CODISP Decodes the command DISP.

DISP Displays all variables.

DISP1 Displays single variables.

PARINT Decodes the command PAR and INIT.

COPLLOT Decodes the command PLOT.

COAXES Decodes the command AXES.

AXES Draws axes on the display.

DEFPAR Defines parameters for current picture.

SCALE Finds scales factors for plotting.

AXIS Draws an axis.

COSIMU Decodes the command SIMU.

SIMU Main routine for the simulation.

INIVAR Initializes certain variables before the simulation.

CALCUL Interprets and executes the pseudo code.

LCOND Computes the skip condition for CALCUL.

INTEG Calls different integration routines.

HAMPC Solves differential equations using Hammings modified predictor - corrector method.

RKADM Solves differential equations using a Runge-Kutta method with automatic step size adjustment.

RKFIX Solves differential equations using a Runge-Kutta method with fix step size.

RK Performs one step of a 4th order Runge-Kutta method.

F Computes derivatives for the Runge-Kutta routines.

OUTPRK Output routine for the Runge-Kutta routines.

DERIV Computes derivatives by calling CALCUL.

OUTP Performs output on display and mass storage for integration routines.

DRAW Plots variables specified by the PLOT command.

STORE Stores specified variables on a file.

COERR Decodes the ERROR command.

COLIST Decodes the command LIST and lists files.

COEDIT Decodes the command EDIT.

COTURN Decodes the command TURN.

COSAVE Decodes the command SAVE.

SAVE Saves parameters and initial values on a file.

COGET Decodes the command GET.

GETV Gets parameters and initial values from a file.

COALG Decodes the command ALGOR.

COSTOR Decodes the command STORE.

COSHOW Decodes command SHOW and plots variables from a file.

ASHOW Decodes the command ASHOW.

COSPLT Decodes the command SPLIT.

COAREA Decodes the command AREA.

COTEXT Decodes the command TEXT.

T5Ø11 Writes the data base for test purposes.

P5Ø11 Prints the pseudo code in symbolic format.

EXTCAL Calls a specified part of all external systems.

DEV Decodes device specification for DISP and LIST commands.

VARIB Decodes variable specification.

LOKFOR Searches the variable table.

PLOMRK Plots curves with marks.

EMPTY

WIFF Writes a line buffer.

IDENT Declares a FORTRAN system.

EDECL Contains entry points for declaration of variables.

DEFV Declares a variable in a FORTRAN system.

EDECLV Contains entry points for declaration of vectors.

DEFVEC Declares a vector.

EDECLM Contains entry points for declaration of matrices.

DEFMAT Declares a matrix.

USRSUB User subroutine.

2. Cross reference (called subroutines)

Calling Called

ACTHD : UNASS
 ALL :
 ALLERR:
 ALLER1: IWRITE
 ALLOC : LCOMPV HSTORV SEARCH ERROR ALLERR DEFVAR DEFLIT
 ALLWRK:
 AREA :
 ASHOW : LHOLL LTERM LFINAM LCOMPV HSTORV PLMODE VARIB FILES
 CRENAM FILRED AXES PLOMRK EMPTY
 AXES : PLMODE PLSET DEFPAR SCALE AXIS
 AXIS : MOVECM LINECM PFLOAT PLSYMB
 BOOLW : LFIND
 CALCUL: LCOND SYSTS
 CHANGE: EMESS TWOSTR COMPAR GAC PAC CROSS DISPL WCURL
 CHECK : ERROR
 CHEDEC: LFIND
 CMPSTA: LEFTP CODE CHECK REONO ERROR ALLOC NODEIN INSTR
 RELOC
 COALG : LHOLLS LTERM WBUFF
 COAREA: LINT LTERM PLMODE DEFPAR
 COAXES: LTERM LHOLLS LNUMB AXES
 CODE : LCOMPV FAC RIFF BOOLW FUNC RCHAR ERROR VARIAB
 LITT OPER
 CODISP: LTERM DEV VARIB HSTORV DISP DISPL
 COEDIT: LFINAM LTERM PLSET
 COERR : LNUMB LTERM
 COGET : LFINAM LTERM FCHECK GETV
 COLECT:
 COLIST: LFINAM LTERM DEV FCHECK HSTORV EJECT FSEEK RBUFF
 WBUFF FCLOSE
 COMLIN: LCOMPV IFAC FAC RIFF HSTOR2 HSTORV RFLOAT
 COMND : LFIND WRLINE IFAC RIFF EMESS
 COMP : ICOMNT IFAC ERROR RESW TYPE DECL HEAD CONDEF
 CMPSTA
 COMPAR: LCOMPV GAC
 CONDEF: ICOMNT LCOMPV LFIND RIFF RCHAR ERROR BOOLW FUNC
 DEFVAR DEPOS
 COPLOT: LTERM LHOLL IADDR VARIB LOKFOR HSTORV PLMODE IWRITE
 COSAVE: LFINAM LHOLL LHOLLS LTERM HSTORV SAVE
 COSHOW: LHOLL LTERM LHOLLS LFINAM LCOMPV HSTORV PLMODE VARIB
 FILES CRENAM FILRED EJECT PLOMRK EMPTY
 COSIMU: LTERM LHOLL LNUMB LHOLLS LFINAM HSTORV PLMODE IBREAK
 COSPLT: LTERM LINT PLSET
 COSTOR: LHOLL LTERM LCOMPV VARIB LOKFOR
 COSYST: LFINAM LTERM LHOLL LHOLLS HSTORV PLSET
 COTEXT: LTERM IFIXR PLMODE GAC PAC MOVECM PLSYMB
 COTURN: LHOLLS LTERM PLINIT PLSET
 CRENAM: HSTORV LENGTH PINT
 CURDIS:
 DECL : LFIND RIFF ERROR BOOLW FUNC DEFVAR
 DEFPAR: AREA PLLIM
 DEFV : HSTORV IADDR
 DEFVAR: ERROR HSTORV IADDR
 DEFVEC: CRENAM DEFV
 DERIV : DEPOS CALCUL FETCH
 DEV : LHOLL LHOLLS
 DIGITS: FAC

DISHDL: ERDIS PLCURS TWRITE TPOS TREAD
 DISP : EJECT IWRITE HSTORV FETCH
 DISPL : WBUFF EJECT
 DISPL : LCOMPV LOKFOR EJECT FETCH PHOLL PFLOAT PSPACE WIFF
 DRAW : EMPTY FETCH PLOMRK
 EDECL : HSTOR2 DEFV
 EDECLV: HSTOR2 DEFVEC
 EDITOR: ICOMNT LCOMPV OPEN HSTORV EMESS WCURL COMND DISPL
 RBUFF CLOSEF ERAS NARG RCURL FINLOC CROSS REINAP
 CHANGE RIFF FCLOSE
 EMESS :
 EQORD : NXTINS INLIST FENTER NODEIN INSTR NODMOV PREORD PREQND
 FCLOSE
 ERROR : CURDIS CROSS WERROR PSPACE PAC PTERM WBUFF
 ESIMN : ISENSW INTRAC INTERR COSYST EXTSYS EXTCAL CODISP PARINT
 COPLOT COAXES COSIMU COERR COLIST COEDIT COTURN COSAVE
 COGET COALG COSTOR COSHOW T5011 ASHOW COSPLT COAREA
 COTEXT WMESS USRSUB LPCOM
 EXTCAL: SYSTS
 EXTSUB: COMP
 EXTSYS: LCOMPV ALL HSTORV SYSTS ALLWRK ALLER1 CHEDEC
 F : DEPOS CALCUL FETCH
 FILDAT: LCOMPV LUFIND RDREAL WRREAL
 FILES : LCOMPV LUFIND FENTER WRINT RDREAL RDINT FSEEK FCLOSE
 FILRED: LUFIND RDREAL
 FINLOC: ONESTR WCURL RCURL DISPL COMPAR EMESS
 FORMAC: LCOMPV LFINF HSTORV RFLOAT READIN FENTER
 FORNXT: LCOMPV LCOMP2 HSTORV SUBST COMLIN IFIXR RFLOAT MACHDL
 HSTOR2 RBUFF
 FREE : LCOMPV FR2 HSTORV HSTOR2
 FR2 : LCOMPV HSTORV
 FUNC : LFINF
 GET : LCOMPV IFIXR HSTORV
 GETV : ICOMNT LCOMPV LFINF FSEEK RBUFF IFAC RCHAR RIFF
 DEPOS IWRITE FCLOSE
 HAMPC : DERIV OUTP
 HEAD : ICOMNT ACTHD ERROR
 IBDATA:
 ICOMNT: FAC
 IFGOLA: LCOMPV LCOMP2 LFINF SUBST IFIXR HSTORV MACHDL HSTOR2
 COMLIN RBUFF
 INFIX : LCOMPV NEWNOD PSPACE PHOLL PFLOAT GAC
 INIVAR: DEPOS FETCH
 INLIST:
 INSTR : ERROR
 INTEG : OUTPRK DERIV OUTP HAMPC RKADM RKFIX
 INTERR: IWRITE
 INTINI:
 INTR : LCOMPV LFINF HSTORV WBUFF WRLINE COMLIN SUBST RECLIN
 INTRAC: MACHDL INTR LPCOM RESEX
 ISIMN : LCOMPV INTINI PLINIT PLSET HSTORV SYSTS HSTOR2
 LARG : HSTORV IFIXR
 LCOND :
 LEFTP : LCOMPV RIFF ERROR BOOLW FUNC VARIAB RCHAR
 LENGTH: LCOMPV GAC
 LET : LCOMPV LFINF SUBST RFLOAT IFIXR HSTORV CRENAM MACHDL
 LFINF : LCOMPV
 LHOLL : LHNAME LDELIM LCOMPV
 LHOLLS: LHOLL
 LIFO :
 LINECM: PLMOVE
 LITT : RIFF ERROR

LOGARG: LARG
 LOKFOR: LCOMPV LFIND
 LPCOM : WRLINE
 LUFIND:
 MACEND: LCOMPV WBUFF FCLOSE FSEEK FENTER RBUFF HSTOR2 MACHDL
 IFAC RIFF
 MACHDL: LCOMPV LFIND FCLOSE FSEEK PUT HSTORV RFLOAT RBUFF
 IFAC RIFF
 MOVECM: PLMOVE
 NARG : ICOMNT RIFF EMESS
 NEWNOD:
 NODEIN:
 NODMOV:
 NXTINS:
 ONESTR: LCOMPV GAC EMESS
 OPEN : FSEEK FENTER FCLOSE
 OPER : LFIND
 OUTP : CALCUL EXTCAL STORE DRAW BREAK
 OUTPRK: OUTP
 PARINT: LHOLL LNUMB LTERM VARIB LOKFOR FETCH DEPOS
 PFLOAT: LCOMPV IMACON PAC PINT GAC
 PHOLL : LCOMPV GAC PAC
 PINT : PAC
 PLINIT: PLDEV
 PLLIM :
 PLOMRK: MOVECM LINECM PLSYMB PLRSET
 PREORD: NXTINS NODMOV PREQND
 PREQND: HSTORV INFIX
 PSPACE: PAC
 PTERM : PAC
 PUT : LCOMPV HSTORV IFIXR RFLOAT
 P5011 :
 RABC : FAC PAC
 RBUFF : LENGTH
 RCHAR : FAC
 RCURL : RBUFF EXTSUB WBUFF
 READ : LFIND HSTORV READIN RFLOAT PUT MACHDL
 READIN: LCOMPV WRLINE COMLIN HSTORV
 RECLIN: LCOMPV PAC PSPACE PHOLL IFIXR PINT PFLOAT GAC
 PTERM HSTORV
 REINAP: ONESTR EMESS WCURL CROSS GAC PAC DISPL
 RELOC :
 REONO: LIFO PUSH POP ERROR
 RESEX : HSTORV SWITCH FORMAC READ IFGOLA LET RESUME FORNXT
 WRITE FREE MACEND WRLINE LPCOM
 RESUME: LCOMP2 MACHDL
 RESW : LFIND RIFF
 RFP : LCOMPV IMACON GAC FAC SIGN1 DIGITS
 RIFF : RFP RABC FAC
 RK :
 RKADM : F RK
 RKFIX : F OUTPRK RK
 SAVE : LCOMPV LFIND FENTER PHOLL WIFF FETCH PFLOAT FCLOSE
 SCALE :
 SEARCH: LCOMPV LFIND ERROR
 SIGN1 : LCOMPV FAC
 SIMNON: ISIMN ESIMN SIMNSY EQORD SIMU EDITOR
 SIMNSY: HSTORV EDITOR COLECT
 SIMU : INIVAR CALCUL FETCH DEPOS STORE INTEG EMPTY EXTCAL
 STORE : FILES FILDAT HSTORV FETCH
 SUBST : LCOMPV HSTORV IFIXR GET
 SWITCH: LFIND SUBST

TWOSTR: LCOMPV GAC EMESS
 TYPE : ICOMNT LFIND RESW ERROR RIFF HSTORV
 T5011 : LTERM HSTORV PREQND P5011
 UNASS : LCOMPV CURDIS WIFF CROSS PHOLL PSPACE DISPL
 USRSUB:
 VARIAB: LCOMPV HSTORV RIFF ERROR RCHAR
 VARIB : LHOLL LHNAME HSTORV
 WBUFF : IWRITE WRHDL
 WERROR:
 WIFF : PTERM WBUFF PSPACE GAC PAC
 WMESS : IWRITE
 WRITE : LCOMPV SUBST WRT LPCOM EJECT WBUFF WRT2
 WRLINE: HSTORV GAC PAC PSPACE PTERM WRHDL WBUFF RBUFF
 TRHDL
 WRT : LCOMPV PHOLL PINT IFIXR PFLOAT PSPACE GAC PAC
 PTERM
 WRT2 : IMACON EJECT PSPACE PINT PTERM WBUFF PHOLL PAC
 IFIXR PFLOAT

3. Cross reference (calling subroutines)

Called Calling

ACTHD : HEAD
 ALL : EXTSYS
 ALLERR: ALLOC
 ALLER1: EXTSYS
 ALLOC : CMPSTA
 ALLWRK: EXTSYS
 AREA : DEFPAR
 ASHOW : ESIMN
 AXES : ASHOW COAXES
 AXIS : AXES
 BOOLW : CODE CONDEF DECL LEFTP
 BREAK : OUTP
 CALCUL: DERIV F OUTP SIMU
 CHANGE: EDITOR
 CHECK : CMPSTA
 CHEDEC: EXTSYS
 CLOSEF: EDITOR
 CMPSTA: COMP
 COALG : ESIMN
 COAREA: ESIMN
 COAXES: ESIMN
 CODE : CMPSTA
 CODISP: ESIMN
 COEDIT: ESIMN
 COERR : ESIMN
 COGET : ESIMN
 COLECT: SIMNSY
 COLIST: ESIMN
 COMLIN: FORNXT IFGOLA INTR READIN
 COMND : EDITOR
 COMP : EXTSUB
 COMPAR: CHANGE FINLOC
 CONDEF: COMP
 COPLOT: ESIMN
 COSAVE: ESIMN
 COSHOW: ESIMN
 COSIMU: ESIMN
 COSPLT: ESIMN
 COSTOR: ESIMN
 COSYST: ESIMN
 COTEXT: ESIMN
 COTURN: ESIMN
 CRENAM: ASHOW COSHOW DEFVEC LET
 CROSS : CHANGE EDITOR ERROR REINAP UNASS
 CURDIS: ERROR UNASS
 DECL : COMP
 DEFLIT: ALLOC
 DEFPAR: AXES COAREA
 DEFV : DEFVEC EDECL
 DEFVAR: ALLOC CONDEF DECL
 DEFVEC: EDECLV
 DEPOS : CONDEF DERIV F GETV INIVAR PARINT SIMU
 DERIV : HAMPC INTEG
 DEV : CODISP COLIST
 DIGITS: RFP
 DISP : CODISP

DISPL : CHANGE EDITOR FINLOC REINAP UNASS
DISPL : CODISP
DRAW : OUTP
EDITOR: SIMNON SIMNSY
EJECT : COLIST COSHOW DISP DISPL DISPL WRITE WRT2
EMESS : CHANGE COMND EDITOR FINLOC NARG ONESTR REINAP TWOSTR
EMPTY : ASHOW COSHOW DRAW SIMU
EQORD : SIMNON
ERAS : EDITOR
ERDIS : DISHDL
ERROR : ALLOC CHECK CMPSTA CODE COMP CONDEF DECL DEFVAR
HEAD INSTR LEFTP LITT REPONO SEARCH TYPE VARIAB
ESIMN : SIMNON
EXTCAL: ESIMN OUTP SIMU
EXTSUB: RCURL
EXTSYS: ESIMN
F : RKADM RKFIX
FAC : CODE COMLIN DIGITS ICOMNT RABC RCHAR RFP RIFF
SIGN1
FCHECK: COGET COLIST
FCLOSE: COLIST EDITOR EQORD FILES GETV MACEND MACHDL OPEN
SAVE
FENTER: EQORD FILES FORMAC MACEND OPEN SAVE
FETCH : DERIV DISP DISPL DRAW F INIVAR PARINT SAVE
SIMU STORE
FILDAT: STORE
FILES : ASHOW COSHOW STORE
FILRED: ASHOW COSHOW
FINLOC: EDITOR
FORMAC: RESEX
FORNXT: RESEX
FREE : RESEX
FR2 : FREE
FSEEK : COLIST FILES GETV MACEND MACHDL OPEN
FUNC : CODE CONDEF DECL LEFTP
GAC : CHANGE COMPAR COTEXT INFIX LENGTH ONESTR PFLOAT PHOLL
RECLIN REINAP RFP TWOSTR WIFF WRLINE WRT
GET : SUBST
GETV : COGET
HAMPC : INTEG
HEAD : COMP
HSTORV: ALLOC ASHOW CODISP COLIST COMLIN COPLOT COSAVE COSHOW
COSIMU COSYST CRENAM DEFV DEFVAR DISP EDITOR EXTSYS
FORMAC FORNXT FREE FR2 GET IFGOLA INTR ISIMN
LARG LET MACHDL PREQND PUT READ READIN RECLIN
RESEX SIMNSY STORE SUBST TYPE T5011 VARIAB VARIB
WRLINE
HSTOR2: COMLIN EDECL EDECLV FORNXT FREE IFGOLA ISIMN MACEND
IADDR : COPLOT DEFV DEFVAR
IBREAK: COSIMU
ICOMNT: COMP CONDEF EDITOR GETV HEAD NARG TYPE
IFAC : COMLIN COMND COMP GETV MACEND MACHDL
IFGOLA: RESEX
IFIXR : COTEXT FORNXT GET IFGOLA LARG LET PUT RECLIN
SUBST WRT WRT2
IMACON: PFLOAT RFP WRT2
INFIX : PREQND
INIVAR: SIMU
INLIST: EQORD
INSTR : CMPSTA EQORD
INTEG : SIMU
INTERR: ESIMN

INTINI: ISIMN
 INTR : INTRAC
 INTRAC: ESIMN
 ISENSW: ESIMN
 ISIMN : SIMNON
 IWRITE: ALLER1 COPLOT DISP GETV INTERR WBUFF WMESS
 LARG : LOGARG
 LCOMPV: ALLOC ASHOW CODE COMLIN COMPAR CONDEF COSHOW COSTOR
 DISPl EDITOR EXTSYS FILDAT FILES FORMAC FORNXT FREE
 FR2 GET GETV IFGOLA INFIX INTR ISIMN LEFTP
 LENGTH LET LFINL LHOLL LOKFOR MACEND MACHDL ONESTR
 PFLOAT PHOLL PUT READIN RECLIN RFP SAVE SEARCH
 SIGN1 SUBST TWOSTR UNASS VARIAB WRITE WRT
 LCOMP2: FORNXT IFGOLA RESUME
 LCOND : CALCUL
 LDELIM: LHOLL
 LEFTP : CMPSTA
 LENGTH: CRENAM RBUFF
 LET : RESEX
 LFINAM: ASHOW COEDIT COGET COLIST COSAVE COSHOW COSIMU COSYST
 LFINL : BOOLW CHEDEC COMND CONDEF DECL FORMAC FUNC GETV
 IFGOLA INTR LET LOKFOR MACHDL OPER READ RESW
 SAVE SEARCH SWITCH TYPE
 LHNAME: LHOLL VARIB
 LHOLL : ASHOW COPLOT COSAVE COSHOW COSIMU COSTOR COSYST DEV
 LHOLLS PARINT VARIB
 LHOLLS: COALG COAXES COSAVE COSHOW COSIMU COSYST COTURN DEV
 LIFO : REONO
 LINECM: AXIS PLOMRK
 LINT : COAREA COSPLT
 LITT : CODE
 LNUMB : COAXES COERR COSIMU PARINT
 LOKFOR: COPLOT COSTOR DISPl PARINT
 LPCOM : ESIMN INTRAC RESEX WRITE
 LTERM : ASHOW COALG COAREA COAXES CODISP COEDIT COERR COGET
 COLIST COPLOT COSAVE COSHOW COSIMU COSPLT COSTOR COSYST
 COTEXT COTURN PARINT T5011
 LUFIND: FILDAT FILES FILRED
 MACEND: RESEX
 MACHDL: FORNXT IFGOLA INTRAC LET MACEND READ RESUME
 MOVECM: AXIS COTEXT PLOMRK
 NARG : EDITOR
 NEWNOD: INFIX
 NODEIN: CMPSTA EQORD
 NODMOV: EQORD PREORD
 NXTINS: EQORD PREORD
 ONESTR: FINLOC REINAP
 OPEN : EDITOR
 OPER : CODE
 OUTP : HAMPC INTEG OUTPRK
 OUTPRK: INTEG RKFIX
 PAC : CHANGE COTEXT ERROR PFLOAT PHOLL PINT PSPACE PTERM
 RABC RECLIN REINAP WIFF WRLINE WRT WRT2
 PARINT: ESIMN
 PFLOAT: AXIS DISPl INFIX RECLIN SAVE WRT WRT2
 PHOLL : DISPl INFIX RECLIN SAVE UNASS WRT WRT2
 PINT : CRENAM PFLOAT RECLIN WRT WRT2
 PLCURS: DISHDL
 PLDEV : PLINIT
 PLINIT: COTURN ISIMN
 PLLIM : DEFPAR
 PLMODE: ASHOW AXES COAREA COPLOT COSHOW COSIMU COTEXT

PLMOVE: LINECM MOVECM
 PLOMRK: ASHOW COSHOW DRAW
 PLRSET: PLOMRK
 PLSET : AXES COEDIT COSPLT COSYST COTURN ISIMN
 PLSYMB: AXIS COTEXT PLOMRK
 POP : REPONO
 PREORD: EQORD
 PREQND: EQORD PREORD T5011
 PSPACE: DISPL ERROR INFIX RECLIN UNASS WIFF WRLINE WRT
 WRT2
 PTERM : ERROR RECLIN WIFF WRLINE WRT WRT2
 PUSH : REPONO
 PUT : MACHDL READ
 P5011 : T5011
 RABC : RIFF
 RBUFF : COLIST EDITOR FORNXT GETV IFGOLA MACEND MACHDL RCURL
 WRLINE
 RCHAR : CODE CONDEF GETV LEFTP VARIAB
 RCURL : EDITOR FINLOC
 RDINT : FILES
 RDREAL: FILDAT FILES FILRED
 READ : RESEX
 READIN: FORMAC READ
 RECLIN: INTR
 REINAP: EDITOR
 RELOC : CMPSTA
 REPONO: CMPSTA
 RESEX : INTRAC
 RESUME: RESEX
 RESW : COMP TYPE
 RFLOAT: COMLIN FORMAC FORNXT LET MACHDL PUT READ
 RFP : RIFF
 RIFF : CODE COMLIN COMND CONDEF DECL EDITOR GETV LEFTP
 LITT MACEND MACHDL NARG RESW TYPE VARIAB
 RK : RKADM RKFIX
 RKADM : INTEG
 RKFIX : INTEG
 SAVE : COSAVE
 SCALE : AXES
 SEARCH: ALLOC
 SIGN1 : RFP
 SIMNSY: SIMNON
 SIMU : SIMNON
 STORE : OUTP SIMU
 SUBST : FORNXT IFGOLA INTR LET SWITCH WRITE
 SWITCH: RESEX
 SYSTS : CALCUL EXTCAL EXTSYS ISIMN
 TPOS : DISHDL
 TREAD : DISHDL
 TRHDL : WRLINE
 TWOSTR: CHANGE
 TWRITE: DISHDL
 TYPE : COMP
 T5011 : ESIMN
 UNASS : ACTHD
 USRSUB: ESIMN
 VARIAB: CODE LEFTP
 VARIB : ASHOW CODISP COPLLOT COSHOW COSTOR PARINT
 WBUFF : COALG COLIST DISPL ERROR INTR MACEND RCURL WIFF
 WRITE WRLINE WRT2
 WCURL : CHANGE EDITOR FINLOC REINAP
 WERROR: ERROR

WIFF : DISPL SAVE UNASS
WMESS : ESIMN
WRHDL : WBUFF WRLINE
WRINT : FILES
WRITE : RESEX
WRLINE: COMND INTR LPCOM READIN RESEX
WRREAL: FILDAT
WRT : WRITE
WRT2 : WRITE

4. Subroutines with entry points

Subr. Entry point

DEFVAR: DEFLIT
DISHDL: EJECT IWRITE PLMODE PLSET TRHDL WRHDL
DISPL : CROSS ERAS
LIFO : POP PUSH
LOGARG: LDELIM LHNAME LINT LNUMB LTERM
OPEN : CLOSEF
PLOMRK: EMPTY
RCURL : WCURL

5. Subroutine trees

The notation

```

SUB1    SUB2    SUB3
          SUB4
          SUB5

```

means that

```

SUB1 calls SUB2 and SUB5,
SUB2 calls SUB3 and SUB4

```

The names in the subroutine trees are sometimes followed by special signs with the following meaning:

- + Installation dependent subroutine.
- Entry point in a subroutine (see appendix 4).
- * The root of an other subroutine tree.

Basic subroutines:

```

RIFF    RFP      LCOMPV+
          IMACON+
          GAC+
          FAC+
          SIGN1   LCOMPV+
                   FAC+
          DIGITS  FAC+
          RABC    FAC+
                   PAC+
          FAC+

```

```

WRLINE  HSTORV+
          GAC+
          PAC+
          PSPACE PAC+
          PTERM  PAC+
          WRHDL*
          WBUFF  IWRITE*
                   WRHDL*
          RBUFF  LENGTH LCOMPV+
                   GAC+
          TRHDL*

```

```

DISHDL  ERDIS+
          PLCURS+
          TWRITE+
          TPOS+
          TREAD+

```

Intrac:

INTRAC	MACHDL*						
	INTR	LCOMPV+					
		LFIND	LCOMPV+				
		HSTORV+					
		WBUF	IWRITE*				
			WRHDL*				
		WRLINE*					
		COMLIN	LCOMPV+				
			IFAC+				
			FAC+				
			RIFF*				
			HSTOR2+				
			HSTORV+				
			RFLOAT+				
	SUBST	LCOMPV+					
		HSTORV+					
		IFIXR+					
		GET	LCOMPV+				
			IFIXR+				
			HSTORV+				
		RECLIN	LCOMPV+				
			PAC+				
			PSPACE	PAC+			
			PHOLL	LCOMPV+			
				GAC+			
				PAC+			
			IFIXR+				
			PINT	PAC+			
			PFLOAT	LCOMPV+			
				IMACON+			
				PAC+			
				PINT	PAC+		
				GAC+			
			GAC+				
			PTERM	PAC+			
			HSTORV+				
	LPCOM	WRLINE*					
	RESEX	HSTORV+					
		SWITCH	LFIND	LCOMPV+			
			SUBST	LCOMPV+			
				HSTORV+			
				IFIXR+			
				GET	LCOMPV+		
					IFIXR+		
					HSTORV+		
		FORMAC	LCOMPV+				
			LFIND	LCOMPV+			
			HSTORV+				
			RFLOAT+				
			READIN	LCOMPV+			
				WRLINE*			
				COMLIN	LCOMPV+		
					IFAC+		
					FAC+		
					RIFF*		
1	2	3	4	5	6	7	8

1	2	3	4	5	6	7	8
					HSTOR2+		
					HSTORV+		
					RFLOAT+		
				HSTORV+			
		READ	FENTER+				
			LFIND	LCOMPV+			
			HSTORV+				
			READIN	LCOMPV+			
				WRLINE*			
				COMLIN	LCOMPV+		
					IFAC+		
					FAC+		
					RIFF*		
					HSTOR2+		
					HSTORV+		
					RFLOAT+		
				HSTORV+			
			RFLOAT+				
			PUT	LCOMPV+			
				HSTORV+			
				IFIXR+			
				RFLOAT+			
		IFGOLA	MACHDL*				
			LCOMPV+				
			LCOMP2+				
			LFIND	LCOMPV+			
			SUBST	LCOMPV+			
				HSTORV+			
				IFIXR+			
				GET	LCOMPV+		
					IFIXR+		
					HSTORV+		
			IFIXR+				
			HSTORV+				
			MACHDL*				
			HSTOR2+				
			COMLIN	LCOMPV+			
				IFAC+			
				FAC+			
				RIFF*			
				HSTOR2+			
				HSTORV+			
				RFLOAT+			
			RBUFF	LENGTH	LCOMPV+		
					GAC+		
		LET	LCOMPV+				
			LFIND	LCOMPV+			
			SUBST	LCOMPV+			
				HSTORV+			
				IFIXR+			
				GET	LCOMPV+		
					IFIXR+		
					HSTORV+		
			RFLOAT+				
			IFIXR+				

1	2	3	4	5	6	7	8
			HSTORV+				
			CRENAM	HSTORV+			
				LENGTH	LCOMPV+		
					GAC+		
				PINT	PAC+		
			MACHDL*				
	RESUME		LCOMP2+				
			MACHDL*				
	FORNXT		LCOMPV+				
			LCOMP2+				
			HSTORV+				
			SUBST	LCOMPV+			
				HSTORV+			
				IFIXR+			
				GET	LCOMPV+		
					IFIXR+		
					HSTORV+		
			COMLIN	LCOMPV+			
				IFAC+			
				FAC+			
				RIFF*			
				HSTOR2+			
				HSTORV+			
				RFLOAT+			
			IFIXR+				
			RFLOAT+				
			MACHDL*				
			HSTOR2+				
			RBUFF	LENGTH	LCOMPV+		
					GAC+		
	WRITE		LCOMPV+				
			SUBST	LCOMPV+			
				HSTORV+			
				IFIXR+			
				GET	LCOMPV+		
					IFIXR+		
					HSTORV+		
			WRT	LCOMPV+			
				PHOLL	LCOMPV+		
					GAC+		
					PAC+		
				PINT	PAC+		
				IFIXR+			
				PFLOAT	LCOMPV+		
					IMACON+		
					PAC+		
					PINT	PAC+	
					GAC+		
				PSPACE	PAC+		
				GAC+			
				PAC+			
				PTERM	PAC+		
			LPCOM	WRLINE*			
			EJECT*				
			WBUFF	IWRITE*			

1	2	3	4	5	6	7	8
			WRT2	WRHDL*			
				IMACON+			
				EJECT*			
				PSPACE	PAC+		
				PINT	PAC+		
				PTERM	PAC+		
				WBUF	IWRITE*		
					WRHDL*		
				PHOLL	LCOMPV+		
					GAC+		
					PAC+		
				IFIXR+			
				PFLOAT	LCOMPV+		
					IMACON+		
					PAC+		
					PINT	PAC+	
					GAC+		
		FREE	LCOMPV+				
			FR2	LCOMPV+			
				HSTORV+			
				HSTOR2+			
		MACEND	LCOMPV+				
			WBUF	IWRITE*			
				WRHDL*			
			FCLOSE+				
			FSEEK+				
			FENTER+				
			RBUF	LENGTH	LCOMPV+		
					GAC+		
			HSTOR2+				
			MACHDL*				
			IFAC+				
			RIFF*				
		WRLINE*					
		LPCOM	WRLINE*				

Subtree:

MACHDL	LCOMPV+		
	LFIND	LCOMPV+	
	FCLOSE+		
	FSEEK+		
	PUT	LCOMPV+	
		HSTORV+	
		IFIXR+	
		RFLOAT+	
	HSTORV+		
	RFLOAT+		
	RBUF	LENGTH	LCOMPV+
			GAC+
	IFAC+		
	RIFF*		

The editor:

```

EDITOR  ICOMNT  FAC+
        LCOMPV+
        OPEN    FSEEK+
                FENTER+
                FCLOSE+

        HISTORV+
        EMESS
        WCURL-  RBUFF  LENGTH  LCOMPV+
                GAC+

                EXTSUB  COMP*
                WBUFF  IWRITE*
                WRHDL*
        COMND   LFIND   LCOMPV+
                WRLINE*
                IFAC+
                RIFF*
                EMESS
        DISPL  WBUFF  IWRITE*
                WRHDL*

                EJECT*
        RBUFF  LENGTH  LCOMPV+
                GAC+

        CLOSEF- FSEEK+
                FENTER+
                FCLOSE+

        ERAS-  WBUFF  IWRITE*
                WRHDL*

                EJECT*
        NARG   ICOMNT  FAC+
                RIFF*
                EMESS
        RCURL  RBUFF  LENGTH  LCOMPV+
                GAC+

                EXTSUB  COMP*
                WBUFF  IWRITE*
                WRHDL*
        FINLOC ONESTR  LCOMPV+
                GAC+
                EMESS
                WCURL-  RBUFF  LENGTH  LCOMPV+
                GAC+

                EXTSUB  COMP*
                WBUFF  IWRITE*
                WRHDL*
        RCURL  RBUFF  LENGTH  LCOMPV+
                GAC+

                EXTSUB  COMP*
                WBUFF  IWRITE*
                WRHDL*
        DISPL  WBUFF  IWRITE*
                WRHDL*

                EJECT*
        COMPAR LCOMPV+
                GAC+
        EMESS

```

1	2	3	4	5	6	7	8
	CROSS-	WBUF	IWRITE*				
			WRHDL*				
	REINAP	EJECT*					
		ONESTR	LCOMPV+				
			GAC+				
		EMESS	EMESS				
		WCURL-	RBUF	LENGTH	LCOMPV+		
					GAC+		
			EXTSUB	COMP*			
			WBUF	IWRITE*			
				WRHDL*			
		CROSS-	WBUF	IWRITE*			
				WRHDL*			
			EJECT*				
		GAC+					
		PAC+					
		DISPL	WBUF	IWRITE*			
				WRHDL*			
			EJECT*				
	CHANGE	EMESS					
		TWOSTR	LCOMPV+				
			GAC+				
			EMESS				
		COMPAR	LCOMPV+				
			GAC+				
		GAC+					
		PAC+					
		CROSS-	WBUF	IWRITE*			
				WRHDL*			
			EJECT*				
		DISPL	WBUF	IWRITE*			
				WRHDL*			
			EJECT*				
		WCURL-	RBUF	LENGTH	LCOMPV+		
					GAC+		
			EXTSUB	COMP*			
			WBUF	IWRITE*			
				WRHDL*			
	RIFF*						
	FCLOSE+						

The compiler:

```

-----
COMP      ICOMNT  FAC+
          IFAC+
          ERROR*
          RESW      LFIND  LCOMPV+
                   RIFF*
          TYPE      ICOMNT  FAC+
                   LFIND  LCOMPV+
                   RESW   LFIND  LCOMPV+
                   RIFF*
                   ERROR*
                   RIFF*
                   HSTORV+
          DECL      LFIND  LCOMPV+
                   RIFF*
                   ERROR*
                   BOOLW  LFIND  LCOMPV+
                   FUNC   LFIND  LCOMPV+
                   DEFVAR  ERROR*
                   HSTORV+
                   IADDR+
          HEAD      ICOMNT  FAC+
                   ACTHD  UNASS  LCOMPV+
                                   CURDIS
                                   WIFF  PTERM  PAC+
                                   WBUFF  IWRITE*
                                   WRHDL*
                                   PSPACE PAC+
                                   GAC+
                                   PAC+
                                   CROSS- WBUFF  IWRITE*
                                   WRHDL*
                                   PHOLL  EJECT*
                                   LCOMPV+
                                   GAC+
                                   PAC+
                                   PSPACE PAC+
                                   DISPL  WBUFF  IWRITE*
                                   WRHDL*
                                   EJECT*
          CONDEF    ERROR*
                   ICOMNT  FAC+
                   LCOMPV+
                   LFIND  LCOMPV+
                   RIFF*
                   RCHAR  FAC+
                   ERROR*
                   BOOLW  LFIND  LCOMPV+
                   FUNC   LFIND  LCOMPV+
                   DEFVAR  ERROR*
                   HSTORV+
                   IADDR+
          CMPSTA    DEPOS+
                   LEFTP  LCOMPV+
                                   RIFF*
                                   ERROR*

```

1

2

3

4

5

6

7

8

1	2	3	4	5	6	7	8
			BOOLW	LFIND	LCOMPV+		
			FUNC	LFIND	LCOMPV+		
			VARIAB	LCOMPV+			
				HSTORV+			
				RIFF*			
				ERROR*			
				RCHAR	FAC+		
				FAC+			
		CODE	RCHAR				
			LCOMPV+				
			FAC+				
			RIFF*				
			BOOLW	LFIND	LCOMPV+		
			FUNC	LFIND	LCOMPV+		
			RCHAR	FAC+			
			ERROR*				
			VARIAB	LCOMPV+			
				HSTORV+			
				RIFF*			
				ERROR*			
				RCHAR	FAC+		
				RIFF*			
				ERROR*			
			LITT				
				RIFF*			
				ERROR*			
				LFIND	LCOMPV+		
		CHECK	OPER				
		REONO	ERROR*				
			LIFO				
			PUSH-				
			POP-				
			ERROR*				
		ERROR*					
		ALLOC	LCOMPV+				
			HSTORV+				
			SEARCH	LCOMPV+			
				LFIND	LCOMPV+		
				ERROR*			
			ERROR*				
			ALLERR				
			DEFVAR	ERROR*			
				HSTORV+			
				IADDR+			
			DEFLIT-	ERROR*			
				HSTORV+			
				IADDR+			
		NODEIN					
		INSTR	ERROR*				
		RELOC					

Subtree:

ERROR	CURDIS		
	CROSS-	WBUFF	IWRITE*
			WRHDL*
		EJECT*	
	WERROR		
	PSPACE	PAC+	
	PAC+		
	PTERM	PAC+	
	WBUFF	IWRITE*	
		WRHDL*	

Simulation:

SIMU	INIVAR	DEPOS+			
		FETCH+			
	CALCUL	LCOND			
		SYSTS+			
	FETCH+				
	DEPOS+				
	STORE	FILES	LCOMPV+		
			LUFIND		
			FENTER+		
			WRINT+		
			RDREAL+		
			RDINT+		
			FSEEK+		
			FCLOSE+		
		FILDAT	LCOMPV+		
			LUFIND		
			RDREAL+		
			WRREAL+		
		HSTORV+			
		FETCH+			
INTEG	OUTPRK	OUTP*			
	DERIV	DEPOS+			
		CALCUL	LCOND		
			SYSTS+		
			FETCH+		
	OUTP*				
	HAMPC	DERIV	DEPOS+		
			CALCUL	LCOND	
				SYSTS+	
			FETCH+		
		OUTP*			
	RKADM	F	DEPOS+		
			CALCUL	LCOND	
				SYSTS+	
			FETCH+		
		RK			
	RKFIX	F	DEPOS+		
			CALCUL	LCOND	
				SYSTS+	
			FETCH+		
		OUTPRK	OUTP*		
		RK			
	DASP3+				
EMPTY-	MOVECM	PLMOVE+			
	LINECM	PLMOVE+			
	PLSYMB+				
	PLRSET+				
EXTCAL	SYSTS+				

Subtree:

OUTP	CALCUL	LCOND		
		SYSTS+		
	EXTCAL	SYSTS+		
	STORE	FILES	LCOMPV+	
			LUFIND	
			FENTER+	
			WRINT+	
			RDREAL+	
			RDINT+	
			FSEEK+	
			FCLOSE+	
		FILDAT	LCOMPV+	
			LUFIND	
			RDREAL+	
			WRREAL+	
		HSTORV+		
		FETCH+		
	DRAW	EMPTY-	MOVECM	PLMOVE+
			LINECM	PLMOVE+
			PLSYMB+	
			PLRSET+	
		FETCH+		
		PLOMRK	MOVECM	PLMOVE+
			LINECM	PLMOVE+
			PLSYMB+	
			PLRSET+	
	BREAK+			

1	2	3	4	5	6	7	8
			DISPL	FETCH+ LCOMPV+ LOKFOR	LCOMPV+ LFIND	LCOMPV+	
				EJECT* FETCH+ PHOLL	LCOMPV+ GAC+ PAC+		
				PFLOAT	LCOMPV+ IMACON+ PAC+ PINT	PAC+	
				PSPACE WIFF	PAC+ PTERM WBUFF	PAC+ IWRITE* WRHDL*	
		PARINT	LHOLL	LHNAME- LDELIM-	LARG	HSTORV+ IFIXR+ HSTORV+ IFIXR+	
			LNUMB-	LCOMPV+ LARG	HSTORV+ IFIXR+		
			LTERM-	LARG	HSTORV+ IFIXR+		
			VARIB	LHOLL	LHNAME- LDELIM-	LARG	HSTORV+ IFIXR+ HSTORV+ IFIXR+
				LHNAME-	LARG	HSTORV+ IFIXR+	
			LOKFOR	HSTORV+ LCOMPV+ LFIND	LCOMPV+		
		COPLOT	LTERM-	LARG	HSTORV+ IFIXR+		
			LHOLL	LHNAME- LDELIM-	LARG	HSTORV+ IFIXR+ HSTORV+ IFIXR+	
			IADDR+ VARIB	LHOLL	LHNAME- LDELIM-	LARG	HSTORV+ IFIXR+ HSTORV+ IFIXR+
					LCOMPV+		
1	2	3	4	5	6	7	8

1	2	3	4	5	6	7	8
				LHNAME-	LARG	HSTORV+	
						IFIXR+	
			LOKFOR	HSTORV+	LCOMPV+		
				LFIND	LCOMPV+		
			HSTORV+				
			PLMODE*				
			IWRITE*				
		COAXES	LTERM-	LARG	HSTORV+		
					IFIXR+		
			LHOLLS	LHOLL	LHNAME-	LARG	HSTORV+
							IFIXR+
					LDELIM-	LARG	HSTORV+
							IFIXR+
			LNUMB-	LARG	LCOMPV+		
					HSTORV+		
			AXES	PLMODE*	IFIXR+		
				PLSET*			
				DEFPAR	AREA		
					PLLIM		
				SCALE			
				AXIS	MOVECM	PLMOVE+	
					LINECM	PLMOVE+	
					PFLOAT	LCOMPV+	
						IMACON+	
						PAC+	
						PINT	PAC+
						GAC+	
		COSIMU	LTERM-	LARG	PLSYMB+		
					HSTORV+		
			LHOLL	LHNAME-	IFIXR+		
					LARG	HSTORV+	
				LDELIM-	LARG	HSTORV+	
						IFIXR+	
				LCOMPV+			
			LNUMB-	LARG	HSTORV+		
					IFIXR+		
			LHOLLS	LHOLL	LHNAME-	LARG	HSTORV+
							IFIXR+
					LDELIM-	LARG	HSTORV+
							IFIXR+
					LCOMPV+		
			LFINAM+				
			HSTORV+				
			PLMODE*				
			IBREAK+				
		COERR	LNUMB-	LARG	HSTORV+		
					IFIXR+		
			LTERM-	LARG	HSTORV+		
					IFIXR+		
		COLIST	LFINAM+				
			LTERM-	LARG	HSTORV+		
					IFIXR+		

1	2	3	4	5	6	7	8
			DEV	LHOLL	LHNAME-	LARG	HSTORV+
					LDELIM-	LARG	IFIXR+
					LDELIM-	LARG	HSTORV+
					LDELIM-	LARG	IFIXR+
				LHOLLS	LHOLL	LHNAME-	LARG
						LDELIM-	LARG
						LCOMPV+	
			FCHECK+				
			HSTORV+				
			EJECT*				
			FSEEK+				
			RBUFF	LENGTH	LCOMPV+		
					GAC+		
			WBUF	IWRITE*			
				WRHDL*			
		COEDIT	FCLOSE+				
			LFINAM+				
			LTERM-	LARG	HSTORV+		
					IFIXR+		
			PLSET*				
		COTURN	LHOLLS	LHOLL	LHNAME-	LARG	HSTORV+
					LDELIM-	LARG	IFIXR+
					LDELIM-	LARG	HSTORV+
					LDELIM-	LARG	IFIXR+
			LTERM-	LARG	LCOMPV+		
					HSTORV+		
					IFIXR+		
			PLINIT	PLDEV+			
			PLSET*				
		COSAVE	LFINAM+				
			LHOLL	LHNAME-	LARG	HSTORV+	IFIXR+
				LDELIM-	LARG	HSTORV+	IFIXR+
				LDELIM-	LARG	HSTORV+	IFIXR+
			LHOLLS	LHOLL	LHNAME-	LARG	HSTORV+
					LDELIM-	LARG	IFIXR+
					LDELIM-	LARG	HSTORV+
					LDELIM-	LARG	IFIXR+
			LTERM-	LARG	LCOMPV+		
					HSTORV+		
					IFIXR+		
			HSTORV+				
			SAVE	LCOMPV+			
				LFIND	LCOMPV+		
				FENTER+			
				PHOLL	LCOMPV+		
					GAC+		
					PAC+		
				WIFF	PTERM	PAC+	
					WBUF	IWRITE*	
						WRHDL*	
					PSPACE	PAC+	
					GAC+		
1	2	3	4	5	6	7	8

1	2	3	4	5	6	7	8
						PAC+	
				FETCH+			
				PFLOAT	LCOMPV+		
					IMACON+		
					PAC+		
					PINT	PAC+	
					GAC+		
				FCLOSE+			
	COGET		LFINAM+				
			LTERM-	LARG	HSTORV+		
					IFIXR+		
			FCHECK+				
			GETV	ICOMNT	FAC+		
				LCOMPV+			
				LFIN	LCOMPV+		
				FSEEK+			
				RBUFF	LENGTH	LCOMPV+	
						GAC+	
				IFAC+			
				RCHAR	FAC+		
				RIFF*			
				DEPOS+			
				IWRITE*			
				FCLOSE+			
	COALG		LHOLLS	LHOLL	LHNAME-	LARG	HSTORV+
							IFIXR+
					LDELIM-	LARG	HSTORV+
							IFIXR+
				LTERM-	LARG	LCOMPV+	
						HSTORV+	
						IFIXR+	
			WBUF	IWRITE*			
				WRHDL*			
	COSTOR		LHOLL	LHNAME-	LARG	HSTORV+	
						IFIXR+	
				LDELIM-	LARG	HSTORV+	
						IFIXR+	
				LCOMPV+			
			LTERM-	LARG	HSTORV+		
					IFIXR+		
			LCOMPV+				
			VARIB	LHOLL	LHNAME-	LARG	HSTORV+
							IFIXR+
					LDELIM-	LARG	HSTORV+
							IFIXR+
					LCOMPV+		
				LHNAME-	LARG	HSTORV+	
						IFIXR+	
				HSTORV+			
			LOKFOR	LCOMPV+			
				LFIN	LCOMPV+		
	COSHOW		LHOLL	LHNAME-	LARG	HSTORV+	
						IFIXR+	
				LDELIM-	LARG	HSTORV+	
						IFIXR+	

1	2	3	4	5	6	7	8
				LTERM-	LCOMPV+ LARG	HSTORV+ IFIXR+	
			LHOLLS	LHOLL	LHNAME-	LARG	HSTORV+ IFIXR+
					LDELIM-	LARG	HSTORV+ IFIXR+
					LCOMPV+		
			LFINAM+ LCOMPV+ HSTORV+ PLMODE*	VARIB	LHOLL	LHNAME-	LARG
						LDELIM-	LARG
						LCOMPV+	
					LHNAME-	LARG	HSTORV+ IFIXR+
			FILES		HSTORV+ LCOMPV+ LUFIND FENTER+ WRINT+ RDREAL+ RDINT+ FSEEK+ FCLOSE+		
			CRENAM		HSTORV+ LENGTH	LCOMPV+ GAC+	
					PINT	PAC+	
			FILRED		LUFIND RDREAL+		
			EJECT* PLOMRK		MOVECM LINECM PLSYMB+ PLRSET+	PLMOVE+ PLMOVE+	
			EMPTY-		MOVECM LINECM PLSYMB+ PLRSET+	PLMOVE+ PLMOVE+	
		T5011	LTERM-		LARG	HSTORV+ IFIXR+	
			HSTORV+ PREQND		HSTORV+ INFIX	LCOMPV+ NEWNOD PSPACE PHOLL	PAC+ LCOMPV+ GAC+ PAC+
						PFLOAT	LCOMPV+ IMACON+
1	2	3	4	5	6	7	8

1	2	3	4	5	6	7	8
			EMPTY-	MOVECM	PLMOVE+		
				LINECM	PLMOVE+		
				PLSYMB+			
				PLRSET+			
		COSPLT	LTERM-	LARG	HSTORV+		
					IFIXR+		
			LINT-	LARG	HSTORV+		
					IFIXR+		
			PLSET*				
		COAREA	LINT-	LARG	HSTORV+		
					IFIXR+		
			LTERM-	LARG	HSTORV+		
					IFIXR+		
			PLMODE*				
			DEFPAR	AREA			
				PLLIM			
		COTEXT	LTERM-	LARG	HSTORV+		
					IFIXR+		
			IFIXR+				
			PLMODE*				
			GAC+				
			PAC+				
			MOVECM	PLMOVE+			
			PLSYMB+				
		COSTAT+					
		WMESS	IWRITE*				
		USRSUB					
		LPCOM	WRLINE*				
	SIMNSY	HSTORV+					
		EDITOR*					
		COLECT					
	EQORD	NXTINS					
		INLIST					
		FENTER+					
		NODEIN					
		INSTR	ERROR*				
		NODMOV					
		PREORD	NXTINS				
			NODMOV				
			PREQND	HSTORV+			
				INFIX	LCOMPV+		
					NEWNOD		
					PSPACE	PAC+	
					PHOLL	LCOMPV+	
						GAC+	
						PAC+	
					PFLOAT	LCOMPV+	
						IMACON+	
						PAC+	
						PINT	PAC+
						GAC+	
					GAC+		
		PREQND	HSTORV+				
			INFIX	LCOMPV+			
				NEWNOD			

1	2	3	4	5	6	7	8
---	---	---	---	---	---	---	---

1	2	3	4	5	6	7	8
				PSPACE	PAC+		
				PHOLL	LCOMPV+		
					GAC+		
					PAC+		
				PFLOAT	LCOMPV+		
					IMACON+		
					PAC+		
					PINT	PAC+	
					GAC+		
				GAC+			
			FCLOSE+				
		SIMU*					
		EDITOR*					

6. Common blocks referenced in the subroutines

Subr.	Common
ACTHD	: CMPVAR SYSINF VARTB1 VARTB3
ALL	: ALLCOM
ALLER1	: ALLCOM DEVICE
ALLOC	: CMPVAR LITTER SYSINF VARIBL VARTB2 VARTB3
ALLWRK	: ALLCOM
AREA	: PLC025
ASHOW	: AXINF AXPARG DEVICE MARKS MESSS STOVAR
AXES	: AXINF AXPARG DEVICE MARKS PLC025 PLT TESTC TEXPAR
AXIS	: PLC025
CALCUL	: DESTIN ENTRY PSCODE USER VALUES
CHANGE	: COMROW CURLIN FLAGS PNTERS
CHEDEC	: VARTB1 VARTB3
CMPSTA	: CMPVAR LIMITS LITTER PSCODE VARIBL
COALG	: ALG DEVICE MESSS
COAREA	: AXPARG MESSS
COAXES	: AXINF MESSS
CODISP	: MESSS SIMN
COEDIT	: EDFILC MESSS
COERR	: ERRWEI MESSS
COGET	: DEVICE MESSS SIMN
COLECT	: LIMITS MESSS NXPNT PNTS1 PNTS2 PNTS3 SIMN STIFF SYSINF VARTB2 VARTB3
COLIST	: DEVICE MESSS
COMND	: COMROW DEVICE FLAGS PNTERS
COMP	: CMPVAR CURLIN PNTERS PSCODE
COMPAR	: COMROW CURLIN FLAGS PNTERS
CONDEF	: CMPVAR VARTB1 VARTB2 VARTB3
COPLT	: DEVICE LINBUF MESSS PLT SIMN TIME
COSAVE	: MESSS SIMN
COSHOW	: AXINF AXPARG DEVICE MARKS MESSS STOVAR
COSIMU	: AXPARG MESSS PLT SIMARG SIMN STOVAR
COSPLT	: AXPARG MESSS
COSTOR	: MESSS SIMN STOVAR
COSYST	: EQFCOM MESSS PLT SIMN STOVAR SYSINF
COTEXT	: AXPARG LINBUF MESSS PLC025 TEXPAR
COTURN	: DEVICE MESSS SIMN TESTC
CURDIS	: FLAGS
DECL	: CMPVAR SYSINF VARTB1 VARTB2 VARTB3
DEFPAR	: AXPARG TEXPAR
DEFV	: DESTIN EXTCOM LIMITS VARTB1 VARTB2 VARTB3
DEFVAR	: CMPVAR LIMITS VALUES VARTB1 VARTB2 VARTB3
DERIV	: DERS PNTS1 PNTS2 STATES TIME VALUES
DEV	: DEVICE
DIGITS	: CRANK RFP003
DISHDL	: DEVICE DISCOM
DISP	: SYSINF VARTB1 VARTB2 VARTB3
DISPL	: CURLIN DEVICE FLAGS PNTERS
DISP1	: DEVICE MESSS
DRAW	: AXPARG MARKS PLT SIMN TIME USER
EDITOR	: COMROW CURLIN DEVICE FILEN FLAGS PNTERS
EMESS	: DEVICE FILEN FLAGS
EQORD	: CMPVAR DEVICE ENTRYS EQFCOM EQOCOM LIMITS PSCODE SIMN SYSINF VARTB1 VARTB3
ERROR	: BUFFER CMPVAR CURLIN DEVICE PNTERS
ESIMN	: COMINF MESSS SIMN USER
EXTCAL	: DESTIN SYSINF USER

```

EXTSYS: DESTIN  EXTCOM  MESSS  SYSINF  USER
F      : PNTS1   PNTS2   TIME    VALUES
FILDAT: FCTCOM
FILES  : FCTCOM
FILRED: FCTCOM
FINLOC: FLAGS   PNTERS
FORMAC: COMINF  LINBUF  MACINF
FORNXT: COMINF  LINBUF  MACINF
FREE   : COMINF  GLOBAL  LINBUF  MACINF
FR2    : GLOBAL
GET    : GLOBAL
GETV   : DEVICE  SYSINF  VARTB1  VARTB2  VARTB3
HEAD   : CMPVAR
IBDATA: COMINF  DEVICE  GLOBAL  LINBUF  MACINF  SUS001  SWI001
IFGOLA: COMINF  LINBUF  MACINF  SUS001
INFIX  : FXTREE  SYSINF  VALUES  VARTB1
INIVAR: PNTS1   PNTS3   SYSINF  TIME    VARTB2  VARTB3
INLIST: PSCODE
INSTR  : CMPVAR  LIMITS  PSCODE
INTEG  : ALG     DERS    ERRWEI  MESSS  STATES  STIFF  TIME  USER
INTERR: COMINF  DEVICE  LINBUF  MACINF
INTINI: COMINF  MACINF  SWI001
INTR   : COMINF  DEVICE  LINBUF  MACINF  SWI001
INTRAC: COMINF  MACINF
ISIMN  : ALG     AXINF   AXPARG  DESTIN  DEVICE  ERRWEI  EXTCOM  NALLOC
        NSYSTS  SIMARG  SIMN    STOVAR  TESTC  USER
LARG   : COMINF  LINBUF
LCOND  : COND
LET    : COMINF  LINBUF  MACINF
LINECM: PLC025
LITT   : LITTER
LOKFOR: SYSINF  VARTB1  VARTB2  VARTB3
LPCOM  : COMINF  DEVICE  LINBUF  MACINF  SWI001
LUFIND: FCTCOM
MACEND: COMINF  LINBUF  MACINF
MACHDL: COMINF  LINBUF  MACINF  SUS001
MOVECM: PLC025
NARG   : PNTERS
NEWNOD: FXTREE
NODEIN: PSCODE
NODMOV: PSCODE
NXTINS: PSCODE
ONESTR: COMROW  PNTERS
OPEN   : DEVICE
OUTP   : ENTRY  ENTRYS  USER
OUTPRK: USER
PARINT: MESSS  SIMN
PLINIT: PLC025
PLLIM  : PLC025
PLOMRK: AXINF   AXPARG
PREORD: DEVICE  EQOCOM  PSCODE  SYSINF  VARTB1  VARTB3
PREQND: PSCODE  SYSINF
PUT    : GLOBAL
P5011 : DEVICE  SYSINF  VALUES  VARTB1
RCURL  : CURLIN  DEVICE  FLAGS   PNTERS
READ   : COMINF  LINBUF  MACINF
READIN: DEVICE  LINBUF
RECLIN: COMINF  LINBUF
REINAP: COMROW  CURLIN  FLAGS   PNTERS
RELOC  : CMPVAR  PSCODE
RESEX  : COMINF  DEVICE  LINBUF  MACINF  SWI001
RESUME: COMINF  LINBUF  MACINF  SUS001

```

```

RFP      : CRANK   RFP003
RKADM   : INTINF
SAVE    : DEVICE  SYSINF  VARTB1  VARTB2  VARTB3
SEARCH:  CMPVAR  SYSINF  VARTB1  VARTB3
SIMNON:  ALG     ALLCOM  AXINF   AXPARG  CMPVAR  COND   DERS   DESTIN
        EDFILC  ENTRY   ENTRYS  ERRWEI  EXTCOM  LIMITS  MARKS  MESSS
        NALLOC  NSYSTS  NXPNT  PLT    PNTS1  PNTS2  PNTS3  PSCODE
        SIMARG  SIMN    STATES  STIFF  STOVAR  SYSINF  TEXPARG TIME
        USER   VALUES VARTB1  VARTB2  VARTB3
SIMNSY:  CMPVAR  DEVICE  EXTCOM  LIMITS  PSCODE  SIMN   SYSINF
SIMU    : AXPARG  COND   ENTRY   ENTRYS  MARKS  MESSS  NXPNT  PLT
        PNTS1  PNTS2  PNTS3  SIMARG  SIMN   STATES  SYSINF  TIME
        USER   VALUES
STORE   : DEVICE  STOVAR  SYSINF  TIME    USER   VARTB1  VARTB2
SUBST   : COMINF  LINBUF  MACINF
SWITCH:  COMINF  LINBUF  MACINF  SWI001
TWOSTR:  COMROW  PNTERS
TYPE    : CMPVAR  SYSINF  VARTB3
T5011  : CMPVAR  DEVICE  ENTRYS  PNTS1  PNTS2  PNTS3  PSCODE  SYSINF
        VARTB1  VARTB2  VARTB3
UNASS   : BUFFER  CMPVAR  DEVICE
USRSUB:  USRCOM
VARIAB:  VARIBL
WBUFF   : DEVICE
WMESS   : COMINF  DEVICE  LINBUF  MACINF  MESSS
WRITE   : COMINF  DEVICE  LINBUF  MACINF
WRLINE:  DEVICE
WRT     : COMINF  LINBUF
WRT2    : COMINF  GLOBAL  MACINF

```

7. Subroutines referencing the common blocks

Common Subroutines

```

ALG      : COALG  INTEG  ISIMN  SIMNON
ALLCOM: ALL    ALLER1 ALLWRK SIMNON
AXINF   : ASHOW  AXES   COAXES COSHOW ISIMN  PLOMRK SIMNON
AXPAR   : ASHOW  AXES   COAREA COSHOW COSIMU COSPLT COTEXT DEFPAR
        DRAW   ISIMN  PLOMRK SIMNON SIMU
BUFFER: ERROR  UNASS
CMPVAR: ACTHD  ALLOC  CMPSTA COMP   CONDEF DECL   DEFVAR EQORD
        ERROR  HEAD   INSTR  RELOC  SEARCH SIMNON SIMNSY TYPE
        T5011  UNASS
COMINF: ESIMN  FORMAC FORNXT FREE   IBDATA IFGOLA INTERR INTINI
        INTR  INTRAC LARG  LET   LPCOM  MACEND MACHDL READ
        RECLIN RESEX  RESUME SUBST SWITCH WMESS WRITE  WRT
        WRT2
COMROW: CHANGE COMND  COMPAR EDITOR ONESTR REINAP TWOSTR
COND    : LCOND  SIMNON SIMU
CRANK   : DIGITS RFP
CURLIN: CHANGE COMP   COMPAR DISPL  EDITOR ERROR  RCURL  REINAP
DERS    : DERIV  INTEG  SIMNON
DESTIN: CALCUL DEFV   EXTCAL EXTSYS ISIMN  SIMNON
DEVICE: ALLER1 ASHOW  AXES   COALG  COGET  COLIST  COMND  COPLOT
        COSHOW COTURN DEV   DISHDL DISPL  DISPL  EDITOR EMESS
        EQORD  ERROR  GETV   IBDATA INTERR INTR  ISIMN  LPCOM
        OPEN  PREORD P5011 RCURL  READIN RESEX  SAVE  SIMNSY
        STORE  T5011  UNASS  WBUFF  WMESS  WRITE  WRLINE
DISCOM: DISHDL
EDFILC: COEDIT SIMNON
ENTRY  : CALCUL OUTP   SIMNON SIMU
ENTRYS: EQORD  OUTP   SIMNON SIMU  T5011
EQFCOM: COSYST EQORD
EQOCOM: EQORD  PREORD
ERRWEI: COERR  INTEG  ISIMN  SIMNON
EXTCOM: DEFV   EXTSYS ISIMN  SIMNON SIMNSY
FCTCOM: FILDAT FILES  FILRED LUFIND
FILEN  : EDITOR EMESS
FLAGS  : CHANGE COMND  COMPAR CURDIS DISPL  EDITOR EMESS  FINLOC
        RCURL  REINAP
FXTREE: INFIX  NEWNOD
GLOBAL: FREE   FR2   GET   IBDATA PUT   WRT2
INTINF: RKADM
LIMITS: CMPSTA COLECT DEFV   DEFVAR EQORD  INSTR  SIMNON SIMNSY
LINBUF: COPLOT COTEXT FORMAC FORNXT FREE   IBDATA IFGOLA INTERR
        INTR  LARG  LET   LPCOM  MACEND MACHDL READ  READIN
        RECLIN RESEX  RESUME SUBST SWITCH WMESS WRITE  WRT
LITTER: ALLOC  CMPSTA LITT
MACINF: FORMAC FORNXT FREE   IBDATA IFGOLA INTERR INTINI INTR
        INTRAC LET   LPCOM  MACEND MACHDL READ  RESEX  RESUME
        SUBST SWITCH WMESS WRITE  WRT2
MARKS  : ASHOW  AXES   COSHOW DRAW   SIMNON SIMU
MESSS  : ASHOW  COALG  COAREA COAXES CODISP COEDIT COERR  COGET
        COLECT COLIST COPLOT COSAVE COSHOW COSIMU COSPLT COSTOR
        COSYST COTEXT COTURN DISPL  ESIMN  EXTSYS INTEG  PARINT
        SIMNON SIMU  WMESS
NALLOC: ISIMN  SIMNON
NSYSTS: ISIMN  SIMNON
NXPNT  : COLECT SIMNON SIMU
PLC025: AREA  AXES   AXIS   COTEXT LINECM MOVECM PLINIT PLLIM

```

```

PLT      : AXES   COPLOT COSIMU COSYST DRAW   SIMNON SIMU
PNTERS:  CHANGE COMND  COMP   COMPAR DISPL  EDITOR ERROR  FINLOC
        NARG   ONESTR RCURL   REINAP TWOSTR
PNTS1   : COLECT DERIV  F      INIVAR SIMNON SIMU   T5011
PNTS2   : COLECT DERIV  F      SIMNON SIMU   T5011
PNTS3   : COLECT INIVAR SIMNON SIMU   T5011
PSCODE:  CALCUL CMPSTA COMP   EQORD  INLIST INSTR  NODEIN NODMOV
        NXTINS PREORD PREQND RELOC  SIMNON SIMNSY T5011
RFP003:  DIGITS RFP
SIMARG:  COSIMU ISIMN  SIMNON SIMU
SIMN    : CODISP COGET  COLECT COPLOT COSAVE COSIMU COSTOR COSYST
        COTURN DRAW   EQORD  ESIMN  ISIMN  PARINT SIMNON SIMNSY
        SIMU
STATES:  DERIV  INTEG  SIMNON SIMU
STIFF   : COLECT INTEG  SIMNON
STOVAR:  ASHOW  COSHOW COSIMU COSTOR COSYST ISIMN  SIMNON STORE
SUS001:  IBDATA IFGOLA MACHDL RESUME
SWI001:  IBDATA INTINI INTR   LPCOM  RESEX  SWITCH
SYSINF:  ACTHD  ALLOC  COLECT COSYST DECL  DISP  EQORD  EXTCAL
        EXTSYS GETV   INFIX  INIVAR LOKFOR PREORD PREQND P5011
        SAVE  SEARCH SIMNON SIMNSY SIMU  STORE  TYPE  T5011
TESTC   : AXES   COTURN ISIMN
TEXPAR:  AXES   COTEXT DEFPAR SIMNON
TIME    : COPLOT DERIV  DRAW   F      INIVAR INTEG  SIMNON SIMU
        STORE
USER    : CALCUL DRAW   ESIMN  EXTCAL EXTSYS INTEG  ISIMN  OUTP
        OUTPRK SIMNON SIMU  STORE
USRCOM:  USRSUB
VALUES:  CALCUL DEFVAR DERIV  F      INFIX  P5011  SIMNON SIMU
VARIBL:  ALLOC  CMPSTA VARIAB
VARTB1:  ACTHD  CHEDEC CONDEF DECL  DEFV  DEFVAR DISP  EQORD
        GETV   INFIX  LOKFOR PREORD P5011 SAVE  SEARCH SIMNON
        STORE  T5011
VARTB2:  ALLOC  COLECT CONDEF DECL  DEFV  DEFVAR DISP  GETV
        INIVAR LOKFOR SAVE  SIMNON STORE  T5011
VARTB3:  ACTHD  ALLOC  CHEDEC COLECT CONDEF DECL  DEFV  DEFVAR
        DISP  EQORD  GETV   INIVAR LOKFOR PREORD SAVE  SEARCH
        SIMNON TYPE  T5011

```