27
7-15-81
240 TIS
SAND80-7154/2
Unlimited Release
UC-66

B5685①

Programmer's Manual for IOSYM: An Input-Oriented Simulation Language for Continuous Systems

MASTER

Volume 2: Subprogram Description

David Michael Smith
Scientific Simulation Corp.

Prepared by Sandia National Laboratories, Albuquerque, New Mexico 87185
and Livermore, California 94550 for the United States Department of Energy
under Contract DE-AC04-76DP00789

Printed June 1981

Prepared for Sandia National Laboratories under Contract No. 077252

Sandia National Laboratories

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

SAND80-7154/2
Unlimited Release
Printed

PROGRAMMERS' MANUAL FOR IOSYM:
AN INPUT-ORIENTED SIMULATION LANGUAGE FOR CONTINUOUS SYSTEMS
VOLUME II

SUBPROGRAM DESCRIPTION

David Michael Smith
Senior Systems Analyst
Scientific Simulation Corporation
P.O. Box 9331
Albuquerque, New Mexico 87119

Formerly Of
The BDM Corporation
1801 Randolph, SE
Albuquerque, New Mexico 87106

ABSTRACT

IOSYM is an extension of the GASP IV simulation language. It permits systems which are sequences of continuous processes to be modeled graphically. Normally the system can be described by data input only. The language permits stochastic sequencing and termination criteria for processes and allows crossing conditions for ending operations that are more general than GASP IV. Extensive capability exists for conditional branching and logical modification of the network. IOSYM has been used to model the cost of geothermal drilling where the various costly processes of drilling are represented by IOSYM operations. The language is much more general however; it retains more of GASP IV's discrete event capabilities and permits easy modeling of continuous processes.

(BDM/TAC-79-673-TR)

DISCLAIMER

This book was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

ACKNOWLEDGEMENT

The author is indebted to Joseph Polito of Pritsker and Associates formerly of Sandia National Laboratories. His careful editing and aid in organizing and preparing figures for this document have substantially improved its final form.

In addition, Dr. Polito shared in the programming and program design tasks. Without his expertise and design, this work would never have begun.

TABLE OF CONTENTS

VOLUME I

		<u>Page</u>
I	INTRODUCTION	1
II	STRUCTURE OF IOSYM	3
III	DATA STRUCTURE AND PROGRAMMING CONSIDERATION	11
IV	DATA DESCRIPTION	62

VOLUME II

V.	SUBPROGRAM DESCRIPTIONS	
----	-------------------------	--

I. INTRODUCTION (From Volume I)

This programmers' manual is intended as an aid to the IOSYM program listing, IOSYM users' manual¹, the GASP-IV Simulation Language², and the Sandia Mathematical Program Library Manual³ (for line printer plot description of XPLOT, XPMNMX, and XPSYMB). This manual is for use by an individual knowledgeable with FORTRAN and the IOSYM users' manual. When used in conjunction with the computer program listing, it will allow IOSYM to be supported, modified, or extended. Due to the complexity of the program, these functions should not be attempted without first obtaining a detailed understanding of the program structure, data flow, and functional flow of IOSYM. The program is well documented with comments indicating the operations being performed.

IOSYM is an extension of the GASP-IV simulation language. It is general enough to be used as a tool for modeling systems that are characterized by discrete and/or continuous processes. IOSYM allows the model to be defined through free form data input rather than user-written programs. The user is not normally required to do any programming. Advanced users who are familiar with GASP will find that virtually all of GASP's capabilities are also available.

For a thorough understanding of how a simulation is performed by IOSYM, an understanding of the GASP-IV language is required. The GASP-IV text is a source of information on the structure, use, and description of GASP-IV.

1. Users' Manual for IOSYM: An Input-Oriented Simulation Language for Continuous Systems, Joseph Polito, SAND80-2000, 1980.

2. The GASP-IV Simulation Language, A. Alan B. Pritsker, John Wiley and Sons, Inc. 1974.

3. Sandia Mathematical Program Library Manual, Sandia Laboratories, SAND75-0545, 1975.

V. SUBPROGRAM DESCRIPTIONS

Each subprogram is described in this section. In part 5 of each entry, "F.P." indicates formal parameters of the program. Labeled common areas are denoted by the name of the block between slashes, e.g. /I02/. Blank common is denoted by two slashes, e.g. //.

1. NAME: AAIROP - 55
2. PURPOSE: AAIROP takes one or two operands and an operator or library function and evaluates the expressions.
3. METHOD: Depending on what operator or library function is being used, processing skips to the appropriate part of the routine and the expression is evaluated.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
A	First operand in expression, such as 6 + 2. "A" would be 6. The value of the expression is stored and returned in variable A.
B	Second operand in expression, such as, 6 + 2. B would be 2.
I	Result of IFIX.
IERR1	Formal parameter returned from routine EEROR, but not used in this routine.
NOP	This is a number corresponding to the operator or library function. 1 = exponentiation 2 = multiplication 3 = division 4 = addition 5 = subtraction 6 = equal 7 = arc tangent (ATAN) 8 = complex (not operational) 9 = pick maximum (AMAX1) 10 = pick minimum (AMIN1) 11 = modulus (AMOD) 12 = SIGN 13 = absolute value (ABS) 14 = arc tangent (ATAN2) 15 = cosine (COS) 16 = e to the power (EXP) 17 = log (ALOG) 18 = sine (SIN) 19 = square root (SQRT) 20 = tangent (TAN) 21 = make integer (IFIX)

PPREC

Precision, Used to check for
zero values in division, log and
square root.

5. I/O VARIABLES:

A.	INPUT	LOCATION
	A	F.P.
	B	F.P.
	NOP	F.P.
	PPREC	/I04/
B.	OUTPUT	LOCATION
	A	F.P.

6. CALLING ROUTINE: EEVAL

7. CALLED ROUTINES:

EEROR

EEXP

1. NAME: AARCST - 539
2. PURPOSE: This routine does the necessary set up before processing an ARC during an IOSYM simulation.
3. METHOD: The IOSYM TABLE AV is cleared by setting it to all zeros. The pointer to the current ARC number is set. The next NODE is determined from the ARC uniform command sequence (UCS) and stored in the next NODE pointer. The OPERATION number and any substitutable parameters are identified. If a TRACE is called for, routine TTRACE is called to process TRACE information. The statistics are set up and the TYPE S commands are done. The TERM conditions are evaluated and filed in the GASP event file. The CROSS conditions are identified. If there is no OPERATION on the ARC, a Termination is scheduled immediately. The TYPE I commands are set up and then control returns to the calling routine.
4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
ATRI B	Array that contains the attribute values being stored into the GASP file. See GASP-IV manual.
DD	The value of the derivative of the corresponding primary variable at the time TNOW.
DDL	The value of the derivative of the corresponding primary variable at the time TTLAS.
I	Pointer, counter, and index used for various purposes.
ICM	Command number for TERM (39) or CROSS (44).
IERR1	Error indicator returned from routine EEROR but not used in this routine.
IIARC	The IOSYM ARC number of the ARC currently being processed.
IINXAC	The IOSYM ARC number of the next ARC to start processing.
IINXND	The IOSYM NODE number of the NODE that will normally be processed next.

IIOP	IOSYM OPERATION number of the OPERATION to process on current ARC.
IIOPR	Array containing the variable identifiers of the substitutable parameters.
L	Pointer to location in the NSET array where the substitutable parameter being processed is located.
LLCROS	Location in the NNWRK array that points to the UCS for the first CROSS command on the current ARC.
LLTRAC	Pointer to the location in the NSET array where the TRACE information is located. If LLTRAC is less than or equal to zero, there is no TRACE command to process.
LLUCSL	Array that contains the lengths of all the standard command UCS's.
LLWRK	Next available location in the NNWRK array to be used.
LSTR	Location in the TTABL array where the storage for the values of the TABLE AV start.
LUCS	Location in the NSET array where the UCS for the current ARC starts.
LUCSOP	Location in the NSET array where the UCS for the current OPERATION starts.
LUCST	Location in the NSET array where the UCS for the first TERM for the current ARC starts.
MMV1	Maximum number of primary variables a NETWORK can have (set in routine DFAUT, also the dimension of the SSIN, DDIN, NNPT1, DD, DDL, SS, and SSL arrays).
MMWRK	Maximum number of locations available in the work storage array (dimension of the NNWRK array set in routine USRDAT).

NDUM	Number of substitutable parameters on the current OPERATION.
NNCROS	Number of CROSS commands on the current ARC.
NNDUM	Number of substitutable parameters that occur in the GLOBAL and the OPERATION for the current ARC.
NNEQD	Number of DD equations (equal to number of primary variables).
NNEQS	Number of state equations which is equal to NNCROS.
NNEQT	$NNCROS = NNEQD + NNEQS$.
NNGDUM	Number of substitutable parameters on the GLOBAL command.
NNPTAC	Array that contains in the first two words the ARC name, and the third word points to the location in the NSET array where the ARC UCS starts.
NNPTB	Array that contains in the first two words the TABLE name, and the third word points to the location in the TTABL array where the TABLE element storage for that TABLE starts.
NNWRK	Work storage array.
NSET	Storage array that contains the link listed NETWORK.
NTERM	Number of TERMS and CROSSes on the current ARC.
SS	Value of the state variables at the time TNOW.
SSL	Value of the state variables at the time TTLAS.
TNOW	Current time of the simulation.

TTABL

Array that contains all of the TABLE
element values.

VVAR2

Array that contains all of the secondary
variable values.

5. I/O VARIABLES:

A. INPUT	LOCATION
DD	/GCOM2/
DDL	/GCOM2/
IERR1	F.P.
IINXAC	/I03/
IIOPR	/I02/
LLTRAC	/I03/
LLUCSL	/I02/
LLWRK	/I03/
MMV1	/I04/
MMWRK	/I04/
NNEQD	/GCOM2/
NNEQS	/GCOM2/
NNEQT	/GCOM2/
NNGDUM	/I02/
NNPTAC	/I02/
NNPTB	/I02/
NSET	/ /
SS	/GCOM2/
SSL	/GCOM2/

TNOW	/GCOM1/
TTABL	/102/
B. OUTPUT	LOCATION
ATRI	/GCOM1/
IIARC	/103/
IINXAC	/103/
IINXND	/103/
IIOP	/103/
IIOPR	/102/
LLCROS	/103/
LLWRK	/103/
NNCROS	/103/
NNDUM	/102/
NNEQS	/GCOM2/
NNEQT	/GCOM2/
NNWRK	/103/
VVAR2	/102/

6. CALLING ROUTINES:

EVNTS
INTLC

7. CALLED ROUTINES:

BBLIST
CCOLC
DDOLST
EEROR
EEVAL

FILEM (GASP)
IIMOV
STATE
TTABU
TTRACE
USTAT (USER)
VVMOV
ZZSET

1. NAME: ADATA - 39
2. PURPOSE: This routine sets the values of some internal variables that the usual user will not need to modify. The variables the user can modify are in routine USRDAT.

3. METHOD: Not applicable.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
DUMP	Indicator. If TRUE, then a printout of all of the IOSYM arrays is to occur. If FALSE, then a DUMP does not occur.
ECHO	Indicator. If TRUE, then the input cards processed are to be printed out on the output file. If FALSE, then the input cards are not ECHOed back.
IITPS	Contains the device code for the different files needed. IITPS(1) is where the SAVE information is located. IITPS(2) is where the GASP information is stored. IITPS(3) is where the IOSYM PLOT information is stored. IITPS(4) is where the IOSYM TABULATE information is stored. IITPS(5) is a scratch file.
LLUCSL	Contains the lengths of all the command uniform command sequences that are always a given length.

LLUCSL(1) = ACTIVATE
 7 = DEACTIVATE
 16 = GOTO
 21 = NODE
 29 = PICK
 35 = SET
 36 = SWITCH
 39 = TERM
 43 = REPLACE
 44 = CROSS
 46 = LOOP
 47 = LABEL
 48 = IF
 53 = JUMP

54 = EXIT
 55 = PRINT
 56 = PUT
 57 = GET
 58 = STAT

MMFUNF	Number of IOSYM menu functions.
MMFUNP	Number of IOSYM probability functions.
NNMNET	Contains the NETWORK name.
NNPTGB	First two words contain the word GLOBAL in the IOSYM notation.
NOINFO	Indicator that tells whether or not informative diagnostics (warnings) are to be printed. If FALSE, the warnings are to be printed. If TRUE, they are not.
NTALP	Alpha field indicator.
NTALPC	Alpha field indicator.
NTDLM	Delimiter field indicator.
NTERR	Error field indicator.
NTFLP	Floating point number field indicator.
NTFUNF	IOSYM menu function field indicator.
NTFUNP	IOSYM probability function field indicator.
NTFUNU	IOSYM user function field indicator.
NTINT	Integer number field indicator.
NTNUL	Null field indicator.
NTOPR	Operator field indicator.
NTTBL	Table field indicator.
NTVARD	Derivative variable field indicator.
NTVARO	Independent variable field indicator.

NTVAR1	Primary variable field indicator.
NTVAR2	Secondary variable field indicator.

5. I/O VARIABLES:

A. INPUT	LOCATION
----------	----------

(None)

B. OUTPUT	LOCATION
-----------	----------

DUMP	/ME1/
------	-------

ECHO	/ME1/
------	-------

IITPS	/I01/
-------	-------

LLUCSL	/I02/
--------	-------

MMFUNF	/I04/
--------	-------

MMFUNP	/I04/
--------	-------

NNMNET	/I03/
--------	-------

NNPTGB	/I02/
--------	-------

NOINFO	/ME1/
--------	-------

NTALP	F. P.
-------	-------

NTALPC	F. P.
--------	-------

NTDLM	F. P.
-------	-------

NTERR	F. P.
-------	-------

NTFLP	F. P.
-------	-------

NTFUNF	F. P.
--------	-------

NTFUNP	F. P.
--------	-------

NTFUNU	F. P.
--------	-------

NTINT	F. P.
-------	-------

NTNUL	F. P.
-------	-------

NTOPR	F.P.	000000
NTTBL	F.P.	000000
NTVARD	F.P.	000000
NTVAR0	F.P.	000000
NTVAR1	F.P.	000000
NTVAR2	F.P.	000000

6. CALLING ROUTINE: CHOICE
7. CALLED ROUTINE: IISSET

1. NAME: ALPFND - 12
2. PURPOSE: This routine looks for an alphanumeric string among a list of literals. If it finds an integer number instead, it calls a routine to convert this number to an alphanumeric string. Once an alphanumeric string has been found (or an integer number converted), this subroutine returns the location of that string.
3. METHOD: It starts searching the literal storage array (LITSTR) at the first available string as given by NDXLIT. If a delimiter is found, a skip to the next field is done, and this field is examined. If a zero is found for string type, it assumes that that location in LITSTR has already been used, and a skip to the next field is done. If an integer number is found, the routine CONVRT2 is called to convert the number to an alphanumeric string and store the resultant string where the number resided. Once an alphanumeric string has been identified, NDXLIT points to the location in LITSTR where that alphanumeric string is located and control returns to the calling routine. If none of the above fields is found, the variable NOADD is set TRUE to indicate that an alphanumeric string was not found, and control returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
IERR1	Error indicator. If TRUE, then fatal error has occurred. If FALSE, no fatal error has occurred up to the current time.
LITMAX	Maximum length of literal storage array LITSTR. Used as variable dimension of LITSTR.
LITNDX	Position of the last defined location in the literal storage array LITSTR.
LITSTR	Array where the literals are stored after being scanned and prior to being parsed.
NDXLIT	Pointer to current position in the literal storage array LITSTR.
NOADD	Comes into routine set FALSE. If upon returning it is still FALSE, then an alphanumeric string was found. If upon returning it is set TRUE, then an alphanumeric string was not found.

NTALPC

Alphanumeric field indicator.

NTDLM

Delimiter field indicator.

NTINT

Integer field indicator.

5. I/O VARIABLES:

A. INPUT

LOCATION

LITMAX

F.P.

LITNDX

F.P.

LITSTR

F.P.

NDXLIT

F.P.

NTALPC

F.P.

NTDLM

F.P.

NTINT

F.P.

B. OUTPUT

LOCATION

IERR1

F.P.

LITSTR

F.P.

NDXLIT

F.P.

NOADD

F.P.

6. CALLING ROUTINE: ALPSRH

7. CALLED ROUTINES:

CONVR2

EEROR

1. NAME: ALPSRH - 96
2. PURPOSE: This routine processes an alpha character in the command syntax.
3. METHOD: Routine ALPFND is called to identify the alpha field, and depending on what command is being processed, different things happen.

If the current command is ARC, then the alpha field wanted is the ARC status. The current status is identified and stored in the work array.

If the current command is GOTO, then the alpha field needed is an A or N to indicate whether an ARC or NODE is processed. The correct one is identified and stored in the work array.

If the current command is TYPE, then the alpha field needed is an S, I, or E to indicate which TYPE. The TYPE is determined and is stored in the work array.

If the current command is SET, then the alpha field needed is a truth condition. The truth condition is identified and is stored in the work array.

If the current command is TABULATE, then the alpha field needed could be two items depending on what part of the TABULATE is being processed. If working at the beginning of the TABULATE, then the time option is needed. It is identified and stored in the work array. If working further along in the TABULATE, then need an alpha to tell whether the variable is to be TABULATED at the END, START, or CHANGE. The current type is identified and stored in the work array.

If the current command is a TERM, CROSS, or PICK command, then an alpha that tells which truth condition is needed is identified and stored in the work array.

If the current command is a SAVE or RESTORE command, then an alpha to determine if a NETWORK summary is to be done is needed and check to see if there is a NETWORK name. These fields are identified and stored in the NNMNET array.

If the current command is an IF command, an alpha is needed to tell which truth condition to use. The correct one is identified and stored in the work array.

If the current command is a NETWORK command, then the NETWORK name needs to be processed. The name is identified and stored in the NNMNET array.

If the current command is CHANGE, then an alpha is needed to identify whether or not a NETWORK summary is to be done. Once identified, it is stored in the NNMNET array.

After processing one of the above commands, control returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
IERR	Error indicator. If FALSE, then an error hasn't occurred in this routine. If TRUE, then a fatal error has occurred in this routine.
ISSET	Indicator to tell which alpha character was found.
LITMAX	Maximum number of fields in the literal storage array (dimension of the LITSTR array).
LITNDX	Number of defined fields in the literal storage array.
LITSTR	Literal storage array where the first word contains the type of field and the last two words are the field values.
LLXWRK	Number of locations in current use in the work storage array. Also, the last position used in that array.
MMWRK	Maximum number of locations in the work storage array (dimension of the MMWRK array, set in routine USRDAT).
NDXLIT	Pointer to the current location in the literal storage array under investigation.
NDX1	Current command number.
NNMNET	Array containing in the first two words the IOSYM representation of the NETWORK name. The third word is an indicator of whether or not a NETWORK summary is to be printed.

NNWRK	Work storage array. Contains the current command parsed into the various elements.
NOADD	Indicator. If FALSE, then an alpha field was found. If TRUE, an alpha field wasn't found.
NTALPC	Alpha field indicator.
NTDLM	Delimiter field indicator.
NTINT	Integer number field indicator.
NTNUL	Null field indicator.

5. I/O VARIABLES:

A.	INPUT	LOCATION
	IERR	F.P.
	LITMAX	F.P.
	LITNDX	F.P.
	LITSTR	F.P.
	LLXWRK	/I03/
	MMWRK	/I04/
	NDXLIT	F.P.
	NDX1	F.P.
	NNMNET	/I03/
	NOADD	F.P.
	NTALPC	F.P.
	NTDLM	F.P.
	NTINT	F.P.
	NTNUL	F.P.

B.	OUTPUT	LOCATION
	IERR	F.P.
	NOXLIT	F.P.
	NNMNET	/I03/
	NNWRK	/I03/
	NOADD	F.P.

6. CALLING ROUTINE: PARSE

7. CALLED ROUTINES: ALPFND
 ARAYCK
 EEROR
 TRUTH

1. NAME: ANET
2. PURPOSE: When using the CDC segmented loader, a subroutine cannot appear in more than one segment unless it is specified on the right hand side of an INCLUDE card. Subroutine NETSUM is used in two places and is the main routine in one of the segments. Routine ANET is just a call to NETSUM to satisfy the first line of this description.
3. METHOD: Not applicable.
4. INTERNAL VARIABLES: (None).
5. I/O VARIABLES: (None).
6. CALLING ROUTINE: CHOICE
7. CALLED ROUTINE: NETSUM

1. NAME: ARANGE - 82
2. PURPOSE: This routine, given the information on the plot file, will calculate "nice" X and Y minimum and maximum values that include all the plot points. If the number of points to plot is greater than 1000, then it will make the X axis length so that the first plot of the data and each subsequent plot will have a maximum of 1000 points per plot.
3. METHOD: The plot file and the minimum and maximum values for the X and Y axis are found. If there are more than 1000 points, the X-axis is scaled so that no plot has more than 1000 points. For the Y-axis, the number of tic marks is set to 5 for the X axis set to 10. The interval between tick marks is calculated. The power of ten between ticks is calculated, and the mantissa is set greater than the mantissa of the largest data point. The smoothed mantissa's are 1, 2, 3, 4, 5, 6, 7, 8, 9, 10. Once the smoothed mantissa is found, it is multiplied by power of ten corresponding to the data, and the minimum and maximum smoothed values are calculated from this.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
A	An array where the Y values are stored temporarily after being read off the plot file.
AIN	The unsmoothed interval between tick marks.
AINL	A real variable that contains the power of ten that equals AIN.
AMAX	Temporary storage for the plot limit maximum value.
ANTV	Number of tick marks per axis.
FAC	Set to 1 initially and used only if there are more than 1000 points to plot. If there are, then FAC is the ratio of plot points to 1000, to show the amount of a full plot the remaining points represent.
I	Used as index in DO loop and in one case indicates what Y value is currently being processed. In another case, it indicates doing X axis (I=2) or Y axis (I=4).

IERR	Error indicator. If FALSE, then an error hasn't occurred in this routine. If TRUE, then a fatal error has occurred in this routine.
INUM	Is the plot number read off the plot file.
IPP	Current plot number being processed.
IRE	Is the number of plot points remaining after all multiples of 1000 have been subtracted.
IST	Indicator to show whether at stopping point for reading plot file. If equal to zero, then not at stopping point. If equal to one, then at end of plot tape.
ITP	Indicates the tape number of the file that contains the plot information.
J	Indicates the mantissa of the smoothed interval, the value of any integer from 1 to 10 inclusive.
JK	Indicates the number of plot points read from plot file for current plot number.
K	Index used to indicate the Y value currently being read.
LOGINT	Integer power of ten that the interval between tick marks is.
MAXEXP	Maximum exponent that a number can have on this computer (machine dependent).
MINLIM	Used to calculate the minimum axis limit.
NV	Indicates the number of dependent variables to be read off the plot file.
PLTLIM	Used to calculate the minimum axis limit.
PPREC	Indicates the computer precision (machine dependent).

IERR	Error indicator. If FALSE, then an error hasn't occurred in this routine. If TRUE, then a fatal error has occurred in this routine.
INUM	Is the plot number read off the plot file.
IPP	Current plot number being processed.
IRE	Is the number of plot points remaining after all multiples of 1000 have been subtracted.
IST	Indicator to show whether at stopping point for reading plot file. If equal to zero, then not at stopping point. If equal to one, then at end of plot tape.
ITP	Indicates the tape number of the file that contains the plot information.
J	Indicates the mantissa of the smoothed interval, the value of any integer from 1 to 10 inclusive.
JK	Indicates the number of plot points read from plot file for current plot number.
K	Index used to indicate the Y value currently being read.
LOGINT	Integer power of ten that the interval between tick marks is.
MAXEXP	Maximum exponent that a number can have on this computer (machine dependent).
MINLIM	Used to calculate the minimum axis limit.
NV	Indicates the number of dependent variables to be read off the plot file.
PLTLIM	Used to calculate the minimum axis limit.
PPREC	Indicates the computer precision (machine dependent).

RANGE	An array that contains (1) not used (2) X minimum (3) X maximum (4) Y minimum (5) Y maximum
SCLINT	Unsmoothed mantissa of the interval between tick marks.
T	Value of maximum X value encountered in last 1000 points.
TICINT	Length of smoothed interval.
T1	Value of independent variable read from plot file.
T2	Final value of independent variable read from plot file.
X1	When processing more than 1000 points, X1 is for a given 1000 points. The difference between the value of the first independent variable point and the value of the 1000th point.
X2	When processing more than 1000 points, X2 is the first independent variable value.
X3	Value of first independent variable read.

5. I/O VARIABLES:

A. INPUT	LOCATION
IPP	F.P.
ITP	F.P.
MAXEXP	/MCHDEP/
PPREC	/IO4/
B. OUTPUT	LOCATION
RANGE	F.P.

6. CALLING ROUTINE: PPRPLT

7. CALLED ROUTINE: (NONE)

1. NAME: ARAYCK - 14
2. PURPOSE: This routine increments array indices and checks to make sure array dimensions have not been exceeded.
3. METHOD: Given the maximum size of an array and the pointer to the current position, the pointer is incremented and checked against the maximum limit. If the dimensions have been exceeded, then an error message is printed.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
IERR1	Error indicator. If TRUE, then fatal error has occurred. If FALSE, no fatal error has occurred up to the current time.
IIER	Each array has a specific error number associated with it to use if the bounds of the array are exceeded. This variable contains that value.
ISUB	Contains the routine number of the routine that called ARAYCK. It is used when processing error.
MXNDX	This is the maximum length that the given array can have.
NDX	On entering the routine, this variable is the value of the current pointer. On leaving the routine, this variable contains the value of the incremented pointer.

5. I/O VARIABLES:

A. INPUT	LOCATION
IIER	F.P.
ISUM	F.P.
MXNDX	F.P.
NDX	F.P.

B. OUTPUT

LOCATION

IERR1

F.P.

NDX

F.P.

6. CALLING ROUTINES:

ALPSRH	FUNC4	NAMFND	SCAN
ARCSTR	FUNC4A	NODSTR	SETSTR
BLDALP	FUNC5	NUMFND	SRHARC
BLDDLML	FUNSTR	NUMSTR	STASTR
BLDOPR	GBLSTR	OPRSTR	TBLFND
COLSTR	GETSTR	PARSE	TBTSTR
DVFND	GTOSTR	PCKCK	TOFSTR
DVSTR	IFSTR	PCKSTR	TRCSTR
ENDNET	JOESTR	PLTSTR	TRMSTR
EVTSTR	LBLSTR	PRTSTR	TYPSTR
FNFND	LITFND	PUTSTR	VARFND
FUNC2	LOLFND	REPSTR	WTHSTR
FUNC3	LOPSTR	RPTSTR	ZERO

7. CALLED ROUTINE: EEROR

1. NAME: ARCSTR - 28
2. PURPOSE: This routine processes the input of the ARC command. It stores the start NODE, end NODE, initial and current ARC status. Sets up the locations for storing the pointer to the Uniform Command Sequence (UCS) for the OPERATION and the pointer to the location of the number of TERMinations for the ARC.
3. METHOD: First, a check is done to make sure there is a start and an end NODE. If there is not one of each, an error is processed. If there is, then the start NODE number is stored in the UCS. A check is done to make sure the ARC does not already exist. The PICK chain is increased so the PICK commands can be located. A pointer to the end NODE is also saved in the UCS. The next location in the UCS is set aside as a pointer to the UCS for the OPR. The next location in the UCS is set aside for the number of TERMS and CROSS commands on this ARC. The current and initial ARC status is stored in the next two locations in the UCS. If none is specified on the ARC command, a default of active is assumed. Processing for the ARC command is then finished.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
FIR2	Indicator to show whether an ARC leaving the given NODE has already taken place. If FIR2(I) is TRUE, then an ARC leaving NODE I has not been processed. If FIR2(I) is FALSE, then an ARC leaving NODE I has been processed.
ID	Dimension of the FIR2 array.
IER	Error number indicator.
IERR	Error indicator. If FALSE, then no error has occurred in this routine. If TRUE, then a fatal error has occurred in this routine.
IERR1	Error indicator. If FALSE, then no error has occurred up to the present time. If TRUE, then a fatal error has occurred.
IFIR	Indicator to show that we have started processing a new ARC.
INDX4	Pointer in the ARC UCS that points to the location in the NSET array where the UCS for the OPR command for this ARC is located.

INDX5	Pointer in the ARC UCS that points to the location in the NSET array where the number of TERMS and CROSSes for this ARC is located.
IN1	Start NODE number as defined by IOSYM.
IN2	ARC number as defined by IOSYM.
IN3	Location in the NSET array where start NODE begins.
IPTCMD	Pointer to the next command. If IPTCMD is zero, then no command UCS points to the current command.
ISNODE	Current start NODE.
LLXSET	Counter of the number of locations in current use in the NSET array. Also, last position in the NSET array used.
LSET	Temporary storage for the number of NSET elements needed.
MMSET	Maximum dimension of the NSET array.
NNPTAC	Contains the ARC name and location in the NSET array where the ARC UCS starts.
NNPTND	Contains the NODE name and location in the NSET array where the NODE UCS starts.
NNWRK	Work storage array. Contains the current command parsed into the various elements.
NSET	Storage array where the link listed NETWORK is stored.
NTNUL	Null field indicator.
OPR	Indicator. If FALSE, then an OPR command for this ARC has not been processed. If TRUE, then OPR command has been processed for this ARC.

PICK

Indicator. If FALSE, then PICK command for this ARC has not been processed. If TRUE, then a PICK command has been processed for this ARC.

TERM

Indicator. If FALSE, then a TERM or CROSS command for this ARC has not been processed. If TRUE, then TERM or CROSS command has been processed for this ARC.

5. I/O VARIABLES:

A. INPUT	LOCATION
FIR2	F.P.
ID	F.P.
LLXSET	/I03/
MMSET	/I04/
NNPTAC	/I02/
NNPTND	/I02/
NNWRK	/I03/
NSET	/ /
NTNUL	F.P.
B. OUTPUT	LOCATION
FIR2	F.P.
IERR1	F.P.
IFIR	F.P.
INDX4	F.P.
INDX5	F.P.
IPTCMD	F.P.
ISNODE	F.P.
LLXSET	/I03/
NNPTAC	/I02/

NNWRK

/I03/

NSET

/ /

OPR

F.P.

PICK

F.P.

TERM

F.P.

6. CALLING ROUTINE: PROCES

7. CALLED ROUTINE:

ARRAYCK

EEROR

1. NAME: AXJ
2. PURPOSE: This routine computes $B(I) = \sum \text{over } J \text{ of } A(I,J) * X(J)$ where A is a matrix and B and X are arrays.
3. METHOD: The A(I,J) matrix is multiplied by array X(J) for all the values of J and stored in the B(I) array. This is done for each I. Control then returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
A	Matrix to be multiplied.
B	Array that is returned with the calculated values.
I	Index for pointing to the array column.
J	Index for pointing to the array rows.
MI	the dimension of the B array.
N	Second dimension of the A matrix and
NI	the dimension of the X array.
X	Array to multiply the A matrix by.
Z	Temporary storage for the calculation of A(I,J) times X(J).

5. I/O VARIABLES:

A. INPUT	LOCATION
A	F.P.
M	F.P.
N	F.P.
X	F.P.
B. OUTPUT	LOCATION
B	F.P.

6. CALLING ROUTINE: (not used at this time)
7. CALLED ROUTINE: (none)

1. NAME: BBLIST - 521
2. PURPOSE: This routine will build a command list to be executed on the current ARC. Only commands of the type IASTYP are inserted in the list. The list is built in the NNWRK array beginning at the location LLIST and consists of pointers to the uniform command sequence (UCS) of the commands.
3. METHOD: The current command TYPE is identified. The list is built first with the commands on GLOBAL, then OPERATION, and finally ARC. Each command is identified and its location is stored in the NNWRK array and the pointer to the next command is identified and the process repeated until all the GLOBAL commands have been processed. All the OPERATION commands are processed. And finally all the ARC commands are processed. Control then returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
IASTYP	Current kind of TYPE commands to be processed. 1 = TYPE S 2 = TYPE I 3 = TYPE E
IATP	Current kind of TYPE commands to be processed. 1 = TYPE S 2 = TYPE I 3 = TYPE E
ICM	Command number of the command currently being processed.
IERR1	Error indicator returned from routine EERQR but not used in this routine.
IIARC	ARC Number of the current ARC.
IIOP	OPERATION number of the current OPERATION.
IKIND	Indicates what particular kind of commands are being processed. 1 = GLOBAL 2 = OPERATION 3 = ARC

ITEM	Pointer to the NSET array used for different purposes.
LIST	Location in the NNWRK array where the next pointer to a command is to be stored.
LLIST	Location in the NNWRK array where the list of pointers to the various commands starts.
LLUCSL	Array containing the length of the UCS for the various commands.
LLWRK	Next location available for use in the NNWRK array.
LOPR	Pointer to the location in the NSET array where the OPERATION UCS is located.
LUCS	Pointer to the location in the NSET array where the current command starts.
MMWRK	Maximum number of locations available in the NNWRK array (dimension of the NNWRK array, set in routine USRDAT).
NNPTAC	Array that contains the ARC name in the first two words and the pointer to the ARC USC in the third word.
NNPTGB	Array that contains the name GLOBAL in the first two words and the third, fourth, and fifth words point to the commands on the TYPE S, TYPE I, and TYPE E commands, respectively.
NNPTOP	Array that contains in the first two words the OPERATION name, the third, fourth, and fifth words point to the commands on the TYPE S, TYPE I, and TYPE E commands, respectively. The sixth word indicates whether or not the OPERATION is a COLLECT OPERATION.
NNWRK	Work storage array.
NSET	Storage array containing the link listed NETWORK.

5. I/O VARIABLES:

A. INPUT	LOCATION
IASTYP	F.P.
IERR1	F.P.
IIOP	/I03/
LLIST	/I03/
LLUCSL	/I02/
LLWRK	/I03/
MMWRK	/I04/
NNPTAC	/I02/
NNPTGB	/I02/
NNPTOP	/I02/
NSET	/ /
B. OUTPUT	LOCATION
LLIST	/I03/
LLWRK	/I03/
NNWRK	/I03/

6. CALLING ROUTINES: AARCST
EVNTS

7. CALLED ROUTINES: EEROR
RREWRK

1. NAME: BLDALP - 17
2. PURPOSE: Builds alphanumeric fields from data cards.
3. METHOD: Given the current position on the data card that contains an alpha character, this routine builds an alphanumeric word of up to 7 characters. If more than seven characters are encountered, a warning message is printed if the alphanumeric string is not the first field on the card. If the first field on the card is longer than 7 characters, it is assumed that the card command was spelled out in which case an error is not justified. If a blank or a delimiter is encountered, the scan for the alphanumeric field stops and control returns to the scanning routine. The alphanumeric word is built four characters at a time. The first four characters go into the first computer word and the last three characters in the second word. The words are built by taking the current contents of the word and multiplying it by 64 and adding the IOSYM character number for the current character. The next character is done in similar fashion. The two words are blank filled right justified four characters.

Example: DRILLIN

IOSYM character number

D = 4

R = 18

I = 9

L = 12

L = 12

I = 9

N = 14

blank = 47

The first computer word contains $((4 \times 64 + 18) \times 64 + 9) \times 64 + 12 = 1122892$

The second computer word contains $((12 \times 64 + 9) \times 64 + 14) \times 64 + 47 = 3183535$

4. INTERNAL VARIABLES:

VARIABLE

DEFINITION

IALPHA

Contains the integer representation of 4 characters of an alphanumeric string.

IEER

Error indicator. If FALSE, then no error has occurred in this routine. If TRUE, then a fatal error has occurred.

IERR1	Error indicator. If FALSE, then no error has occurred up to the present time. If TRUE, then a fatal error has occurred.
IMXBYT	Maximum number of characters that can be stored in each computer word. This is set to 4.
I2ND	Indicator. If FALSE, then processing first four characters of alphanumeric string. If TRUE, then processing second set of four characters.
K	Index used to tell which character currently processing.
KARDD	Is the array containing the characters of the data card currently being worked on.
KHAR	Array containing all of the IOSYM characters A - Z, and 0 - 9 and . (period).
KNTCHR	Number of characters in alphanumeric string currently processing.
KRD	Current character number working on.
LITMAX	Maximum number of fields in the literal storage array (dimension of the LITSTR array).
MAXCHR	Maximum number of characters an alphanumeric string can have; currently set to seven.
NCODE	Array containing the type of fields processed.
NDXVAL	This is the number of fields that have been processed.
NK	Indicates the position in the KHAR array where the last alphanumeric character is (., period is the last character) currently set to 37.
NTAB	Column in data card currently being processed.

NTALPC Alphanumeric field indicator.

NVAL Is the array containing the values of
the fields read.

5. I/O VARIABLES:

A. INPUT LOCATION

 IERR1 F.P.

 KARDD F.P.

 KHAR F.P.

 LITMAX F.P.

 NDXVAL F.P.

 NK F.P.

 NTAB F.P.

 NTALPC F.P.

B. OUTPUT LOCATION

 IALPHA F.P.

 IERR1 F.P.

 NCODE F.P.

 NTAB F.P.

 NVAL F.P.

6. CALLING ROUTINE: SCAN

7. CALLED ROUTINES: ARAYCK
 EEROR

1. NAME: BLDDL - 15
2. PURPOSE: Deciphers delimiter fields from data cards.
3. METHOD: Given the current position on the data card that contains a delimiter (parenthesis, or comma), it assigns a number denoting which delimiter it is.
4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
IERR	Indicator. If FALSE, then no error has occurred in this routine. If TRUE then an error has occurred in this routine.
IERR1	Error indicator. If FALSE, then no error has occurred up to the present time. If TRUE, then fatal error has occurred.
K	Index used to tell which delimiter is currently being processed.
KARDD	Is the array containing the characters of the data card that is currently being worked on.
KHAR	Array containing all of the IOSYM delimiters (left parenthesis, right parenthesis, or a comma).
LITMAX	Maximum length of literal storage array LITS'R.
NCODE	Array containing the type of fields processed.
NDLM	Contains the delimiter number.
NDXVAL	This is the number of fields that have been processed.
NK	Indicates the position in the KHAR array where the delimiters are, currently set to 47.
NTAB	Column on the data card currently processing.

NTDLM	Delimiter field indicator.
NTNUL	Null field indicator.
NTOPR	Operator field indicator.
NVAL	Is the array containing the values of the fields read.

5. I/O VARIABLES:

A. INPUT	LOCATION
----------	----------

IERR1	F.P.
KARDD	F.P.
KHAR	F.P.
LITMAX	F.P.
NCODE	F.P.
NDXVAL	F.P.
NK	F.P.
NTAB	F.P.
NTDLM	F.P.
NTNUL	F.P.
NTOPR	F.P.
NVAL	F.P.

B. OUTPUT	LOCATION
-----------	----------

IERR1	F.P.
NCODE	F.P.
NDLM	F.P.
NVAL	F.P.

6. CALLING ROUTINE: SCAN

7. CALLED ROUTINES: ARAYCK
EEROR

1. NAME: BLDNUM - 16
2. PURPOSE: Builds numeric field from data cards.
3. METHOD: Given the current position on the data card that contains a numeric, this routine builds a number of up to MAXNUM number of digits with MAXEXP as the largest possible exponent. This routine deciphers all types of numbers integer (1, 4, 972), floating point (1.2, 6.3, 0.006), or exponent (1.2E3, 3.4E-4). If the number of digits in a number is greater than MAXNUM and the digit occurs on the left hand side of the decimal point, an error message is printed. If the digit occurs on the right hand side of the decimal point, then it is truncated.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
BUILD	When TRUE, still building the mantissa of the number; when FALSE, finished building the number.
DEC	Contains the value of the decimal part of the number.
EXP	Indicator. When TRUE, then are working on an exponent; when FALSE, then are not working on an exponent.
FK	The value of the current character that is being processed.
FNUM	The value of the number processed including the numbers both left and right of the decimal point.
IDIG	The integer value of the current character that is being processed.
IERR1	Error indicator. If FALSE, then no error has occurred up to the present time; if TRUE, then fatal error has occurred.
IEXP	The computer representation of the letter "E" used to detect an exponent indicator.
IMIN	The computer representation of the character "-" used to detect a minus sign.

IPLUS	The computer representation of the character "+" used to detect a plus sign.
ISGN	Indicates the sign of the exponent.
K	Index used to tell which character currently processing.
KARDD	Is the array containing the characters of the data card that is currently being worked on.
KHAR	Array containing all of the IOSYM characters 0-9 and . (decimal point).
KNTCHR	Number of characters in the number string currently processing.
KRD	Current character number working on.
MAXEXP	Value of the maximum exponent allowed on this machine.
MAXNUM	Maximum number of digits in a number allowed on this machine.
NDEC	Indicates the power of ten to the right of the decimal place.
NK	Indicates the position in the KHAR array where the last numeric character is, currently set to 37.
NTAB	Column in the data card currently being processed.
NTERR	Error field indicator.
NTFLD	Return type of field built (integer or floating point).
NTFLP	Floating point number field indicator.
NTINT	Integer number field indicator.
NUM	When returning to the calling routine, this contains the number that was built.

5. I/O VARIABLES:

A. INPUT	LOCATION
IERR1	F.P.
KARDD	F.P.
KHAR	F.P.
MAXEXP	/MCHDEP/
MAXNUM	/MCHDEP/
NK	F.P.
NTAB	F.P.
NTERR	F.P.
NTFLP	F.P.
NTINT	F.P.
B. OUTPUT	LOCATION
IERR1	F.P.
NTAB	F.P.
NTFLD	F.P.
NUM	F.P.

6. CALLING ROUTINE: SCAN

7. CALLED ROUTINE: EEROR

1. NAME: BLDOPR - 23
2. PURPOSE: Deciphers operator fields from data cards.
3. METHOD: Given the current position on the data card that contains an operator (+, -, *, /, ** or =), it assigns a number denoting which operator it is.
4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
IERR	Indicator. If FALSE, then no error has occurred in this routine; if TRUE, then an error has occurred in this routine.
IERR1	Error indicator. If FALSE, then no error has occurred up to the present time. If TRUE, then fatal error has occurred.
K	Index used to tell which operator currently processing.
KARDD	Is the array containing the characters of the data card that is currently being worked on.
KHAR	Array containing all of the IOSYM operators (+, -, *, /, **, or =).
LITMAX	Maximum length of literal storage array LITSTR.
NCODE	Array containing the type of fields processed.
NDXVAL	This is the number of fields that have been processed.
NK	Indicates the position in the KHAR array where the operators are; currently set to 43.
NOPR	When returning to calling routine, this contains the operator number.
NTAB	Column on the data card currently processing.

NTDLM	Delimiter field indicator.
NTNUL	Null field indicator.
NTOPR	Operator field indicator.
NVAL	This array contains the values of the fields read.

5. I/O VARIABLES:

A. INPUT	LOCATION
IERR1	F.P.
KARDD	F.P.
KHAR	F.P.
LITMAX	F.P.
NCODE	F.P.
NDXVAL	F.P.
NK	F.P.
NTAB	F.P.
NTDLM	F.P.
NTNUL	F.P.
NTOPR	F.P.
NVAL	F.P.
B. OUTPUT	LOCATION
IERR1	F.P.
NCODE	F.P.
NOPR	F.P.
NTAB	F.P.
NVAL	F.P.

6. CALLING ROUTINE: SCAN

7. CALLED ROUTINE: ARAYCK

1. NAME: BLOCK DATA - 97
2. PURPOSE: This routine is not ANSI. This routine contains all of the DATA statements that initialize labeled COMMON BLOCKS.
3. METHOD: The KEYWORD array is initialized to contain the command names, the command level, valid command levels, and pointers to the command syntax. The NTSFMT array is initialized to contain the syntax of all the commands. The LIBFUN is initialized to contain all of the FORTRAN library function names. The ICHAR array is initialized to contain all of the IOSYM characters. And, the NNPT2 array is initialized to contain the first nine secondary variables.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
ICHAR	This array contains all of the IOSYM characters in internal IOSYM representation.
KEYWORD	This array contains in the first two words the command name; the third word contains the command level; the fourth word is the valid command levels this command is valid under, and the last word points to the location in the NTSFMT array where the command syntax is stored.
LIBFUN	This array contains the names of all the FORTRAN library functions.
NNPT2	This array contains the names of the secondary variables.
NTSFMT	This array contains the syntax of each command. The last word in the KEYWORD array points to this array. That location contains the number of fields in the command syntax. The next, however many elements in the NTSFMT array, contain the syntax for those fields.

5. I/O VARIABLES:

A. INPUT	LOCATION
(None)	

B.	OUTPUT	LOCATION
	ICHAR	/ME1/
	KEYWORD	/ME1/
	LIBFUN	/ME1/
	NNPT2	/IO2/
	NTSFMT	/ME1/

6. CALLING ROUTINE: None. This routine is loaded automatically.
7. CALLED ROUTINE: None.

1. NAME: BXDATN - 98
2. PURPOSE: When using the CDC segmented loader, a subroutine cannot appear in more than one segment unless it is specified on the right hand side of an INCLUDE card. Subroutine XXDATN is used in two places. Routine BXDATN is just a call to XXDATN to satisfy the first line of this description.
3. METHOD: Not applicable.
4. INTERNAL VARIABLES: (None)
5. I/O VARIABLES: (None)
6. CALLING ROUTINE: DDOSIM
7. CALLED ROUTINE: XXDATN

1. NAME: CCOLC - 533
2. PURPOSE: This routine will collect the data necessary to do COLLECT statistics.
3. METHOD: Depending on where in processing a simulation is, this routine does the COLLECT processing differently.

If a simulation is just starting, then all the storage area for the COLLECT variables is initialized to zero by a call to routine ZZSET. Control then returns to the calling routine.

If a run is just starting, then the storage area for the COLLECT variables collected on a run is initialized to zero by a call to routine ZZSET. Control then returns to the calling routine.

If the processing is currently at the beginning of an OPERATION, then the initial values of the COLLECT variables are saved. This is first done by checking to see if the OPERATION is a COLLECT OPERATION or not. If not, control returns to the calling routine. If it is a COLLECT OPERATION, the current values of the COLLECT variables are determined and the values stored in the QSET array. Control then returns to the calling routine.

If the processing is currently at the end of an OPERATION, then the changes in the COLLECT variable values are saved. First, a check is done to see if the OPERATION is a COLLECT OPERATION; if not, control is returned to the calling routine. If the OPERATION is a COLLECT OPERATION, the COLLECT variables are identified and their values calculated. The initial value is subtracted from the current value and added to the total changes calculated so far. Control then returns to the calling routine.

If the processing is currently at the end of a simulation, then the end values and total changes are accumulated. This information is stored in the QSET array to be used later to print a summary of the COLLECT statistics. Control is then returned to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Index to the NNCOL array where the current variable being processed is located.
IDV	IOSYM variable identifier of the variable being processed.
IFN	Activity specification 1 = Start of simulation 2 = Start of run

	3 = Beginning of OPERATION 4 = End of OPERATION 5 = End of simulation
IFUNC	Activity specification -1 = Start of simulation 0 = Start of run 1 = Beginning of OPERATION 2 = End of OPERATION 3 = End of simulation
IIOP	OPERATION number of the current OPERATION.
IISTAT	Indicator to show whether IOSYM statistics are collected or not 0 = Do not collect statistics 1 = Do collect statistics
INDX1 INDX2 INDX3	Indices used to point to various locations in the QSET array.
K L LL	Index and pointers used for various purposes.
LLCOL	The first word available for COLLECT statistics storage in the QSET array.
LLL	Index used to point to the QSET array.
NNCOL	Number of COLLECT variables.
NNCOLC	Array containing the variable identifiers of the COLLECT variables.
NNCOP	Number of COLLECT OPERATIONS.
NNPTOP	Array that contains in the first two words the OPERATION name; the third, fourth, and fifth words point to the commands in the OPERATION, and the sixth word indicates whether the OPERATION is a COLLECT OPERATION or not. If the value is greater than zero, then the OPERATION is a COLLECT OPERATION.
PIK	Percent change of the given variable on the given OPERATION.

PPREC	Precision of the computer set in routine USRDAT.
QSET	This array contains the COLLECT statistic information.
XTI	Difference of the given value between its initial value and its current value.
YIK	Amount the given variable changed on the given OPERATION.

5. I/O VARIABLES

A. INPUT	LOCATION
IFUNC	F.P.
IIOP	/I03/
IISTAT	/I03/
LLCOL	/I03/
NNCOL	/I03/
NNCOLC	/I01/
NNCOP	/I03/
NNPTOP	/I02/
PPREC	/I04/
QSET	/ /
B. OUTPUT	LOCATION
QSET	/ /

6. CALLING ROUTINES: AARCST

DDOSIM
EVNTS
INTLC
OTPUT

7. CALLED ROUTINES: VVARBL

ZZSET

1. NAME: CHOICE - 67
2. PURPOSE: CHOICE controls the flow of processing from initialization to input processing to execution to input processing, etc.
3. METHOD: A variable called IICHOI is used to indicate where processing is to go next. Initially, it is set to one to indicate that initialization is next. After initialization, input processing is next so IICHOI is set to two. Depending on the results of the input processing the value of IICHOI could be 2, -2, 3, 4 or -4. The number three indicates that an IOSYM simulation should take place next; the number four indicates that IOSYM processing is to end; the numbers with minus signs indicate that, in addition to the usual function of the specific number, a NETWORK summary is to be done. After an IOSYM simulation input, processing again takes place. If IICHOI is four, that indicates IOSYM processing is to end.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
DUMP	Indicator. If TRUE, then a printout of all of the IOSYM arrays is to occur. If FALSE, then a DUMP does not occur.
FIRST	Indicator which indicates that processing the first IOSYM record hasn't started yet, if TRUE. If FALSE, then input processing has already started.
IICHOI	Determines what procedure is to occur next: 1 = initialization 2 = input processing -2 = do NETWORK summary and proceed with input processing 3 = do IOSYM simulation 4 = end IOSYM execution -4 = do NETWORK summary, then end IOSYM execution
NTALP	Alphanumeric indicator.
NTALPC	Alphanumeric continuation indicator.
NTDLM	Delimiter field indicator.
NTERR	Error field indicator.
NTFLP	Floating point number field indicator.

NTFUNF	Menu function field indicator.
NTFUNP	Probability function field indicator.
NTFUNU	User function field indicator.
NTINT	Integer number field indicator.
NTNUL	Null field indicator.
NTOPR	Operator field indicator.
NTTBL	Table field indicator.
NTVARD	Derivative variable field indicator.
NTVARO	Independent variable indicator.
NTVAR1	Primary variable indicator.
NTVAR2	Secondary variable indicator.

5. I/O VARIABLES:

A. INPUT	LOCATION
IICHOI	F.P.
B. OUTPUT	LOCATION
FIRST	/ME1/
IICHOI	F.P.

6. CALLING ROUTINE: IOSYM (Main program)

7. CALLED ROUTINES: ADATA
 ANET
 DDOSIM
 DDUMP
 PARSE

1. NAME: CHOIC2 - 68
2. PURPOSE: CHOIC2 controls the flow of input processing.
3. METHOD: A variable called NDX1 (current command number) determines what type of input processing is to be done. If NDX1 indicates that GASP processing is to occur (NDX1=13), then routine CXDATN is called to do the GASP processing. If NDX1 indicates that IOSYM memory is to be initialized by a CLEAR command (NDX1=4) or by a NETWORK command (NDX1=19) then routine ZERO is called to do initialization. If NDX1 is anything other than the above, then routine PROCES is called to continue IOSYM command processing.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
FIR2	Indicator to show whether an ARC leaving the given NODE has already taken place. If FIR2(I) is TRUE, then an ARC leaving NODE I has not been processed. If FIR2(I) is FALSE, then an ARC leaving NODE I has been processed.
ICLEVL	Is an array containing the current command hierarchy from level 1 to level LEVEL.
ID	Dimension of the FIR2 array.
IERR1	Error indicator. If FALSE, then no error has occurred up to present time. If TRUE, then a fatal error has occurred.
IICHOI	Indicates what procedure is to occur next in routine CHOICE.
IPICK	Location of the pointer in a PICK command for an ARC, leaving a NODE, to the next ARC leaving the same NODE.
ITYP1	Indicates what TYPE command is now being processed: 0 = not doing TYPE commands. 1 = Start commands. 2 = Immediate commands. 3 = End commands.

LEVEL	Indicates what level of command hierarchy is being processed. There are five levels: 0th, 1st, 2nd, 3rd, and 4th.
LLXSET	Counter of the number of locations in current use in the NSET array. Also, last position in NSET used.
NDX1	Current command number.
NNSET	Amount of NSET storage that GASP knows about.
NTFLP	Floating point number field indicator.
NTINT	Integer number field indicator.
NTNUL	Null field indicator.
NTTBL	Table field indicator.
NTVARD	Derivative variable field indicator.
NTVARO	Independent variable field indicator.
NTVAR1	Primary variable field indicator.
NTVAR2	Secondary variable field indicator.

5. I/O VARIABLES:

A,	INPUT	LOCATION
	FIR2	F.P.
	ICLEVL	/ME1/
	ID	F.P.
	IERR1	F.P.
	IICHOI	F.P.
	IPICK	F.P.
	ITYP1	F.P.
	LEVEL	/ME1/
	NDX1	/ME1/

NNSET	/GCOM5/
NTFLP	F.P.
NTINT	F.P.
NTNUL	F.P.
NTTBL	F.P.
NTVARO	F.P.
NTVAR1	F.P.
NTVAR2	F.P.
B. OUTPUT	LOCATION
LLXSET	/I03/

6. CALLING ROUTINE: PARSE
7. CALLED ROUTINES: CXDATN
PROCES
ZERO

1. NAME: COLSTR - 53
2. PURPOSE: This subroutine processes the COLLECT and NOCOLLECT commands into IOSYM nomenclature for later execution by IOSYM during a simulation.
3. METHOD: Given the COLLECT information parsed into the NNWRK array, COLSTR stores the variables that statistics are to be collected on, in the NNCOLC array. Given the NOCOLLECT information parsed into the NNWRK array, COLSTR determines if there are any variables specified on the NOCOLLECT command to remove. If no specific variables are specified, then the number of COLLECT variables is set to zero. If variables are specified on the NOCOLLECT command, then the current COLLECT variables are searched for equivalent variables, and they are eliminated. If a variable specified on a NOCOLLECT command is not a current COLLECT variable, then a warning is printed stating that that variable was not found and is ignored. If there are COLLECT variables left after a NOCOLLECT command is processed and there are no COLLECT OPERATIONS, then all the OPERATIONS are reset to COLLECT OPERATIONS.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Index to NNWRK array.
IERR	Error indicator. If TRUE then a fatal error has occurred in this routine. If FALSE, then in error hasn't occurred in this routine.
IERR1	Error indicator. If TRUE, then fatal error has occurred. If FALSE, no fatal error has occurred up to the current time.
IIDV	The identifying number of the current variable being worked on.
ISUB	Indicator. If greater than zero, then doing a COLLECT command. If less than or equal to zero, then doing a NOCOLLECT command.
J	Index and counter for NNCOLC array.
LLXWRK	Indicates the last defined location in NNWRK array.
MMCOLC	Maximum number of COLLECT variables allowed.

NNCOL	Number of current COLLECT variables.
NNCOLC	Array for storing IIDVs of COLLECT variables.
NNCOP	Number of COLLECT OPERATIONS.
NNOP	Number of OPERATIONS.
NNPTOP	Array containing (1,x) = first four characters of OPERATION name. (2,x) = next three characters of OPERATION name. (3,x) = pointer to TYPE S commands on OPERATION. (4,x) = pointer to TYPE I commands on OPERATION. (5,x) = pointer to TYPE E commands on OPERATION. (6,x) = indicator to show whether or not this OPERATION is a COLLECT OPERATION. If (6,x) is less than or equal to zero, then it is not a COLLECT OPERATION. If it is greater than zero, then it is a COLLECT OPERATION.
NNWRK	Work storage array. Contains the current command parsed into the various elements.
NTNUL	Null field indicator.
NTTBL	Table field indicator.
NTVARD	Derivative variable field indicator.
NTVAR0	Independent variable field indicator.
NTVAR1	Primary variable field indicator.
NTVAR2	Secondary variable field indicator.

5. I/O VARIABLES:

A. INPUT	LOCATION
ISUB	F.P.
LLXWRK	/I03/

MMCOLC	/I04/
NNCOL	/I03/
NNCOLC	/I01/
NNCOP	/I03/
NNOP	/I03/
NNWRK	/I03/
NTNUL	F.P.
NTTBL	F.P.
NTVARD	F.P.
NTVAR0	F.P.
NTVAR1	F.P.
NTVAR2	F.P.
B. OUTPUT	LOCATION
IERR1	F.P.
NNCOL	/I03/
NNCOLC	/I01/
NNCOP	/I03/
NNPTOP	/I02/

6. CALLING ROUTINE: PROCES

7. CALLED ROUTINES: ARAYCK

EEROR

SSHIFT

1. NAME: CONVGJ
2. PURPOSE: This routine returns the $\text{NORM2}(X1-X2)/\text{MIN}(\text{NORM2}(X1), \text{NORM2}(X2))$, where $\text{NORM2}(\)$ is the L2 NORM.
3. METHOD: The difference between each element in the X1 array and the X2 array is squared and added together. The inner products of arrays X1 and X2 are calculated and the square root of the minimum is taken. If less than ACCC, then it is set to one. Then CONVGJ is calculated by taking the square root of the differences squared divided by the square root of the minimum inner product.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
ACCC	Accuracy for denominator.
CONVGJ	Function value returned to the calling routine.
I	Do loop index.
N NN	Number of array elements to process.
X1	First array to process.
X2	Second array to process.
Z	Denominator.
Z2	Inner product of second array.

5. I/O VARIABLES:

A. INPUT	LOCATION
ACCC	F.P.
NN	F.P.
X1	F.P.
X2	F.P.
B. OUTPUT	LOCATION
CONVGJ	FUNCTION

6. CALLING ROUTINE: (None. Not used at this time)
7. CALLED ROUTINE: DOTJ

1. NAME: CONVRT - 40
2. PURPOSE: This routine takes an IOSYM Menu, Probability or User function and deciphers the name into the type of function it is and which one in that type.
3. METHOD: The function name is stored in four characters in the first alphanumeric word for that function. The letter corresponding to the function type is first striped off. The resulting alphanumeric is then shifted toward the right to remove any blank characters until only the alphanumerics of the numbers are in from one to three character positions starting at the right of the word. Now the alphanumeric field is converted to a numeric field. First, a tens digit is deciphered; if present, then ones digit. Once both of these have been individually defined, they are recombined into a numeric field from one to ninety-nine.

4. INTERNAL VARIABLES:

VARIABLE

DEFINITION

IVAL1

Upon entering this routine, this is the function name and number in the form of IXXB where I is the type of function, XX is the function number and B is a blank. When leaving this routine, this contains the function number as a number instead of an alphanumeric.

IVAL2

This contains the letter corresponding to the type of function being deciphered:
F = menu function.
P = probability function.
U = user function.

N1

Contains the tens digit; if there is not a tens digit, it is set to the number zero.

N2

Contains the ones digit.

5. I/O VARIABLES:

A. INPUT

LOCATION

IVAL1

F.P.

IVAL2

F.P.

B. OUTPUT

LOCATION

IVAL1

F.P.

6. CALLING ROUTINE: FUNEND

7. CALLED ROUTINE: (NONE)

1. NAME: CONVR2 - 13
2. PURPOSE: This routine, given an integer number as a name, will convert the number into an alphanumeric field.
3. METHOD: Given an integer number, it is first truncated to seven digits; if it is longer than seven, a warning is issued telling that it was truncated by dropping the least significant digits (right-hand most digits). The magnitude of the number is then found to know how many characters to decipher. The number is then broken into individual digits and converted to alphanumeric characters. These characters are then recombined into two alphanumeric words. The first word contains the first four characters, and the second word contains three characters plus a blank.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Counter to show which character currently processing.
IERR1	Error indicator. If TRUE, then fatal error has occurred; if FALSE, no fatal error has occurred up to the current time.
II	Index to show which character in the N array is being processed.
IJ1	Indicates the number of digits in the number being converted.
INTINT	Temporary storage for the IOSYM representation of the number,
INTIN1	On entering this routine, it contains the integer number to be deciphered. During final phase of this routine and upon returning to calling routine, it contains the alphanumeric IOSYM representation of the first four characters.
INTIN2	Returns as the IOSYM representation of the last three characters and a blank.
IT	Power of ten of the current digit working on.

J Index to show which of the two words in the array INTINT is being worked on.

JJ Indicator to show if the number has been truncated before processing: 0 = number not truncated.
1 = number has been truncated.

K IOSYM internal representation of the numeric character being coded.

N Array containing the individual numeric characters.

5. I/O VARIABLES;

A. INPUT	LOCATION
INTIN1	F.P.
B. OUTPUT	LOCATION
IERR1	F.P.
INTIN1	F.P.
INTIN2	F.P.

6. CALLING ROUTINE: ALPFND

7. CALLED ROUTINE: EEROR

1. NAME: CXDATN - 69
2. PURPOSE: When using the CDC segmented loader, a subroutine cannot appear in more than one segment unless it is specified on the right hand side of an INCLUDE card. Subroutine XXDATN is used in two places. Routine CXDATN is just a call to XXDATN to satisfy the first line of this description.
3. METHOD: Not applicable.
4. INTERNAL VARIABLES: (none)
5. I/O VARIABLES: (none)
6. CALLING ROUTINE: CHOIC2
7. CALLED ROUTINE: XXDATN

1. NAME: DATIN - 541
2. PURPOSE: This is an IOSYM version of the GASP-IV DATIN routine. The main parts of the GASP routine are contained in the IOSYM routines XXDATN and XXINTL.
3. METHOD: Not applicable.
4. INTERNAL VARIABLES: None.
5. I/O VARIABLES: None.
6. CALLING ROUTINE: GASP (main GASP-IV routine).
7. CALLED ROUTINE: INTLC

1. NAME: DDOCMD - 530
2. PURPOSE: This routine does the processing of a chain of commands.
3. METHOD: By starting at the command given by the pointer LSTRT in the NNWRK array, the command number and position in the NSET array are found. This information is passed to routine PPFCMD which does the actual execution of the command. The next command is identified and the pointers found and passed to PPFCMD; this continues until there are no commands left in the list and control returns to the calling routine. If the command identified is an IF or a LOOP command, then the next command after them is found by their LABEL commands which in turn identify the next command.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
ICM	Command number of the current command.
INX	Location in the NSET array where the next command starts.
LLUCSL	Array that contains the lengths of all the standard command uniform command sequences.
LSTRT	Location in the NSET array where the pointer to where the initial command starts is located.
LUCS	Location in the NSET array where the next command starts.
NSET	Storage array where the link listed NETWORK is stored.

5. I/O VARIABLES:

A. INPUT	LOCATION
LLUCSL	/I02/
LSTRT	F.P.
NSET	/ /
B. OUTPUT	LOCATION
ICM	F.P.
LUCS	F.P.

6. CALLING ROUTINE: EVNTS
NNODST
7. CALLED ROUTINE: PPFCMD

1. NAME: DDOEVT - 525
2. PURPOSE: This routine will perform the EVENT command.
3. METHOD: This routine performs the EVENT command whose uniform command sequence starts at the location given by LUCS. The parameters on the EVENT, if any, are identified and their values stored in the work array. The user written routine UEVENT is then called to process the user EVENTS with the parameter values. Control then returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	LOCATION
I	Index and counter for the parameter being processed.
IDV	Variable identifier of the parameter being processed.
IERR1	Error indicator returned from the routine EEROR but not used in this routine.
IEVT	EVENT number of the current EVENT being processed.
LLEVNT	Location in the work array that holds the first parameter for an EVENT command.
LLWRK	Next location in the work array that is unused at the current time.
LUCS	Location in the NSET array where the EVENT command UCS starts.
LWRK	Pointer to different locations in the work array.
LWRKE	Pointer to the NSET array where the variable identifier for the current variable is located.
MMWRK	Maximum number of elements that can be used in the work array (dimension of the NNWRK and the WWRK arrays, set in routine USRDAT).
NNWRK	Work storage array equivalenced to the WWRK array.

NPARS	Number of EVENT parameters.
NSET	Storage array containing the link listed NETWORK.
WWRK	Work storage array equivalenced to the NNWRK array.

5. I/O VARIABLES:

A. INPUT LOCATION

IERR1	F.P.
LLWRK	/I03/
LUCS	F.P.
MMWRK	/I04/
NNWRK	/I03/
NSET	/ /

B. OUTPUT LOCATION

LLEVNT	/I03/
LLWRK	/I03/
WWRK	/I03/

6. CALLING ROUTINE: PPFCMD

7. CALLED ROUTINES: EEROR

RREWRK

UEVENT (user written)

VVARBL

1. NAME: DDOLST - 531
2. PURPOSE: This routine performs the commands that were stored in the command list built by routine BBLIST.
3. METHOD: The pointer to the command uniform command sequence (UCS) is identified, and the pointers are passed to routine PPFCMD which actually does the command execution. This is repeated until all the commands in the list have been completed; then control returns to the calling routine.
4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
ICM	Command number of the command currently being processed.
LIST	Location in the NNWRK array where the pointer to the current command UCS is located.
LLIST	Location in the NNWRK array where the list of pointers to the various commands start.
LUCS	Location in the NSET array where the current command UCS starts.
NNWRK	Work storage array where the command list is stored starting at location LLIST.
NSET	Storage array where the link listed NETWORK is stored.

5. I/O VARIABLES:

A. INPUT	LOCATION
LLIST	/I03/
NNWRK	/I03/
NSET	/ /
B. OUTPUT	LOCATION
ICM	F.P.
LUCS	F.P.

6. CALLING ROUTINES: AARCST

EVNTS

STATE

7. CALLED ROUTINE: PPFCMD

1. NAME: DDOSIM - 542
2. PURPOSE: This is the main routine in the execution phase of doing an IOSYM simulation. It will perform DO simulations or MINIMIZE simulations.
3. METHOD: This routine determines whether this is a DO or MINIMIZE by checking the value of the variable NNSIM. If a DO command is to be done, then the pointers are established, and processing begins. Routine TTAPES is called to renew the GASP storage area before each run. GASP input is read; if there is any, all the READ variables are read. If there is any, the user input routine UREAD is called; statistics storage is prepared, and routine GASP is called to do the run. Routines OTPUT and PPRCOL are called to process any necessary output. After all the runs are finished, control returns to the calling routine.

If doing a MINIMIZE simulation, then control just returns to the calling routine because at this time the MINIMIZE option is not yet available.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
DUMP	Indicator. If TRUE, then a printout of all of the IOSYM arrays is to occur. If FALSE, then a DUMP is not to occur.
I	Indicator to determine if a WITH variable is also a READ variable. If I is greater than zero, then the variable is a WITH variable only. If I is less than or equal to zero, then it is also a READ variable.
ICHAR	Array containing all of the IOSYM characters in IOSYM notation.
IERR1	Error indicator returned from routine EEROR but not used in this routine.
II	Index used to point to different positions in the IRAY array.
IIGASP	Indicator if equal to one. Then GASP input is to be read before each run.
IISIM	The number of the current simulation.

IISTAT	Indicator if equal to one. Then IOSYM statistics are to be done. If equal to zero, IOSYM statistics are not to be done.
IRAY	Array that contains the NETWORK name as one character per word.
IRUN	Number of runs in this simulation.
ISIM	Number of the current simulation.
JGASP	Word containing the string GASP.
JREAD	Word containing the string READ.
KODE	Not used at current time.
KREAD	Character string read from input card.
L	Pointer and index to locations in the NNWRK where the WITH variable information is located.
LEND	Location in the NNWRK where the beginning of the WITH variable information is located.
LLWITH	Start of the WITH information in the NNWRK array.
NNMNET	Array containing the NETWORK name in the first two words and an indicator to show whether a NETWORK summary should be done or not in the third word.
NNREAD	Number of READ variables there are to be processed.
NNRNS	Number of runs remaining in the simulation, including the one being processed.
NNRUN	Number of the current run.
NNSIM	Number of simulations to do if greater than zero and if less than or equal to zero, then a MINIMIZE command is to be done.

NNWITH	Number of WITH variables.
NNWRK	Work storage array.
NPRNT	Device code number where the output file is to be written.
VAL	Value of the READ variable processed.
WRK	Work storage array equivalenced to the NNWRK array.

5. I/O VARIABLES:

A. INPUT	LOCATION
DUMP	/ME1/
ICHAR	/ME1/
IERR1	F.P.
IIGASP	/I03/
LLWITH	/I03/
NNMNET	/I03/
NNREAD	/I03/
NNRNS	/GCOM5/
NNSIM	/I03/
NNWITH	/I03/
NNWRK	/I03/
NPRNT	/GCOM1/

B. OUTPUT	LOCATION
IISIM	/I03/
IISTAT	/I03/
NNRUN	/GCOM5/
WRK	/I03/

6. CALLING ROUTINE: CHOICE

7. CALLED ROUTINES: BXDATN

CCOLC

EEROR

FFAJ

FFPUDJ

FFRJ

GASP (main routine in the GASP program)

PPRCOL

PPRRPT

RREPRT

SSTWRK

TTAPES

UNCO1

UREAD (user written routine)

1. NAME: DDPRINT - 72
2. PURPOSE: This routine executes PRINT commands during an IOSYM simulation and stores appropriate information on the TABULATE file, does an immediate PRINT or bypasses PRINT altogether.
3. METHOD: Given the Uniform Command Sequence (UCS) for the PRINT command, this routine decides what type of PRINT is to be done. The PRINT information is deciphered from the PRINT UCS. The location of the variables to be printed, the current time, the values of the variables, the current ARC and OPERATION or NODE are compiled. Then, depending on the value of the second word in the UCS, the information is written to the TABULATION file or routine PPPRNT is called to print the PRINT information immediately.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
AN1	Temporary storage for floating point value of the second word in the PRINT UCS which determines the type of PRINT command to do. See definition of IN1.
IDV	Pointer to the NSET array where the variable identifiers (IIDVs) are located.
IIARC	Current ARC Number.
IINXND	Current NODE number.
IIOP	Current OPERATION number.
IN1	Integer value of the second word of the PRINT UCS, used to determine the type of PRINT to do. -1 = do no PRINT, return to calling routine. 0 = do an immediate PRINT. >0 = store PRINT information on TABULATION IN1 file.
IOPNO	An array containing the current OPERATION and ARC numbers or the current NODE number. If current processing is on an ARC, then the first word contains the OPERATION number, and the second word is the

	ARC number. If currently processing a NODE, then the first word is zero, and the second word is the negative of the NODE number.
LUCS	Pointer to the NSET array where the UCS for the PRINT command starts.
NNTABU	<p>Array containing the TABULATION information:</p> <p>(1,x) = Time option specification if equal to one, then time option is off; if equal to two, the time option is on; if equal to minus one, then TABULATION has not been defined yet.</p> <p>(2,x) = The TABULATION variable identifier (IIDV).</p> <p>(3,x) = Is the type of TABULATION; if it is equal to one, then it is a start type TABULATION; if equal to two, then it is an End type TABULATION. If equal to three, then it is an Immediate type TABULATION. (For more explanation of the TABULATION, see IOSYM User's Manual).</p> <p>(4,x) = Corresponding IIDVs for next</p> <p>(6,x) three variables.</p> <p>(8,x)</p> <p>(5,x) = Corresponding TABULATION</p> <p>(7,x) types for next three variables.</p> <p>(9,x)</p>
NSET	Storage array where the link listed NETWORK is stored.
NV	Number of PRINT variables.
PPREC	Machine precision, used to check and make sure integer numbers are truncated to the correct integer and not integer minus one.
T	Array containing in the first word, the location in the NSET array where the IIDVs of the variables are stored. The second word contains the time the PRINT statement was executed.

TNOW

Current time of the IOSYM Simulation.

VAL

An array that contains the current values of the PRINT variables.

5. I/O VARIABLES:

A. INPUT	LOCATION
IIARC	/I03/
IINXND	/I03/
IIOP	/I03/
LUCS	F.P.
NNTABU	/I01/
NSET	/ /
PPREC	/I04/
B. OUTPUT	LOCATION
IOPNO	F.P.
NV	F.P.
T	F.P.
VAL	F.P.

6. CALLING ROUTINE: PPFCMD

7. CALLED ROUTINES: PPPRNT

VVARBL

WWRIT

1. NAME: DDRVJ
2. PURPOSE: This routine will evaluate the directional derivative of a function F at a point X in the direction V.
3. METHOD: If an estimate of the gradient of F at X is known, then the value is calculated directly. If the estimate of the gradient isn't known, it must be calculated first before the final value is calculated.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
DDRVJ	Function name returns the value calculated.
DFV	Value of the function F at WORK.
DFX	Value of the function F at X.
DH	Precision of calculation.
FX	Value of the function F at X.
H	Intermediate variable.
HH	Precision of calculation.
I	Do loop index.
ITEST	Indicator. If equal to one, the gradient is known. If equal to two, the gradient isn't known.
J	Location in the X array where the largest magnitude number is located.
N NV	Number of elements in the X, W, and V arrays.
PREC	Machine precision.
V	A vector containing the direction.
VNORM	Square root of the inner product of array V.

WORK

Working array.

X

An array containing the point at which to evaluate the directional derivative.

5. I/O VARIABLES:

A. INPUT

LOCATION

FX

F.P.

ITEST

F.P.

N

F.P.

PREC

/ALLJ/

V

F.P.

WORK

F.P.

X

F.P.

B. OUTPUT

LOCATION

DDRVJ

F.P.

HH

F.P.

NV

F.P.

WORK

F.P.

6. CALLING ROUTINE: (None. Not used at this time)

7. CALLED ROUTINES:

DOTJ

F (user-written routine)

VADDJ

1. NAME: DDUMP - 94
2. PURPOSE: This routine does a DUMP of IOSYM memory in a readable fashion. It does a DUMP of all the link listed NETWORK.
3. METHOD: Different print routines are called to print out various sections of the IOSYM arrays.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
CCONS	This array contains all of the constants in a NETWORK
I	This array contains the IOSYM array names to print out in the DUMP leadings.
IBLANK	FORTTRAN representation of a blank.
IIRPT	This array contains the IOSYM REPORT statistics variable identifiers.
LLXSET	Last position in the NSET array used to store the link listed NETWORK.
MMPLOT	Maximum number of PLOTs commands that can be stored in the NNPLLOT array. (Dimension of the NNPLLOT array.)
MMTABU	Maximum number of TABULATE commands that can be stored in the NNTABU array. (Dimension of the NNTABU array.)
NNARC	This is the number of ARCs that were processed in the NETWORK.
NNCNS	This is the number of constants that are stored in the CCONS array.
NNCOL	This is the number of COLLECT variables in the NETWORK.
NNCOLC	This array contains the IOSYM variable identifiers for all the variables to be COLLECTed and reported in the COLLECT statistics.

NNNODE	This is the number of NODEs in the NETWORK.
NNOP	This is the number of OPERATIONS in a NETWORK.
NNPLOT	This array contains the PLOT information.
NNPTAC	This array contains the ARC information.
NNPTB	This array contains the TABLE information.
NNPTGB	This array contains the GLOBAL information.
NNPTND	This array contains the NODE information.
NNPTOP	This array contains the OPERATION information.
NNPTO	This array contains the independent variable information.
NNPT1	This array contains the primary variable information.
NNPT2	This array contains the secondary, variable information.
NNRPT	Number of REPORT variables.
NNTABU	This array contains the TABULATE information.
NNV1	Number of primary variables.
NNV2	Number of secondary variables.
NNXTB	Number of TABLEs.
NSET	Contains the link listed NETWORK.
TTABL	This array contains the TABLE element values.

5. I/O VARIABLES:

A. INPUT	LOCATION
CCONS	/I02/
IIRPT	/I01/
LLXSET	/I03/
MMPL0T	/I04/
MMTABU	/I04/
NNARC	/I03/
NNCNS	/I03/
NNCOL	/I03/
NNCOLC	/I01/
NNNODE	/I03/
NNOP	/I03/
NNPLOT	/I01/
NNPTAC	/I02/
NNPTB	/I02/
NNPTGB	/I02/
NNPTND	/I02/
NNPTOP	/I02/
NNPT0	/I02/
NNPT1	/I02/
NNPT2	/I02/
NNRPT	/I03/
NNTABU	/I01/
NNV1	/I03/
NNV2	/I03/

NNXTB /I03/

NSET / /

TTABL /I02/

B. OUTPUT LOCATION

None.

6. CALLING ROUTINE: CHOICE

7. CALLED ROUTINES: PIVECJ

PKMATJ

PVEQJ

1. NAME: DFAUT
2. PURPOSE: This is a GASP routine with some modifications to initialize IOSYM variables and change some of the GASP default values.
3. METHOD: For further discussion of this routine, see the GASP-IV book.
4. INTERNAL VARIABLES: (See GASP-IV book).
5. I/O VARIABLES: (See GASP-IV book).
6. CALLING ROUTINES: XXDATN
ZERO
7. CALLED ROUTINES: (None).

1. NAME: DOSTR - 553
2. PURPOSE: This routine stores the information provided by the DO command for later use in an IOSYM simulation.
3. METHOD: The number of simulations is stored. A check is done to make sure it is a valid number. Pointers to various items in the work array are setup, and an indicator that indicates whether GASP input is to be processed or not is set.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
IERR1	Error indicator returned from routine EEROR, but not used in this routine.
IIGASP	Indicator. If equal to zero, then no GASP input is to be processed before each run. If equal to one, then GASP input is to be read before each run.
LLWITH	Indicator. If equal to zero, then there is no WITH or MINIMIZE information in the work array. If greater than zero, then LLWITH is the pointer to the first location in the work array that contains WITH or MINIMIZE information.
LLWRK	Next available location in the work array (NNWRK).
NNSIM	If greater than zero, then it indicates the number of simulations to do. If less than zero, indicates a minimization is to be done.
NNSM	Number of simulations to do.
NNWITH	Number of WITH variables.

5. I/O VARIABLES:

A. INPUT	LOCATION
IERR1	F.P.
NNSM	F.P.

B.	OUTPUT	LOCATION
	IIGASP	/I03/
	LLWITH	/I03/
	LLWRK	/I03/
	NNSIM	/I03/
	NNWITH	/I03/

6. CALLING ROUTINE: PROCES

7. CALLED ROUTINE: EEROR

1. NAME: DOTJ
2. PURPOSE: This routine computes the inner products of vectors X and Y.
3. METHOD: This routine multiplies each element of X by the corresponding element in Y and sums the results.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
DOTJ	This is the FUNCTION name and contains the inner product value.
I	Do Loop index.
N, NN	Number of array elements to process.
PREC	Machine precision.
X, Y	Arrays to multiply together.
YY	Intermediate results of $X(I) * Y(I)$.

5. I/O VARIABLES:

A. INPUT	LOCATION
N	F. P.
PREC	/ALLJ/
X	F. P.
Y	F. P.
B. OUTPUT	LOCATION
DOTJ	FUNCTION

6. CALLING ROUTINES:

DDRVJ
ZVECJ

7. CALLED ROUTINES: (None)

1. NAME: DVARFD - 50
2. PURPOSE: This routine takes a possible alphanumeric variable name and determines if it is a derivative variable and, if so, processes it.
3. METHOD: The first letter is stripped off and determined whether it is a D. If not, processing returns to the calling routine because the name is not a derivative. If the first letter is a D, then further processing is needed to determine if it is a derivative variable. The name is rebuilt without the leading D. If the resulting name matches the given primary variable name, then the original name was a derivative variable, so it is noted and control returns to the calling routine. If the names do not match, control returns for further processing.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Pointer to the current position in the primary variable storage array that is being compared to by the suspected derivative variable.
IFLD	Contains the derivative field indicator if the variable was a derivative; if not, it contains a zero.
IND1	Contains the first four letters of the suspected name.
IND2	Contains the next four letters of the suspected name.
IN1	Is the first four characters of the suspected name with the D stripped off and characters shifted over one.
IN2	Is the next three characters of the suspected name with the D stripped off and a blank added after the first character was shifted to IN1.
IPT	Is the first character of the name that is stored in IND1 or IND2.
NNPT1	Array containing the names of all the primary variables. The first word contains the first four characters of the

name, and the second word contains the next three characters plus a trailing blank.

NTVARD

Derivative variable field indicator.

5. I/O VARIABLES:

A. INPUT

LOCATION

I

F.P.

IND1

F.P.

IND2

F.P.

NNPT1

/I02/

NTVARD

F.P.

B. OUTPUT

LOCATION

IFLD

F.P.

6. CALLING ROUTINES: DVFND

FUNC2

FUNC4A

7. CALLED ROUTINE: (None)

1. NAME: DVFND - 2
2. PURPOSE: This routine, given an alphanumeric field and in some cases a numeric field, checks to see if the alphanumeric field is any kind of defined variable; if not, it is assumed to be a new secondary variable. In some instances, a number may be substituted for a defined variable, and this routine checks and does the substitution.
3. METHOD: A search is done to find the first non-delimiter field. If the field is a number and number substitutions are not allowed, an error is processed. If number substitutions are allowed, then this substitution is done, and control is returned to calling routine. If the first field is an alphanumeric, then a check is done to see if this alphanumeric field name matches any other defined variables. If names do match, then that variable type and number are noted and stored for later use, and control returns to the calling routine. If the current command level is OPERATION, ARC or GLOBAL, a check is also done to see if the alphanumeric name matches any of the DUMM variables. If work is currently being done on REPORT, NOREPORT, COLLECT or NOCOLLECT commands, the alphanumeric name must be checked against TABLE names with no arguments in addition to normal check of single TABLE arguments. If, in any of the cases, the name is not found to agree with any current names, it is assumed that the alphanumeric should be added to the secondary variable list. If currently there is a need for a list of defined variables (LDV) instead of a single defined variable (DV), then the calling routine the above sequence is repeated until there are no more variables in the LITSTR array to check.
4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Do loop index and pointer.
ICLEVL	Array containing the current command hierarchy from level 1 to level LEVEL.
IER	If certain errors occur, this contains the error number.
IERR	Error indicator. IF FALSE, then no error has occurred in this routine. If TRUE, then a fatal error has occurred in this routine.
IERR1	Error indicator. IF FALSE, then no error has occurred up to the present time. If TRUE, then a fatal error has occurred.

IFLD	Contains the type of variable field that has been observed.
II	Do loop index and pointer.
IIN	Is the command level once removed from the current command level.
IND9	When processing a TABLE array under commands REPORT, NOREPORT, COLLECT or NOCOLLECT, this is the actual length of that table.
INUM	Indicator 0 = only alphanumeric defined variables are allowed. 1 = either alphanumeric defined or numbers are allowed.
LDV	Indicator. If TRUE, then are looking for a list of defined variables (LDV); if FALSE, then are only looking for one defined variable (DV).
LEVEL	Indicates level of command hierarchy being processed.
LITMAX	Maximum length of literal storage array LITSTR. Also used as a variable dimension of LITSTR.
LITNDX	Position of last defined location in literal storage array LITSTR.
LITSTR	Array where literals are stored after being scanned and prior to being parsed.
LLXWRK	Indicates the last defined location in NNWRK array. Also used as a pointer to NNWRK array.
LOL	List of literal indicator set to FALSE before call to NUMFND to indicate only want one number stored and returned.
LWRK	Temporary counter for work array NNWRK.
MMV2	Maximum number of secondary variables possible. (Set by user in USRDAT).

MMWRK	Maximum length of work storage array NNWRK, set by user in the routine USRDAT.
NDXLIT	Pointer to current position in the literal storage array LITSTR.
NNPTB	Array containing TABLE information (1,x) = the first four characters of the TABLE name. (2,x) = the next three characters of the TABLE name and a blank to fill it to four characters. (3,x) = pointer to the TTABL array where the storage of values for TABLE X starts.
NNPTO	Array containing the independent vari- able name. The first four characters in the first word and the next three characters plus a blank in the second word.
NNPT1	Array containing all of the primary variable names with the first four characters in the first word (1,x) and the next three characters plus a blank in the second word (2,x).
NNPT2	Array containing all of the secondary variable names with the first four characters in the first word (1,x) and the next three characters plus a blank in the second word (2,x).
NNV1	The number of currently defined primary variables.
NNV2	The number of currently defined secondary variables.
NNWRK	Work storage array. Contains the current command parsed into the various elements.
NNXTB	Number of currently defined TABLEs in use.
NTALPC	Alphanumeric field indicator.
NTDLM	Delimiter field indicator.
NTFLP	Floating point number field indicator.
NTINT	Integer number field indicator.

NTNUL	Null field indicator.
NTOPR	Operator field indicator.
NTTBL	Table variable field indicator.
NTVARD	Derivative variable field indicator.
NTVARO	Independent variable field indicator.
NTVAR1	Primary variable field indicator.
NTVAR2	Secondary variable field indicator.

5. I/O VARIABLES:

A. INPUT	LOCATION
ICLEVL	/ME1/
IERR1	F.P.
INUM	F.P.
LDV	F.P.
LEVEL	/ME1/
LITMAX	F.P.
LITNDX	F.P.
LITSTR	F.P.
LLXWRK	/I03/
MMV2	/I04/
MMWRK	/I04/
NDXLIT	F.P.
NNPTB	/I02/
NNPT0	/I02/
NNPT1	/I02/
NNPT2	/I02/
NNV1	/I03/

NNV2	/I03/
NNWRK	/I03/
NNXTB	/I03/
NTALPC	F.P.
NTDLM	F.P.
NTFLP	F.P.
NTINT	F.P.
NTNUL	F.P.
NTOPR	F.P.
NTTBL	F.P.
NTVARD	F.P.
NTVAR0	F.P.
NTVAR1	F.P.
NTVAR2	F.P.
B. OUTPUT	LOCATION
IERR1	F.P.
LLXWRK	/I03/
LOL	F.P.
NDXLIT	F.P.
NNPT2	/I02/
NNWRK	/I03/

6. CALLING ROUTINES: PARSE
LOLFND
7. CALLED ROUTINES: ARAYCK
DVARFD
EEROR
NUMFND

1. NAME: DVSTR - 46
2. PURPOSE: This routine, given a number of defined variables parsed into the work array NNWRK, will take them and transform them into variable identifiers (IIDVs) and store them in the NSET array.
3. METHOD: The variable is first checked to see whether it is a variable or table; if it is neither, an error is printed to that effect, and processing returns to the calling routine. If it is determined that it is a variable, then the variable type and variable number are encoded into an IIDV, and this is stored in the NSET array. If it is determined that it is a table, then the table IIDV is stored in the NSET array. The above is continued until no more defined variables are present in the NNWRK array.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
IERR	Error indicator. If FALSE, then no error has occurred in this subroutine. If TRUE, then a fatal error has occurred in this routine.
IERR1	Error indicator. If FALSE, then no error has occurred up to the present time. If TRUE, then a fatal error has occurred.
IND1	Number of defined variables to store.
IND2	Number of defined variables to store.
IN1	Index that points to location in the work storage array (NNWRK) where the current field being worked on is located.
IN2	Location in NNWRK array where the variable number or table element number is stored.
J	Counter to keep track of how many defined variables have been processed.
LLXSET	Counter for number of locations in current use in NSET array. Also, last position in NSET array used.
MMSET	Maximum dimension of NSET (set in USRDAT).

NNWRK	Work storage array. Contains the current command parsed into the various elements.
NSET	Storage array where the link listed NETWORK is stored.
NTFLP	Floating point number indicator.
NTINT	Integer number indicator.
NTTBL	Table field indicator.
NTVARD	Derivative variable field indicator.
NTVAR0	Independent variable field indicator.
NTVAR1	Primary variable field indicator.
NTVAR2	Secondary variable field indicator.

5. I/O VARIABLES:

A. INPUT	LOCATION
IND1	F.P.
LLXSET	/I03/
MMSET	/I04/
NNWRK	/I03/
NTFLP	F.P.
NTINT	F.P.
NTTBL	F.P.
NTVARD	F.P.
NTVAR0	F.P.
NTVAR1	F.P.
NTVAR2	F.P.
B. OUTPUT	LOCATION
IERR1	F.P.
NSET	/ /

6. CALLING ROUTINE: EVTSTR

7. CALLED ROUTINES: ARAYCK

EEROR

1. NAME: EEOF - 75
2. PURPOSE: This is a machine specific routine. This routine checks the status of a computer file. If that file is at the end of file position, a non-zero value is returned. If the file is at some other position, a zero value is returned.
3. METHOD: This routine calls a machine specific routine. In the case of CDC machines routine EOF is called which returns the status of a file. This is then returned as the value of the function.
4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
EEOF	This contains the status of the file. If zero, the file is at some position other than an end of file. If non-zero, the file is at the end of file position.
ITP	The file number of the file to be checked.

5. I/O VARIABLES:

A. INPUT	LOCATION
ITP	F.P.
B. OUTPUT	LOCATION
EEOF	Function Name
ITP	F.P.

6. CALLING ROUTINES: RSAVIN, SCAN, PACWJ
7. CALLED ROUTINE: EOF (machine specific)

1. NAME: EEPRNT - 90
2. PURPOSE: This routine is used to do PRINT information during the processing of warnings and errors.
3. METHOD: The information to do the PRINT is collected and put together in a form that is usable by the PRINT execution routine PPPRNT.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Do loop index.
IIARC	Current ARC number.
IINXND	Current NODE number.
IIOP	Current OPERATION number.
IOPNO	An array containing the current OPERATION and ARC numbers or the current NODE number. If current processing is on an ARC, then the first word contains the OPERATION number, and the second word is the ARC number. If currently processing a NODE, then the second word is the negative of the NODE number.
NSET	Storage array where the link listed NETWORK is stored.
NV	Number of PRINT variables. This is used by PPPRNT but is set to zero for this case.
T	Array containing in the first word the location in NSET where the variable identifiers are stored. For this routine, there are no variables so this is set to zero. The second word contains the time the PRINT statement was executed.
TNOW	Current time of the IOSYM simulation.
VAL	An array that contains the current values of the PRINT variables. In this case, there are no PRINT variables so this array is set to zeroes.

5. I/O VARIABLES:

A. INPUT	LOCATION
IIARC	/I03/
IINXND	/I03/
IIOP	/I03/
NSET	/ /
TNOW	/GCOM1/
B. OUTPUT	LOCATION
IOPNO	F.P.
NV	F.P.
T	F.P.
VAL	F.P.

6. CALLING ROUTINE: EEROR

7. CALLED ROUTINE: PPRNT

1. NAME: EEROR - 41
2. PURPOSE: This routine processes warning and error messages.
3. METHOD: This routine is called whenever an error or warning message is to be printed. A routine number and an error number are passed to this routine by the calling routine. If the error number is even, then a warning message is printed. If the error number is odd, then an error message is printed. If the error number is negative, then routine EEPRNT is called to do PRINT information. If the error number is fatal, then the fatal error indicator (IERR1) is set TRUE. If the no information diagnostic message indicator (NOINFO) is set TRUE, and if the error is warning, no message is printed; likewise, no PRINT information is done either. If the input card echo command (ECHO) is set FALSE and a NETWORK is being processed, then the card that has the error is found and is printed out before the error or warning message is printed out. If the error number is odd and negative, it indicates the error occurred during an IOSYM simulation so all processing must stop. This is accomplished by doing a square root of a minus one to terminate. IOSYM will attempt to print accumulated statistics before ending the job.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
A	Real variable used to call SQRT function with a minus one.
ECHO	Indicator to show whether input cards read while defining a NETWORK are printed out or ECHOed.
ICLEVL	Array containing the current command hierarchy from level 1 to level LEVEL.
IER	Error number corresponding to the error or warning that has occurred.
IERO	Error number corresponding to the error or warning that has occurred.
IERR1	Error indicator. If FALSE, then no error has occurred up to the present time. If TRUE, then a fatal error has occurred.

IPART	The closest lower or equal even integer to the error number.
ISUB	The routine number of the routine in which the error occurred.
KARDD	The array containing the characters of the data card that is currently being worked on.
NCARD	The card number of the data card currently being worked on.
NCRDR	The input device code number for reading input. (Set in USRDAT.)
NOINFO	Indicator that tells whether or not informative diagnostics (warnings) are to be printed out. If FALSE, the warnings are to be printed; if TRUE, they are not.
NPRNT	The output device code number for writing output. (Set in USRDAT.)

5. I/O VARIABLES:

A.	INPUT	LOCATION
	ECHO	/ME1/
	ICLEVL	/ME1/
	IER	F.P.
	ISUB	F.P.
	KARDD	/ME1/
	NCARD	/ME1/
	NCRDR	/GCOM1/
	NOINFO	/ME1/
	NPRNT	/GCOM1/

B. OUTPUT

LOCATION

IERR1

F.P.

KARDD

/ME1/

NCARD

/ME1/

6. CALLING ROUTINES:

AAIROP
AARCST
ALPFND
ALPSRH
ARAYCK
ARCSTR
BBLIST
BLDALP
BLDDLML
BLDNUM
COLSTR
CONVR2
DDOEVT
DDOSIM
DOSTR
DVFND
DVSTR
EEVAL
EEXP
ENDNET
FF
FNFND
FUNC1
FUNC2
FUNC3
FUNC4
FUNC4A
FUNC5

GBLSTR
GETSTR
GGGET
GGOTO
GGPAR
GGTAB
GGVAR
IIACST
KWFND
LBLCHK
LITFND
LOLFND
LOPSTR
NAMFND
NNODST
NNXNOD
NODSTR
NUMFND
OPRSTR
PARSE
PCKCK
PCKSTR
PLTSTR
PP
PPFCMD
PPIDF
PPPAR

PPPUT
PPRCOL
PPTAB
PPVAR
PROCES
PRTSTR
PUTSTR
RESEQ
RPTSTR
RREPLC
SAVE
SCAN
SSAVE
SSTWRK
SSWARC
STASTR
TBLFND
TBTSTR
TOFSTR
TRCSTR
TRMSTR
TRUTH
TYPSTR
UNCODE
VARFND
VVARBL
WTHSTR

7. CALLED ROUTINES: EEPRNT

OTPUT

UEEROR

1. NAME: EEVAL - 54
2. PURPOSE: This routine takes a Forward Polish Notation (FPN) expression and evaluates and returns the resulting answer to the calling routine.
3. METHOD: A pointer to the NSET array where the FPN expression is located is passed as a formal parameter to this routine. The number of elements in the expression is found from the first word pointed to in the NSET array. Checks are done to make sure there is enough room in the work array NNWRK to do the processing. A scan is then started for operators (these include +, -, *, /, **, =, and all functions).
 - (A) If the operator is a plus, minus, multiple, divide or exponentiation, the two operands that are to be operated on are identified. If these operands are not numbers, then the current value of the variables is determined. Routine AAIROP is then called to perform the operation on the operands. The result is stored, and scanning for the next operator continues.
 - (B) If the operator is an IOSYM function (F, U or P), then these functions are processed. The function number is first identified. The first item in the function is found; this will tell how many arguments there are in the function. All of the arguments are then evaluated if they are not already numbers, and all the values stored in NNWRK. The start of the TABLE references in the function is found. The first element tells how many tables there are to process. Then, the start of the parameter sets are found along with how many of them there are. The number of entities in the function is found along with the starting location identified. Now a determination of which type of function is to be processed. If a probability function is to be processed, then FUNCTION PP is called. If a menu function is to be processed, then FUNCTION FF is called. If a user function is to be processed, then FUNCTION U is called. Once the functions have been performed, a truth condition check is done by calling FUNCTION EEVLTF. The result of the function operation is then stored, and scanning for the next operator continues.
 - (C) If the operator is a FORTRAN library function, then these functions are processed. All of the library functions have either one or two arguments. The first one is identified, and a determination is done to see if there is another one; if so, it is also identified. The exact library function number is identified, and the arguments are evaluated if they are not numbers. Routine AAIROP is called to do the actual processing. The result of the function operation is then stored, and scanning for the next operator continues.
 - (D) If the operator is an equal sign, then the location of the last operand is identified. It is evaluated; if it is not a number, the value is stored in the variable FNC and is returned as a formal parameter to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE

DEFINITION

A

The first operand to be operated on or the first argument in a FORTRAN library function. Example: $A + B = C$ or $\text{MINO}(A,B)$. A in both cases is the first operand passed to routine AAIROP.

B

The second operand to be operated on or the second argument in FORTRAN library functions that have two arguments. Example: $A + B = C$ or $\text{MINO}(A,B)$. B in both cases is the second operand passed to routine AAIROP.

CCONS

This array contains all of the constant values, i.e. numbers. An IOSYM identifier (IIDV) of 8000 to 9000 indicates a number. This number is found by taking the IIDV subtracting 8000 from it, and using the result as a pointer to the CCONS array to find the value of the IIDV.

FNC

This is the resulting value of the expression as calculated by routine EEVAL.

I

Pointer to indicate where in the NSET array current processing of the expression is being done.

IA

The IIDV of the first operand (see definition of A above).

IB

The IIDV of the second operand (see definition of B above).

IERR1

Formal parameter returned from routine EEROR, but not used in this routine.

IFF

F, U or P function number.

II

The IIDV of the resulting answer to the expression, used to find the value of FNC (see definition of FNC above).

IND

The position of the last element in the current expression to be evaluated.

IND1	Pointer to the CCONS array for the value of constants found while processing IOSYM functions.
IN3	Pointer to NNWRK array where operands have been temporarily stored.
I6	Index counter for number of arguments in an IOSYM function.
I7	Pointer to NNWRK where the IIDV of a constant is stored as an argument of an IOSYM function.
LFNC	Pointer to the NSET array where the information to process an expression is located.
LLWRK	Pointer to the NNWRK array where the first available position in NNWRK array is located.
MMFUNF	Number of IOSYM menu functions (type F).
MMSET	Maximum dimension of the NSET array.
MMWRK	Maximum dimension of the NNWRK and WRK arrays.
N	Index used to find locations in NNWRK array where items for IOSYM function processing are located.
NCNS	Temporary storage for the number of constants stored in CCONS array prior to processing of expression. This is used to reset NNCNS after the expression has been evaluated.
NE	Number of entities that an IOSYM function has.
NNCNS	Pointer to the CCONS array where the last position used is located.
NNWRK	Work storage array. The elements in this array from position LLWRK to MMWRK contain temporary storage for intermediate results.

NOP

The operator number of the operator passed to routine AAIRP:

- 1 = exponentiation
- 2 = multiplication
- 3 = division
- 4 = addition
- 5 = subtraction
- 6 = equals sign
- 7 = arc tangent (ATAN)
- 8 = complex (not operational)
- 9 = pick maximum (AMAX1)
- 10 = pick minimum (AMIN1)
- 11 = modulus (AMOD)
- 12 = SIGN
- 13 = absolute value (ABS)
- 14 = arc tangent (ATAN2)
- 15 = cosine (COS)
- 16 = e to the power (EXP)
- 17 = log (ALOG)
- 18 = sine (SIN)
- 19 = square root (SQRT)
- 20 = tangent (TAN)
- 21 = make integer (IFIX)

NP

Number of parameter sets that an IOSYM function has.

NSET

Storage array where the link listed NETWORK is stored. This is from where the expression is retrieved and executed.

NT

Number of TABLEs that an IOSYM function has.

NX

Number of arguments that an IOSYM function has.

N1

Position in the NSET array where the TABLEs for an IOSYM function begin.

N2

Position in the NSET array where the parameter sets for an IOSYM function begin.

N3

Position in the NSET array where the entities for an IOSYM function begin.

N4

Position in the NNWRK array where the result of an IOSYM function is stored.

N5

Position in the NNWRK array where the truth condition for an IOSYM function is stored.

WWRK

Array that is equivalent to the NNWRK array. The values of the arguments for IOSYM functions are stored in this array.

5. I/O VARIABLES:

A. INPUT	LOCATION
A	F.P.
CCONS	/I02/
IERR1	F.P.
LFNC	F.P.
LLWRK	/I03/
MMFUNF	/I04/
MMSET	/I04/
MMWRK	/I04/
NNCNS	/I03/
NNWRK	/I03/
NSET	/ /
WWRK	/I03/
B. OUTPUT	LOCATION
A	F.P.
B	F.P.
CCONS	/I02/
FNC	F.P.
IFF	F.P.
NE	F.P.
NNCNS	/I03/

NNWRK	/I03/
NOP	F.P.
NP	F.P.
NT	F.P.
NX	F.P.
WWRK	/I03/

6. CALLING ROUTINE: AARCST

GGSTAT

NNODST

PPFCMD

STATE

7. CALLED ROUTINES: AAIROP

EEROR

EEVLTF

FF

PP

U

VVARBL

1. NAME: EEVLTF - 524
2. PURPOSE: This routine evaluates the truth condition and returns the result in EEVLTF.
3. METHOD: The expression value and the truth condition code are passed to this function. This routine evaluates whether or not the truth condition is satisfied. If it is satisfied, a one is returned. If it isn't, a zero is returned. If there is no truth condition, then the expression value is returned.
4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
EEVLTF	This contains the result of this function upon returning to the calling routine.
FF	Indicator. If equal to zero, the truth condition wasn't satisfied. If equal to one, the truth condition was satisfied.
FNC	Value of the expression to do truth condition test on.
ITF	Truth condition code number: <ul style="list-style-type: none"> 1 = Less than or equal to zero. 2 = Less than zero. 3 = Equal to zero. 4 = Greater than zero. 5 = Greater than or equal to zero.
PPREC	Precision of the computer being run on; set in USRDAT.

5. I/O VARIABLES:

A. INPUT	LOCATION
FNC	F.P.
ITF	F.P.
PPREC	/I04/
B. OUTPUT	LOCATION
EEVLTF	FUNCTION

6. CALLING ROUTINES: EEVAL
NNODST
PPFCMD

7. CALLED ROUTINES: (None).

1. NAME: EEXP - 511
2. PURPOSE: This routine is a FORTRAN function and evaluates an exponential expression such as X to the Y power. It does both integer and floating point exponentiation.
3. METHOD: If the expression is X to the Y power, the X value is checked to see if it is zero. If so, the function value returned is zero. If X is not zero, then Y is checked to see if it is an integer or not. If Y is an integer number, then integer exponentiation is done. If Y is a floating point number, then floating point exponentiation is done. The resulting value is returned in EEXP.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
EEXP	This contains the result of the expression upon returning to the calling routine.
IERR1	Error indicator returned from routine EEROR but not used in this routine.
PPREC	Precision of the computer being used, set in routine USRDAT.
X	The variable in the expression X to the Y power.
Y	The exponent in the expression X to the Y power.
YI	Integer number, part of the Y exponent.
YY	Floating point number, part of the Y exponent.

5. I/O VARIABLES:

A. INPUT	LOCATION
IERR1	F.P.
PPREC	/I04/
X	F.P.
Y	F.P.

B. OUTPUT**LOCATION****EEXP****FUNCTION**

6. **CALLING ROUTINES:** AAIROP
FF

7. **CALLED ROUTINE:** EEROR

1. NAME: ENDNET - 56
2. PURPOSE: This routine is called after processing a NETWORK set of input cards. This routine initializes some variables before IOSYM executes a simulation. It resequences the ARC PICK commands leaving the NODEs so that they are checked in the proper order when finishing a NODE.
3. METHOD: Event A. A check is first done to see if the COLLECT variables and OPERATIONS have been processed; if so, control skips down to Event B. If not, then a check is done to see if any COLLECT variables have been processed; if not, control skips down to Event B. If some variables have been processed, but no OPERATIONS, then it is assumed that the variables are to be COLLECTed on all OPERATIONS. This initialization then takes place.

Event B. Routine TTAPES is called to write all of the GASP information to tape.

Event C. A check is then made to make sure all of the OPERATIONS are defined. If an OPERATION is not defined, then it is assumed to be a dummy OPERATION with no statements and is so initialized.

Event D. The sequence in which ARCs are searched to see which one should be taken when leaving a NODE is examined, and resequenced if necessary according to the PICK commands. Each NODE is studied individually. If a NODE has not been defined, it is assumed to be a dummy NODE and is so initialized. The first PICK is identified; if there is not one, then no further processing of this NODE is necessary, and the next NODE is investigated. If a valid PICK was found, then the position of this PICK in the NSET array is stored in the first available position in the first word of array INNOD. The location of the pointer pointing to this PICK is stored in the second word of array INNOD. The next PICK is identified, and the sequence is repeated. Once all of the PICKs have been found and stored, a check is made to make sure that they are the correct number; if not, an error message is printed.

Now the resequencing of the PICKs is started. The first pointer in the INNOD array is removed and stored in IA. The second pointer to the PICKs is removed and stored in IB. If PICK IA comes after PICK IB, the sequence is repeated with PICK IB becoming PICK IA, and a new pointer removed and stored in IB. If IB comes after IA, the pointers are exchanged, and a new PICK IB is removed from the INNOD, and the sequence is repeated. This is continued until all of the PICKs have been examined once, and the smallest PICK number is stored in the last position in array INNOD. This PICK pointer is then removed from the INNOD array and put back in the proper place in the NSET array. This is repeated until all of the PICK pointers have been put back into the NSET array in correct order. This is done for each NODE; then control returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
FIR2	Indicator to show whether an ARC leaving the given NODE has already taken place. If FIR2(I) is TRUE, then an ARC leaving NODE I has not been processed. If FIR2(I) is FALSE, then an ARC leaving NODE I has been processed.
I	Index used for the pointer to the NNPTOP array (OPERATIONS) and the NNPTND array (NODEs).
IA	A PICK command PICK number.
IB	A PICK command PICK number.
IERR	Error indicator. If TRUE, a fatal error has occurred in this routine. If FALSE, a fatal error hasn't occurred.
IERR1	Error indicator. If TRUE, a fatal error has occurred. If FALSE, a fatal error hasn't occurred up to the present time.
IIJ	Location in NSET array where the pointer to the first PICK from a NODE is stored.
IJK	Number of PICK commands for a given NODE.
IND	Position in the NSET array where a PICK command starts.
INDX11	Location in the NSET array where the number of ARCs leaving a NODE is stored.
INDX12	Location in the NSET array where the pointer to the first PICK for a NODE is stored.
INNOD	Array that contains in the first word a pointer to the PICK command, and the second word is a pointer to the NSET array where the pointer to the PICK command in the first word is located.

INP	Position in the NSET array where the current NODE uniform command sequence (UCS) starts.
INY	Is the number of PICK commands leaving the current NODE.
INZ	Is the pointer in the NSET array that points to the start of the next PICK.
INJ	Pointer to the location in the NSET array where the ARC UCS begins.
IPIK	Location in the NSET array where the pointer to the next PICK command is stored.
J	Pointer and counter for INNOD array.
JJ	Pointer and counter for INNOD array.
K	Index used for pointer to NNPTAC array (ARC).
LLXSET	Counter of the number of locations in current use in NSET array. Also, last position in NSET used.
LSET	Temporary storage for number of NSET array elements needed.
MMSET	Maximum dimension of the NSET array.
NNARC	Number of defined ARCs. Used as a counter for the NNPTAC array.
NNCOL	The number of COLLECT variables that have been processed on a COLLECT command.
NNCOP	The number of OPERATIONS that the COLLECT variables are to be collected on.
NNNODE	Number of defined NODEs. Used as a counter for the NNPTND array.
NNOP	Number of defined OPERATIONS. Used as a counter for the NNPTOP array.

NNPTAC

The first two words of this array contain the ARC name, and the third contains a pointer to the location in the NSET array where the ARC UCS begins.

NNPTND

The first two words of this array contain the NODE name, and the third word contains a pointer to the location in the NSET array where the NODE UCS begins.

NNPTOP

The first two words of this array contain the OPERATION name. The third, fourth, and fifth words contain the TYPE S, I, E commands that occur under the OPERATION respectively. The sixth word contains an indicator to show whether the COLLECT variables are COLLECTED on this OPERATION. If this is a positive number, then the variables are COLLECTED; if less than or equal to zero, they are not.

NSET

Storage array where the link listed NETWORK is stored.

5. I/O VARIABLES:

A. INPUT	LOCATION
IERR1	F.P.
LLXSET	/I03/
MMSET	/I04/
NNARC	/I03/
NNCOL	/I01/
NNCOP	/I03/
NNNODE	/I03/
NNOP	/I03/
NNPTAC	/I02/
NNPTND	/I02/

	NNPTOP	/I02/
	NSET	/ /
B.	OUTPUT	LOCATION
	IERR1	F.P.
	LLXSET	/I03/
	NNCOP	/I03/
	NNPTND	/I02/
	NNPTOP	/I02/
	NSET	/ /

6. CALLING ROUTINE: PROCES

7. CALLED ROUTINES: ARAYCK

EEROR

TTAPES

1. NAME: EVNTS - 532
2. PURPOSE: This routine processes all IOSYM state events, TERM events, and user events.
3. METHOD: If an IOSYM state event is identified, a CROSS event is to be processed. The user routine UEVNTS is called so the user can process any event. The CROSS command is identified; the TYPE E commands are performed; any auxiliary or user actions are performed; statistics are collected; the CROSS commands are cleared; any TERM commands are cleared, and the next step is setup. NNODST is called to do NODE, or AARCST is called to do ARC processing. Control returns to the calling routine.

If an IOSYM TERM event is identified, the TYPE E commands, any auxiliary command, and user actions are performed; statistics are collected; the CROSS and TERM commands are cleared from the work storage array; and processing for the next step is setup. Control then returns to the calling routine.

If a user event is identified, routine UEVNTS is called. Then, depending on what type of event was processed, the TYPE E command, auxiliary command, and user commands are performed; statistics are collected; TERM and CROSS commands cleared; and setup for the next step is done. Or, control returns to the calling routine for more processing on current step.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
ATRI	Array containing the attributes of the event with the third word containing the TERM pointer.
I	Index and pointer used for various purposes.
IEND	Indicator. If equal to zero, this event doesn't end the current ARC processing. If any other value, the current ARC is ended.
IIARC	Current ARC number. If equal to zero, then there is no current ARC.
IINXAC	Number of the next ARC. If equal to zero, then an ARC isn't processed next.

IVT	Event code number. If equal to one, an IOSYM state event is being processed. If equal to two, an IOSYM TERM is being processed. If greater than two, a user event is being processed.
IX	Event code number.
LFLAG	State condition flag (see GASP manual for more explanation).
LLCROS	Location in the work array that holds the pointer to the uniform command sequence for the first CROSS command on the current ARC.
LLIST	Location in the work array where the list of TYPE E commands on the current ARC start.
LSTR	Pointer to various locations in the work array.
LUCS	Pointer and index used for various purposes.
MSTOP	Indicator for specifying the method of ending the simulation (see GASP manual for further description).
NCROS	Number of CROSS commands on the current ARC plus one when processing a state event. It equals zero when not processing a state event.
NNCROS	Number of CROSS commands on the current ARC.
NNEQS	Number of state variables defined by state equations (see GASP book for more description).
NNEQT	Largest subscript used for state variables. Number of state, derivative, and CROSS equations.
NNWRK	Work storage array. Contains the pointer to the CROSS command if processing a state event.

TNOW

Current time of simulation.

VVAR2

Array containing in the first word the current values of all the secondary variables; the second word contains the initial value of all the secondary variables.

5. I/O VARIABLES:

A. INPUT	LOCATION
ATRIB	/GCOM1/
IEND	F.P.
IINXAC	/I03/
IX	F.P.
LFLAG	/GCOM2/
LLCROS	/I03/
LLIST	/I03/
MSTOP	/GCOM1/
NNCROS	/I03/
NNEQS	/GCOM2/
NNEQT	/GCOM2/
NNWRK	/I03/
TNOW	/GCOM1/
VVAR2	/I02/
B. OUTPUT	LOCATION
IEND	F.P.
IIARC	/I03/
NNCROS	/I03/
NNEQS	/GCOM2/
NNEQT	/GCOM2/
VVAR2	/I02/

6. CALLING ROUTINE: GASP (main routine in the GASP program).

7. CALLED ROUTINES:

AARCST
BBLIST
CCOLC
DDOCMD
DDOLST
NFIN (GASP routine)
NNODST
RMOVE (GASP routine)
RREWRK
TTABU
UEVNTS (user written routine)
USTAT (user written routine)
UTERM (user written routine)

1. NAME: EVTSTR - 47
2. PURPOSE: This routine processes EVENT commands.
3. METHOD: The information stored in the NNWRK array for the event command is examined. The EVENT command number is stored in the NSET array, and a check is done to see if another command points to the EVENT command. If so, the pointer for it is set up. Next to EVENT number, the number of parameters and the parameters themselves are stored in the NSET array. Last a location in the NSET array is set aside to point to the next command.
4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
CCONS	Array contains the values of all the IOSYM constants.
IERR	Error indicator. If FALSE, then no error has occurred in this subroutine. If TRUE, then a fatal error has occurred in this routine.
IERR1	Error indicator. If FALSE, then no error has occurred up to the current time. If TRUE, then a fatal error has occurred.
IN1	Pointer to the location in the CCONS array where the EVENT number is located.
IPTCMD	Pointer to the next command. If IPTCMD is zero, then no command points to the current command. If IPTCMD is greater than zero, then the pointer to the next command is stored in NSET(IPTCMD).
LLXSET	Counter of the number of locations in current use in the NSET array. It is also the last position in the NSET array used.
LLXWRK	Last position in the NNWRK array that is used at current time.
LSET	Temporary storage for the number of NSET elements needed.
MMSET	Maximum dimension of the NSET array.

NNWRK	Work storage array. Contains the current command parsed into the various elements.
NSET	Storage array where the link listed NETWORK is stored.
NTFLP	Floating point number field indicator.
NTINT	Integer field indicator.
NTNUL	Null field indicator.
NTTBL	Table field indicator.
NTVARD	Derivative variable field indicator.
NTVAR0	Independent variable field indicator.
NTVAR1	Primary variable field indicator.
NTVAR2	Secondary variable field indicator.

5. I/O VARIABLES:

A. INPUT	LOCATION
CCONS	/I02/
IERR	F.P.
IPTCMD	F.P.
LLXSET	/I03/
LLXWRK	/I03/
MMSET	/I04/
NNWRK	/I03/
NTFLP	F.P.
NTINT	F.P.
NTNUL	F.P.
NTTBL	F.P.

	NTVARD	F.P.
	NTVAR0	F.P.
	NTVAR1	F.P.
	NTVAR2	F.P.
B.	OUTPUT	LOCATION
	IERR1	F.P.
	IPTCMD	F.P.
	LLXSET	/I03/
	LSET	F.P.
	NSET	/ /

6. CALLING ROUTINE: PROCES

7. CALLED ROUTINES:

ARRAYCK

DVSTR

1. NAME: FF - 512
2. PURPOSE: This routine processes all of the IOSYM menu functions.
3. METHOD: Depending on which menu function is to be processed, different areas of the routine are executed. There are twenty-nine functions, each of which has its own section of programming. Once the function has been evaluated, its value is returned to the calling routine as FF.
4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
FF	This contains the result of the function upon returning to the calling routine.
I	Index and pointer used for various purposes.
IAC	Array containing all of the fields in the entities of the function.
IERR1	Error indicator returned from the routine EEROR but not used in this routine.
IFF	Function number of the menu function to process.
ITAB	Contains the TABLE number of the current TABLE being processed. If equal to zero, then a TABLE with all ones is assumed.
ITB	TABLE number currently working on.
ITBB	Array containing the current TABLE being worked on.
IX	Pointer to the current argument in the argument list of a function.
I1 I2	Pointers to different elements in a TABLE.
J	Index and counter used for various purposes.

K	Pointer to various TABLEs and TABLE elements.
KGO	Variable that has been ASSIGNED various statement labels to go to.
LEN	Length of a TABLE being processed.
MMFUNF	Number of IOSYM menu functions.
NA	Number of entities in a function.
NN	Integer variable used for various purposes.
NNPTB	Array that contains in the first two words the TABLE names, and the third word points to the location in the TTABL array where the TABLE elements are stored.
NT	Number of TABLEs in a function.
NTT	Number of TABLEs in a function.
NT1 NT2	Lengths of certain TABLEs.
NX	Number of arguments in the arguments section of a function.
PPREC	Precision of the computer being run on, set in routine USRDAT.
SIG	Indicates the sign that is supposed to be used in the calculations.
TEMP TEMP2 TEMP3	Variable used to store various intermediate results.
X	Array that contains all of the values of the arguments in the arguments section of a function.
Z	Variable used to store various intermediate results.

5. I/O VARIABLES:

A. INPUT	LOCATION
IAC	F.P.
IERR1	F.P.
IFF	F.P.
ITAB	F.P.
MMFUNF	/I04/
NA	F.P.
NNPTB	/I02/
NT	F.P.
NX	F.P.
PPREC	/I04/
X	F.P.
B. OUTPUT	LOCATION
FF	FUNCTION

6. CALLING ROUTINE: EEVAL

7. CALLED ROUTINES: EEROR
EEXP
GGTAB
PPIDF
PPTAB

1. NAME: FFAJ
2. PURPOSE: This routine will read an alphanumeric string in free form input.
3. METHOD: Routine PACWJ is called to build the free form input into the WORDJ array. The WORDJ array is written out and read into the LALPH word as A4.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
IITPS	Array containing the device code numbers. IITPS(5) is the device code number for the scratch tape.
ITP	Device code number for the scratch tape.
LALPH	Free form alphanumeric string.
WORDJ	Packed variable array that contains the free form input word.

5. I/O VARIABLES:

A. INPUT	LOCATION
IITPS	/I01/
WORDJ	/FFJ/
B. OUTPUT	LOCATION
LALPH	F. P.

6. CALLING ROUTINE: DDOSIM

7. CALLED ROUTINE: PACWJ

1. NAME: FFDJ
2. PURPOSE: This routine will read a double precision value in free form input.
3. METHOD: Routine PACWJ is called to build the free form input into the WORDJ array. The WORDJ array is written out and read into variable DVAR as D20.0.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
DVAR	Free form variable value.
IITPS	Array containing the device code numbers. IITPS(5) is the device code number for the scratch tape.
ITP	Device code number for scratch tape.
WORDJ	Packed variable array that contains the free form input word.

5. I/O VARIABLES:

A. INPUT	LOCATION
IITPS	/I01/
WORDJ	/FFJ/
B. OUTPUT	LOCATION
DVAR	F. P.

6. CALLING ROUTINE: (None)

7. CALLED ROUTINE: PACWJ

1. NAME: FFIJ
2. PURPOSE: This routine will read an integer value in free form input.
3. METHOD: Routine PACWJ is called to build the free form input into the WORDJ Array. The WORDJ array is written out and read into variable IVAR as I20.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
IITPS	Array containing the device code numbers. IITPS(5) is the device code number for the scratch tape.
ITP	Device code number for scratch tape.
IVAR	Free form variable value.
WORDJ	Packed variable array that contains the free form input word.

5. I/O VARIABLES:

A. INPUT	LOCATION
IITPS	/I01/
WORDJ	/FFJ/
B. OUTPUT	LOCATION
IVAR	F. P.

6. CALLING ROUTINE:

7. CALLED ROUTINE: PACWJ

1. NAME: FFLJ
2. PURPOSE: This routine will read a logical value in free form input.
3. METHOD: Routine PACWJ is called to build the free form input into the WORDJ array. The WORDJ array is written out and read into variable TF as L20.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
IITPS	Array containing the device code numbers. IITPS(5) is the device code number for the scratch tape.
ITP	Device code number for the scratch tape.
TF	Free form logical value.
WORDJ	Packed variable array that contains the free form input word.

5. I/O VARIABLES:

A. INPUT	LOCATION
IITPS	/I01/
WORDJ	/FFJ/
B. OUTPUT	LOCATION
TF	F. P.

6. CALLING ROUTINE: (None)

7. CALLED ROUTINE: PACWJ

1. NAME: FFPUDJ
2. PURPOSE: This routine clears the free form input buffer.
3. METHOD: Variables IFFJ and NFFJ are set to one and zero, respectively.
4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
IFFJ	Location in the LWJ array where the first non-blank character is.
NFFJ	Location in the LWJ array where the last non-blank character is.

5. I/O VARIABLES:

A. INPUT	LOCATION
(None)	
B. OUTPUT	LOCATION
IFFJ	/FFJ/
NFFJ	/FFJ/

6. CALLING ROUTINES: DDOSIM
FFRJ
7. CALLED ROUTINES: (None)

1. NAME: FFRJ
2. PURPOSE: This routine will read a real value in free form input.
3. METHOD: Routine PACWJ is called to build the free form input into the WORDJ array. The WORDJ and is written out and read into variable VAR as G20.0.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
IITPS	Array containing the device code numbers. IITPS(5) is the device code number for the scratch tape.
ITP	Device code number for the scratch tape.
VAR	Free form variable value.
WORDJ	Packed variable array that contains the free form input word.

5. I/O VARIABLES:

A. INPUT	LOCATION
IITPS	/I01/
WORDJ	/FFJ/
B. OUTPUT	LOCATION
VAR	F. P.

6. CALLING ROUTINES: DDOSIM
7. CALLING ROUTINE: PACWJ

1. NAME: FNFND - 4
2. PURPOSE: This routine processes an IOSYM function found in the literal storage array and stores it in the proper format to be later executed.
3. METHOD: A search is done until a valid function field is found. The specific function number and type are coded and stored in the work storage array. Routine FUNC1 is called to define the positions in the literal storage array where the function is stored. Routine FUNC2 is then called to process any arguments in the function. If at this time the end of the function has been reached, default values are stored for all the other fields. If not, routine FUNC3 is called to process any TABLEs in the function. If the end of the function has been reached, default values are stored for all the other fields. If not at the end, routine FUNC4 is called to process the parameter sets in the function. If the end of the function has been reached this time, default values are stored for all other fields. If not at the end, routine FUNC4A is called to process any entities in the function. If at the end of the function now, then the default truth condition is stored. If not at the end of the function, routine FUNC5 is called to process the truth condition. Control is then returned to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
IEND1	Indicator. If TRUE, then the end of the function has been reached. If FALSE, then there are still more fields in the function to process.
IERR	Error indicator. If FALSE, then no error has occurred while processing this routine. If TRUE, then a fatal error has occurred in this routine.
IERR1	Error indicator. If FALSE, then no error has occurred up to the present time. If TRUE, then a fatal error has occurred.
IERR3	Error indicator. If FALSE, then the function limits of storage in the literal storage array have been defined. If TRUE, then a closing parenthesis which defines the end of the function wasn't found.

IVAL	<p>The function type and number are coded into this word.</p> <p>If IVAL is negative, then the function is a probability function, and the function number is the negative of IVAL. If IVAL is positive and less than or equal to MMFUNF, then the function is a menu function, and the function number is IVAL.</p> <p>If IVAL is greater than MMFUNF, then the function is a user function and the function number is IVAL-MMFUNF.</p>
LITMAX	Maximum length of the literal storage array LITSTR.
LITNDX	Maximum defined literal storage in array LITSTR.
LITSTR	Array where the literals are stored prior to being used by these routines to build a function.
LLXWRK	Indicates the last defined location in the NNWRK array.
MMFUNF	Number of IOSYM menu functions (type F functions).
MMWRK	Maximum dimension of the NNWRK array.
NDXLIT	Pointer to current position in the literal storage array LITSTR.
NDX7	Location in the LITSTR where the end of the function is located.
NNWRK	Work storage array. Contains the current command parsed into the various elements.
NTALPC	Alpha field indicator.
NTDLM	Delimiter field indicator.
NTFLP	Floating point number field indicator.
NTFUNF	Menu function field indicator.
NTFUNP	Probability function field indicator.

NTFUNU	User function field indicator.
NTINT	Integer number field indicator.
NTNUL	Null field indicator.
NTOPR	Operator field indicator.
NTTBL	Table field indicator.
NTVARO	Derivative variable field indicator.
NTVAR1	Independent variable field indicator.
NTVAR2	Primary variable field indicator.
	Secondary variable field indicator.

5. I/O VARIABLES:

A. INPUT	LOCATION
IEND1	F.P.
IERR	F.P.
IERR1	F.P.
IERR3	F.P.
LITMAX	F.P.
LITNOX	F.P.
LITSTR	F.P.
LLXWRK	/I03/
MMFUNF	/I04/
MMWRK	/I04/
NDXLIT	F.P.
NDX7	F.P.
NTALPC	F.P.
NTDLM	F.P.
NTFLP	F.P.

	NTFUNF	F.P.
	NTFUNP	F.P.
	NTFUNU	F.P.
	NTINT	F.P.
	NTNUL	F.P.
	NTOPR	F.P.
	NTTBL	F.P.
	NTVARD	F.P.
	NTVAR0	F.P.
	NTVAR1	F.P.
	NTVAR2	F.P.
B.	OUTPUT	LOCATION
	IEND1	F.P.
	IERR	F.P.
	IERR1	F.P.
	NDXLIT	F.P.
	NNWRK	/I03/

6. CALLING ROUTINES: LOLFND
PARSE

7. CALLED ROUTINES: ARAYCK
EEROR
FUNC1
FUNC2
FUNC3
FUNC4
FUNC4A
FUNC5

1. NAME: FORSTR - 52
2. PURPOSE: This routine processes the FOR and NOFOR commands into IOSYM nomenclature for later execution by IOSYM during a simulation.
3. METHOD:

A) FOR commands

The OPERATIONS are all set to no COLLECT status by setting the sixth word, in the OPERATION storage array, for each OPERATION to zero. Then the number of OPERATIONS needed to do the COLLECTing is determined from the work array (NNWRK). Next the OPERATIONS that were identified on the FOR command are set to COLLECT status by setting the sixth word of the appropriate OPERATION (in NNPTOP array) to the number corresponding to its order on the FOR command. The number of OPERATIONS that are going to be COLLECTed is stored in NNCOP, and control is returned to the calling routine.

B) NOFOR commands

A check is done to see if there are any OPERATIONS to delete from COLLECT status. If there are no OPERATIONS listed on the NOFOR command, then all OPERATIONS are to be deleted. In this case, all the OPERATIONS are set to no COLLECT status by setting the sixth word of each OPERATION to zero (NNPTOP array). If there are OPERATIONS to delete, then the OPERATION array is searched for each OPERATION on the NOFOR command and the sixth word (in array NNPTOP) for these OPERATIONS is set to zero. As each OPERATION is set to no COLLECT status, the other OPERATIONS that have COLLECT status are resequenced to fill in the blanks left by the deleted OPERATION. After all the given OPERATIONS are deleted, control returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Index pointer and counter used to identify which location in the array is being processed.
IN1	The OPERATION number of the current OPERATION.
IN2	Position in NNWRK array where the last OPERATION to give COLLECT status is located.
ISUB	Indicator. If less than or equal to zero, then the current command is NOFOR. If greater than zero, then the current command is FOR.

J	Index pointer and counter used to identify which location in an array is being processed.
K	Index pointer and counter used to identify which location in an array is being processed.
LLXWRK	Indicates the last defined location in the NNWRK array.
NNCOP	The number of COLLECT OPERATIONS that statistics for COLLECT variables will be done.
NNOP	The number of OPERATIONS in the current NETWORK.
NNPTOP	Array that contains the names and pointers for the OPERATIONS. The first word is the first four characters of the OPERATION name stored in IOSYM notation. The second word is the next three characters of the name. The third, forth, and fifth words point to the NSET array where the various commands to be performed on the OPERATIONS are located. The sixth word, if less than or equal to zero, indicates that that OPERATION is not used to COLLECT statistics for the COLLECT command. If the sixth word is greater than zero, then that number indicates in which order the OPERATIONS are used to do the COLLECT statistics.
NNWRK	Work storage array. Contains the current command parsed into the various elements.

5. I/O VARIABLES:

A.	INPUT	LOCATION
	ISUB	F.P.
	LLXWRK	/I03/
	NNCOP	/I03/

NNOP /I03/

NNPTOP /I02/

NNWRK /I03/

B. OUTPUT LOCATION

NNCOP /I03/

NNPTOP /I02/

6. CALLING ROUTINE: PROCES

7. CALLED ROUTINE: (None)

1. NAME: FUNC1 - 5
2. PURPOSE: This routine searches the literal storage array for the closing parenthesis that ends an IOSYM function.
3. METHOD: The LITSTR array is searched from the current position (the beginning of the function) until it finds the matching closing parenthesis. If an open parenthesis is found, it is added to the open parenthesis stack; if a close parenthesis is found, it is added to the closed parenthesis stack. When the stacks are equal (ICLOSE=IOPEN), then the closing parenthesis for the function has been found and processing returns to the calling routine. If the end of the LITSTR array is found before the matching closing parenthesis is found, an error is printed, and processing returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
ICLOSE	Counts the number of closing parenthesis found.
IEND1	Indicator. If TRUE, then the end of the function has been reached. If FALSE, then there are still more fields in the function to process.
IERR1	Error indicator. If FALSE, then no error has occurred up to the present time. If TRUE, then a fatal error has occurred.
IERR3	Error indicator. If FALSE, then the function limits of storage in the literal storage array have been defined. If TRUE, then the closing parenthesis which defines the end of the function wasn't found.
IOPEN	The number of open parenthesis found.
LITMAX	Maximum length of the literal storage array LITSTR.
LITNDX	Position of the last defined location in the literal storage array LITSTR.
LITSTR	Array where the literals are stored prior to being used to build a function.

NDXLIT

Pointer to current position in the
literal storage array LITSTR.

NDX7

The position in the literal storage
array where the last element for the
function is located.

NTDLM

Delimiter field indicator.

5. I/O VARIABLES

A. INPUT

LOCATION

LITMAX

F.P.

LITNDX

F.P.

LITSTR

F.P.

NDXLIT

F.P.

NTDLM

F.P.

B. OUTPUT

LOCATION

IEND1

F.P.

IERR1

F.P.

IERR3

F.P.

NDX7 F.P.

6. CALLING ROUTINE: FNFND

7. CALLED ROUTINE: EEROR

1. NAME: FUNC2 - 10
2. PURPOSE: This routine builds the argument list for IOSYM functions. By argument, this does not imply that it processes all the fields in the function but just the fields termed "arguments" in the IOSYM user manual.
3. METHOD: This routine starts at the beginning of the function to build an argument list. Each argument may consist of more than one field. These fields must be built into an argument and then counted as only one item in the argument list. A check is done to see what each field is, and processing continues according to that field type.

If the field is a null field, then the number of arguments processed is zero, and this is stored in the work array and control returns to the calling routine to continue processing the other fields in the function.

If the field is an integer or floating point number, a check is done to see if an argument is currently being built. If an argument is not being built, an indicator is set, and the argument counter is increased by one. The number of words in that argument is initialized to two (because each field is defined by two words). If an argument is being built, then the number of words in that argument is increased by two. Then control returns to check the next field.

If the field is an alpha field, a check is done to see what type of variable or TABLE it is. If it is a variable or single table element, a check is done to see if an argument is being built. If so, then the number of words for that argument is increased by two. If not building an argument, one is started. After processing the argument, control returns to check the next field. If the field is just a TABLE name with no element reference, then we are through processing the arguments of the function and have found the first TABLE. If it was a TABLE, then a check is done to see if the last field before it was a slash; if so, it is deleted from the last argument. Control then skips out of checking the fields and goes to store the arguments (this will be discussed later). If the field wasn't a defined variable, TABLE element or a TABLE name, then it is assumed to be a secondary variable and is so defined and processed.

If the field is a delimiter, a check is done to see if it is an open or close parenthesis; if so, then it is part of the current argument and is included by increasing the number of words in that argument by two. If an argument isn't being processed, then the start of a new argument is assumed and processing continues. If the delimiter isn't an open or close parenthesis, then the building of the current argument is ended. Control then returns to check the next field.

If the field is an operator, a check is done to see if the last two fields have been slashes. If so, then we are through processing arguments and control skips out of the checking field

loop and goes to store the arguments. If the last two fields haven't been slashes, then a check is done to see if an argument is being built; if so, the number of fields in that argument is increased by two; if not, then a check is done to see if the operator is a slash. If so, then we are through processing the arguments and control skips out of checking fields and goes to store the arguments. If the operator isn't a slash, then it must be an error, and an error message is printed.

To store the arguments the following is done. First the number of arguments is stored in the work array. If the number of arguments is zero, then control returns to the calling routine. If not, the arguments are then stored. First the number of fields in the argument is stored, and then the fields for that argument are stored as IOSYM identifiers. This is repeated for each of the arguments. Control then returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Index pointer and counter.
IARG	Number of arguments in the function that have been processed.
IEND1	Indicator. If TRUE, then the end of the function has been reached. If FALSE, then there are still more fields in the function to process.
IERR	Error indicator. If TRUE, then a fatal error has occurred in this routine. If FALSE, a fatal error hasn't.
IERR1	Error indicator. If TRUE, then a fatal error has occurred. If FALSE, a fatal error has occurred up to the present time.
IIARG	Indicator. If TRUE, then are currently building an argument with a number of fields. If FALSE, then are not building an argument.
IN1	Index used to point to the literal storage array where the field being stored is located, before storing in the work array.
IN2	Index that points to the location in the literal storage array where a table element is located.

IN3	The number of the TABLE currently being stored.
IWORD	Is the number of fields in the current argument.
LITMAX	Maximum length of the literal storage array LITSTR.
LITSTR	Array where the literals are stored prior to being used to build a function.
LLXWRK	Indicates the last defined location in the NNWRK array.
LXWRK	Temporary counter for the number of locations needed in the NNWRK array.
MMV2	Maximum number of secondary variables that are allowed (dimension of NNPT2 and VVAR2 arrays).
MMWRK	Maximum number of locations in the NNWRK array.
NDXLIT	Pointer to current position in the literal storage array.
NDX10	Counter to keep track of which field in the argument is being stored at the current time.
NDX7	Position in the literal storage array where the current function ends.
NDX8	Current position in the literal storage array that is being processed as a possible argument in the current function.
NDX9	Array index for processing different variables.
NNPTB	Array that contains in the first two words the TABLE names and the third word points to the location in the ITABL array where the TABLE values are stored.
NNPT0	Array that contains the independent variable name.

NNPT1	Array that contains in the first word the first four characters of the primary variable name, and the second word contains the next three characters of the name.
NNPT2	Array that contains in the first word the first four characters of the secondary variable name, and the second word contains the next three characters.
NNV1	Number of primary variables.
NNV2	Number of secondary variables.
NNWRK	Work storage array. Contains the current command parsed into the various elements.
NNXTB	Number of defined TABLEs.
NTALPC	Alpha field indicator.
NTDLM	Delimiter field indicator.
NTFLP	Floating point number field indicator.
NTINT	Integer number field indicator.
NTNUL	Null field indicator.
NTOPR	Operator field indicator.
NTTBL	Table field indicator.
NTVARD	Derivative variable field indicator.
NTVAR0	Independent variable field indicator.
NTVAR1	Primary variable field indicator.
NTVAR2	Secondary variable field indicator.
NWORD	Contains the number of words in each argument.

5. I/O VARIABLES:

A. INPUT	LOCATION
LITMAX	F.P.

LITSTR	F.P.
LLXWRK	/I03/
MMV2	/I04/
MMWRK	/I04/
NDXLIT	F.P.
NDX7	F.P.
NNPTB	/I02/
NNPT0	/I02/
NNPT1	/I02/
NNPT2	/I02/
NNV1	/I03/
NNV2	/I03/
NNWRK	/I03/
NNXTB	/I03/
NTALPC	F.P.
NTDLM	F.P.
NTFLP	F.P.
NTINT	F.P.
NTNUL	F.P.
NTOPR	F.P.
NTTBL	F.P.
NTVARD	F.P.
NTVAR0	F.P.
NTVAR1	F.P.
NTVAR2	F.P.

B.	OUTPUT	LOCATION
	IEND1	F.P.
	IERR1	F.P.
	LITSTR	F.P.
	LLXWRK	/I03/
	NDXLIT	F.P.
	NNPT2	/I02/
	NNWRK	/I03/

6. CALLING ROUTINE: FNFND

7. CALLED ROUTINES: ARAYCK

DVARFD

EEROR

NUMSTR

1. NAME: FUNC3 - 11
2. PURPOSE: This routine processes the TABLEs in an IOSYM function.
3. METHOD: This routine starts at the first field after processing the function arguments. It looks for the TABLEs and stores them. It does this by looking at each field.

If the field is a null field, then the null field is reset to be a default TABLE by setting the TABLE number to zero. The number of TABLEs found is incremented by one, and the next field is searched.

If the field is an alpha field, a search is done to see if it is a TABLE; if not, an error is processed and control returns to the calling routine. If it is a TABLE, the field indicator is changed to a TABLE, and the number of TABLEs processed on this function is increased by one. The next field is then searched.

If the field is an operator, that indicates that the processing of the TABLEs on the function is finished, and control skips down to store the TABLEs.

If the field isn't an operator, delimiter, alpha or null field, an error is processed, and control returns to the calling routine.

If the end of the function occurs, then an indicator is set, and control proceeds to store the TABLEs.

To store the TABLEs, first the number of TABLEs processed is stored. If the number of TABLEs was zero, then control returns to the calling routine. If there was at least one TABLE, they are stored in the work array, and then control returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Pointer to the current TABLE in the NNPTB array being investigated.
IEND1	Indicator. If TRUE, then the end of the function has been reached. If FALSE, then there are still more fields in the function to process.
IERR	Error indicator. If FALSE, then no error has occurred in this routine. If TRUE, then a fatal error has occurred in this routine.
IERR1	Error indicator. If FALSE, then no error has occurred up to the present time. If TRUE, then a fatal error has occurred.

IN1	Pointer to the current location in the literal storage array of the field being processed.
ITAB	Number of TABLEs that have been processed.
LITMAX	Maximum length of the literal storage array LITSTR.
LITSTR	Array where the literals are stored prior to being used to build a function.
LLXWRK	Indicates the last defined location in the NNWRK array.
LWRK	Temporary storage for the number of work storage locations needed.
MMWRK	Maximum number of locations in the NNWRK array.
NDXLIT	Pointer to the current position in the literal storage array.
NDX7	Position in the literal storage array where the current function ends.
NDX8	Current position in the literal storage array that is being processed as a possible TABLE field in the current function.
NDX9	Array index for processing different TABLEs.
NNPTB	Array that contains in the first two words the IOSYM representation of the TABLE names. The third word is the location in TTABL array where the TABLE elements are stored.
NNWRK	Work storage array. Contains the current command parsed into the various elements.
NNXTB	Number of defined TABLEs.
NTALPC	Alpha field indicator.
NTDLM	Delimiter field indicator.

NTNUL	Null field indicator.
NTOPR	Operator field indicator.
NTTBL	Table field indicator.

5. I/O VARIABLES:

A. INPUT	LOCATION
LITMAX	F.P.
LITSTR	F.P.
LLXWRK	/I03/
MMWRK	/I04/
NDXLIT	F.P.
NDX7	F.P.
NNPTB	/I02/
NNXTB	/I03/
NTALPC	F.P.
NTDLM	F.P.
NTNUL	F.P.
NTOPR	F.P.
NTTBL	F.P.
B. OUTPUT	LOCATION
IEND1	F.P.
IERR1	F.P.
LITSTR	F.P.
LLXWRK	/I03/
NDXLIT	F.P.
NNWRK	/I03/

6. CALLING ROUTINE: FNFND

7. CALLED ROUTINES: ARAYCK

EEROR

1. NAME: FUNC4 - 6
2. PURPOSE: This routine processes the parameter sets in an IOSYM function.
3. METHOD: This routine starts at the first field after processing the function TABLEs. It looks for valid parameter sets and stores them. It does this by looking at each field in the literal storage array.
 If the field is a null field, then the null field is taken to be a null parameter set. The number of parameter sets is increased by one. Control then continues looking at each field.
 If the field is an integer number, the number of parameter sets is increased by one. Control then continues looking at each field.
 If the field is a delimiter, it is skipped, and control continues looking at each field.
 If the field is an operator, the end of the parameter sets has been reached, and control goes to store the parameter sets.
 If the end of the function occurs, then an indicator is set, and control proceeds to store the parameter sets.
 If the field isn't a null, integer, delimiter, or operator, an error is processed, and control returns to the calling routine.
 To store the parameter sets, first the number of parameter sets is stored. If the number is zero, then control returns to the calling routine. If there is at least one parameter set, they are stored in the work array, and control returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
IEND1	Indicator. If TRUE, then the end of the function has been reached. If FALSE, then there are still more fields in the function to process.
IERR	Error indicator. If FALSE, then no error has occurred in this routine. If TRUE, then a fatal error has occurred in this routine.
IERR1	Error indicator. If FALSE, then no error has occurred up to the current time. If TRUE, then a fatal error has occurred.
IND1	Pointer to current location in the LITSTR array.
IPAR	Number of parameter sets processed.

LITMAX	Maximum length of the literal storage array LITSTR.
LITSTR	Array where the literals are stored prior to being used to build a function.
LLXWRK	Indicates the last defined location in the NNWRK array.
LWRK	Temporary counter for the number of locations needed in the NNWRK array.
MMWRK	Maximum number of locations in the NNWRK array.
NDXLIT	Pointer to the current position in the literal storage array.
NDX7	Position in the literal storage array where the current function ends.
NDX8	Current position in the literal storage array that is being processed as a possible parameter set field in the current function.
NDX9	Index and counter.
NNWRK	Work storage array. Contains the current command parsed into the various elements.
NTDLM	Delimiter field indicator.
NTINT	Integer number field indicator.
NTNUL	Null field indicator.
NTOPR	Operator field indicator.

5. I/O VARIABLES

A. INPUT	LOCATION
LITMAX	F.P.
LITSTR	F.P.
LLXWRK	/I(3/

MMWRK	/I04/
NDXLIT	F.P.
NDX7	F.P.
NTDLM	F.P.
NTINT	F.P.
NTNUL	F.P.
NTOPR	F.P.
B. OUTPUT	LOCATION
IEND1	F.P.
IERR1	F.P.
LITSTR	F.P.
LLXWRK	/I03/
NDXLIT	F.P.
NNWRK	/I03/

6. CALLING ROUTINE: FNFND

7. CALLED ROUTINES:

ARAYCK

EEROR

1. NAME: FUNC4A - 81
2. PURPOSE: This routine processes the entities in an IOSYM function.
3. METHOD: This routine starts at the first field after processing the function parameter sets. It looks for valid entities and stores them. It does this by looking at each field in the literal storage array.

If the field is a null field or delimiter field, the pointer to the literal storage array is incremented, and the next field is investigated.

If the field is an integer number, routine CONVR2 is called to change the number name into an alpha name. A check is done to see if the name is an ARC or NODE name. If not, an error is processed, and control returns to the calling routine. If it is an ARC or NODE name, the number of entities is increased by one, and the ARC or NODE number is stored in the work array. Control then proceeds to check the next field.

If the field is an operator, the end of processing the fields entities is finished. Control then returns to the calling routine.

If the field is an alpha field, a check is done to see what type of alpha name it is: independent variable, primary variable, secondary variable, TABLE, DUMM (see users' manual), ARC or NODE. If the field is any of those, its IOSYM identifier is stored in the work array, and the entity counter is increased by one. If the field is not one of those, it is assumed to be an undefined secondary variable. Control then returns to check the next field.

If the field is not a null, integer number, delimiter, operator or alpha field, an error is processed, and control returns to the calling routine.

If the end of the fields in the literal storage array is reached, an indicator is set. Before returning to the calling routine, the number of entities is stored in the work array.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Array pointer and index value.
IEND1	Indicator. If TRUE, then the end of the function has been reached. If FALSE, then there are still more fields in the function to process.

IENT	Number of entities, in the current function, processed.
IERR	Error indicator. If FALSE, then no error has occurred in this routine. If TRUE, then a fatal error has occurred in this routine.
IERR1	Error indicator. If FALSE, then no error has occurred up to the present time. If TRUE, then a fatal error has occurred.
IFLD	Indicator to determine whether a field returned from the DVARFD routine is a derivative variable or not.
II	Pointer to the secondary variable array NNPT2 used to find matching variable names for TABLE variable references.
INN1	Location in the NNWRK where the number of entities in this function will be stored.
I1	First four characters in the alpha name found.
I2	Last three characters in the alpha name followed by a blank.
J	Pointer to the primary variable array and the TABLE array.
LITMAX	Maximum length of literal storage array LITSTR.
LITSTR	Array where the literals are stored prior to being used to build a function.
NNV1	Number of defined primary variables.
NNV2	Number of defined secondary variables.
NNWRK	Work storage array. Contains the current command parsed into the various elements.
NNXTB	Number of defined TABLEs.

NTALPC	Alpha field indicator.
NTDLM	Delimiter field indicator.
NTINT	Integer number field indicator.
NTNUL	Null field indicator.
NTOPR	Operator field indicator.
NTVARD	Derivative variable field indicator.

5. I/O VARIABLES:

A. INPUT	LOCATION
LITMAX	F.P.
LITSTR	F.P.
LLXWRK	/I03/
MMV2	/I04/
MMWRK	/I04/
NDXLIT	F.P.
NDX7	F.P.
NNNODE	/I03/
NNPTAC	/I02/
NNPTB	/I02/
NNPTND	/I02/
NNPT0	/I02/
NNPT1	/I02/
NNPT2	/I02/
NNV1	/I03/
NNV2	/I03/
NNXTB	/I03/
NTALPC	F.P.

NTDLM	F. P.
NTINT	F. P.
NTNUL	F. P.
NTOPR	F. P.
NTVARD	F. P.
B. OUTPUT	LOCATION
IEND1	F. P.
IERR1	F. P.
NDXLIT	F. P.
NNPT2	/I02/
NNWRK	/I03/

6. CALLING ROUTINE: FNFND

7. CALLED ROUTINES:

ARRAYCK

CONVR2

DVARFD

EEROR

1. NAME: FUNC5 - 7
2. PURPOSE: This routine processes the truth condition in an IOSYM function.
3. METHOD: This routine investigates the first field after processing entitles on an IOSYM function, looking for a valid truth conditon. It does this by looking at the field in the literal storage array.

If the field is an alpha field, routine TRUTH is called to determine which truth condition the field is. The truth conditon is then stored in the NNWRK array, and control returns to the calling routine.

If the field is a null field, a zero is stored as the truth condition in the NNWRK array. (Zero indicates that there is not truth conditon.) Control then returns to the calling routine.

If the field is a delimiter field, a check is done to see if there is another field in the function. If not, control returns to the calling routine. If so, this next field is checked to see what it is.

4. INTERNAL VARIABLES

VARIABLE	DEFINITION
IEND1	Indicator is TRUE. Then the end of the function has been reached. If FALSE, then there are still more fields in the function to process.
IERR	Error indicator. If FALSE, then no error has occurred in this routine. If TRUE, then a fatal error has occurred in this routine.
IERR1	Error indicator. If FALSE, then no error has occurred up to the present time. If TRUE, then a fatal error has occurred.
LITMAX	Maximum length of the literal storage array LITSTR.
LITSTR	Array where the literals are stored prior to being used to build a function.
LLXWRK	Indicates the last defined location in the NNWRK array.

MMV2	Maximum number of secondary variables allowed (dimension of VVAR2 and NNPT2).
MMWRK	Maximum number of locations in the NNWRK array.
NDXLIT	Pointer to the current position in the literal storage array.
NDX7	Position in the literal storage array where current function ends.
NDX8	Current position in the literal storage array that is being processed as a possible entity field in the current function.
NNNODE	Number of defined NODEs.
NNPTAC	The first two words contain the ARC names, and the third word is the location in the NSET array where it is located.
NNPTB	Array that contains in the first two words the TABLE names and the third word points to the location in the TTABL array where the TABLE values are stored.
NNPTND	Array that contains in the first two words the NODE names and the third word is the location in the NSET array where the NODE is located.
NNPT0	Array that contains the independent variable name.
NNPT1	Array that contains the primary variable names.
NNPT2	Array that contains the secondary variable names.
LLXWRK	Indicates the last defined location in the NNWRK array.
MMWRK	Maximum number of locations in the NNWRK array.

NDXLIT	Pointer to the current position in the literal storage array.
NDX7	Position in the literal storage array where the current function ends.
NNWRK	Work storage array. Contains the current command parsed into the various elements.
NTALPC	Alpha field indicator.
NTDLM	Delimiter field indicator.
NTNUL	Null field indicator.

5. I/O VARIABLES

A. INPUT	LOCATION
LITMAX	F.P.
LITSTR	F.P.
LLXWRK	/I03/
MMWRK	/I04/
NDXLIT	F.P.
NDX7	F.P.
NTALPC	F.P.
NTDLM	F.P.
NTNUL	F.P.
B. OUTPUT	LOCATION
IEND1	F.P.
IERR1	F.P.
LITSTR	F.P.
NDXLIT	F.P.
NNWRK	/I03/

6. CALLING ROUTINE: FNFND

7. CALLED ROUTINES:

ARAYCK

EEROR

TRUTH

1. NAME: FUNFND-24
2. PURPOSE: This routine searches the NCODE array for alpha fields and checks to see if they are IOSYM functions.
3. METHOD: This routine checks all the defined locations (from 1 to NDXVAL) for an alpha field. If one is found, it is checked to see if the last three characters are blanks. Blanks in the last three characters indicate that the alpha name contains four characters, the most an IOSYM function can have. The first of the four characters is removed. If the result is not an alpha string of 001 to 999, then the string isn't an IOSYM function. If it is, then the first character is checked to see if it is an F, P, or U. If it is none of these, it isn't an IOSYM function. If it is, then the alpha field code is reset to the type of function. Routine CONVRT is called to convert the three characters after the letter from alpha to numeric characters. The resulting number will be the function number and is stored in the appropriate location in routine CONVRT. After all of the fields have been checked, control returns to the calling routine.
4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Is the index that points to the current location in the NCODE array that is being investigated.
IBLNK	Contains the IOSYM internal representation of four blanks.
IFUNC	Is the internal IOSYM value of the appropriate function letter. F = 6 P = 16 U = 21
IN1	Internal IOSYM representation of the three characters after the function letter in the alpha field.
NCODE	Array that contains all the current fields being processed.
NDXVAL	Number of fields that have been processed and stored in the NCODE and NVAL arrays.

NTALPC	Alpha field indicator.
NTFUNF	Menu function field indicator.
NTFUNP	Probability function field indicator.
NTFUNU	User function field indicator.
NVAL	Is the array containing the values of the current fields processed.

5. I/O VARIABLES:

A. INPUT	LOCATION
NCODE	F.P.
NDXVAL	F.P.
NTALPC	F.P.
NTFUNF	F.P.
NTFUNP	F.P.
NTFUNU	F.P.
NVAL	F.P.
B. OUTPUT	LOCATION
IFUNC	F.P.
NCODE	F.P.

6. CALLING ROUTINE: PARSE

7. CALLED ROUTINE: CONVRT

1. NAME; FUNSTR - 22
2. PURPOSE: While processing commands that contain IOSYM functions, routine FUNSTR is called to store the function in the NSET array.
3. METHOD: FUNSTR is sent as a formal parameter the position of the beginning of the function in the NNWRK array. (The work array.) This routine copies the function fields from the work array to permanent storage in the NSET array. The first word in the location passed as a formal parameter is the number of fields in the function. This is first stored in the NSET array. Then the next IND fields are copied from the NNWRK array to the NSET array. Control then returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Index to show which field is being copied from the NNWRK array to the NSET array.
IERR	Error indicator. If FALSE, then no error has occurred in this routine. If TRUE, then a fatal error has occurred in this routine.
IERR1	Error indicator. If FALSE, then no error has occurred up to the present time. If TRUE, then a fatal error has occurred.
IND	Number of fields in the function being copied.
INDX	Pointer to the NNWRK where the function information starts. Also NSET(INDX) is the location of the number of fields in the function.
IND1	Index that points to the current location in the NNWRK array being copied.
LLXSET	Counter of the number of locations in current use in the NSET array. Also the last position in the NSET array used.
LSET	The number of locations in the NSET array needed.

MMSET	Maximum dimension of the NSET array.
NNWRK	Work storage array. Contains the current command parsed into the various elements.
NSET	Storage array where the link listed NETWORK is stored.

5. I/O VARIABLES:

A. INPUT	LOCATION
INDX	F.P.
LLXSET	/I03/
MMSET	/I04/
NNWRK	/I03/
B. OUTPUT	LOCATION
IERR1	F.P.
LLXSET	/I03/
LSET	F.P.
NSET	/ /

6. CALLING ROUTINES:

IFSTR
PCKSTR
SETSTR
STASTR
TRMSTR

7. CALLED ROUTINE; ARAYCK

1. NAME: GBLSTR - 65
2. PURPOSE: This routine processes the GLOBAL commands. This routine does the GLOBAL command initialization and storage of substitutable parameters if there are any.
3. METHOD: First the pointers to the TYPE commands under a GLOBAL command are initialized to zero. A check is done to see if there are any substitutable parameters. If not, the number of parameters (zero) is stored, and control returns to the calling routine. If there are substitutable parameters, the number of them is found and stored in NGOPR. A check is then done to see if there is enough room to store the parameters in the IIOPR array. If not, an error is processed, and control returns to the calling routine. If there is enough room, then NGOPR number of parameters are stored in the IIOPR array. And NNGDUM is set to the number of parameters there are, and control returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Index to show what position in the NNWRK array is being investigated as a substitutable parameter.
IERR	Error indicator. If FALSE, then no error has occurred in this routine. If TRUE, then a fatal error has occurred in this routine.
IERR1	Error indicator, If FALSE, then no error has occurred up to the present time. If TRUE, then a fatal error has occurred.
IIOPR	The first word contains the IIDV of the parameter being substituted, and the second word is the IIDV of the substituted parameter.
IPTCMD	Pointer to the next command. Set to zero to show that GLOBAL command doesn't point to any other command and initializes the pointer for use under TYPE commands.
ITYP1	Initializes the pointer to the kind of TYPE command done on the GLOBAL command to zero to show that a TYPE command hasn't been processed yet.

LEN	Position of the last defined location in the NNWRK array.
MMOPR	Maximum number of substitutable parameters that can be saved in the IIOPR array.
NGOPR	Number of substitutable parameters to process.
NNGDUM	Number of substitutable parameters processed on a GLOBAL command.
NNPTGB	Array that contains in the first two words the IOSYM internal representation of the word GLOBAL. The third, fourth and fifth words point to the location in the NSET array where the TYPE S, TYPE I and TYPE E commands start, respectively.
NNWRK	Work storage array. Contains the current command parsed into the various elements.

5. I/O VARIABLES:

A. INPUT	LOCATION
IERR	F.P.
MMOPR	/I03/
NNGDUM	/I02/
NNWRK	/I03/
B. OUTPUT	LOCATION
IERR1	F.P.
IIOPR	/I02/
IPTCMD	F.P.
ITYP1	F.P.
NGOPR	F.P.

NNGDUM

/I02/

NNPTGB

/I02/

6. CALLING ROUTINE: PROCES

7. CALLED ROUTINES:

ARAYCK

EEROR

1. NAME: GETSTR - 86
2. PURPOSE: This routine processes GET commands. The file number or pointer, the indicator to show the command worked, the TABLE number, and the TABLE pointer are all stored in the NSET array to be processed during IOSYM execution.
3. METHOD: If another command points to this command, then it is set to indicate where this command starts. The current command number is stored in the NSET array. A check is done to see if there are enough fields for this command. If not, an error is processed, and control returns to the calling routine. If so, then the file identifier is stored in the uniform command sequence (UCS) for the GET command in the NSET array. If the file identifier is a number, it is checked to see if it is a valid file number. If not, the number of files is increased to make it a valid file number. The identifier to show that the GET operation worked or didn't work is stored in the UCS. The starting location identifier to store the fields is stored in the UCS. The TABLE that the starting location is in is also stored in the UCS. And lastly, a position in the NSET array is set aside to point to the next command.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
CCONS	Array that contains all of the constants in a NETWORK.
I	Is the current TABLE number being investigated as the TABLE being used on a GET command.
IERR	Error indicator. If FALSE, then no error has occurred in this routine. If TRUE, then a fatal error has occurred in this routine.
IERR1	Error indicator. If FALSE, then no error has occurred up to the present time. If TRUE, then a fatal error has occurred.
IN1	Number of the file being used to retrieve the information from.
IPTCMD	Pointer to the next command. Location in the NSET array where the pointer to the next command is located. If IPTCMD is zero, then no command UCS points to the current command.

LLXSET	Counter of the number of locations in current use in the NSET array. Also the last position in the NSET array used.
LXSET	Temporary storage for the number of NSET elements needed.
MMSET	Maximum number of elements in the NSET array (the dimension of the NSET array).
NNFIL	Number of GASP files to be used.
NNPTB	Array that contains in the first two words the TABLE names. The third word is the pointer to the TTABL array where the start of the storage elements for the TABLE is located.
NNWRK	Work storage array. Contains the current command parsed into the various elements.
NNXTB	Number of defined TABLEs.
NSET	Storage array where the link listed NETWORK is stored.

5. I/O VARIABLES:

A. INPUT	LOCATION
CCONS	/I02/
IERR	F.P.
IPTCMD	F.P.
LLXSET	/I03/
MMSET	/I04/
NNFIL	/GCOM1/
NNPTB	/I02/
NNWRK	/I03/
NNXTB	/I03/

B.	OUTPUT	LOCATION
	IERR1	F.P.
	IPTCMD	F.P.
	LLXSET	/I03/
	LXSET	F.P.
	NNFIL	/GCOM1/
	NSET	/ /

6. CALLING ROUTINE: PROCES

7. CALLED ROUTINES:

ARAYCK

EEROR

1. NAME: GGGET - 87
2. PURPOSE: This routine executes a GET command during an IOSYM simulation.
3. METHOD: The TABLE number that is being written to is retrieved from the GET command uniform command sequence (UCS). The starting point in the TABLE is retrieved. The amount of room left in the TABLE is calculated. If there is not enough room to store the GET information, an error is processed. The file number is then retrieved from the command UCS. If the file number is greater than the number of files allowed, an error is processed. If there are no entries left in the file, an indicator to show that is set to zero, and control returns to the calling routine. If there is an entry in the file,, it is removed and the values copied from the GASP array ATRIB to the TABLE. The indicator is then set to show that the transfer worked as requested, and control returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
ATRI	Array that contains the attribute values being retrieved from the files.
I	Index to which attribute is being copied from the ATRIB array to the TABLE.
IERR1	Error indicator used in call to EEROR, but not used in this routine.
LEN	Used as a pointer to find information from various arrays.
LEN1	Position in the TTABL array where the items from the ATRIB array are to be stored.
LEN2	Position in the NSET (QSET) array where the first entry in the file is located.
LUCS	Position in the NSET array where the UCS for the GET command starts.
MFE	Array that contains the relative addresses of the first entry in the different files.

NNATR	Number of attributes in each entry in the files.
NNFIL	Number of GASP attribute files.
NNPTB	Array that contains in the first two words the TABLE names, and the third word contains the pointer to the location in the TTABL array where the TABLE values are stored.
NNQ	Is an array that tells how many entries are in each file.
NSET	Storage array where the link listed NETWORK is stored.
TTABL	Array that contains in the first word the current values of all the TABLE elements and the second word contains the initial values.
VVAR2	Array that contains in the first word the current values of all the secondary variables and the second word contains the initial values.

5. I/O VARIABLES:

A. INPUT	LOCATION
ATTRIB	/GCOM1/
LUCS	F.P.
MFE	/GCOM1/
NNATR	/GCOM1/
NNFIL	/GCOM1/
NNPTB	/I02/
NNQ	/GCOM1/
NSET	/ /
VVAR2	/I02/

B. OUTPUT

LOCATION

LEN1

F.P.

TTABL

/I02/

6. CALLING ROUTINE: PPFCMD

7. CALLED ROUTINES:

EEROR

PPIDF

RMOVE

VVARBL

1. NAME: GGOTO-529
2. PURPOSE: This routine performs a GOTO command during an IOSYM simulation.
3. METHOD: First, a determination is made on whether a NODE or ARC is what is being gone to. If an ARC is to be processed, the ARC number is identified, and the pointer to the next ARC is set to the ARC number. Control then returns to the calling routine.

If a NODE is to be processed, the NODE number is identified; the pointer to the next NODE is set to the NODE number, and the next ARC pointer is set to zero to indicate an ARC is not next. Control then returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Location in the NSET array where the uniform command sequence is located.
IAND	Indicator. If positive, then it is the ARC number. If negative, then it is the negative of the NODE number.
IARCND	Indicator. If positive, then it is the ARC number. If negative, then it is the negative of the NODE number.
IERR1	Error indicator returned from routine EEROR but not used in this routine.
IINXAC	Next ARC number. If zero, then an ARC is not performed next.
IINXND	Next NODE number.
NNARC	Number of ARCs in a NETWORK.
NNNODE	Number of NODEs in a NETWORK.
NNPTAC	Array containing the first two words in the ARC name, and the third word points to the location of the uniform command sequence.
NNPTND	Array containing in the first two words the NODE name, and the third word points to the location of the uniform command sequence.

NSET

Storage array where the link listed
NETWORK is stored.

5. I/O VARIABLES:

A. INPUT	LOCATION
IARCND	F.P.
IERR1	F.P.
NNARC	/I03/
NNNODE	/I03/
NNPTAC	/I02/
NNPTND	/I02/
NSET	/ /
B. OUTPUT	LOCATION
IINXAC	/I03/
IINXND	/I03/

6. CALLING ROUTINE: PPFCMD

7. CALLED ROUTINE: EEROR

1. NAME: GGPAR-503
2. PURPOSE: This routine returns elements of the parameter sets.
3. METHOD: If the parameter set number is less than or equal to zero, then a default parameter set of all ones is assumed. If the parameter set number is greater than zero, the parameter set value is retrieved from the PPARM array.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
GGPAR	This contains the value of the parameter set upon returning to the calling routine.
IERR1	Error indicator returned from the EEROR routine but not used in this routine.
IPAR	Parameter set number.
LMENT	Element in parameter set desired.
MMPAR	Number of parameter sets.
PPARM	Array that contains the values of all the parameter set elements.

5. I/O VARIABLES:

A. INPUT	LOCATION
IERR1	F.P.
IPAR	F.P.
LMENT	F.P.
MMPAR	/I04/
PPARM	/GCOM1/
B. OUTPUT	LOCATION
GGPAR	FUNCTION

6. CALLING ROUTINE: (None. Not used at this time).
7. CALLED ROUTINE: EEROR

1. NAME: GGSTAT-558
2. PURPOSE: This routine will process the IOSYM - GASP statistics encountered during a simulation.
3. METHOD: The type of statistic is first identified, and the expression is evaluated. If collect statistics are to be done, GASP routine COLCT is called. If time stat statistics are to be done, GASP routine TIMST is called. If histograms are to be done, GASP routine HISTO is called. If time statistics are to be done, GASP routine TIMSA is called. Control then returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
ITYP	Type of statistic to do. 1 = COLCT 2 = TIMST 3 = HISTO 4 = TIMSA
LUCS	Location in the NSET array where the uniform command sequence for the STATistic command is stored.
NSET	Storage array where the link listed NETWORK is stored.
VV	Value of the variable to be collected.

5. I/O VARIABLES:

A. INPUT	LOCATION
LUCS	F.P.
NSET	/ /
VV	F.P.
B. OUTPUT	LOCATION
(None)	

6. CALLING ROUTINE: PPFCMD

7. CALLED ROUTINES:

COLCT (GASP routine)
EEVAL
HISTO (GASP routine)
TIMSA (GASP routine)
TIMST (GASP routine)
VVARBL

1. NAME: GGSTEP-557
2. PURPOSE: This routine estimates the step size for use by GASP when there are CROSS commands.
3. METHOD: If there is no CROSS command, the normal GASP logic is used to determine the step size. If there is a CROSS command, then the step size is calculated by taking the minimum of the full step size or the time from current time to next CROSS event.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
DT	Current delta time step size.
DTFUL	Step size of the full time step.
DTNOW	Time at TNOW minus time at TTLAS.
I	Index to point to the values of the different CROSS commands.
IBT	GASP indicator to use half or full step size.
J	Pointer to the state condition flags.
L	Location of where the values for the first CROSS command are located in the SS and SSL arrays.
LFLAG	State condition flags (see GASP book).
NNCROS	Number of CROSS commands.
NNEQT	Number of primary variables and CROSS commands.
SS	Value of the primary variables and CROSS conditions at time TNOW.
SSL	Value of the primary variables and CROSS conditions at time TTLAS.

5. I/O VARIABLES:

A. INPUT	LOCATION
DTFUL	/GCOM2/

DTNOW /GCOM2/

IBT F. P.

LFLAG /GCOM2/

NNCROS /I03/

NNEQT /GCOM2/

SS /GCOM2/

SSL /GCOM2/

B. OUTPUT LOCATION

DTFUL /GCOM2/

6. CALLING ROUTINE: GASP

7. CALLED ROUTINE: (None)

1. NAME: GG TAB-501
2. PURPOSE: This routine identifies the value of a TABLE element and returns it to the calling routine.
3. METHOD: The TABLE number is identified. If it is less than one, a default TABLE containing all ones is assumed. If it is greater than zero, then the TABLE element desired is identified. If the TABLE element number is positive, then the current value of the TABLE element is returned. If the TABLE element is less than zero, then the initial value of the TABLE element is returned. If the TABLE element is zero, then the length of the TABLE is returned.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
GGTAB	Contains the value of the TABLE element or the TABLE length upon returning to the calling routine.
I	Indicates the current value is desired if equal to one, and the initial value is desired if equal to two.
IERR1	Error indicator returned from routine EEROR but not used in this routine.
ITABL	TABLE number of the TABLE being investigated.
ITB	TABLE number of the TABLE being investigated.
LMNT	The TABLE element desired.
LMNT	If positive, it indicates the current value of the TABLE element LMNT is desired. If negative, it indicates the initial value of the TABLE element -LMNT is desired. If zero, indicates the length of the TABLE is desired.
NNPTB	Array where the first two words are the TABLE name, and the last word points to the location in the TTABL array where the TABLE element values are stored.

NNXTB

Number of defined TABLEs.

TTABL

Array containing all the TABLE
element values.

5. I/O VARIABLES:

A. INPUT

LOCATION

IERR1

F.P.

ITABL

F.P.

LMNT

F.P.

NNPTB

/I02/

NNXTB

/I03/

TTABL

/I02/

B. OUTPUT

LOCATION

GGTAB

FUNCTION

6. CALLING ROUTINES: FF
PP
VVARBL

7. CALLED ROUTINE: EEROR

1. NAME: GGVAR-509
2. PURPOSE: This routine will return the value of a simple variable.
3. METHOD: The variable type and variable number are identified. If the value of the independent variable is desired, TNOW is returned as current value or TTBEQ is returned for initial value.

If the value of a primary variable is desired, then the current value, from the SS array, or the initial value, from the SSIN array, is returned.

If the value of a secondary variable is desired, then the current value, from the first word of the VVAR2 array, or the initial value, from the second word of the VVAR2 array, is returned.

If the value of a derivative variable is desired, then the current value, from the DD array, or the initial value, from the DDIN array, is returned.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
DD	The value of the derivative variable at the time TNOW.
DDIN	The initial value of the derivative variable at time TTBEQ.
GGVAR	Contains the variable value, upon returning to the calling routine.
IERR1	Error indicator returned from the routine EEROR but not used in this routine.
IP	Type of variable being investigated: <ul style="list-style-type: none"> 1 = Independent 2 = Primary 3 = Secondary 4 = Derivative
IVAR	Variable number. If positive, want current value. If negative, want initial value.

I0123

Type of variable being investigated:

0 = Independent
1 = Primary
2 = Secondary
3 = Derivative

NNV1

Number of defined primary variables.

NNV2

Number of defined secondary variables.

NV

Variable number.

SS

Array containing the values of the primary variables at the time TNOW.

SSIN

Array containing the initial values of the primary variables at the time TTBEQ.

TNOW

Current time of the simulation.

TTBEQ

Initial time.

VVAR2

Array containing in the first word the current value of the secondary variable, and the second word is the initial value of the secondary variable.

5. I/O VARIABLES:

A. INPUT

LOCATION

DD

/GCOM2/

DDIN

/I02/

IERR1

F.P.

IVAR

F.P.

I0123

F.P.

NNV1

/I03/

NNV2

/I03/

SS

/GCOM2/

SSIN /I02/

TNOW /GCOM1/

TTBEG /GCOM1/

VVAR2 /I02/

B. OUTPUT LOCATION

GGVAR FUNCTION

6. CALLING ROUTINES: PPIDF
VVARBL

7. CALLED ROUTINE: EEROR

1. NAME: GTOSTR-48
2. PURPOSE: This routine processed GOTO commands. The ARC or NODE pointer is stored to be processed during an IOSYM simulation.
3. METHOD: If another command points to this command, then it is set to indicate where the command is stored in the NSET array. The ARC number is stored in the NSET array if the GOTO command sends control to an ARC. The negative of the NODE number is stored in the NSET array if the GOTO command sends control to a NODE. Last a position in the NSET array is set aside to point to the next command.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
IERR	Error indicator. If FALSE, then no error has occurred in this routine. If TRUE, then a fatal error has occurred in this routine.
IERR1	Error indicator, If FALSE, then no error has occurred up to the present time. If TRUE, then a fatal error has occurred.
IPTCMD	Pointer to the next command location in the NSET array where the pointer to the next command is located. If IPTCMD is zero, then no command points to the current command.
LLXSET	Counter of the number of locations in current use in the NSET array. Also the last position in the NSET array used.
LSET	Temporary storage for the number of NSET elements needed.
MMSET	Maximum number of elements in the NSET array (the dimension of the NSET array).
NNWRK	Work storage array. Contains the current command parsed into the various elements.
NSET	Storage array where the link listed NETWORK is stored.

5. I/O VARIABLES:

A.	INPUT	LOCATION
	IERR	F.P.
	IPTCMD	F.P.
	LLXSET	/I03/
	MMSET	/I03/
	NNWRK	/I03/
B.	OUTPUT	LOCATION
	IERR1	F.P.
	IPTCMD	F.P.
	LLXSET	/I03/
	NSET	/ /

6. CALLING ROUTINE: PROCES

7. CALLED ROUTINE: ARAYCK

1. NAME: IFSTR-59
2. PURPOSE: This routine processes IF commands. The identifier for the left hand side variable, truth condition, pointer to right hand expression, pointer to the TRUE side of the IF, pointer to the FALSE side of the IF, and a pointer to the IF LABEL are all stored to be processed during an IOSYM simulation.
3. METHOD: If another command points to this command, then it is set to indicate where this command is stored in the NSET array. The command number is stored in the NSET array as part of this command's uniform command sequence (UCS). The identifier of the left side variable is retrieved from the work array and stored in the NSET array. The truth condition is also retrieved and stored. A pointer to the right hand side expression is put aside as is the pointer for the TRUE side of the IF, the FALSE side of the IF, and the LABEL location. These pointers will be used later to point to the appropriate information, as that information is processed. The LABEL number, TRUE pointer, FALSE pointer and the LABEL pointer are stored in the IIFPT array to be used when processing the TRUE, FALSE or LABEL information for this IF command. Routine FUNSTR is called to process the right hand side expression and store it in the NSET array. And the pointer to the RHS is set to point to the location of the start of this expression. Control then returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
IERR	Error indicator. If FALSE, then no error has occurred in this routine. If TRUE, then a fatal error has occurred in this routine.
IERR1	Error indicator. If FALSE, then no error has occurred up to the present time. If TRUE, then a fatal error has occurred.
IIFPT	Array that contains the IF command pointers. Word one contains the negative of the LABEL number to show this information is for an IF. (LOOP information is also stored in this array, but the LABEL number is positive). Word two contains the location in the NSET array to store the pointer to the TRUE part of the IF command. Word three contains the location in the NSET array to store the pointer to the FALSE part of the IF command. Word four contains the location in the NSET array to store the pointer to the IF LABEL.

INDF	Is the location in the NSET array to store the pointer to the FALSE part of the IF command.
INDL	Is the location in the NSET array to store the pointer to the IF LABEL.
INDT	Is the location in the NSET array to store the pointer to the TRUE part of the IF command.
IND1	Is the location in the NSET array to store the pointer to the right hand side expression.
IPTCMD	Pointer to the next command. Location in the NSET array where the pointer to the next command is located. If IPTCMD is zero, then no command points to the current command.
LLXSET	Counter of the number of locations in current use in the NSET array. Also the last position in the NSET array used.
LSET	Temporary storage for the number of NSET elements needed.
MMFPT	Maximum number of IF or LOOP commands that can be stacked. (Dimension of IIFPT array.)
MMSET	Maximum number of elements in the NSET array (the dimension of the NSET array).
NNFPT	Number of stacked IF or LOOP commands.
NNWRK	Work storage array. Contains the current command parsed into the various elements.
NSET	Storage array where the link listed NETWORK is stored.
NTFLP	Floating point number field indicator.
NTINT	Integer number field indicator.

NNTBL	Table field indicator.
NTVARD	Derivative variable field indicator.
NTVARO	Independent variable field indicator.
NTVAR1	Primary variable field indicator.
NTVAR2	Secondary variable field indicator.

5. I/O VARIABLES:

A. INPUT	LOCATION
IERR	F.P.
IPTCMD	F.P.
LLXSET	/I03/
MMFPT	/I04/
MMSET	/I04/
NNFPT	/I03/
NNWRK	/I03/
NTFLP	F.P.
NTINT	F.P.
NTTBL	F.P.
NTVARD	F.P.
NTVARO	F.P.
NTVAR1	F.P.
NTVAR2	F.P.
B. OUTPUT	LOCATION
IERR1	F.P.
IIFPT	/I02/
IPTCMD	F.P.
LLXSET	/I03/

LSET

F.P.

NSET

/ /

6. CALLING ROUTINE: PROCES

7. CALLED ROUTINES:

ARRAYCK

FUNSTR

1. NAME: IIACST-506
2. PURPOSE: This routine returns the current status of the given ARC.
3. METHOD: The ARC number is identified, and the uniform command sequence is checked to find the current ARC's status.
4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
IARC	ARC number of the ARC to check.
IERR1	Error indicator returned from routine EEROR but not used in this routine.
IIACST	Upon returning to the calling routine, contains a one if the ARC is active or a minus one if the ARC is inactive.
L	Pointer to the location in the NSET array where the current ARC status is located.
NNARC	Number of defined ARCs in the current NETWORK.
NNPTAC	Array containing in the first two words the ARC name, and the third word is the location in the NSET array where the ARC is stored.
NSET	Storage array where the link listed NETWORK is stored.

5. I/O VARIABLES:

A. INPUT	LOCATION
IARC	F.P.
IERR1	F.P.
NNARC	/I03/
NNPTAC	/I02/
NSET	/ /

B. OUTPUT

LOCATION

IIACST

FUNCTION

6. CALLING ROUTINES: (None. Not used at the present time).
7. CALLED ROUTINE: EEROR

1. NAME: IIMOV-519
2. PURPOSE: This routine takes an array and multiplies each element by a factor and stores it in another array.
3. METHOD: Two arrays of length N and an integer multiplier IMUL are passed to this routine. N elements in the second array are multiplied by IMUL and stored in the first array. Control then returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Index and counter to show which array element is being processed.
IM	Multiplication factor.
IMUL	Multiplication factor.
I1	First array. This is the array that results are stored in.
I2	Second array. This is the array that is multiplied by the multiplication factor.
N	Number of elements in the I1 and I2 arrays.
NN	Number of elements to multiply by the multiplication factor.

5. I/O VARIABLES:

A. INPUT	LOCATION
IMUL	F.P.
I2	F.P.
N	F.P.
B. OUTPUT	LOCATION
I1	F.P.

6. CALLING ROUTINES: AARCST
SSHIFT

7. CALLED ROUTINE: (None)

1. NAME: ISET - 517
2. PURPOSE: This routine initializes the elements in an array to a constant value.
3. METHOD: N elements of the array are set to the value stored in IVAL. Then control returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Index and counter to the array element being processed.
IV	Value to set array elements to.
IVAL	Value to set array elements to.
IZ	The array that is to be initialized.
N	Number of elements in the IZ array to initialize.
NN	Number of elements in the IZ array to initialize.

5. I/O VARIABLES:

A. INPUT	LOCATION
IVAL	F.P.
N	F.P.
B. OUTPUT	LOCATION
IZ	F.P.

6. CALLING ROUTINES: ADATA
OUTPUT
PARSE
ZERO

7. CALLED ROUTINE: (None)

1. NAME: IMOVJ
2. PURPOSE: This routine calculates $INEW = MULT * IOLD$, where INEW and IOLD are arrays, for each IOLD and INEW element.
3. METHOD: Each IOLD element is multiplied by MULT and stored in the corresponding location in the INEW vector.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Do loop index.
INEW	Array to store the new values in.
IOLD	Array to be multiplied by MULT.
MUL, MULT	Multiplication factor.
N, NN	Number of elements to process in the arrays.

5. I/O VARIABLES

A. INPUT	LOCATION
IOLD	F.P.
MULT	F.P.
N	F.P.
B. OUTPUT	LOCATION
INEW	F.P.

6. CALLING ROUTINE: (None. Not used at this time)
7. CALLING ROUTINE: (None)

1. NAME: IMULJ
2. PURPOSE: This routine calculates $IJ = MULT * IJ$, where IJ is an integer array, for each IJ element.
3. METHOD: Each IJ element is multiplied by MULT and stored in the IJ element.
4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Do loop index.
IJ	Array to process.
MUL, MULT	Multiplication coefficient.
N, NN	Number of IJ array elements to process.
5. I/O VARIABLES

A. INPUT	LOCATION
IJ	F.P.
MULT	F.P.
N	F.P.
B. OUTPUT	LOCATION
IJ	F.P.
6. CALLING ROUTINE: VMULJ
7. CALLED ROUTINE: (None)

1. NAME: INTLC - 540
2. PURPOSE: This routine initializes the variables and sets up for the next run.
3. METHOD: The NODE and ARC pointers are setup. The IOSYM input variables are initialized. The ARC status for all the ARCs is reset to the initial values. The statistics, the GASP variables, and the initial NODEs and ARCs are setup. The PLOT and TABULATE tapes are initialized. Control then returns to the calling routine.
4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
DD	This array contains the derivative values at the time TNOW.
DDIN	This array contains the initial derivative values.
I	Index used to point to current ARC or NODE.
IIGO	This is the NODE number of the NODE where the simulation starts.
IINXAC	The ARC number for the next ARC. If equal to zero, then the next step isn't ARC; it is a NODE instead.
IINXND	NODE number of the next NODE to process.
LUCS	Pointer to the NSET array where the ARC uniform command sequence starts.
MMARC	Maximum number of ARCs a NETWORK can have.
MMNODE	Maximum number of NODEs a NETWORK can have.
MMTAB	Maximum number of TABLE elements a NETWORK can have.
NNPTAC	Array that contains in the first two words the ARC name, and the third word points to the location in the NSET array where the ARC uniform command sequence begins.

NNPTND	Array that contains in the first two words the NODE name, the third word points to the location in the NSET array where the NODE uniform command sequence begins, and the fourth word indicates whether the NODE has been replaced or not. If equal to zero, the NODE hasn't been replaced.
NNV1	Number of primary variables in the NETWORK.
NNV2	Number of secondary variables in the NETWORK.
NNWITH	Number of WITH variables.
NSET	Storage array that contains the link listed NETWORK.
SS	This array contains the primary variables current value at the time TNOW.
SSIN	This array contains the primary variables initial value at time TTBEQ.
TTABL	Array that contains the current and initial values of the TABLE elements in the first and second words, respectively.
TTBEG	Beginning time of the simulation.
TTIME1	Beginning time of the simulation.
VVAR2	Array that contains the current and initial values of the secondary variables in the first and second words, respectively.

5. I/O VARIABLES:

A. INPUT	LOCATION
DD	/GCOM2/
DDIN	/I02/

IIG0	/I03/
MMARC	/I04/
MMNODE	/I04/
MMTAB	/I04/
NNPTAC	/I02/
NNV1	/I03/
NNV2	/I03/
NNWITH	/I03/
SS	/GCOM2/
SSIN	/I02/
TTABL	/I02/
TTIME1	/I02/
VVAR2	/I02/
B. OUTPUT	LOCATION
IINXAC	/I03/
IINXND	/I03/
NNPTND	/I02/
NSET	/ /
TTBEG	/GCOM1/

6. CALLING ROUTINE: DATIN

7. CALLED ROUTINES:

AARCST
 CCOLC
 NNODST
 TTAPES
 UINTLC (user written and supplied)
 VVMOV
 WTHMN
 XXINTL
 ZZSET

1. NAME: IOSYM
2. PURPOSE: This is the main program. This routine is machine specific. It sets up the files that will be used. Routine SETJ is called to initialize some data. Routine USRDAT is called to set all of the machine dependent variables and array dimensions. Routine CHOICE is called which makes the decisions on the flow of control through the whole IOSYM program.

3. METHOD: Not applicable.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
DEBUG	This tells the CDC computer where to put the FORTRAN DEBUG statements. In this case, they will go on the file called OUTPUT.
INPUT	This tells the CDC computer how much buffer space to allow for the file input. Also, it is the file from which the input will be read.
OUTPUT	This tells the CDC computer how much buffer space to allow for the file output. This also is the file where the output will be put.
TAPE10	This is the file on which the IOSYM plot information is stored.
TAPE11	This is the file on which the IOSYM tabulation information is stored.
TAPE12	This is a scratch file used when processing the SAVE and RESTORE commands in routine SAVE.
TAPE5	This is the file that input is read from.
TAPE6	This is the file that output is written to.
TAPE8	This is the file that the NETWORK information is stored on when a SAVE command has been processed. Used as input when a RESTORE command is to be processed.

TAPE9

File that is used to store the GASP information.

TAPE98

This is the tape that is used to store the user routine UPLOT's plot information to be further processed.

5. I/O VARIABLES: None.
6. CALLING ROUTINE: Not applicable.
7. CALLED ROUTINES: CHOICE
SETJ
USRDAT

1. NAME: JOESTR - 70
2. PURPOSE: This routine processes JUMP and EXIT commands. The command number and the LABEL name are stored in the NSET array to be used during an IOSYM simulation.
3. METHOD: If another command points to this command, then it is set to indicate where this command is stored in the NSET array. The command number for either the JUMP or EXIT command is stored in the NSET array. Also stored is the name of the LABEL that the JUMP or EXIT is to do. Control then returns to the calling routine.
4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
IERR	Error indicator. If FALSE, then no error has occurred in this routine. If TRUE, then a fatal error has occurred in this routine.
IERR1	Error indicator. If FALSE,, then no error has occurred up to the present time. If TRUE, then a fatal error has occurred.
IN1	LABEL number that the JUMP or EXIT refers to.
IPTCMD	Pointer to the next command. Location in the NSET array where the pointer to the next command is located. If IPTCMD is zero, then no command points to the current command.
LLARY	Array containing in the first word the first four characters of the LABEL name in internal IOSYM representation. The second word is the last three characters of the LABEL name.
LLXSET	Counter of the number of locations in current use in the NSET array. Also the last position in the NSET array used.
LXSET	Temporary storage for the number of NSET elements needed.

MMSET	Maximum number of elements in the NSET array (the dimension of the NSET array).
NNWRK	Work storage array. Contains the current command parsed into the various elements.
NSET	Storage array where the link listed NETWORK is stored.

5. I/O VARIABLES:

A. INPUT	LOCATION
IERR	F.P.
IPTCMD	F.P.
LLARY	/I02/
LLXSET	/I03/
MMSET	/I04/
NNWRK	/I03/
B. OUTPUT	LOCATION
IERR1	F.P.
IPTCMD	F.P.
LLXSET	/I03/
LXSET	F.P.
NSET	/ /

6. CALLING ROUTINE: PROCES

7. CALLED ROUTINE: ARAYCK

1. NAME: KWFND - 42
2. PURPOSE: This routine searches the NCODE array for keyword names. If one is found, it is processed and stored in the keyword array.
3. METHOD: This routine checks all the defined locations (from one to NDXVAL) in the NCODE array for keywords. Each field is checked to see if it is an equal sign (indicates a SET command). If it is, the SET command number is stored in the keyword array (KWSTR). Control then returns to the calling routine. If not an equal sign, the field is checked to see if it is an alpha field, if not, the next field is checked. If it is an alpha field, then a check is done to see if the name matches any keyword names. If not, the next field is checked, if so, then the keyword number is stored in the keyword array and check of each field continues. When the end of the defined locations in NCODE is reached, control returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Index that shows which field in the NCODE and NVAL arrays is being tested for a keyword.
IERR1	Error indicator. If FALSE, then no error has occurred up to the present time. If TRUE, then a fatal error has occurred.
J	Index that shows what command number is being processed.
KWNDX	Indicates the number of keywords found on one input record.
KWSTR	Array where the keywords are stored after they have been identified.
KYWORD	The first two words contain the keyword names in internal IOSYM representation; third word contains the level the command brings control to; the fourth word is the level this command is valid under, and the last word points to the NTSFMT array where the syntax for this command is located.
MAXKW	Is the number of valid keywords recognized by IOSYM.

NCODE	Array containing the type of fields that were processed from the input record.
NDXVAL	This is the number of fields read from the current input record.
NTALPC	Alpha field indicator.
NTOPR	Operator field indicator.
NVAL	Is the array containing the value of the fields read from the current input record.

5. I/O VARIABLES:

A. INPUT	LOCATION
KEYWORD	/ME1/
MAXKW	F.P.
NCODE	F.P.
NDXVAL	F.P.
NTALPC	F.P.
NTOPR	F.P.
NVAL	F.P.
B. OUTPUT	LOCATION
IERR1	F.P.
KWSTR	F.P.
NCODE	F.P.

6. CALLING ROUTINE: PARSE

7. CALLED ROUTINE: EEROR

1. NAME: KWORDJ
2. PURPOSE: This routine will locate the entry in the KEY array that matches the alpha name KWORD.
3. METHOD: The KEY array is searched for a match to the KWORD name. If a match is found, the value returned is the location in the KEY array where the match was. If a match isn't found, a new KWORD is requested.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
KEY	Array containing the names to match.
KWORD	Name to match against the KEY array.
KWORDJ	Function value that contains the location in the KEY array where the match was found.
NKEY	Number of names in the KEY array.
NKNOWN	Error message array.

5. I/O VARIABLES:

A. INPUT	LOCATION
KEY	F.P.
KWORD	F.P.
NKEY	F.P.
B. OUTPUT	LOCATION
KWORDJ	FUNCTION

6. CALLING ROUTINES: (None. Not used at this time).

7. CALLED ROUTINES: FFAJ
FFPUDJ
PSTRJ

1. NAME: LBLCHK - 61
2. PURPOSE: This routine checks to see that when any END command is processed that all LOOP and IF LABELs have been found and processed.
3. METHOD: As each LOOP or IF command is processed, information on the LABEL needed is added to the IIFPT array. As each LABEL is processed, the appropriate information is removed from the IIFPT array. The counter that keeps track of the number of entries in the array is NNFPt. In this routine, a check is done to see if NNFPt is zero. If so, control returns to the calling routine. If not, an error is processed, and the counter is set to zero for the next command.
4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
IERR1	Error indicator. If FALSE, then no error has occurred up to the present time. If TRUE, then a fatal error has occurred.
NNFPt	Number of stacked IF or LOOP commands.
5. I/O VARIABLES:

A. INPUT	LOCATION
IERR1	F.P.
NNFPt	/I03/
B. OUTPUT	LOCATION
NNFPt	/I03/
6. CALLING ROUTINE: PROCES
7. CALLED ROUTINE: EEROR

1. NAME: LBLSTR - 58
2. PURPOSE: This routine processes LABEL commands. It stores the appropriate LABEL information in the NSET array and checks to see if this LABEL ends any IF or LOOP commands. The TRUE or FALSE information is stored for the IF command, and the information needed to end both the IF and LOOP commands is stored.
3. METHOD: If another command points to this command, then it is set to indicate where the current command starts. The command number is stored in the NSET array along with the internal IOSYM representation of the LABEL nmae in the next two words after the command number. A location in the NSET array is also set aside to point to the next command. If currently there are no LOOP or IF commands pending, control returns to the calling routine. If this LABEL doesn't belong to the end of the IF or LOOP command pending, control returns to the calling routine. If this LABEL does end an IF or LOOP, set pointers in these commands to point to this LABEL's location in the NSET array. If the LABEL goes with an IF command, check the pointers to the TRUE and FALSE commands and reset if necessary. Decrease the number of stacked LOOPS or IFs and return to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
IERR	Error indicator. If FALSE, then no error has occurred in this routine. If TRUE, then a fatal error has occurred in this routine.
IERR1	Error indicator. If FALSE, then no error has occurred up to the present time. If TRUE, then a fatal error has occurred.
IIFPT	Array that contains the IF and LOOP commandpointers. Word one contains the negative of the LABEL number if it is for an IF command, and the LABEL number (positive) if it is for a LOOP command. Word two contains the location in the NSET array to store the pointers to the TURE part of the IF command. Word three contains the location in the NSET array to store the pointer to the FALSE part of the IF command. Word four contains the location in the NSET array to store the pointer to the IF or LOOP LABEL.

IINI	Location in the NSET array where the pointer to the TRUE or FALSE part of an IF is stored.
INI	Pointer to the NSET array that points to this LABEL command.
IPT	Position in the NSET array that points to this LABEL command.
IPTCMD	Pointer to the next command. Location in the NSET array where the pointer to the next command is located. If IPTCMD is zero, then no command points to the current command.
LLARY	Array that contains the internal representation of the LABEL name. The first four characters are in the first word and the last three characters are in the second word.
LLXSET	Counter of the number of locations in current use in the NSET array. Also the last position in the NSET array used.
LXSET	Temporary storage for the number of NSET elements needed.
MMSET	Maximum number of elements in the NSET array (the dimension of the NSET array).
NNARY	Number of LABELs that have yet to be processed.
NNFPT	Number of stacked IF and LOOP commands.
NNWRK	Work storage array. Contains the current command parsed into the various elements.
NSET	Storage array where the link listed NETWORK is stored.

5. I/O VARIABLES:

A. INPUT	LOCATION
IERR	F.P.

IIFPT	/I02/
IPTCMD	F.P.
LLARY	/I02/
LLXSET	/I03/
MMSET	/I04/
NNARY	/I03/
NNFPT	/I03/
NNWRK	/I03/
NSET	/ /

B. OUTPUT LOCATION

IERR1	F.P.
IPTCMD	F.P.
LLXSET	/I03/
LXSET	F.P.
NNARY	/I03/
NNFPT	/I03/
NSET	/ /

6. CALLING ROUTINE: PROCES

7. CALLED ROUTINE: ARAYCK

1. NAME: LIBFND - 25
2. PURPOSE: This routine searches the NCODE array for alpha fields and checks them to see if they are FORTRAN library functions. If so, they are set to the appropriate operation number.
3. METHOD: This routine checks all of the defined locations (from one to NDXVAL) for an alpha field. If the field is an alpha, the next field is checked. If the field is an alpha field, it is checked against all of the FORTRAN library function names for a match. If there isn't a match, the next field is checked. If there is a match, the field is changed from an alpha to an operator field (FORTRAN library functions are considered operators along with the +, -, =, /, *, and **). The appropriate operator number is stored in the corresponding NVAL location. After all fields are checked, control returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Index that points to the current literal field in the NCODE array being processed.
J	Index to the current FORTRAN library function being investigated.
LIBFUN	Contains the internal IOSYM representation of the FORTRAN library function names.
MAXLIB	Number of FORTRAN library functions. Also the dimension of LIBFUN.
NCODE	Array that contains all the current fields being processed.
NDXVAL	Number of fields that have been processed and stored in the NCODE and NVAL array.
NTALPC	Alpha field indicator.
NTOPR	Operator field indicator.
NVAL	The array containing the values of the current fields processed.

5. I/O VARIABLES:

A. INPUT	LOCATION
LIBFUN	F.P.
MAXLIB	F.P.
NCODE	F.P.
NDXVAL	F.P.
NTALPC	F.P.
NTOPR	F.P.
NVAL	F.P.
B. OUTPUT	LOCATION
NCODE	F.P.
NVAL	F.P.

6. CALLING ROUTINE: PARSE

7. CALLED ROUTINE: (None)

1. NAME: LITFND - 18
2. PURPOSE: This routine takes all of the non-zero fields in the NCODE and NVAL arrays and stores them in the LITSTR (literal storage) array.
3. METHOD: The LITSTR, NCODE and NVAL arrays have been EQUIVALENCED in the PARSE routine. This routine is really just eliminating the blank fields and restructuring the method by which the fields are stored. The restructuring is done to aid in later command processing. All of the defined locations (from one to NDXVAL) in the NCODE array are checked for valid field types (non-zero). If a valid field is located, its field type is stored in the first word of the LITSTR array, and the value of the field in the second and third words. For numbers the numeric value is stored in the second word and the third word is left blank; for operators, functions or delimiters the associated number is stored in the second word, and the third word is left blank. For alpha fields the first four characters in internal IOSYM representation are stored in the second word and the next three characters in the third word. After all the fields have been processed, control returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLES	DEFINITION
I	Index that points to the current literal field in the NCODE array being processed.
IERR	Error indicator. If FALSE, then no error has occurred in this routine. If TRUE, then a fatal error has occurred in this routine.
IERR1	Error indicator. If FALSE, then no error has occurred up to the current time. If TRUE, then a fatal error has occurred.
LITMAX	Maximum number of literals that can be stored in the literal storage array (dimension of the LITSTR array).
LITNDX	Index pointing to the current position in the literal storage array being processed.

LITSTR	Literal storage array where the first word is the literal field type, and the second and third words contain the field value.
NCODE	Array that contains all of the current fields being processed.
NDXVAL	Number of fields that have been processed and stored in the NCODE and NVAL arrays.
NTALPC	Alpha field indicator.
NTERR	Error field indicator.
NTFUNU	User function field indicator.
NTINT	Integer number field indicator.
NTNUL	Null field indicator.
NVAL	Array containing the values of the current fields being processed.

5. I/O VARIABLES:

A. INPUT	LOCATION
IERR	F.P.
LITMAX	F.P.
NCODE	F.P.
NDXVAL	F.P.
NTALPC	F.P.
NTERR	F.P.
NTFUNU	F.P.
NTINT	F.P.
NTNUL	F.P.
NVAL	F.P.

B.	OUTPUT	LOCATION
	IERR1	F.P.
	LITNDX	F.P.
	LITSTR	F.P.

6. CALLING ROUTINE: PARSE

7. CALLED ROUTINES:

ARAYCK

EEROR

1. NAME: LNFDJ
2. PURPOSE: This routine advances the output file one line at a time.
3. METHOD: This routine prints a line feed character how many times specified.
4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Index counter.
JPRT	Device code where the output is written.
LF	Number of line feeds to process.
5. I/O VARIABLES:

A. INPUT	LOCATION
JPRT	/ALLJ/
LF	F.P.
B. OUTPUT	LOCATION
(none)	
6. CALLING ROUTINE: (Not used at this time)
7. CALLED ROUTINE: (None)

1. NAME: LOCEXJ
2. PURPOSE: This routine will return the minimum or maximum value of an array.
3. METHOD: Depending on whether a minimum or maximum value is desired XJ is set plus one or minus one, respectively. Then a search is done on the array to find the maximum or minimum value.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Do loop index.
LOCEXJ	Returns as the maximum or minimum value.
MINMAX	Minimum or maximum indicator. If equal to one, the maximum is found. If not equal to one, the minimum is found.
N NN	Length of the array to process.
XJ	Indicator. If minimum is needed, XJ equals minus one. If maximum is needed, XJ equals plus one.
Z	Array to process.

5. I/O VARIABLES:

A. INPUT	LOCATION
MINMAX	F.P.
NN	F.P.
Z	F.P.
B. OUTPUT	LOCATION
LOCEXJ	FUNCTION

6. CALLING ROUTINE: (None. Not used at this time)
7. CALLED ROUTINE: (None)

1. NAME: LOLFND - 9
2. PURPOSE: Certain commands (TRACE, NOTRACE, PRINT, OPR, PICK, SET, GLOBAL, WITH, CONO-CON9, CROSS, TERM, LOOP, IF, PUT, GET, and STAT) can have as part of their syntax a list of various kinds of literals, such as IOSYM functions, operators, variables, constants or TABLEs. This routine processes a List Of Literals (LOL) and stores the appropriate information in the NNWRK, CCONS and other arrays.
3. METHOD: The literals are processed by checking each field and processing it according to what type of field it is.

If the field is a delimiter, it is checked to see if it is a comma. Commas are not needed so they are skipped, and the next field is investigated. If it isn't a comma, then it must be an open or closed parenthesis; these are stored as internal IOSYM representations.

If the field is an integer or floating point number, it is stored in the CCONS array, and the pointer to the location in the CCONS array where the number is stored is added to 8000, and this coded identifier is stored in the work array. The next field is then investigated.

If the field is any IOSYM function (F, U or P), routine FNFND is called to process and store the function information. The next field is then investigated.

If the field is an alpha field, a check is done to see if the current command is STAT; if so, if working on the last argument in syntax then need to increment the number of arguments by one. Routine DVFND is called to determine what variable or TABLE the alpha field is. With the pointers and indicators returned, the variable or TABLE is coded and stored in the work array. The next field is then investigated.

If the field is an operator (any FORTRAN library function or +, -, =, *, /, or **), a check is done to see if the current command is GLOBAL, OPR, LOOP, PRINT, WITH or IF. If so, store the operator depending on whether it is a unitary operator or part of an expression. If the current command isn't one of those commands, store the coded operator in the work array and investigate the next field.

If the field is a null field, the coded null field indicator is stored in the work array, and the next field is investigated.

If the field wasn't a delimiter, number, IOSYM function, alpha, operator or null field, an error is processed, and control returns to the calling routine.

If the end of the defined literal storage array is reached and there are no more fields to process, the number of lists is incremented by one, and the number of words in the list is stored in the work array along with the number of lists. If the current command is PICK, SET, TERM, CROSS or IF, then routine RESEQ is called to resequence any arithmetic expressions into forward polish notation. Control is then returned to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
CCONS	Array that contains all of the constants in a NETWORK.
IERR	Error indicator. If FALSE, then no error has occurred in this routine. If TRUE, then a fatal error has occurred in this routine.
IERR1	Error indicator. If FALSE, then no error has occurred up to the present time. If TRUE, then a fatal error has occurred.
INDX1	Location in the work storage array where the number of lists in the list of literals is stored.
INDX2	Location in the work storage array where the number of locations in the work array used to store the list of literals is located.
INDX3	Number of lists of literals.
IN1	Location in the CCONS array where the number that was last stored is located.
IN2	Coded identifier of the variable or TABLE returned from the DVFND routine.
IPAR	Number of open parentheses that have been processed that haven't had corresponding closing parenthesis processed.

LAWRK	Location in work array to be used as the starting point for storing fields after processing an alpha field.
LDV	Indicator. If TRUE, indicates that a list of defined variables is being processed. If FALSE, only one defined variable is being processed by routine DVFND.
LITMAX	Maximum number of literals that can be stored in the literal storage array (dimension of the LITSTR array).
LITNDX	Index pointing to the last defined location used in the literal storage array (LITSTR).
LITSTR	Literal storage array where the first word is the literal field type and the second and third words contain the field value.
LLXWRK	Counter of the number of locations in current use in the NNWRK (work storage array). Also the last position in the NNWRK array used.
MMCNS	Maximum number of constants that can be stored in the constants array (dimension of the CCONS array).
MMWRK	Maximum number of elements in the NNWRK array (the dimension of the NNWRK array).
NDXLIT	Pointer to the current location in the literal storage array being investigated.
NDX1	The current command number being processed.
NNCNS	Number of constants currently stored in the CCONS array.
NNWRK	Work storage array. Contains the current command parsed into the various elements.

NTALPC	Alpha field indicator.
NTDLM	Delimiter field indicator.
NTFLP	Float point number field indicator.
NTFUNF	IOSYM menu function field indicator.
NTFUNP	IOSYM probability function field indicator.
NTFUNU	IOSYM user function field indicator.
NTINT	Integer number field indicator.
NTNUL	Null field indicator.
NTOPR	Operator field indicator.
NTTBL	Table field indicator.
NTVARD	Derivative variable field indicator.
NTVAR0	Independent variable field indicator.
NTVAR1	Primary variable field indicator.
NTVAR2	Secondary variable field indicator.

5. I/O VARIABLES:

A. INPUT	LOCATION
IERR	F.P.
INDX1	F.P.
INDX2	F.P.
INDX3	F.P.
LDV	F.P.
LITMAX	F.P.
LITNDX	F.P.
LITSTR	F.P.
LLXWRK	/I03/

MMCNS	/I04/
MMWRK	/I04/
NDXLIT	F.P.
NDX1	F.P.
NNCNS	/I03/
NNWRK	/I03/
NTALPC	F.P.
NTDLM	F.P.
NTFLP	F.P.
NTFUNF	F.P.
NTFUNP	F.P.
NTFUNU	F.P.
NTINT	F.P.
NTNUL	F.P.
NTOPR	F.P.
NTTBL	F.P.
NTVARD	F.P.
NTVARO	F.P.
NTVAR1	F.P.
NTVAR2	F.P.
B. OUTPUT	LOCATION
CCONS	/I02/
IERR1	F.P.
INDX3	F.P.
LLXWRK	/I03/

NDXLIT

F.P.

NNWRK

/I03/

6. CALLING ROUTINE: PARSE

7. CALLED ROUTINES:

ARRAYCK

DVFND

EEROR

FNFND

NUMSTR

RESEQ

1. NAME: LOPSTR - 57
2. PURPOSE: This command processes LOOP commands. It stores the appropriate LOOP information in the NSET array to be processed during an IOSYM simulation.
3. METHOD: If another command points to this command, then it is set to indicate where the current command starts. The LOOP command number is stored in the NSET array along with the identifier to be used for the LOOP index. If there isn't a start value or an end value for the index, an error is processed. The identifiers for the start and end values are stored in the NSET array. If there is an index increment value it is also stored, if not a value of one is assumed and it is stored. A pointer to the next command is set aside to be used later. The number of LOOP or IF commands stacked is incremented by one and the information for this LOOP is stored in the IIFPT array. The LABEL number is stored in the first word and the pointer to that LABEL is stored in the fourth word. Control then returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
CCONS	Array that contains all of the constants in a NETWORK.
IERR	Error indicator. If FALSE, then no error has occurred in this routine. If TRUE, then a fatal error has occurred in this routine.
IERR1	Error indicator. If FALSE, then no error has occurred up to the present time. If TRUE, then a fatal error has occurred.
IIFPT	Array that contains the LOOP command pointer. Word one contains the LABEL number. Words two and three are used just for IF commands. Word four contains the location in the NSET array to store the pointer to the LOOP LABEL.
IPTCMD	Pointer to the next command. Location in the NSET array where the pointer is located. If IPTCMD is zero, then no command points to the current command.

LLXSET	Counter of the number of locations in current use in the NSET array. Also the last position in the NSET array used.
LSET	Temporary storage for the number of NSET elements needed.
MMCNS	Maximum number of constants that can be stored (dimension of CCONS array).
MMFPT	Maximum number of IF or LOOP commands that can be stacked (dimension of IIFPT array).
MMSET	Maximum number of elements in the NSET array (the dimension of the NSET array).
NNCNS	The number of constants currently stored in the CCONS array. Pointer to the last position in the CCONS array used.
NNFPT	Number of stacked IF or LOOP commands. Pointer to the IIFPT array where the last LOOP or IF LABEL information was stored.
NNWRK	Work storage array. Contains the current command parsed into the various elements.
NSET	Array containing the link listed NETWORK.
NTTBL	Table field indicator.
NTVARD	Derivative variable field indicator.
NTVAR0	Independent variable field indicator.
NTVAR1	Primary variable field indicator.
NTVAR2	Secondary variable field indicator.

5. I/O VARIABLES:

A. INPUT	LOCATION
IERR	F.P.

	IPTCMD	F.P.
	LLXSET	/I03/
	MMCNS	/I04/
	MMFPT	/I04/
	MMSET	/I04/
	NNCNS	/I03/
	NNFPT	/I03/
	NNWRK	/I03/
	NTTBL	F.P.
	NTVARD	F.P.
	NTVAR0	F.P.
	NTVAR1	F.P.
	NTVAR2	F.P.
B.	OUTPUT	LOCATION
	CCONS	/I02/
	IERR1	F.P.
	IIFPT	/I02/
	IPTCMD	F.P.
	LLXSET	/I03/
	LSET	F.P.
	NSET	/ /

6. CALLING ROUTINE: PROCES

7. CALLED ROUTINES:

ARAYCK

EEROR

1. NAME: NAMFND - 26
2. PURPOSE: This routine checks an alpha field against a given array to see if it matches any names. If not, it is added to the array.
3. METHOD: Routine ALPFND is called to find an alpha field. The elements in array IRAY1 (the array to be checked) are compared against the alpha field for a match. If a match is found, the index is stored in the next position in the work array. If a match isn't found, the alpha field is stored in the IRAY1 array, and the number of fields in the array is incremented by one, and this pointer is stored in the work array. Control then returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
IDIM	First dimension of the array IRAY1 passed to this routine to check.
IERR	Error indicator. If FALSE, then no fatal error has occurred in this routine. If TRUE, then a fatal error has occurred in this routine.
IERR1	Error indicator. If FALSE, then no error has occurred up to the present time. If TRUE, then a fatal error has occurred.
IIER	Is the error message associated with exceeding the bounds of the IRAY1 array as defined by the calling routine.
IRAY1	The array passed to this routine to check the alpha fields against. The first two words contain the alpha name to be compared.
LITMAX	Maximum number of literals that can be stored in the literal storage array (dimension of the LITSTR array).
LITNDX	Index pointing to the last defined location used in the literal storage array (LITSTR).

LITSTR Literal storage array where the first word is the literal field type and the second and third words contain the field value.

LLXWRK Counter of the number of locations in current use in the NNWRK (work storage array) array. Also the last position in the NNWRK array used.

MMWRK Maximum number of elements in the NNWRK array (the dimension of the NNWRK array).

NDXLIT Pointer to the current location in the literal storage array being investigated.

NDX3 Index to the current name in the IRAY1 array being investigated.

NDX4 Number of defined fields in the IRAY1 array.

NDX5 Maximum number of fields in the IRAY1 array (one of the dimensions of the IRAY1 array).

NNWRK Work storage array. Contains the current command parsed into the various elements.

NOADD Indicator. If FALSE, then an alpha field was found by routine ALPFND. If TRUE, then an alpha field wasn't found.

NOADD1 Indicator. If TRUE, then an alpha field is desired, but not needed. If FALSE, then an alpha field is a must and if one isn't found, it is an error.

NTALPC Alpha field indicator.

NTDLM Delimiter field indicator.

NTINT Integer field indicator.

5. I/O VARIABLES:

A. INPUT	LOCATION
IDIM	F.P.

IERR	F.P.
IIER	F.P.
IRAY1	F.P.
LITMAX	F.P.
LITNDX	F.P.
LITSTR	F.P.
LLXWRK	/I03/
MMWRK	/I04/
NDXLIT	F.P.
NDX4	F.P.
NDX5	F.P.
NOADD1	F.P.
NTALPC	F.P.
NTDLM	F.P.
NTINT	F.P.
B. OUTPUT	LOCATION
IERR1	F.P.
IRAY1	F.P.
LITSTR	F.P.
NNWRK	/I03/
NOADD	F.P.
NOADD1	F.P.

6. CALLING ROUTINES:

PARSE

TBLFND

7. CALLED ROUTINES:

ALPFND

ARRAYCK

EEROR

1. NAME: NETSUM - 76
2. PURPOSE: This routine does NETWORK summaries after processing a NETWORK or CHANGE command.
3. METHOD: This routine processes a number of output pages. The method for each one is described below.

A. NETWORK Summary Page

The NETWORK name is identified and decoded by calling UNCO1. This first page is written and includes the page number (NPAG), the NETWORK name (IRAY(I), I=1, 7), number of NODEs in the NETWORK (NNNODE), the number of ARCs in the NETWORK (NNARC), the number of OPERATIONS in the NETWORK (NNOP), the number of primary (NNV1) and secondary (NNV2) variables in the NETWORK and the number of TABLEs in the NETWORK (NNXTB).

B. NODE Summary Page

Each NODE, the name, the internal IOSYM number, and the number of ARCs leaving it is determined. These are printed out as the number (I), name (IRAY(K), K=1, 7) and the number of ARCs leaving (IN1).

C. ARC Summary Page

Each NODE that has ARCs leaving it is processed. The ARCs are processed according to PICK order. The ARC name is decoded; the OPERATION name is found and decoded; the end NODE is identified and decoded; the start NODE name is decoded; the PICK number is identified, and the ARC status is determined. This information is printed out as: start NODE name (IRAY(K3), K3=15, 21), pick number (IN2), the internal ISOYM ARC number (I), the ARC name (IRAY(K3), K3=1, 7), OPERATION name (IRAY(K3), K3=8, 14), the END NODE name (IRAY(K3), K3=22, 28) and the ARC status (IRAY(K3), K3=29, 36).

D. OPERATION Summary Page

For each OPERATION the OPERATION name is decoded. All the ARCs are searched to see if any use the OPERATION. The OPERATION information is printed as internal IOSYM OPERATION number (I), the OPERATION name (IRAY(KK), KK=1, 7), and the ARC name (IRAY(KK), KK=8, 14).

E. VARIABLE Summary Page

The independent variable name is decoded; its starting value is found, and the information printed as internal IOSYM variable number (IZERO), the variable name (IRAY(K), K=12, 18) and the initial value (AN1).

The primary variables are processed next. Each variable name is decoded, the initial value found, and the information printed as internal IOSYM variable number (I), the variable name (IRAY(K), K=12, 18), and the initial value (AN1).

The secondary variables are processed last. Each variable name is decoded, the initial value identified, and the information printed as internal IOSYM variable number (I), the variable name (IRAY(K), K=12, 18) and the initial value (AN1).

F. TABLE Summary Page

The TABLE name is first decoded. The number of elements in the TABLE is calculated. A check is done to see if there are enough lines left on the page to print all of the current TABLE's information; if not, skip to the next page. The TABLE information is written out at five TABLE elements per line as TABLE number (I), TABLE name (IRAY(K), K=1, 7), number of TABLE elements (IN3), and first five element values (TTABL(K,2), K=IN1, IN4). If after printing out the first five elements there are still more, they are printed as each set of five elements (TTABL(K,2), K=KK, IN5).

G. STATISTICS Summary Page

The TABULATE statistics are processed first. The time option is determined, the variables are decoded and what type of TABULATE statistics type is determined. The TABULATE statistics part of the STATISTICS summary page is written as the TABULATE number (I), the time option (IRAY(K), K=1, 3), and the four variables and their TABULATE type (IRAY(K), K=4, 63).

The REPORT statistics are processed next. The REPORT variables are processed six at a time. The six variable names are decoded and written (IRAY(K), K=1, 90).

The COLLECT statistics are processed last. The COLLECT statistics are broken into two parts.

The first part is the COLLECT variables. They are processed six at a time. The variable names are decoded and written out (IRAY(K), K=1, 90).

The COLLECT OPERATIONS are processed next. The COLLECT OPERATIONS are processed ten at a time. The OPERATION names are decoded and written out (IRAY(K), K=1, 70).

H. PLOT Summary Page

Each plot number is processed one at a time. If the current plot number is active, the variables to be PLOTEd are decoded, and

the plot information is written out as plot number (I), the independent variable (IRAY(K), K=1, 15) and five dependent variables (IRAY(K), K=16, 90).

Control then returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
AN1	The initial value for the independent primary and secondary variables.
I	Index and counter for processing most of the arrays.
IAR	OPERATION number on the current ARC being checked to see if it uses the current OPERATION.
IARCST	Pointer to the location in the NSET array where the current ARC starts.
IAR1	Indicator to show whether an OPERATION has been processed. 1 = have processed OPERATION 0 = haven't processed OPERATION
ICHAR	Array that contains the internal IOSYM representation of all the IOSYM recognizable characters. Locations 1-26 are A-Z 27-36 are 0-9 37 is period or decimal point 38-43 is *, *, /, +, -, = 44-48 is), (, ,, (blank), \$ 49-50 not used.
IIRPT	Array containing the variable identifiers of the REPORT variables.
IN1	Pointer used for various purposes.
IN10	Start NODE number.
IN2 IN3 IN4 IN5	Pointers used for various purposes.

IPICST	Location in NSET where the current PICK command being processed begins.
IRAY	Array that contains a number of names or words stored as one character per word.
IRAY1	Array that contains a name or word stored as one character per word.
IWT	Number of blanks needed to fill a word.
IZERO	The value of integer number zero.
J	Pointers used for various purposes.
JJ	
JJJ	
K	
K1	
K3	
MMLINE	Maximum number of lines that can fit on one computer page. (Set in USRDAT.)
MMPLLOT	Maximum number of plots allowed. (Set in USRDAT, also the dimension of NNPLLOT.)
MMTABU	Maximum number of TABULATIONS allowed. (Set in USRDAT, also the dimension of NNTABU.)
NLINE	Number of lines left on a computer page that can be written on.
NNARC	Number of defined ARCs.
NNCOL	Number of COLLECT variables.
NNCOLC	IOSYM identifiers of all the COLLECT variables.
NNCOP	Number of COLLECT OPERATIONS.
NNMNET	Array that contains in the first two words the internal IOSYM representation of the NETWORK name.
NNNODE	Number of defined NODEs.

NNOP	Number of defined OPERATIONS.
NNPLOT	Number of defined PLOTS.
NNPTAC	Array that contains in the first two words the internal IOSYM representation of the ARC names, and the third word points to the location in the NSET array where the ARC starts.
NNPTB	Array that contains in the first two words the internal IOSYM representation of the TABLE names, and the third word points to the location in the TTABL array where the TABLE elements for this TABLE are stored.
NNPTND	Array that contains in the first two words the internal IOSYM representation of the NODE names, the third word points to the location in the NSET array where the NODE starts, and the fourth word indicates whether this NODE has been replaced.
NNPTOP	Array that contains in the first two words the internal IOSYM representation of the OPERATION names, the third, fourth and fifth words indicate where the TYPE S, I and E commands start in the NSET array for this OPERATION respectively. The sixth word indicates whether this OPERATION is COLLECTed or not. If the sixth word is greater than zero, then it is COLLECTed.
NNPT0	Contains the internal IOSYM representation of the independent variable name.
NNPT1	Contains the internal IOSYM representation of the primary variable names.
NNPT2	Contains the internal IOSYM representation of the secondary variable names.
NNRPT	Number of REPORT variables.

NNTABU

The first word indicates the time option to be used.

1 = time option off
2 = time option on

The second, fourth, sixth and eighth words are the TABULATE variables, with the third, fifth, seventh, and ninth being the report type for the four variables, respectively.

NNV1

Number of defined primary variables.

NNV2

Number of defined secondary variables.

NNXTB

Number of defined TABLEs.

NPAG

The NETWORK summary page currently being worked on.

NPRNT

The file number of the device where the output is to be printed.

NSET

Array that contains the link listed NETWORK.

NWT

Number of letters in a decoded name.

TTABL

Array containing the initial and current values of the TABLE elements.

TTIME1

Initial value of the independent variable.

5. I/O VARIABLES:

A. INPUT

LOCATION

ICHAR

/ME1/

IIRPT

/I01/

IRAY

F.P.

IRAY1

F.P.

MMLINE

/I04/

MMPL0T

/I04/

MMTABU	/I04/
NNARC	/I03/
NNCOL	/I03/
NNCOLC	/I01/
NNCOP	/I03/
NNMNET	/I03/
NNNODE	/I03/
NNOP	/I03/
NNPLOT	/I01/
NNPTAC	/I02/
NNPTB	/I02/
NNPTND	/I02/
NNPTOP	/I02/
NNPT0	/I02/
NNPT1	/I02/
NNPT2	/I02/
NNRPT	/I03/
NNTABU	/I01/
NNV1	/I03/
NNV2	/I03/
NNXTB	/I03/
NPRNT	/GCOM1/
NSET	/ /
NWT	F.P.
TTABL	/I02/
TTIME1	/I02/

B. OUTPUT

LOCATION

(None)

6. CALLING ROUTINES:

ANET

SAVE

7. CALLED ROUTINES:

UNCODE

UNCO1

VVARBL

1. NAME: NNODST - 538
2. PURPOSE: This routine does the NODE processing and the next ARC setup during an IOSYM simulation.
3. METHOD: The IOSYM TABLE NV is initialized to zeros. If a TRACE is being done, routine TTRACE is called to process the TRACE information. The current NODE indicator and pointers are setup, and the commands at this NODE are performed. The PICKs for the ARCs leaving this NODE are examined, and the next ARC is selected. If there are no active outgoing ARCs, the simulation is ended. If there are active ARCs, the PICK functions are evaluated, and the first one that is satisfied identifies the next ARC. Control then returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Pointer to the location in the NSET array where the pointer to the next PICK command is located.
IACTV	Indicator. If less than zero, then there was no active ARC. If equal to zero, the next ARC has been identified. If greater than zero, then there were active ARCs but none with satisfied PICK commands.
IERR1	Error indicator returned from routine EEROR but not used in this routine.
IINXAC	ARC number of the next ARC to process.
IINXND	NODE number of the next NODE to process.
INOD	Current NODE number.
LASTP	Pointer to where the last PICK command processed is located in the NSET array.
LCMD	Pointer to where the first command on the NODE begins in the NSET array.
LLTRAC	Location in the NSET array where the TRACE command information is located. If LLTRAC is less than or equal to zero, there is no current TRACE command.

LLUCSL	Array that contains the lengths of the various command uniform command sequences.
LPICK	Location in the NSET array where the current PICK command starts.
LSTR	Pointer to the TTABL array where the NV TABLE elements are stored.
LUCS	Pointer to where the ARC or NODE command starts.
MSTOP	If equals minus one, the end of the simulation has been reached (see GASP book for more information).
NNPTB	Array that contains in the first two words the TABLE names, and the third word points to the location in the TTABL array where the TABLE elements are stored.
NNPTND	Array that contains in the first two words the NODE names; the third word points to the location in the NSET array where the NODE uniform command sequence starts, and the fourth word indicates if the NODE has been replaced or not.
NSET	Storage array that contains the link listed NETWORK.
TTABL	Array containing all of the TABLE element values.
VAL	Value of the evaluated PICK command.

5. I/O VARIABLES:

A. INPUT	LOCATION
IERR1	F.P.
IINXAC	/I03/
IINXND	/I03/
LLTRAC	/I03/

LLUCSL	/I02/
NNPTB	/I02/
NNPTND	/I02/
NSET	/ /
TTABL	/I02/
B. OUTPUT	LOCATION
IINXAC	/I03/
MSTOP	/GCOM1/

6. CALLING ROUTINES:

EVNTS
INTLC

7. CALLED ROUTINES:

DDOCMD
EEROR
EEVAL
EEVLTF
NNXNOD
TTRACE
ZZSET

1. NAME: NNXNOD - 528
2. PURPOSE: This routine processes NODE replacements if replacements are necessary.
3. METHOD: This routine checks to see if the next NODE to perform is replaced. If so, then it is replaced. The new NODE is then checked to see if it is replaced. This continues until a NODE isn't replaced and that NODE is the next NODE. Control then returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
ICNT	Counter to make sure there isn't a circular reassignment of NODEs. If ICNT is greater than NNNODE (number of NODEs), then must be circular.
IERR1	Error indicator returned from routine EEROR but not used in this routine.
IINXND	NODE number of the next NODE to process.
IXNOD	The original next NODE to process.
NNNODE	Number of NODEs in the current NETWORK.
NNPTND	Array that contains in the first two words the NODE name; the third word points to the location of the uniform command sequence of the NODE, and the fourth word indicates if the NODE is replaced or not. If the fourth word is less than or equal to zero, then the NODE isn't replaced. If it is greater than zero, then it is the NODE number of the replacement NODE.

5. I/O VARIABLES:

A. INPUT	LOCATION
IERR1	F.P.
IINXND	/I03/
NNNODE	/I03/
NNPTND	/I02/

B. OUTPUT

LOCATION

IINXND

/I03/

6. CALLING ROUTINE: NNODST

7. CALLED ROUTINE: EEROR

1. NAME: NODSTR - 34
2. PURPOSE: This routine processes the NODE command. It sets aside locations, for the first PICK pointer, the number of ARCs leaving this NODE, and a pointer to the commands to be processed on the NODE, in the NSET array.
3. METHOD: A check is first done to see if this NODE has already been defined; if so, an error is processed, and control returns to the calling routine. The location of this NODE in the NSET array is stored in the NODE pointing array (NNPTND). The location of the number of ARCs leaving this NODE is set aside along with the pointer to the other commands to process on the NODE.

Each ARC is checked to see if it leaves the current NODE. If it does and it is the first ARC to leave this NODE, then the pointer to the pick on that ARC is stored in the uniform command sequence of the NODE (see previous paragraph). If the ARC leaves the NODE, the number of ARCs leaving this NODE is increased by one. Each ARC is checked until all ARCs have been processed.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
FIR2	Indicator to show whether an ARC leaving the given NODE has been processed. If FIR2(I) is TRUE, then an ARC leaving NODE I has not been processed. If FIR2(I) is FALSE, then an ARC leaving NODE I has been processed.
I	Index to indicate which ARC is being checked as leaving the current NODE.
IERR	Error indicator. If FALSE, then no error has occurred in this routine. If TRUE, then a fatal error has occurred in this routine.
IERR1	Error indicator. If FALSE, then no error has occurred up to the present time. If TRUE, then a fatal error has occurred.
INDX11	Location in the NSET array where the number of ARCs leaving the current NODE is stored.

INDX12	Location in the NSET array where the pointer to the first PICK command for an ARC leaving this NODE is stored.
IN1	Location in NSET where start of the given ARC is stored.
IN2	Current NODE number.
IPTCMD	Pointer to the next command. If IPTCMD is zero then no command uniform command sequence points to the current command.
LLXSET	Counter of the number of locations in current use in the NSET array. The last position in the NSET array used.
LSET	Temporary storage for the number of NSET elements needed.
MMSET	Maximum number of NSET elements available (dimension of NSET defined in USRDAT).
NNARC	Number of defined ARCs.
NNPTAC	Contains the ARC name in the first two words and the pointer to the location in the NSET array where the ARC starts, in the third word.
NNPTND	Contains the NODE name in the first two words, the pointer to the location in the NSET array where the NODE starts, and a number indicating if the NODE has been replaced.
NNWRK	Work storage array. Contains the current command parsed into the various elements.
NSET	Storage array where the link listed NETWORK is stored.
NTNUL	Null field indicator.

5. I/O VARIABLES:

A. INPUT	LOCATION
FIR2	F.P.
IERR	F.P.
LLXSET	/I03/
MMSET	/I04/
NNARC	/I03/
NNPTAC	/I02/
NNPTND	/I02/
NNWRK	/I03/
NSET	/ /
NTNUL	F.P.

B. OUTPUT	LOCATION
FIR2	F.P.
IERR1	F.P.
IPTCMD	F.P.
LLXSET	/I03/
LSET	F.P.
NNPTND	/I02/
NSET	/ /

6. CALLING ROUTINE: PROCES

7. CALLED ROUTINES:

ARRAYCK

EEROR

1. NAME: NUMFND - 3
2. PURPOSE: This routine checks the literal storage array for an integer or floating point number. If one is found, the pointer to the location in the CCONS array where it is located is returned; if one isn't found, a negative one is returned.
3. METHOD: This routine looks at the next location in the literal storage array for a number. Depending on what the field is, it does different processing.

If the field is a zero (indicating it has already been used) or a delimiter, the pointer to the literal storage array is increased by one,, and the next field is investigated.

If the field is a null field, the pointer to the CCONS array, where the value would be stored, is set to minus one to indicate that nothing is stored. Control returns to the calling routine.

If the field is a plus or minus sign, an integer value is set to minus one if a minus sign was found. The next field is then investigated. This indicator is then used to multiply the number to get the correct sign.

If the field is an integer or floating point number, the number is stored in the next location in the CCONS array, and the pointer to that location is stored in the work array.

If the field isn't a zero, delimiter, null, plus, minus, or number, an error is processed, and control returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
ALIT	Real number to store as floating point number.
CCONS	Array that contains all the constants in a NETWORK.
IERR	Error indicator. If FALSE, then no error has occurred in this routine. If TRUE, then a fatal error has occurred in this routine.
IERR1	Error indicator. If FALSE, then no error has occurred up to the present time. If TRUE, then a fatal error has occurred.

ILIT	Integer number to store in the CCONS array.
ISIGN	Negative or positive one used to put correct sign on the number to store by multiplying by ISIGN.
LITMAX	Maximum number of literals that can be stored in the literal storage array (dimension of LITSTR set in USRDAT).
LITNDX	Index pointing to the last defined location used in the literal storage array (LITSTR).
LITSTR	Literal storage array where the first word is the literal field type and the second and third words contain the field value.
LLXWRK	Counter of the number of locations in current use in the NNWRK (work storage array). Also, the last position in the NNWRK array used.
LOL	List of literal indicator. If TRUE, then a position in the literal storage array has already been identified as a number, and it just has to be stored.
MMCNS	Maximum number of constants that can be stored in the CCONS array (dimension of the CCONS array set in USRDAT).
MMWRK	Maximum number of work storage elements (the dimension of the NNWRK array set in USRDAT).
NDXLIT	Pointer to the current location in the literal storage array being investigated.
NDX6	Pointer that indicates what location in the CCONS array the number is stored. If NDX6 is minus one, that indicates a number wasn't found.

NNCNS	Number of constants that have been defined in the CCONS array. Also, the last position in the CCONS array used.
NNWRK	Work storage array. Contains the current command parsed into the various elements.
NTDLM	Delimiter field indicator.
NTFLP	Floating point number field indicator.
NTINT	Integer number field indicator.
NTNUL	Null field indicator.
NTOPR	Operator field indicator.

5. I/O VARIABLES:

A. INPUT	LOCATION
IERR	F.P.
LITMAX	F.P.
LITNDX	F.P.
LITSTR	F.P.
LLXWRK	/I03/
LOL	F.P.
MMCNS	/I04/
MMWRK	/I04/
NDXLIT	F.P.
NNCNS	/I03/
NTDLM	F.P.
NTFLP	F.P.
NTINT	F.P.

	NTNUL	F.P.
	NTOPR	F.P.
B.	OUTPUT	LOCATION
	CCONS	/I02/
	IERR1	F.P.
	NDXLIT	F.P.
	NNWRK	/I03/

6. CALLING ROUTINES:

DVFND
 PARSE
 TBLFND
 VARFND

7. CALLED ROUTINES:

ARAYCK
 EEROR

1. NAME: NUMSTR - 27
2. PURPOSE: This routine stores numbers in the CCONS array (constant storage array).
3. METHOD: The pointer to the CCONS array is increased by one, and the number is stored in the CCONS array. The pointer to the CCONS array is returned to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
ANUM	Real number to store as a floating point number.
CCONS	Array that contains all the constants in a NETWORK.
IERR	Error indicator. If FALSE, then no error has occurred in this routine. If TRUE, then an error has occurred in this routine.
IERR1	Error indicator. If FALSE, then no error has occurred up to the present line. If TRUE, then a fatal error has occurred.
IFLD	Field type of the number. 4 = integer number. 5 = floating point number.
INUM	Enters the routine as the number to store. Returned as the pointer to the CCONS array where the number is stored.
INUM1	Number to store in the CCONS array if the number is a floating point number. This variable (INUM1) is equivalent to ALIT which is the floating point number stored in the CCONS array.
MMCNS	Maximum number of constants that can be stored in the CCONS array (dimension of the CCONS array set in USRDAT).
NNCNS	Number of constants that have been defined in the CCONS array. Also, the last position in the CCONS array used.

5. I/O VARIABLES:

A. INPUT	LOCATION
IERR	F.P.
IFLD	F.P.
INUM	F.P.
MMCNS	/I04/
NNCNS	/I03/
B. OUTPUT	LOCATION
CCONS	/I02/
IERR1	F.P.
INUM	F.P.

6. CALLING ROUTINES: FUNC2
LOLFND

7. CALLED ROUTINE: ARAYCK

1. NAME: OPRSTR - 21
2. PURPOSE: This routine stores the uniform command sequence (UCS) for the OPR command which includes the command number, the OPERATION number, the number of substitutable OPR parameters which are TABLE elements or variable names, and also stores the parameters that are going to be inserted.
3. METHOD: If a PICK command hasn't been processed for the current ARC, then routine PCKCK is called to process the necessary information about PICKs. The OPR command number is stored in the NSET array. A pointer from the ARC UCS is stored to point to the start of this OPR command. The OPERATION number of this OPR command is stored in the NSET array. A location is set aside to store the number of substitutable parameters. If there are any substitutable parameters to store, they are stored, and the number of parameters processed is stored in the NSET array. Control then returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Index used to show what location in the work array is being processed as a substitutable parameter.
ID	Dimension of the IPICK array.
IERR	Error indicator. If FALSE, then no error has occurred in this routine. If TRUE, then a fatal error has occurred in this routine.
IERR1	Error indicator. If FALSE, then no error has occurred up to the present time. If TRUE, then a fatal error has occurred.
INDX4	Location in the NSET array where the pointer to the OPR command is to be stored.
IND1	Location in the NSET array where the number of substitutable parameters is to be stored.
IPICK	Location of the pointer in a PICK command for an ARC leaving a NODE to the next ARC leaving the same NODE.

ISNODE	Current NODE position in the NNPTND array.
LEN	Number of substitutable parameters in the NNWRK array to process.
LLXSET	Counter of the number of locations in current use in the NSET array. Also, the last position in the NNWRK array used.
LXSET	Temporary storage for the number of NSET elements needed.
MMSET	Maximum number of NSET elements needed (dimension of the NSET array set in USRDAT).
NNWRK	Work storage array. Contains the current command parsed into the various elements.
NSET	Storage array where the link listed NETWORK is stored.
NTNUL	Null field indicator.
OPR	Indicator. If TRUE, then an OPR command for the current ARC has been processed. If FALSE, then an OPR command hasn't been processed.
PICK	Indicator. If TRUE, then a PICK command on the current ARC has been processed. If FALSE, then a PICK command wasn't processed.

5. I/O VARIABLES:

A. INPUT	LOCATION
ID	F.P.
IERR	F.P.
INDX4	F.P.
IPICK	F.P.
ISNODE	F.P.

	LLXSET	/I03/
	MMSET	/I04/
	NNWRK	/I03/
	NTNUL	F.P.
	PICK	F.P.
B.	OUTPUT	LOCATION
	IERR1	F.P.
	LLXSET	/I03/
	LXSET	F.P.
	NSET	/ /
	OPR	F.P.

6. CALLING ROUTINE: PROCES

7. CALLED ROUTINES:

ARRAYCK
EEROR
PCKCK

1. NAME: OUTPUT - 546
2. PURPOSE: This routine finishes a run, ends the PLOT and the TABULATE tapes, and finishes statistics.
3. METHOD: A sentinel is put at the end of both the PLOT and the TABULATE tapes. Routines CCOLC, RREPRT, PPRTAB, PPRPLT, COLCT, TIMST, and HISTO are called to finish and process any statistics. Routine UOTPUT, a user written routine, is called to do any user processing. Control then returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
ANUM	Sentinel number to put on the end of the PLOT tape.
IITPS	Array that contains the tape numbers of the different tapes. IITPS(4) is the TABULATE tape number, and IITPS(3) is the PLOT tape number.
INUM	Sentinel number to put on the end of the PLOT and TABULATE tapes.
IONE	Integer number one, written on the end of the PLOT tape.
ITP	Tape number of the tape writing on.
NNCLT	Number of GASP COLCT variables.
NNFIL	Number of GASP files.
NNHIS	Number of GASP histograms.
NNPLT	Number of GASP plots.
NNQ	Current number of entries in the GASP file.
NNSTA	Number of GASP TIMST and TIMSA variables.
XX	Dummy array passed to GASP routines.
ZERO	Floating point number zero, written on the end of the PLOT tape.

5. I/O VARIABLES:

A. INPUT	LOCATION
IITPS	/I01/
NNCLT	/GCOM4/
NNFIL	/GCOM1/
NNHIS	/GCOM4/
NNPLT	/GCOM4/
NNQ	/GCOM1/
NNSTA	/GCOM4/
B. OUTPUT	LOCATION
XX	F.P.

6. CALLING ROUTINES:

EEROR
GASP (the main routine in the GASP program)

7. CALLED ROUTINES:

CCOLC
COLCT (a GASP routine)
GPLOT (a GASP routine)
HISTO (a GASP routine)
IISSET
PPRPLT
PPRTAB
RREPRT
TIMST (a GASP routine)
UOTPUT (a user written routine)
USTAT (a user written routine)

1. NAME: PACWJ
2. PURPOSE: This routine does the free form input reading and packing into the WORDJ array.
3. METHOD: This routine reads the input records and processes one field on that record for each call to this subroutine. If it doesn't have a field, it reads another input record. Commas and blanks are interpreted as delimiters. A field may not continue on another card.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Index and counter used for various purposes.
IAP	This variable contains the representation of a quote.
IBK	This variable contains the representation of a blank.
ICM	This variable contains the representation of a comma.
IDL	This variable contains the representation of a dollar sign.
IEND	Number of end of files used.
IFFJ	Next field to search.
IFRST	Location in the LWJ array where the first character for the current field is located.
IIPS	Array containing the device code numbers. IIPS(5) is the scratch tape device code number.
ILAST	Location in the LWJ array where the last character for the current field is located.
IPLUS	This variable contains the representation of a plus sign.
IQUOT	Number of quote marks found in a field.

ITP	Scratch tape device code.
I10	Indicator. If equal to zero, then pack into two words. If equal to one, pack up to five words.
JPRT	Device code number for output.
JRD	Device code number for input.
K	Index pointer to the LWJ array.
L	Location in the LWJ array where the last character for the current field is located.
LWJ	Array containing the input record.
NBKS	Number of blanks in the packed word.
NCHAR	Number of characters in the packed word.
NFFJ	Location of the last non-blank character in the LWJ array.
WORDJ	Array containing the packed words.

5. I/O VARIABLES:

A. INPUT	LOCATION
IFFJ	/FFJ/
IITPS	/I01/
I10	F.P.
JPRT	/ALLJ/
JRD	/ALLJ/
LWJ	/FFJ/
NFFJ	/FFJ/
B. OUTPUT	LOCATION
IFFJ	/FFJ/
LWJ	/FFJ/

NFFJ

/FFJ/

WORDJ

/FFJ/

6. CALLING ROUTINES: FFAJ
FFDJ
FFIJ
FFLJ
FFRJ

7. CALLED ROUTINE: EEOF

1. NAME: PARSE - 1
2. PURPOSE: This routine is the main routine for the input processing section of IOSYM. It calls the scanning routines, set-up routines, and does the bookkeeping for the different commands.
3. METHOD: Routine SCAN is called to read the input card and build the different fields. If the end of input processing has been reached, then control returns to the calling routine. Routine KWFND is called to identify all of the keywords in the fields that were stored by the SCANNing routine. Routine LIBFND is called to identify all of the FORTRAN library functions, routine FUNFND is called to identify all of the IOSYM functions (F, U, or P), and routine LITFND is called to restructure the fields and eliminate blanks left by the removal of the keywords. The command number of the input record is identified and checked to see if it is a valid command at the current command level. A check is done to see if the command changes the current level; if so, the level is changed. A check is done to see if there is other syntax for the command to be processed. If so, the syntax is identified from the NTSFMT table. Each item in the syntax is processed, depending on what is needed. At this time multiple keywords on one input record are not allowed. The items in the syntax are processed one at a time.

If an ARC, NODE, LABEL, OPERATION, or VARIABLE name is needed, then routine NAMFND is called with the appropriate parameters. If a matching name isn't found, the alpha field is added to the appropriate name list.

If a TABLE command is being processed, routine TBLFND is called to process the syntax for it due to its complexity. If a VARIABLE command is being processed, routine VARFND is called for the same reason.

If a defined variable or number or list of defined variables and numbers is needed, routine DVFND is called to process that syntax.

If a single number or list of numbers is needed, then routine NUMFND is called to process the numeric fields and store the appropriate information.

If an IOSYM function has to be processed, routine FNFND is called to process all of the fields in those type of functions.

If a list of literal fields (this could be any combination of alpha, function, operator, numeric, etc. fields), routine LOLFND is called.

If an alpha field with a single character is to be processed, routine ALPSRH is called to process it.

After all of the syntax for the current command has been processed and the information stored in the proper temporary locations, routine CHOIC2 is called to do all of the setting up of the command and link listing the command with the rest of the NETWORK. This is done by calling numerous other routines from routine CHOIC2 and its successors.

After control has returned from CHOIC2, a decision is made about the next step. An indicator IICHOI is returned from CHOIC2 and is used in this determination. If IICHOI is two, that indicates that there is more input to process. If IICHOI is minus two, the decision to read more input can't be made here so control is returned to the calling routine where the decision can be made. If IICHOI is three, then it is time to stop reading input and time to simulate the NETWORK so control returns to the calling routine for further processing. If IICHOI is four, we are done with all of IOSYM processing so control returns to the calling routine. If IICHOI is minus four, then an error in data input has occurred so control returns to the calling routine for final processing before the IOSYM processing is halted.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
CCONS	Array containing all of the constants in a NETWORK.
DUMP	Indicator. If TRUE, then a printout of all of the useful IOSYM arrays is to occur. If FALSE, then a DUMP is not to occur.
FIRST	Indicator. If TRUE, then haven't finished processing the first IOSYM input record yet. If FALSE, then the first input record has already been processed.
FIR2	Indicator to show whether an ARC leaving the NODE I has already taken place. If FIR2(I) is TRUE, then an ARC leaving NODE I has not been processed. If FIR2(I) is FALSE, then an ARC leaving NODE I has been processed.
I	For the most part, it is a pointer to the NTSFMT array where the syntax for a given command is being processed.

IARGF	The location in the NTSFMT array where the first item in the command syntax is located.
IARGL	The location in the NTSFMT array where the last item in the command syntax is located.
ICLEVL	Is an array that contains the current command hierarchy from level one to level LEVEL.
IEND	Indicator. If TRUE, then the last input record has been read. If FALSE, then the last input record hasn't been processed yet.
IERR	Error indicator. If FALSE, then an error has not occurred in this routine. If TRUE, then a fatal error has occurred in this routine.
IERRFF	End of file error indicator. If TRUE, then the end of the input file was reached prematurely.
IERR1	Error indicator. If FALSE, then an error hasn't occurred up to the present time. If TRUE, then a fatal error has occurred.
II	Index used when storing multiple numbers to show which one is being processed.
IICHOI	Determines what processing is to occur next.

2	= More IOSYM input to process.
-2	= Decision to read more input can't be made here so return to calling routine.
3	= Time to start an IOSYM simulation.
4	= IOSYM processing is finished so return to the calling routine.

-4 = There was an error in IOSYM
input so stop processing
and return to calling
routine.

INDX1	Location in the work storage array to be used later to store the number of arguments in the list of literal storage.
INDX2	Location in the work storage array where the number of locations in the work array used to store the list of literals is located.
INDX3	Number of lists of literals.
INTS	Contains the type of field in the syntax to be processed.
INUM	Indicator if: 0 = Only alpha-numeric defined variables are allowed. 1 = Either alpha-numeric defined variables or numbers are allowed.
IN1	Indicator. If zero, then only processing one number at a time. If greater than zero, then are processing IN1 numbers at a time.
IPICK	Location of the pointer in a PICK command for an ARC leaving a NODE to the next ARC leaving the same NODE.
ITYP1	Indicates what type of command is now being processed. 0 = Not doing TYPE commands. 1 = Start TYPE commands. 2 = Immediate TYPE commands. 3 = End TYPE commands.

KWSTR	Array where the keywords are stored after they have been identified.
KYWORD	The first two words contain the keyword names in internal IOSYM representation; the third word contains the level to which this command brings control; the fourth word is the levels under which this command is valid; and the last word points to the NTSFMT array where the syntax for this command is located.
LDV	Indicator. If TRUE, then processing a list of defined variables. If FALSE, then need only one defined variable.
LEVEL	Indicates what level of command hierarchy is being processed. There are five levels: 0th, 1st, 2nd, 3rd, and 4th.
LIBFUN	Contains the internal IOSYM representation of the FORTRAN library function names.
LITMAX	Maximum number of literals that can be stored in the literal storage array (dimension of the LITSTR array).
LITNDX	Index pointing to the current position in the literal storage array being processed.
LITSTR	Literal storage array where the first word is the literal field type and the second and third words contain the field value.
LLARY	Array containing the LABEL names.
LLXWRK	Counter of the number of locations in current use in the NNWRK (work storage array). Also, the last position in the NNWRK array used.
LN	Indicator. If TRUE, then a list of numbers is to be processed. If FALSE, then only one number is to be processed.

LOL	Indicator. If TRUE, then a list of literals is desired. If FALSE, then a list of literals is not needed.
LOPR	Indicator. If TRUE, then a list of OPERATION names is needed. If FALSE, only one OPERATION name is needed.
MAXKW	The number of valid keywords recognized by IOSYM.
MAXLIB	The number of FORTRAN library functions recognized by IOSYM.
MMARC	Maximum number of ARCs allowed in a NETWORK (dimension of the NNPTAC array set in USRDAT).
MMARY	Maximum number of LABELs allowed in a section of a NETWORK (dimension of the LLARY array set in USRDAT).
MMNODE	Maximum number of NODEs allowed in a NETWORK (dimension of the NNPTND array set in USRDAT).
MMOP	Maximum number of OPERATIONs allowed in a NETWORK (dimension of the NNPTOP array set in USRDAT).
MMWRK	Maximum number of work storage locations available (dimension of NNWRK set in USRDAT).
MNODE	Maximum number of NODEs in a NETWORK (dimension of FIR2 set in PARSE).
NCODE	Array containing the type of fields that were processed from the input record.
NDXLIT	Pointer to the current position in the literal storage array LITSTR being investigated.
NDXVAL	The number of fields that are stored in the NVAL and NCODE arrays.
NDX1	Current command number.

NDX2	Pointer to location in the NTSFMT array where the information for the command syntax is stored. If NDX2 is zero, then command is a command with no syntax.
NDX3	Number of items in the command syntax.
NNARC	Number of defined ARCs in the NETWORK.
NNARY	Number of defined LABELs in the current section of IOSYM.
NNCNS	Number of defined constants in the NETWORK.
NNNODE	Number of defined NODEs in the NETWORK.
NNOP	Number of defined OPERATIONS in the NETWORK.
NNPTAC	The first two words of this array contain the ARC name, and the third word contains a pointer to the location in the NSET array where the ARC uniform command sequence begins.
NNPTND	The first two words of this array contain the NODE name, and the third word contains a pointer to the location in the NSET array where the ARC uniform command sequence begins.
NNPTOP	The first two words of this array contain the OPERATION name. The third, fourth, and fifth words contain the pointers to the TYPE S, I, and E commands that occur under the OPERATION, respectively. The sixth word contains an indicator to show whether the COLLECT variables are COLLECTed on this OPERATION. If this is a positive number, then the variables are COLLECTed; if less than or equal to zero, they are not.
NNWRK	Work storage array. Contains the current command parsed into the various elements.

NOADD	Indicator. If FALSE, then an alpha field was found. If TRUE, then an alpha field wasn't found.
NPRNT	The output device code number for writing output (set in USRDAT).
NTALP	Alpha field indicator.
NTALPC	Alpha field indicator.
NTDLM	Delimiter field indicator.
NTERR	Error field indicator.
NTFLP	Floating point number field indicator.
NTFUNF	IOSYM menu function field indicator.
NTFUNP	IOSYM probability function field indicator.
NTFUNU	IOSYM user function field indicator.
NTINT	Integer number field indicator.
NTNUL	Null field indicator.
NTOPR	Operator field indicator.
NTSFMT	Array that contains the syntax for all the commands. The fifth word in the KEYWORD array points to the location in the NTSFMT array where the syntax for that command begins.
NTTBL	Table field indicator.
NTVARD	Derivative variable field indicator.
NTVARO	Independent variable field indicator.
NTVAR1	Primary variable field indicator.
NTVAR2	Secondary variable field indicator.
NVAL	The array containing the values of the fields read from the input record.

5. I/O VARIABLES:

A. INPUT	LOCATION
CCONS	/I02/
DUMP	/ME1/
FIRST	/ME1/
ICLEVL	/ME1/
IERR	F.P.
IERR1	F.P.
IICHOI	F.P.
KWSTR	F.P.
KEYWORD	/ME1/
LEVEL	/ME1/
LIBFUN	/ME1/
LITMAX	F.P.
LITNDX	F.P.
LITSTR	F.P.
LLXWRK	/I03/
MMARC	/I04/
MMARY	/I04/
MMNODE	/I04/
MMOP	/I04/
MMWRK	/I04/
NCODE	F.P.
NDXLIT	F.P.
NDXVAL	/ME1/

NNARC	/I03/
NNARY	/I03/
NNCNS	/I03/
NNNODE	/I03/
NNOP	/I03/
NNPTAC	/I02/
NNPTND	/I02/
NNPTOP	/I02/
NNWRK	/I03/
NOADD	F.P.
NPRNT	/GCOM1/
NTALP	F.P.
NTALPC	F.P.
NTDLM	F.P.
NTERR	F.P.
NTFLP	F.P.
NTFUNF	F.P.
NTFUNP	F.P.
NTFUNU	F.P.
NTINT	F.P.
NTNUL	F.P.
NTOPR	F.P.
NTSFMT	/ME1/
NTTBL	F.P.
NTVARO	F.P.

NTVAR1	F.P.
NTVAR2	F.P.
NVAL	F.P.
B. OUTPUT	LOCATION
FIR2	F.P.
ICLEVL	/ME1/
IEND	F.P.
IERR	F.P.
IERR1	F.P.
IICHOI	F.P.
INDX1	F.P.
INDX2	F.P.
INDX3	F.P.
LDV	F.P.
LEVEL	/ME1/
LITMAX	F.P.
LITNDX	F.P.
LLXWRK	/I03/
LN	F.P.
LOL	F.P.
LOPR	F.P.
MAXKW	F.P.
MAXLIB	F.P.
MNODE	F.P.
NDXLIT	F.P.

NDXVAL	/ME1/
NDX1	/ME1/
NNCNS	/I03/
NNWRK	/I03/
NOADD	F.P.

6. CALLING ROUTINE: CHOICE

7. CALLED ROUTINES:

ALPSRH
ARRAYCK
CHOIC2
DVFND
EEROR
FNFND
FUNFND
IISSET
KWFND
LIBFND
LITFND
LOLFND
NAMFND
NUMFND
SCAN
TBLFND
VARFND

1. NAME: PCKCK - 44
2. PURPOSE: This routine processes the PICK information when there is no PICK command processed on an ARC.
3. METHOD: This routine checks to see if there are other ARCs leaving the NODE that the current ARC leaves. If so, an error is processed because if more than one ARC leaves a NODE, they must all have PICK commands. Pointer to this ARC is stored to check future ARCs. The NODE that this ARC leaves is checked to see if it is defined. If so, then a pointer in the NODE uniform command sequence (UCS) is pointed to this PICK UCS. The PICK command number is stored in the NSET array. A zero is stored to indicate there is no PICK command on this ARC; a minus one is stored to indicate that there is no information in the PICK UCS; a zero is stored to indicate there is no truth condition, and last, another zero is stored to indicate there are no other ARCs leaving the NODE this ARC does. Control then returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
ID	Dimension of the FIR2 array.
IERR	Error indicator. If FALSE, an error in this routine hasn't occurred. If TRUE, an error has occurred in this routine.
IERR1	Error indicator. If FALSE, an error hasn't occurred up to the present time. If TRUE, then a fatal error has occurred.
IN3	Location in the NODE UCS where the number of ARCs leaving the NODE is stored.
IPICK	Location of the pointer in a PICK command for an ARC leaving a NODE to the next ARC leaving the same NODE.
ISNODE	Current NODE number.
LLXSET	Counter of the number of locations in current use in the NSET array. Also, the last position in the NSET array used.

LSET	Temporary storage for the number of NSET elements needed.
MMSET	Maximum dimension of the NSET array set in USRDAT.
NNPTND	The first two words contain the NODE name, and the third word is the location in the NSET array where the NODE is stored.
NSET	Storage array where the link listed NETWORK is stored.
PICK	Indicator. If TRUE, then a PICK command on the current ARC has been processed. If FALSE,, then a PICK command hasn't been processed.

5. I/O VARIABLES:

A. INPUT	LOCATION
ID	F.P.
IERR	F.P.
IERR1	F.P.
IPICK	F.P.
ISNODE	F.P.
LLXSET	/I03/
LSET	F.P.
MMSET	/I04/
NNPTND	/I02/
B. OUTPUT	LOCATION
IERR1	F.P.
IPICK	F.P.
LLXSET	/I03/
LSET	F.P.

NSET

/ /

PICK

F.P.

6. CALLING ROUTINES:

OPRSTR
TRMSTR
TYPSTR

7. CALLED ROUTINES:

ARAYCK
EEROR

1. NAME: PCKSTR - 29
2. PURPOSE: This routine processes the PICK command which includes storing the PICK command number, PICK number, pointer to location where the PICK expression starts, TRUTH condition, and the pointer to the next pick command.
3. METHOD: The PICK command number is first stored in the NSET array. A check is done to see if other ARCs leave the NODE this ARC does. If so, the last PICK command for that NODE is pointed to this PICK command. If not, the NODE this ARC leaves is pointed to this PICK command. The PICK number is stored next. A pointer to the PICK expression is set aside for later use. A truth condition is stored (if there isn't one, a -1 is stored). Set aside a pointer to the next PICK command leaving the same NODE. The pointer to the PICK expression is set to the next location in the NSET array. Routine RESCC is called to remove constants from the CCONS array that are no longer needed. Routine FUNSTR is called to store the PICK expression. Control then returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
CCONS	Array containing all of the constants in a NETWORK.
ID	Dimension of the FIR2 array.
IERR	Error indicator. If FALSE, then an error has not occurred in this routine. If TRUE, then a fatal error has occurred in this routine.
IERR1	Error indicator. If FALSE, then an error hasn't occurred up to the present time. If TRUE, then a fatal error has occurred.
INDX22	NSET array pointer that points to the location in the NSET array where the PICK expression starts.
IN1	NSET array pointer where information about the last PICK is stored.
IN2	Pointer to the CCONS array where the PICK number is stored.
IN3	NSET array pointer where information about the NODE, the current ARC leaves, is stored.

IN9	Number of CCONS array positions no longer needed. They can be reinitialized and reused.
IPICK	Location of the pointer in a PICK command for an ARC leaving a NODE to the next ARC leaving the same NODE.
ISNODE	Current NODE number.
LLXSET	Counter of the number of locations in current use in the NSET array. Also, the last position in the NSET array used.
LSET	Temporary storage for the number of NSET elements needed.
MMSET	Maximum dimension of the NSET array, set in routine USRDAT.
NNPTND	The first two words contain the NODE name and the third word is the location in the NSET array where the NODE is stored.
NNWRK	Work storage array. Contains the current command parsed into the various elements.
NSET	Storage array where the link listed NETWORK is stored.
NTNUL	Null field indicator.
PICK	Indicator. If TRUE, then a PICK command on the current ARC has been processed. If FALSE, then a PICK command wasn't processed.

5. I/O VARIABLES:

A. INPUT	LOCATION
CCONS	/I02/
ID	F.P.
IERR	F.P.
IERR1	F.P.

	IPICK	F.P.
	ISNODE	F.P.
	LLXSET	/I03/
	LSET	F.P.
	MMSET	/I04/
	NNPTND	/I02/
	NNWRK	/I03/
	NSET	/ /
	NTNUL	F.P.
B.	OUTPUT	LOCATION
	CCONS	/I02/
	IERR1	F.P.
	IPICK	F.P.
	LLXSET	/I03/
	LSET	F.P.
	NSET	/ /
	PICK	F.P.

6. CALLING ROUTINE: PROCES

7. CALLED ROUTINES: ARAYCK
EEROR
FUNSTR
RESCC

1. NAME: PIVARJ
2. PURPOSE: This routine prints a simple integer variable with an introductory label line.
3. METHOD: The array containing the label characters is identified (up to twenty characters), and the label and the variable are printed.
4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Do loop index.
ICHAR	Array containing the carriage control characters first word contains a blank and will skip one line before printing. The second word is a zero and will skip two lines before printing.
ICRTL	Determines which carriage control character to be printed.
JJ	Variable value to print.
JPRT	Device code to print output.
L, LEN	Length of the label array.
LINE	Array containing the label to print, four characters per word.

5. I/O VARIABLES:

A. INPUT	LOCATION
ICRTL	F. P.
JJ	F. P.
JPRT	/ALLJ/
LEN	F. P.
LINE	F. P.
B. OUTPUT	LOCATION
(None)	

6. CALLING ROUTINES: READJ

7. CALLED ROUTINE: (None)

1. NAME: PIVECJ
2. PURPOSE: This routine will print an integer vector with labels and an introductory line.
3. METHOD: First, the introductory line is printed, and then the vector values and labels are printed. Control then returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Index and counter used for various purposes.
IBLOCK	Determines which group of values is being printed out.
ICOL	Number of columns of data to print.
IROW	Number of rows of data to print.
IX	Array containing the vector values to print.
I72	Indicator to determine how many columns of data to print. If equal to zero, three columns are printed. If equal to one, five columns are printed per line.
J	Index and counter.
JPRT	Device code to print information on.
J2	Last column number to print on current line.
L	Number of A4 words in the array LINE.
LA	Single character label.
LEN	Number of A4 words in the array LINE.
LINE	Alpha string of less than sixty (terminal) or eighty (line printer) characters to be printed as an introductory label.
N	Number of vector values to print.
NI	

5. I/O VARIABLES:

A. INPUT	LOCATION
IX	/ALLJ/
I72	/ALLJ/
JPRT	/ALLJ/
LAB	F.P.
LEN	F.P.
LINE	F.P.
N	F.P.
B. OUTPUT	LOCATION
(None)	

6. CALLING ROUTINE: DDUMP

7. CALLED ROUTINE: (None)

1. NAME: PKMATJ
2. PURPOSE: This routine will print an integer matrix with labels and an introductory label.
3. METHOD: First, the introductory line is printed, and then the matrix is printed by columns, up to seven columns per line. After all the rows for the first columns are printed, the next group of columns is processed. After all the columns have been processed, control returns to the calling routine.
4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Index used to print out arrays.
IA	First dimension of the KA array.
I1	Column number of the first column to print on line.
I2	Last column number of the line being printed on.
I72	Indicator. If equal to zero, then only sixty characters (terminal). If equal to one, then eighty characters are printed on the introductory line.
J	Index pointer.
JPRT	Device code number of the output file.
KA	Matrix to print out.
L	Counter used for various purposes.
LEN	Length of introductory label array.
LINE	Array containing the introductory label.
MI	be printed.
N	Second dimension of the KA array.
NI	

5. I/O VARIABLES:

A. INPUT	LOCATION
IA	F.P.

I72 /ALLJ/

JPRT /ALLJ/

KA F.P.

LEN F.P.

LINE F.P.

M F.P.

N F.P.

B. OUTPUT LOCATION

(None)

6. CALLING ROUTINE: DDUMP

7. CALLED ROUTINE: (None)

1. NAME: PLOGJ
2. PURPOSE: This routine prints a logical variable with a label line.
3. METHOD: The array containing the label characters is identified, up to twenty characters, and the label and the logical variable are printed.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Do Loop index.
ICHAR	Array containing the carriage control characters. The first word contains a blank and will skip one line before printing. The second word is a zero, and will skip two lines before printing.
ICRTL	Determines which carriage control character to be printed.
JPRT	Device code to print output.
L, LEN	Length of the label array.
LINE	Array contains the label to print, four characters per word.
TF	Logical variable to be printed.

5. I/O VARIABLES:

A. INPUT	LOCATION
ICRTL	F. P.
JPRT	/ALLJ/
LEN	F. P.
LINE	F. P.
TF	F. P.
B. OUTPUT	LOCATION
(None)	

6. CALLING ROUTINE: PVARJ
7. CALLED ROUTINE: (None)

1. NAME: PLTSTR - 35
2. PURPOSE: This routine stores the PLOT command information to be used during an IOSYM simulation. It also processes the NOPLOT command information.
3. METHOD: A) PLOT commands. The number of plots to be done is incremented by one; the plot number is stored. If positive, then only line printer plots are done; if negative, line printer plots are done, and routine UPLOT is also called to do any other user desired plots. The independent and dependent variables are coded and stored. Control then returns to the calling routine.
 B) NOPLOT commands. The number of plots to be done is decreased by one, and the plot number is set to zero. Control then returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
APLT	Rounding number to obtain PLOT number.
CCONS	Array containing all of the constants in a NETWORK.
I	Array pointer and index counter.
IERR	Error indicator. If FALSE, then an error has not occurred in this routine. If TRUE, then a fatal error has occurred in this routine.
IERR1	Error indicator. If FALSE, then an error hasn't occurred up to the present time. If TRUE, then a fatal error has occurred.
IN1	IOSYM Identifier of the PLOT variable being processed.
IN2	Pointer to the location in the NNPLOT array where the PLOT variable identifier is to be stored.
ISUB	Indicator. If plus one, a PLOT command is being processed. If negative one, a NOPLOT command is being processed.
LLPLOT	Current PLOT number.
LPLLOT	Current PLOT number.

MMPLOT	Maximum number of PLOTS allowed (dimension of NNPLOT, set in USRDAT).
NNCNS	Number of defined constants in a NETWORK.
NNPLOT	Array where the first word is the number of plots; the second word is the IOSYM identifier for the independent variable, and the third, fourth, fifth, sixth and seventh words are the IOSYM identifiers for the dependent variables.
NNPLTS	Number of plots to be made.
NNWRK	Work storage array. Contains the current command parsed into the various elements.
NTNUL	Null field indicator.
NTTBL	Table field indicator.
NTVARD	Derivative variable field indicator.
NTVARO	Independent variable field indicator.
NTVAR1	Primary variable field indicator.
NTVAR2	Secondary variable field indicator.

5. I/O VARIABLES:

A. INPUT	LOCATION
CCONS	/I02/
IERR	F.P.
IERR1	F.P.
ISUB	F.P.
MMPLOT	/I04/
NNCNS	/I03/
NNPLOT	/I01/
NNPLTS	/I03/

NNWRK /I03/

NTNUL F.P.

NTTBL F.P.

NTVARD F.P.

NTVAR0 F.P.

NTVAR1 F.P.

NTVAR2 F.P.

B. OUTPUT LOCATION

CCONS I02

IERR1 F.P.

NNCNS /I03/

NNPLOT /I01/

NNPLTS /I03/

6. CALLING ROUTINE: PROCES

7. CALLED ROUTINES: ARAYCK
EEROR

1. NAME: PMATJ
2. PURPOSE: This routine will print a real matrix with labels and an introductory line.
3. METHOD: First, the introductory line is printed, and then the matrix is printed by columns, up to seven columns per line. After all the rows for the first columns are printed, the next group of columns is processed. After all the columns have been processed, control returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
A	Matrix to print out.
I	Index used to print out arrays.
IA	First dimension of the A array.
I1	Column number of the first column to print on line.
I2	Last column number of the line being printed on.
I72	Indicator. If equal to zero, then only sixty characters (terminal). If equal to one, then eighty characters are printed on the introduction line.
J	Index pointer.
JPRT	Device code number of the output file.
L	Counter used for various purposes.
LEN	Length of introductory label array.
LINE	Array containing the introductory label.
M	Number of rows of the A matrix to be printed.
MI	
N	Second dimension of the A array.
NI	

5. I/O VARIABLES:

A. INPUT	LOCATION
A	F.P.
IA	F.P.
I72	/ALLJ/
JPRT	/ALLJ/
LEN	F.P.
LINE	F.P.
M	F.P.
N	F.P.

B. OUTPUT	LOCATION
(None)	

6. CALLING ROUTINE: (Not used at this time)

7. CALLED ROUTINE: (None)

1. NAME: PP - 513
2. PURPOSE: This routine processes an IOSYM probability function and returns a sample from a probability distribution.
3. METHOD: Depending on which probability function is to be processed, different areas of the routine are executed. There are twelve IOSYM probability functions, each of which has its own section of programming. Once the function has been evaluated, its value is returned to the calling routine as PP.
4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Index and pointer used for various purposes.
IERR1	Error indicator returned from routine EEROR but not used in this routine.
IPAR	An array containing the parameter set numbers.
IPP	The probability function number.
ISM	Stream number to use for taking samples (see GASP manual for further description).
ITAB	Array containing the TABLEs specified in the probability function.
ITB1	TABLE number of the first TABLE in the probability function.
ITB2	TABLE number of the second TABLE in the probability function.
ITP	The parameter set to use for the probability function.
MMFUNP	Number of IOSYM probability functions.
MMPAR	Number of parameter sets.
NL	Number of elements in a TABLE.

NNPTB	Array containing in the first two words the TABLE name, and the third word points to the location in the TTABL array where the TABLE values are stored.
NP	Number of parameter sets in the probability function.
NT	Number of TABLEs in the probability function.
NX	Number of arguments in the probability function.
PP	This contains the result of the function upon returning to the calling routine.
PPREC	Precision of the computer being run on, set in routine USRDAT.
RN	Random number returned by the GASP routine DRAND.
X	Array containing the values of the function arguments.

5. I/O VARIABLES:

A. INPUT	LOCATION
IERR1	F.P.
IPAR	F.P.
IPP	F.P.
ITAB	F.P.
MMFUNP	/I04/
MMPAR	/I04/
NNPTB	/I02/
NP	F.P.
NT	F.P.

NX	F.P.
PPREC	/I04/
X	F.P.
B. OUTPUT	LOCATION
PP	FUNCTION

6. CALLING ROUTINE: EEVAL

7. CALLED ROUTINES:

BETA (GASP routine)
 DRAND (GASP routine)
 EEROR
 ERLNG (GASP routine)
 EXPON (GASP routine)
 GAMA (GASP routine)
 GGTAB
 NPSSN (GASP routine)
 RLOGN (GASP routine)
 RNORM (GASP routine)
 TRIAG (GASP routine)
 UNFRM (GASP routine)
 WEIBL (GASP routine)

1. NAME: PPFCMD - 556
2. PURPOSE: This routine will call the appropriate routine to execute a given command during an IOSYM execution.
3. METHOD: This routine will perform the command at NSET(LLUCS) with the command identifier IICM. A check is first done to see if the command is an IF command. If so, the level of stacked IFs and LOOPS is increased by one. The IF condition is evaluated, and the location of the TRUE or FALSE part is identified. The new command is then processed like an original one.

A check is done to see if the command is a LOOP command. If so, the level of stacked IFs and LOOPS is increased by one. The LOOP index is initialized, and routine PPIDF is called to store the index. The LOOP is then processed with the commands being executed as any other command in this routine.

A check is done to see if the command is a LABEL command. If so, the LOOP index corresponding to that LABEL is incremented and processing continues.

If the command isn't an IF, LOOP, or LABEL, the command type is determined, and control passes to the section of this routine that processes that type of command.

If the command is an ACTIVATE, DEACTIVATE, or SWITCH, the correct pointer is setup, and routine SSWARC is called to process the ARC status.

If the command is a SET command, routines EEVAL and PPIDF are called to evaluate and store the result.

If the command is a GOTO command, routine GGOTO is called to process the command.

If the command is a REPLACE command, routine RREPLC is called to process the command.

If the command is an EVENT command, routine DDOEVT is called to process the command.

If a JUMP or an EXIT command is identified, the correct LABEL is identified, and the appropriate LOOP and/or IF is ended. The level of stacked IFs and LOOPS is determined, and processing continues.

If the command is a PRINT command, routine DDPRNT is called to process the command.

If the command is a PUT or GET command, routine PPPUT or GGGET, respectively, is called to process the command.

If the command is a STAT command, routine GGSTAT is called to process the command.

After the command has been processed, the next command in the string is identified, and it is processed in a like manner. After all the LOOPS, IFs, and other commands in the string have been processed, control returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Pointer, index, and counter used for various purposes.
ICM	Command number of the current command being processed.
IDO	Number of stacked IFs and LOOPS minus one.
IDV	Location in the NSET array where the SET command starts.
IDVX	Initial Loop index value.
IDO	Index to which type of ACTIVATE, INACTIVATE or SWITCH is to take place. 1 = ACTIVATE 2 = DEACTIVATE 3 = SWITCH
IERR1	Error indicator returned from routine EEROR but not used in this routine.
IFUCS	Pointer to where the IF uniform command sequence starts.
IICM	Command number of the first command.
LABUCS	Location in the NSET array where the LABEL uniform command sequence starts.
LEV	Current level of stacked IFs and LOOPS after processing a JUMP or EXIT command.
LEVEL	Current level of stacked IFs and LOOPS.

LEVS	Array that contains pointers to the stacked IF command uniform command sequences.
LLUCS	Pointer to the location in the NSET array where the first command uniform command sequence begins.
LLUCSL	Array that contains the lengths of the various command uniform command sequences.
LPUCS	Pointer to the NSET array where the LOOP command uniform command sequence starts.
LUCS	Location in the NSET array where the command is stored.
MAXLEV	Maximum number of stacked IFs and LOOPS a section of a NETWORK can have.
NSET	Storage array that contains the link listed NETWORK.
NUCS	Pointer to the location in the NSET array for the command uniform command sequence.
VAL	Value of an expression from either an IF or SET command.
VALEND	Value of the LOOP index when the LOOP is finished.
VALINC	Value of the LOOP increment.
VALNDX	Current value of the LOOP index.
VALI	Initial value of the LOOP index.

5. I/O VARIABLES:

A. INPUT	LOCATION
IERR1	F.P.
IICM	F.P.

LLUCS

F.P.

LLUCSL

/I02/

NSET

/ /

B. OUTPUT

LOCATION

(None)

6. CALLING ROUTINES:

DDOCMD

DDOLST

7. CALLED ROUTINES:

DDOEV

DDPRNT

EEROR

EEVAL

EEVLTF

GGGET

GGOTO

GGSTAT

PPIDF

PPPUT

RREPLC

SSWARC

VVARBL

1. NAME: PPIDF - 515
2. PURPOSE: This routine will change the value at an IOSYM variable specified by a variable identifier.
3. METHOD: The first determination is what variable is to be set. If the variable identifier (IDV) is less than zero, it must be a TABLE element with a variable index. The correct TABLE element is identified, and the current value is stored.

If the IDV is a primary, secondary, or derivative variable, the correct one is identified, and routine PPVAR is called to store the value.

If the IDV is a TABLE element with a numeric index, the current value is stored in the correct location in the TTABL array.

If the variable is a substitutable parameter, it is identified, and the above processing takes place for the substituted variable.

Control then returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Index used to point to various arrays.
IDV	Variable identifier of the variable to set.
IDVV	Variable identifier of the variable to set.
IERR1	Error indicator returned by routine EEROR but not used in this routine.
IIOPR	Array containing the IDVs of the substitutable variables.
ITBL	TABLE number of the TABLE to process.
LMNT	TABLE element in the TTABL array to change.
NNDUM	Numer of substitutable parameters.
TTABL	Array containing the values of all the TABLE elements.
VAL	New value of the variable.

5. I/O VARIABLES:

A. INPUT	LOCATION
IDV	F. P.
IERR1	F. P.
IIOPR	/I02/
NNDUM	/I02/
VAL	F. P.
B. OUTPUT	LOCATION
TTABL	/I02/

6. CALLING ROUTINES:

FF
GGGET
PPFCMD
WTHMN

7. CALLED ROUTINES:

EEROR
GGVAR
PPTAB
PPVAR

1. NAME: PPPAR - 504
2. PURPOSE: This routine will assign a value to a parameter set element.
3. METHOD: Given the parameter set number and the element in that parameter set, this routine will assign a given value to it.
4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
IERR1	Error indicator returned from routine EEROR but not used in this routine.
IPAR	Parameter set number.
LMENT	Parameter set element number.
MMPAR	Maximum number of parameter sets.
PPARM	Array that contains all the parameter set values.
VAL	Value to set the parameter set element to.

5. I/O VARIABLES:

A. INPUT	LOCATION
IERR1	F.P.
IPAR	F.P.
LMENT	F.P.
MMPAR	/I04/
VAL	F.P.
B. OUTPUT	LOCATION
PPARM	/GCOM1/

6. CALLING ROUTINE: (Not used at this time).
7. CALLED ROUTINE: EEROR

1. NAME: PPRNT - 89
2. PURPOSE: This routine executes a PRINT command during an IOSYM simulation either immediately or from a TABULATE tape.
3. METHOD: Depending on which type of PRINT command is being done (PRINT, ERROR, TRACE or USER), the PRINT header is set up. If doing the PRINT on a NODE, the NODE name is deciphered; if doing the PRINT on an ARC and OPERATION, their names are decoded. All of the PRINT variables are deciphered. The PRINT information is printed out, and control returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLES	DEFINITION
ICHAR	Array containing all of the IOSYM characters A-Z, 0-9, ., **, *, /, +, -, =,), (, ,, \$.
IDV	Pointer to the location in the IRAY2 array where the IOSYM variable identifiers are located.
IOPN01	Contains the OPERATION number of the OPERATION the PRINT occurred on.
IOPN02	If positive, it is the ARC number of the ARC the PRINT occurred on. If negative, it is the negative of the NODE number of the NODE the PRINT occurred on.
IPNT	Contains the type of PRINT to be done 1 = PRINT 2 = ERROR 3 = TRACE 4 = USER
IRAY	Array containing the ARC and OPERATION or NODE name information.
IRAY1	Array containing the independent variable name information.
IRAY2	Array containing the IOSYM PRINT variable identifiers (This array is subset of a formal parameter array).
IRAY3	Array containing the type of PRINT information name being done.

IWT	Number of blanks needed to fill name.
J	Pointer and counter used for a variety of purposes.
K	Pointer and counter used for a variety of purposes.
KK	Index used to point to locations in arrays.
MIOP	Node number of the NODE that the PRINT is being executed on.
NNPTAC	Array where the first two words are the ARC name in internal IOSYM notation and the third word points to the NSET array where the ARC is stored.
NNPTND	Array where the first two words are the NODE name in internal IOSYM notation; the third word points to the NSET array where the NODE is stored, and the fourth word indicates if this NODE has been replaced or not.
NNPTOP	Array where the first two words are the OPERATION name in internal IOSYM notation; the third, fourth and fifth words are pointers to the NSET array where the OPERATION commands are located, and the sixth word indicates if this is a COLLECT OPERATION.
NNPTO	Contains the internal IOSYM notation of the independent variable name.
NPRNT	The output device code number for writing output, set in USRDAT.
NV	Number of variables to print on PRINT command.
NWT	Number of characters in a name.
T	Array where the first word points to the location in the IRAY2 array where the IOSYM variable identifiers are located, the second word is the simulation time; when the PRINT was performed.
VAL	Contains the values of the PRINT variables.

5. I/O VARIABLES:

A. INPUT	LOCATION
ICHAR	/ME1/
IOPN01	F.P.
IOPN02	F.P.
IPNT	F.P.
IRAY	F.P.
IRAY1	F.P.
IRAY2	F.P.
NNPTAC	/I02/
NNPTND	/I02/
NNPTOP	/I02/
NNPTO	/I02/
NPRNT	/GCOM1/
NV	F.P.
NWT	F.P.
T	F.P.
VAL	F.P.

B. OUTPUT	LOCATION
IRAY2	F.P.

6. CALLING ROUTINES: DDPRNT
 EEPRNT
 PPRTAB
 TTRACE
 UUPRNT

7. CALLED ROUTINE: UNCODE

1. NAME: PPPUT - 88
2. PURPOSE: This routine executes a PUT command during an IOSYM simulation.
3. METHOD: The TABLE number that is being written from is retrieved from the PUT command uniform command sequence (UCS). The length of the TABLE is identified and checked to make sure there are enough elements to write on the GASP file. The information is copied from the TABLE to the ATRIB array; the routine FILEM is called to store the information in the GASP file.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
ATRIB	Array that contains the attribute values being stored into the GASP file.
I	Index to which attribute is being copied from the TABLE to the ATRIB array.
IERR	Error indicator used in call to EEROR, but not used in this routine.
LEN	Used as a pointer to find information from various arrays.
LEN1	Position in the TABLE array where the items to be copied to the ATRIB array are located.
LUCS	Position in the NSET array where the UCS for the PUT command starts.
NNATR	Number of attributes in each entry in the GASP files.
NNFIL	Number of GASP attribute files.
NNPTB	Array that contains in the first two words the internal IOSYM notation of the TABLE names; the third word contains the pointer to the location in the TTABL array where the TABLE values are stored.
NSET	Storage array where the link listed NETWORK is stored.

TTABL Array that contains in the first word the current values of all the TABLE elements, and the second word contains the initial values.

VVAR2 Array that contains in the first word the current values of all the secondary variables, and the second word contains the initial values.

5. I/O VARIABLES:

A. INPUT LOCATION

 IERR1 F.P.

 LUCS F.P.

 NNATR /GCOM1/

 NNFIL /GCOM1/

 NNPTB /I02/

 NSET / /

 TTABL /I02/

 VVAR2 /I02/

B. OUTPUT LOCATION

 ATRIB /GCOM1/

6. CALLING ROUTINE: PPFCMD

7. CALLED ROUTINES: EEROR
 FILEM (GASP routine)
 VVARBL

1. NAME: PPRCOL - 535
2. PURPOSE: This routine prints the COLLECT statistics.
3. METHOD: A check is done to see if there are any COLLECT variables or COLLECT OPERATIONS. If there aren't at least one of each, control returns to the calling routine.

The COLLECT statistics part one page heading is printed. The means and standard deviations for each variable are calculated and printed along with the deciphered variable name.

The COLLECT statistics part two page heading is printed. Then for each COLLECT variable the average value, standard deviation, average percent and standard deviation percent are calculated on each OPERATION and printed along with the deciphered variable and OPERATION names.

Control then returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
AV	Average value of a given variable.
AVP	Average percentage of a given variable.
AVY	Average value of a given variable.
I	Index to show which variable is currently being processed.
ICHAR	Array containing the internal IOSYM representation of all the characters A-Z, 0-9, ., **, *, /, +, -, =,), (, ., \$.
ICOL	Pointer to where the COLLECT information is stored in the QSET (NSET) array.
IDV	The IOSYM variable identifier under current consideration.
IERR1	Error indicator returned by routine EEROR, but not used in this routine.
II	Index and counter used for a number of purposes.

IISIM	Current simulation number.
IJ	Index that points to the location in the QSET array where the current variable value is located.
ILINE	Indicator. If equal to zero, then haven't started working on first variable on current page; if equal to one, then have already started first variable.
IOP	Index that indicates which OPERATION is currently being processed.
IRAY	Array that contains the NETWORK name information.
IRAY1	Array that contains the different name information.
IWT	Number of blanks needed to fill a name.
J	Array index and counter.
K	Array index and counter.
KK	Array index and counter
KKK	Array index and counter.
LLCOL	First word in the QSET (NSET) array used for storage of COLLECT information.
MMLINE	Maximum number of lines that will fit on a computer page.
NLINES	Number of lines on current page left.
NNCOL	Number of COLLECT variables.
NNCOLC	Array that contains the IOSYM identifiers of the COLLECT variables.
NNCOP	Number of COLLECT OPERATIONS.

NNMNET	Array that contains in the first two words the NETWORK name in IOSYM notation, and the third word indicates whether a NETWORK summary is to be done.
NNOP	Number of OPERATIONS in the current NETWORK.
NNPTOP	Array that contains in the first two words the OPERATION name in IOSYM notation; the third, the fourth and fifth words contain pointers to the commands on the OPERATION, and the sixth word indicates if the OPERATION is a COLLECT OPERATION or not. If greater than zero, then it is a COLLECT OPERATION.
NNRUN	Number of runs in current simulation.
NPAG	COLLECT statistic page number currently working on.
NPRNT	The output device code number for writing output (set in USRDAT).
NWT	Number of characters in a name.
P	Percentage of variable explained by current OPERATION.
P2	Percentage of variable explained by current OPERATION squared.
QSET	Array equivalenced to NSET, the part of this array being used is where the COLLECT information is stored.
SDP	Standard deviation percent for a given variable.
SDY	Standard deviation for a given variable.
STD	Standard deviation for a given variable.
SUM	Sum of variable.
SUM2	Sum of variable squared.
XN	Number of runs.

Y Sum of variables.

Y2 Sum of variables squared.

5. I/O VARIABLES:

A. INPUT LOCATION

ICHAR /ME1/

IERR1 F.P.

IISIM /I03/

IRAY F.P.

IRAY1 F.P.

LLCOL /I03/

MMLINE /I04/

NNCOL /I03/

NNCOLC /I01/

NNCOP /I03/

NNMNET /I03/

NNOP /I03/

NNPTOP /I02/

NNRUN /GCOM5/

NPRNT /GCOM1/

NWT F.P.

QSET / /

B. OUTPUT LOCATION

(None)

6. CALLING ROUTINE: DDOSIM

7. CALLED ROUTINES: EEROR
UNCODE
UNC01

1. NAME: PPRPLT - 78
2. PURPOSE: This routine executes PLOT commands and does line printer plots and calls user written routine to do any user type of plots desired.
3. METHOD: The proper tape containing the PLOT information is identified. The PLOTS are done one at a time. If the given PLOT is not active, the next PLOT is investigated. Routine ARANGE is called to calculate nice line printer plot limits. Each line printer plot contains a maximum of 1200 plot points. If there are more points to plot, they are done on subsequent pages. The PLOT information is read from the plot tape, and the independent and dependent variable values are stored in the appropriate arrays. The NETWORK name is decoded and the PLOT heading is printed. Routine XPLOT is called to do the actual line printer plot. The information for the PLOT legend is deciphered and printed. If the PLOT number was negative, then the user plot routine UPLT is called. All of the plots are processed in this manner; then control returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Index used for various purposes.
ICHAR	Array containing the internal IOSYM representation of all the characters A-Z, 0-9, ., **, *, +, -, =,), (, ,, \$.
IERR	Error Indicator used in more than 1000 points and independent variable not monotonically increasing.
II	Indicator. If equal to one, then working on first part of plot; if equal to two, then working on plot continuation.
IITPS	Array that contains the device code numbers for the various tapes. IITPS(3) contains the PLOT tape device code.
INUM	PLOT number read from the PLOT tape.
IPLT	Number of PLOTS done.
IRAY	Array that contains the various name information.

ITP	Device code where PLOT information is located.
IWT	Number of blanks needed to fill name.
J	Plot point being processed.
JG	Array index pointer.
JJ	Array index pointer.
JK	Number of PLOT points processed.
K	Index pointer and counter.
KK	Index pointer and counter.
LAST	Indicator. If TRUE, then working on last PLOT. If FALSE, not working on last PLOT.
MMPLLOT	Maximum number of PLOTs (Dimension of NNPLLOT set in USRDAT).
NNMNET	First two words contain the NETWORK name in IOSYM notation, and the last word indicates whether a NETWORK summary is to be done.
NNPLLOT	Array where the first word is the plot number; the second word is the IOSYM identifier for the independent variable, and the third, fourth, fifth, sixth and seventh words are the IOSYM identifiers for the dependent variables.
NNPLTS	Number of PLOTs to be done.
NPRNT	The output device code number for writing output (set in USRDAT).
NUMPLT	Number of dependent variables on PLOT.
NV	Number of dependent variables on PLOT.
NWT	Number of characters in name.

RANGE

Array where the first word has a one to indicate that the axis limits follow. Second word is the minimum x-axis limit; third word is the maximum x-axis limit; fourth word is the minimum y-axis limit, and the fifth word is the maximum y-axis limit.

T

Independent plot variable value.

VAL

Dependent plot variable values.

X1

Array containing all of the independent variable values.

Y1

Array containing all of the dependent variable values.

5. I/O VARIABLES:

A. INPUT	LOCATION
ICHAR	/ME1/
IITPS	/I01/
IRAY	F.P.
MMPLOT	/I04/
NNMNET	/I03/
NNPLOT	/I01/
NNPLTS	/I03/
NPRNT	/GCOM1/
NWT	F.P.
RANGE	F.P.
B. OUTPUT	LOCATION
RANGE	F.P.
X1	F.P.
Y1	F.P.

- 6. CALLING ROUTINE: OTPUT
- 7. CALLED ROUTINES: ARANGE
UNCODE
UNC01
UPLOT (user written or supplied)
XPLOT

1. NAME: PRRPT - 548
2. PURPOSE: This routine prints the REPORT statistics.
3. METHOD: If there are no REPORT statistics, control returns to the calling routine. The NETWORK name is decoded, and the REPORT page header is printed. For each REPORT variable, its average and standard deviation are calculated; the variable name is decoded, and the REPORT information is printed. This is repeated for each REPORT variable; then control returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
AV	Average of the given variable.
I	Index counter for which variable is being processed.
ICHAR	Array that contains the internal IOSYM representation of all the characters A-Z, 0-9, ., **, *, /, +, -, =,), (, ,, \$.
IIRPT	Array that contains all of the REPORT variable identifiers.
IISIM	Simulation number.
IISTAT	If equal to zero, do REPORT statistics. If equal to one, don't do REPORT statistics.
IRAY1	Array that contains different name information.
J	Index pointer and counter.
L	Location in the WWRK (NNWRK) where the REPORT variable values are located.
LLRPT	Location in the WWRK (NNWRK) where the REPORT variable information starts.
MMLINE	Maximum number of computer lines that will fit on a computer page (set in USRDAT).

NLINES	Number of lines processed on current page.
NNMNET	Array where the first two words are the NETWORK name, and the third word indicates whether or not a NETWORK summary is to be done.
NNRPT	Number of REPORT variables.
NNRUN	Number of runs in this simulation.
NNWRK	Work storage array equivalenced to WWRK (see WWRK definition).
NPAG	Current REPORT page being worked on.
NPRNT	The output device code number for writing output (set in USRDAT).
NWT	Number of characters in a name.
STD	Standard deviation of the given variable.
WWRK	Work storage array equivalenced to NNWRK. The locations used in this routine contain the REPORT variable value information.
XN	Number of runs in this simulation.

5. I/O VARIABLES:

A. INPUT	LOCATION
ICHAR	/ME1/
IIRPT	/I01/
IISIM	/I03/
IISTAT	/I03/
IRAY1	F.P.
LLRPT	/I03/
MMLINE	/I04/

NNMNET	/I03/
NNRPT	/I03/
NNRUN	/GCOM5/
NNWRK	/I03/
NPRNT	/GCOM1/
NWT	F. P.
WWRK	/I03/ Equivalenced to NNWRK

B. OUTPUT LOCATION
 (none)

6. CALLING ROUTINE: DDOSIM
7. CALLED ROUTINES: UNCODE
 UNCO1

1. NAME: PPRTAB - 549
2. PURPOSE: This routine produces and prints the TABULATE output.
3. METHOD: If there are no tabulations, control is returned to the calling routine. The tape containing the TABULATE information is identified. The tabulations are processed one at a time. If the current tabulation is not active, the next one is investigated. The NETWORK name is decoded, and the tabulation page heading is printed. The independent variable name is decoded, and the tabulation subheading is printed. The variables and their report type are decoded, and the second subheading is printed. A record is read off the TABULATE tape. If the information is for this tabulation number and is PRINT information, routine PPRNT is called to process PRINT information. If the information is not for this TABULATE, then the next record is read. If the information is for this TABULATE, the time option is determined, the ARC and OPERATION names are decoded and the TABULATE information is printed. The next record is read off of the TABULATE tape and processed the same way. This continues until the end of the TABULATE tape is reached. Then it is rewound, and the next TABULATE is processed. When all of the TABULATES have been processed, control returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
A	Length of time while processing on the given ARC and OPERATION.
DUR	Length of time while processing on the given ARC and OPERATION.
I	Index pointing to current TABULATE processing.
IA	Indicator. If equal to 999, then read PRINT information off the TABULATE tape. If equal to -191919, then have reached the end of the TABULATE tape. If it is equal to anything else, then TABULATE information was read from TABULATE tape.
ICHAR	Array that contains the internal ISOYM representation of all the characters A-Z, 0-9, ., **, *, /, +, -, =,), (, ,, \$.
IDDUR	Number of days worked on OPERATION.

IDT1	Number of days worked when started OPERATION.
IDT2	Number of days worked when finished OPERATION.
IHDUR	Number of hours worked on OPERATION after taking out the number of days.
IHT1	Number of hours worked when started OPERATION after taking out the number of days.
IHT2	Number of hours worked when finished OPERATION after taking out the number of days.
IISIM	Simulation number.
IISTAT	Indicator. If equal to one, then do TABULATE statistics; if equal to zero, don't.
IITPS	Array that contains the device code number for the various tapes. IITPS (4) contains the TABULATE tape device code.
IMDUR	Number of minutes worked on OPERATION after taking out the days and hours.
IMT1	Number of minutes worked when started OPERATION, after taking out the days and hours.
IMT2	Number of minutes worked after finished OPERATION, after taking out the days and hours.
INUM	TABULATE number of information that is being processed.
IOPN01	Internal IOSYM OPERATION number of the OPERATION, the TABULATE or PRINT was performed on.
IOPN02	If positive, then it is the internal IOSYM ARC number of the ARC the TABULATE or PRINT was performed on. If negative, then it is the negative of the internal IOSYM NODE number of the NODE the PRINT was performed on.

IPNT	Contains the type of PRINT to be done 1=PRINT 2=EEROR 3=TRACE 4=USER
IRAY	Array that contains the different name information.
IRAY1	Array that contains the different name information.
IRAY2	Array that contains the different name information.
IREAD	If equal to zero, then haven't read any information from the TABULATE tape yet, any other value and have read information.
ITIME	Tabulation line number.
ITOP	Time option indicator. If equal to zero, time option is off; if equal to one, then time option is on.
ITP	TABULATE tape device code.
IVAL	The values of the PRINT variables.
IWT	Number of blanks needed to fill name.
J	Index pointer and counter.
JJ	Index pointer and counter.
K	Index pointer and counter.
KK	Index pointer and counter.
L	Index pointer and counter.
MMLINE	Maximum number of lines that can be printed on a computer page (set in USRDAT).
MMTABU	Maximum number of TABULATE statistics (dimension of NNTABU, set in USRDAT).

NLINES	Number of lines used on current computer page.
NNMNET	Array containing in the first two words the NETWORK name, and the third word contains an indicator to tell whether a NETWORK summary is to be done.
NNPTAC	Array that contains in the first two words the ARC name, and the third word points to the location in the NSET array where the ARC is stored.
NNPTOP	Array that contains in the first two words the OPERATION name; the third, fourth and fifth words are pointers to the OPERATION commands, and the sixth word is an indicator to tell whether or not the OPERATION is a COLLECT OPERATION.
NNPTO	Array containing the independent variable name.
NNRUN	Number of runs in this simulation.
NNTABU	The first word indicates the time option to be used 1=time option off 2=time option on, the second, fourth, sixth and eighth words are the TABULATE variables, with the third, fifth, seventh and ninth words being the report types for the four variables, respectively.
NPAG	Page number of current page.
NPRNT	The file number of the device where output is to be written (set in USRDAT).
NV	Number of variables to print on PRINT command.
NWT	Number of characters in the name.
T	Array where the first word points to the location in the NSET array where the PRINT variables are stored, and the second word is the time the PRINT was processed.

VAL

Is the PRINT variable values.

5. I/O VARIABLES:

A. INPUT LOCATION

ICHAR /ME1/

IISIM /I03/

IISTAT /I03/

IITPS /I01/

IRAY F.P.

IRAY1 F.P.

IRAY2 F.P.

MMLINE /I04/

MMTABU /I04/

NNMNET /I03/

NNPTAC /I02/

NNPTOP /I02/

NNPTO /I02/

NNRUN /GCOM5/

NNTABU /I01/

NPRNT /GCOM1/

NWT F.P.

B. OUTPUT LOCATION

(none)

6. CALLING ROUTINE: OPUT

7. CALLED ROUTINES: PPRNT
UNCODE
UNC01

1. NAME: PPTAB - 502
2. PURPOSE: This routine will assign the value of a TABLE element.
3. METHOD: The TABLE number and element number are identified, and the location for the value of the TABLE element in the TTABL array is found, and the value is stored.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
IERR1	Error indicator returned from routine EEROR but not used in this routine.
ITABL	TABLE number of the TABLE element being set.
L	Pointer to the TTABL array.
LMENT	The TABLE element number.
NNPTB	Array containing in the first two words the TABLE name, and the third word points to the location in the TTABL array where that TABLE's element values are stored.
NNXTB	Number of defined TABLEs.
TTABL	Array containing all the TABLE element values.
VAL	The value the TABLE element is to be set to.

5. I/O VARIABLES:

A. INPUT	LOCATION
IERR1	F.P.
ITABL	F.P.
LMENT	F.P.
NNPTB	/I02/
NNXTB	/I03/
VAL	F.P.

B. OUTPUT

LOCATION

TTABL

/I02/

6. CALLING ROUTINES: FF
PPIDF
7. CALLED ROUTINES: EEROR

1. NAME: PPVAR - 510
2. PURPOSE: This routine will assign a value to a simple variable.
3. METHOD: The variable type and variable number are identified. If the variable is a primary variable, then the value is stored in the corresponding location in the SS array. If the variable is a secondary variable, then the value is stored in the corresponding location in the VVAR2 array. If the variable is a derivative variable, then the value is stored in the corresponding location in the DD array. After storing the value in the appropriate array, control returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
DD	This array contains the derivative variable values at the time TNOW.
IERR1	Error indicator returned from routine EEROR but not used in this routine.
IVAR	Variable number.
I123	Variable type: 1 = primary variable 2 = secondary variable 3 = derivative variable
NNV1	Number of defined primary variables.
NNV2	Number of defined secondary variables.
SS	Array containing the values of the primary variables at the time TNOW.
VAL	The value to set the variable to.
VVAR2	Array containing the current value of the secondary variable in the first word, and the initial value in the second word.

5. I/O VARIABLES:

A. INPUT	LOCATION
IERR1	F.P.

IVAR	F.P.
I123	F.P.
NNV1	/I03/
NNV2	/I03/
VAL	F.P.
B. OUTPUT	LOCATION
DD	/GCOM2/
SS	/GCOM2/
VVAR2	/I02/

6. CALLING ROUTINE: PPIDF

7. CALLED ROUTINE: EEROR

1. NAME: PROCES - 40
2. PURPOSE: This routine takes the commands as they are stored in the work array (NNWRK) and by calling the different routines processes the information and stores it in the NSET and other arrays.
3. METHOD: Depending on what command is being processed, different routines are called.

Routine SRHARC is called to process ACTIVATE, DEACTIVATE and SWITCH commands.

Routine ARCSTR is called to process ARC commands.

Routine COLSTR is called to process COLLECT and NOCOLLECT commands.

Routine DOSTR is called to process DO commands.

Routine GBLSTR is called to process GLOBAL commands.

Routine GTOSTR is called to process GOTO commands.

Routine TYPSTR is called to process TYPE commands.

Routine NODSTR is called to process NODE commands.

Routine PLTSTR is called to process PLOT and NO PLOT commands.

Routine RPTSTR is called to process REPORT and NOREPORT commands.

Routine OPRSTR is called to process OPERATION commands.

Routine PCKSTR is called to process PICK commands.

Routine TBTSTR is called to process TABULATE and NOTABULATE commands.

Routine SAVE is called to process SAVE and RESTORE commands.

Routine SETSTR is called to process SET or arithmetic statements (statement with an equal sign in it).

Routine TRMSTR is called to process TERM or CROSS commands.

Routine WTHSTR is called to process WITH commands.

Routine EVTSTR is called to process EVENT commands.

Routine REPSTR is called to process REPLACE commands.

Routine FORSTR is called to process FOR and NOFOR commands.

Routine LOPSTR is called to process LOOP commands.

Routine IFSTR is called to process IF commands.

Routine TOFSTR is called to process TRUE and FALSE commands.

Routine JOESTR is called to process JUMP and EXIT commands.

Routine PRTSTR is called to process PRINT commands.

Routine PUTSTR is called to process PUT commands.

Routine GETSTR is called to process GET commands.

Routine STASTR is called to process STAT commands.

Routine TRCSTR is called to process TRACE and NOTRACE commands.

The DUMP and NODUMP commands are processed by setting the variable DUMP equal to TRUE or FALSE respectively.

The ECHO and NOECHO commands are processed by setting the variable ECHO equal to TRUE or FALSE respectively.

The INFO and NOINFO commands are processed by setting the variable NOINFO equal to FALSE or TRUE respectively.

If a GASP command is processed under a DO command level, then the variable IIGASP is set to one to indicate that before each run GASP input is to be processed.

If a GO command is processed, a check is done to make sure there is a NODE specified to GO to. If there were any WITH variables processed, the work storage needed to process the GO command is shifted out by a call to routine SSHIFT. If no fatal error has been processed, then variable IICHOI is set to three to indicate it is time to do a simulation.

If a READ command is encountered by routine PROCES, an error is processed because a READ can only be encountered after a GO command, and any commands after GO are processed by routine DDOSIM until control is returned to routine PARSE and PROCES.

The QUIT command is processed by setting variable IICHOI to four which indicates that all IOSYM processing is finished.

The END command is used to end a command LEVEL. If a TYPE command is being ended, a check is done to see if any information followed the TYPE command if not an indicator is set to show that. If an ARC, GLOBAL,

IFIR Indicator to show that the start of a new ARC has begun.

IICHOI Determines what processing is to occur next.
 2 = More IOSYM input to process.
 -2 = Indicates a CHANGE command was ended and a NETWORK summary is desired.
 3 = Time to start an IOSYM simulation.
 4 = IOSYM processing is finished so return to the main program.
 -4 = There was an error in IOSYM input so stop processing and return to the calling routine.

IIGASP Indicator. If equal to one, then GASP input is to read before each run.

IIGO The internal IOSYM NODE number of the start NODE as specified on the GO command.

IIOP Current OPERATION number being processed.

INDX4 Pointer in the ARC uniform command sequence (UCS) that points to the location in the NSET array where the UCS for the OPR command for this ARC is located.

INDX5 Pointer in the ARC UCS that points to the location in the NSET array where the number of TERMS and CROSSES for the ARC is located.

IN1 Location in the CCONS array where the number of runs to do in the current simulation, as specified by the DO command, is located.

IN2 Number of runs to do in the current simulation, as specified by the DO command.

IPICK Location of the pointer in a PICK command for an ARC leaving a NODE to the next ARC leaving the same NODE.

OPERATION, TERM or NODE command is being ended, routine LBLCHK is called to check that all the LABELS have been defined. If an ARC command is being ended, routine TRMSTR is called to process a TERM command for the ARC. If a NETWORK command is being ended, routine ENDNET is called to do some NETWORK finishing and resequencing, and the variable IICHOI is set to show that the NETWORK has been processed.

If the command being ended is OPERATION or GLOBAL, then some final post initialization is done.

The current command level is set to zero, and the command level is decreased by one.

After one of the above paragraphs is processed, control then returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
CCONS	Array that contains all of the constants in a NETWORK.
DUMP	Indicator. If TRUE, then a print-of of all of the IOSYM arrays is to occur. If FALSE, then a dump is not to occur.
ECHO	Indicator. If TRUE, then the input cards processed are to be printed out on the output file. If FALSE, then the input cards are not ECHOed back.
FIR2	Indicator to show whether an ARC leaving the NODE I has already taken place. If FIR2(I) is TRUE, then an ARC leaving NODE I has not been processed. If FIR2(I) is FALSE, then an ARC leaving NODE I has been processed.
IC	Current command level.
ICLEVL	Is an array that contains the current command hierarchy from level one to level LEVEL.
ID	Dimension of the IFIR array.
IERR1	Error indicator. If FALSE, then an error hasn't occurred up to the present time. If TRUE, then a fatal error has occurred.

IPTCMD	Pointer to the next command. If IPTCMD is zero, then no command uniform command sequence points to the current command.
ISNODE	Current start NODE.
ITYPI	Indicates what type of command is now being processed. 0 = not doing TYPE commands. 1 = Start commands. 2 = Immediate commands. 3 = End commands.
LEVEL	Indicates what level of command hierarchy is being processed. There are five levels 0th, 1st, 2nd, 3rd, and 4th.
LLWITH	Location in the work storage array (NNWRK) where the WITH command information is stored.
LLWRK	Next available position in the work array (NNWRK) that can be used.
LLXSET	Counter of the number of locations in current use in the NSET array. Also, the last position in the NSET array used.
NDX1	Current command number.
NNCNS	Number of defined constants in the NETWORK.
NNMNET	Array that contains in the first two words the internal IOSYM representation of the NETWORK name. The third word is an indicator to indicate whether or not a NETWORK summary is done.
NNPTGB	Array that contains in the first two words the IOSYM internal representation of the word GLOBAL. The third, fourth and fifth words point to the location in the NSET array where the TYPE S, TYPE I and TYPE E commands start, respectively.

NNPTOP	The first two words of this array contain the OPERATION name. The third, fourth, and fifth words contain the pointers to the TYPE S, TYPE I, and TYPE E commands that occur under the OPERATION, respectively. The sixth word contains an indicator to show whether or not the COLLECT variables are COLLECTED on this OPERATION.
NNWRK	Work storage array. Contains the current command parsed into the various elements.
NOINFO	Indicator that tells whether or not informative diagnostics (warnings) are to be printed. If FALSE, the warnings are to be printed; if TRUE, they are not.
NPRNT	The output device code number for writing output (set in routine USRDAT).
NSET	Storage array where the link listed NETWORK is stored.
NTFLP	Floating point number field indicator.
NTINT	Integer number field indicator.
NTNUL	Null field indicator.
NTTBL	Table field indicator.
NTVARD	Derivative variable field indicator.
NTVARO	Independent variable field indicator.
NTVAR1	Primary variable field indicator.
NTVAR2	Secondary variable field indicator.
OPR	Indicator. If TRUE, then an OPR command for the current ARC has been processed. If FALSE, then an OPR command hasn't been processed.

PICK

Indicator. If TRUE, then a PICK command on the current ARC has been processed. If FALSE, then a PICK command hasn't been processed.

TERM

Indicator. If TRUE, then a TERM command in the current ARC has been processed. If FALSE, then a TERM command wasn't processed.

5. I/O VARIABLES:

A. INPUT	LOCATION
CCONS	/I02/
FIR2	F.P.
ICLEVL	/ME1/
ID	F.P.
IERR1	F.P.
IFIR	F.P.
IIOP	/I03/
IPICK	F.P.
IPTCMD	F.P.
ISNODE	F.P.
ITYP1	F.P.
LEVEL	/ME1/
LLWITH	/I03/
LLWRK	/I03/
LLXSET	/I03/
NDX1	/ME1/
NNCNS	/I03/
NNMNET	/I03/
NNPTGB	/I02/

NNPTOP	/I02/
NNWRK	/I03/
NPRNT	/GCOM1/
NSET	/ /
NTFLP	F.P.
NTINT	F.P.
NTNUL	F.P.
NTTBL	F.P.
NTVARD	F.P.
NTVAR0	F.P.
NTVAR1	F.P.
NTVAR2	F.P.
OPR	F.P.
PICK	F.P.
TERM	F.P.
B. OUTPUT	LOCATION
CCONS	/I02/
DUMP	/ME1/
ECHO	/ME1/
ICLEVL	/ME1/
IICHOI	F.P.
IIGASP	/I03/
IIGO	/I03/
IIOP	/I03/
IPTCMD	F.P.
ITYP1	F.P.

LEVEL	/ME1/
LLWITH	/I03/
LLWRK	/I03/
NNCNS	/I03/
NNMNET	/I03/
NNPTGB	/I02/
NNPTOP	/I02/
NOINFO	/ME1/
NSET	/ /

6. CALLING ROUTINE: CHOIC2

7. CALLED ROUTINES:

- ARCSTR
- COLSTR
- DOSTR
- EEROR
- ENDNET
- EVTSTR
- FORSTR
- GBLSTR
- GETSTR
- GTOSTR
- IFSTR
- JOESTR
- LBLCHK
- LBLSTR
- LOPSTR
- NODSTR
- OPRSTR
- PCKSTR
- PLTSTR
- PRTSTR
- PUTSTR
- REPSTR
- RPTSTR
- SAVE
- SETSTR
- SRHARC
- SSHIFT
- STASTR
- TBTSTR
- TOFSTR
- TRCSTR
- TRMSTR
- TYPSTR
- WTHSTR

1. NAME: PRTSTR - 71
2. PURPOSE: This routine processes PRINT commands. It stores the indicator for what type of PRINT is to be done (Tabulation or Immediate), whether it is done on an ARC or NODE, and the variables to PRINT.
3. METHOD: If another command points to this command, it is set to indicate where this command starts. The current command number is stored in the NSET array. The TABULATE number or immediate type PRINT number is stored. An indicator to show whether this PRINT occurs on a NODE or ARC is stored, and the variable identifiers of the variable are stored. A position in the NSET array is set aside to point to the next command.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Array pointer to the current variable on the PRINT command being processed.
ICLEVL	Is an array that contains the current command hierarchy from level one to level LEVEL.
IERR	Error indicator. If FALSE, then no error has occurred in this routine. If TRUE, then a fatal error has occurred in this routine.
IERR1	Error indicator. If FALSE, then no error has occurred up to the present time. If TRUE, then a fatal error has occurred.
IN1	Number of variables processed on the PRINT command.
IPTCMD	Pointer to the next command. If IPTCMD is zero, then no command uniform sequence points to the current command.
LEVEL	Indicates what level of command hierarchy is being processed. There are five levels 0th, 1st, 2nd, 3rd, and 4th.

LLXSET	Counter of the number of locations in current use in the NSET array. Also, the last position in the NSET array used.
LXSET	Temporary counter of the number of locations in the NSET array needed.
MMSET	Maximum number of locations in the NSET array (dimension of the NSET array, set in routine USRDAT).
NNWRK	Work storage array. Contains the current command parsed into the various elements.
NSET	Storage array where the link listed NETWORK is stored.
NTNUL	Null field indicator.

5. I/O VARIABLES:

A.	INPUT	LOCATION
	ICLEVL	/ME1/
	IERR	F.P.
	IPTCMD	F.P.
	LEVEL	/ME1/
	LLXSET	/I03/
	LXSET	F.P.
	MMSET	/I04/
	NNWRK	/I03/
	NTNUL	F.P.
B.	OUTPUT	LOCATION
	IERR1	F.P.
	IPTCMD	F.P.
	LLXSET	/I03/

LXSET

F.P.

NSET

6. CALLING ROUTINE: PROCES

7. CALLED ROUTINES: ARAYCK
EEROR

1. NAME: PSTRJ
2. PURPOSE: This routine prints out a label.
3. METHOD: If the label is being printed at a terminal, then fifteen A4 words is the maximum that can be printed. If the label is being printed on a line printer, then twenty A4 words is the maximum that can be printed.
4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Index pointer used to print label array.
I72	If equal to zero, then fifteen A4 words is the maximum that can be printed. If equal to one, then twenty A4 words is the maximum.
JPRT	Device code number of the print file.
LE LEN	Number of words of the LINE array to be printed.
LINE	Array that contains the label to be printed.
5. I/O VARIABLES:
 - A. INPUT LOCATION

I72	/ALLI/
JPRT	/ALLI/
LEN	F.P.
LINE	F.P.
 - B. OUTPUT LOCATION

(none)	
--------	--
6. CALLING ROUTINE: (not used at this time).
7. CALLED ROUTINE: (none)

1. NAME: PUTSTR = 84
2. PURPOSE: This routine processes PUT commands. The file number or pointer, the TABLE number, and the TABLE pointer are all stored in the NSET array to be processed during IOSYM execution.
3. METHOD: If another command points to this command, then it is set to indicate where this command starts. The current command number is stored in the NSET array. A check is done to see if there are enough fields for this command. If not, an error is processed, and control returns to the calling routine. If so, then the file identifier is stored in the uniform command sequence (UCS) for the PUT command in the NSET array. If the file number is a number, it is checked to see if it is a valid file number. If not, the number of files is increased to make it a valid file number. The starting location identifier, is stored in the UCS. The TABLE that the starting location is in is also stored in the UCS. A position is also set aside in the NSET array to point to the next command.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
CCONS	Array that contains all of the constants in a NETWORK.
I	The current TABLE number being investigated as the TABLE being used on a PUT command.
IERR	Error indicator. If FALSE, then no error has occurred in this routine. If TRUE, then a fatal error has occurred in this routine.
IERR1	Error indicator. If FALSE, then no error has occurred up to the present time. If TRUE, then a fatal error has occurred.
IN1	Number of the file being used to store the information on.
IPTCMD	Pointer to the next command. Location in the NSET array where the pointer to the next command is located. If IPTCMD is zero, then no command UCS points to the current command.

LLXSET	Counter of the number of locations in current use in the NSET array. Also the last position in the NSET array used.
LXSET	Temporary storage for the number of NSET elements needed.
MMSET	Maximum number of elements in the NSET array (dimension of the NSET array set in routine USRDAT).
NNFIL	Number of GASP files to be used.
NNPTB	Array that contains in the first two words the TABLE names; the third word is the pointer to the TTABL array where the start of the storage elements for the TABLE is located.
NNWRK	Work storage array. Contains the current command passed into the various elements.
NNXTB	Number of defined TABLEs.
NSET	Storage array where the link listed NETWORK is stored.

5. I/O VARIABLES;

A.	INPUT	LOCATION
	CCONS	/I02/
	IERR	F.P.
	IPTCMD	F.P.
	LLXSET	/I03/
	LXSET	F.P.
	MMSET	/I04/
	NNFIL	/GCOM1/
	NNPTB	/I02/
	NNWRK	/I03/
	NNXTB	/I03/

B.	OUTPUT	LOCATION
	IERR	F.P.
	IERR1	F.P.
	IPTCMD	F.P.
	LLXSET	/I03/
	LXSET	F.P.
	NNFIL	/GCOM1/
	NSET	/ /

6. CALLING ROUTINE: PROCES

7. CALLED ROUTINES: ARAYCK
EEROR

1. NAME: PVARJ
2. PURPOSE: This routine prints a simple variable with a label line.
3. METHOD: The array containing the label characters is identified, up to twenty characters, and the label and the variable are printed.
4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Do loop index.
ICHAR	Array containing the carriage control characters. The first word is a blank and will skip one line before printing. The second word is a zero and will skip two lines before printing.
ICRTL	Determines which carriage control character is to be printed.
JPRT	Device code to print output.
L, LEN	Length of the label array.
LINE	Array containing the label to print, four characters per word.
X	Variable value to print.

5. I/O VARIABLES:

A. INPUT	LOCATION
ICRTL	F. P.
JPRT	/ALLI/
LEN	F. P.
LINE	F. P.
X	F. P.
B. OUTPUT	LOCATION
(None)	

6. CALLING ROUTINE: (None. Not used at this time)

7. CALLED ROUTINE: (None)

1. NAME: PVEQJ
2. PURPOSE: This routine will print a real vector with labels and an introductory line.
3. METHOD: First the introductory line is printed, and then the vector values and labels are printed. Control then returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Index and counter used for various purposes.
IBLOCK	Determines which group of values is being printed out.
ICOL	Number of columns of data to print.
IROW	Number of rows of data to print.
I72	Indicator to determine how many columns of data to print. If equal to zero, three columns are used. If equal to one, five columns are done.
J	Index and counter.
JPRT	Device code to print information on.
J2	Last column number to print on current line.
L	Number of A4 words in the array LINE.
LA, LAB	Single character label.
LEN	Number of A4 words in the array LINE.
LINE	Alphanumeric string of less than sixty (terminal) or eighty (line printer) characters to be printed as an introductory label.
N, NI	Number of vector values to print.
X	Array containing the vector values to print.

5. I/O VARIABLES:

A. INPUT	LOCATION
I72	/ALLI/
JPRT	/ALLJ/
LAB	F. P.
LEN	F. P.
LINE	F. P.
N	F. P.
X	F. P.
B. OUTPUT	LOCATION
(None)	

6. CALLING ROUTINE: DDUMP

7. CALLED ROUTINE: (None)

1. NAME: READJ
2. PURPOSE: This routine reads a vector or a row or column of a matrix from free form input and multiplies it by the value of XMUL.
3. METHOD: Routine FFIJ is called to identify an integer value. Routine FFRJ is called to read a real value. This real value is multiplied by XMUL and stored in the location specified by the integer value. This continues until the integer number read is a zero.
4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
IROW	Row of the matrix to process.
J	Column in the matrix to process.
M	First dimension of the X array.
N	Second dimension of the X array.
X	Array to store the values read.
XMUL	Multiplication factor.
Z	Real value read that is to be multiplied.

5. I/O VARIABLES:

A. INPUT	LOCATION
IROW	F.P.
J	F.P.
M	F.P.
N	F.P.
XMUL	F.P.
Z	F.P.
B. OUTPUT	LOCATION
X	F.P.

6. CALLING ROUTINE: (None - Not used at this time)

7. CALLED ROUTINES:

FFIJ

FFPUDJ

FFRJ

PIVARJ

PSTRJ

1. NAME: REPSTR - 45
2. PURPOSE: This routine processes the REPLACE commands. It stores the replacement NODE number and the NODE number of the new NODE.
3. METHOD: If another command points to this command, then it is set to indicate where this command starts. The current command number is stored in the uniform command sequence (UCS). The internal NODE number of the NODE being replaced is stored as well as the replacement NODE number. A position is also set aside in the NSET array to point to the next command.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
IERR	Error indicator. If FALSE, then no error has occurred in this routine. If TRUE, then a fatal error has occurred in this routine.
IERR1	Error indicator. If FALSE, then no error has occurred up to the present time. If TRUE, a fatal error has occurred.
IPTCMD	Pointer to the next command. Location in the NSET array where the pointer to the next command is located. If IPTCMD is zero, then no command UCS points to the current command.
LLXSET	Counter of the number of locations in current use in the NSET array. Also the last position in the NSET array used.
LSET	Temporary storage for the number of NSET elements needed.
MMSET	Maximum number of elements in the NSET array (the dimension of the NSET array, set in routine USRDAT).
NNWRK	Work storage array. Contains the current command parsed into the various elements.
NSET	Storage array where the link listed NETWORK is stored.

5. I/O VARIABLES:

A. INPUT	LOCATION
IERR	F.P.
IPTCMD	F.P.
LLXSET	/I03/
LSET	F.P.
MMSET	/I04/
NNWRK	/I03/
B. OUTPUT	LOCATION
IERR	F.P.
IERR1	F.P.
IPTCMD	F.P.
LLXSET	/I03/
LSET	F.P.
NSET	/ /

6. CALLING ROUTINE: REPSTR

7. CALLED ROUTINE: ARAYCK

1. NAME: RESCC - 73
2. PURPOSE: This routine removes unneeded constants from the storage array (CCONS) and moves the remaining constants over to fill the blanks and fixes the pointers to point to the new locations.
3. METHOD: The number of words in the CCONS storage array from the first position to be removed to the end of the defined constants is calculated. Routine SSHIFT is called to shift out the unneeded constants. The constants at the end that are no longer needed are reinitialized. The pointers to the constants are corrected to point to the new locations.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
CCONS	Array that contains all of the constants in a NETWORK.
I	Index used to point to the different locations in the NNWRK array.
INUM	Number of constants to remove.
INUM1	Number of constants left to reinitialize.
ISTART	Location in the CCONS array where the first constant to remove is located.
LENGTH	Number of elements in the CCONS array from the first constant to remove to the end of the defined constants.
LLXWRK	Counter of the number of locations in current use in the NNWRK array. Also, the last position in the NNWRK array used.
NNCNS	Number of defined constants in the CCONS array. Also, the last position in the CCONS array used.
NNWRK	Work storage array. Contains the current command parsed into the various elements.

5. I/O VARIABLES:

A.	INPUT	LOCATION
	CCONS	/I02/
	INUM	F.P.
	ISTART	F.P.
	LLXWRK	/I03/
	NNCNS	/I03/
	NNWRK	/I03/
B.	OUTPUT	LOCATION
	CCONS	/I02/
	NNCNS	/I03/
	NNWRK	/I03/

6. CALLING ROUTINES: PCKSTR
STASTR
TRMSTR

7. CALLED ROUTINE: SSHIFT

1. NAME: RESEQ - 51
2. PURPOSE: This routine takes an algebraic expression and decodes it and stores it in the more efficient form of Forward Polish Notation (FPN).
3. METHOD: A check is done to make sure there is enough working storage space to do the FPN processing. If not, an error is processed, and control returns to the calling routine. The NNWRK array is broken into three parts. Part one contains the algebraic expression before processing; part two contains the space necessary to store the expression in FPN notation, and part three contains space for temporary storage for items before they are stored in the FPN expression.

An equal sign is stored as the first field in the FPN expression and the pointers incremented to point to the next position in the algebraic and FPN expressions. If a TABLE, constant, or variable is found, it is stored in the FPN expression.

If an IOSYM menu (F), probability (P), or user (U) function is found, a code of 10003 is stored to indicate the start of a function. The arguments in the argument part of the function are stored as FPN expressions. The function number is stored in the FPN expression as are the function TABLEs, parameter sets, entities, and truth conditions.

If the field is a closed parenthesis, then all the fields in temporary storage are copied into the FPN expression until the first open parenthesis is found.

If the field is an open parenthesis, then the open parenthesis indicator is stored in temporary storage.

If the field is an operator or library function, a check is done to see if it is a plus or minus sign. If so, and the previous field has an equal or open parenthesis, then a unary operator is processed. A check is done to see if the next operator in temporary storage has higher priority than the plus or minus. If so, then the field in temporary storage is stored in the FPN expression. If not, the plus or minus sign is stored in temporary storage.

If the operator is a multiply or divide, then a check is done to see if the next operator in temporary storage has higher priority than the multiply or divide. If so, then the field in temporary storage is stored in the FPN expression. If not, the multiply or divide is stored in temporary storage.

If the field is an exponential, then the field is stored in temporary storage.

If the field is not an operator, then the library function is processed by storing the function type and arguments in FPN.

After all of the algebraic expressions have been processed into FPN, then the FPN information is copied to where the algebraic information was previously stored, and control returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
CCONS	Array that contains all of the constants in a NETWORK.
I	Index used to point to various elements in the NNWRK array.
IERR1	Error indicator. If FALSE, then no fatal error has occurred up to the present time, If TRUE, then a fatal error has occurred.
ILIB	Indicator to show the number of stacked library functions that are being simultaneously worked on.
INDX1	Location in the NNWRK array where the FPN information is to be stored before returning to the calling routine.
IN1	Pointer to the current position in the NNWRK array where the last field in FPN is stored.
IN2	Pointer to the current position in the NNWRK array where the last field in temporary storage is stored.
IN3	Pointer to the current position in the NNWRK array where the current field being processed in the algebraic is located.
IN4 IN5 IN6	Pointers to various locations in the NNWRK array.
IN7	First position in the NNWRK array where the first FPN field is stored.

J	Pointers and index used for various
JJ	purposes.
LFNC	Location in the NNWRK array where the number of fields that are in the algebraic expression is stored.
LLXWRK	Counter of the number of locations in current use in the NNWRK array. Also, the last position in the NSET array used.
MMWRK	Maximum number of elements in the NNWRK array (dimension of the NNWRK set in routine USRDAT).
NMWRP	Location in the NNWRK where the last field in the algebraic expression is stored.
NNCNS	Number of locations in current use in the CCONS array. Also, the last position in the CCONS array used.
NNWRK	Work storage array. Initially contains the current command with its algebraic expression parsed into the various elements. During the execution of this routine, two other sections are added. The first one begins after the last word in the parsed command and is the location where the FPN expression fields are stored as the algebraic expression is processed. The last section begins after the FPN section and is temporary storage for fields before they are processed into the FPN expression.
N1	Number of items in an IOSYM function left to process.
N2	Number of arguments in the IOSYM function.
N3	Number of fields in the IOSYM function argument.

5. I/O VARIABLES:

A. INPUT LOCATION

INDX1 F.P.

LFNC F.P.

LLXWRK /I03/

MMWRK /I04/

NNCNS /I03/

NNWRK /I03/

B. OUTPUT LOCATION

CCONS /I02/

IERR1 F.P.

LLXWRK /I03/

NNCNS /I03/

NNWRK /I03/

6. CALLING ROUTINES: LOLFND
STASTR

7. CALLED ROUTINE: EEROR

1. NAME: RPTSTR - 36
2. PURPOSE: This routine processes REPORT and NOREPORT commands. The REPORT command stores variables to be REPORTed on later, and the NOREPORT command deletes some or all of those variables.
3. METHOD: If processing a REPORT command, each variable is coded into its variable identifier and stored in the report array (IIRPT). After all of the variables have been stored, control returns to the calling routine.

If processing a NOREPORT command and no variables are specified, then all the REPORT variables are deleted. If variables are specified, then a comparison of the variable identifiers in the REPORT array and the given variable is done to see if there is a match. If there is, it is deleted from the REPORT. This continues until all the given variables have been removed, and control returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Index used to point to location in the NNWRK and IIRPT arrays.
IERR	Error indicator. If FALSE, then an error has not occurred in this routine. If TRUE, then a fatal error has occurred in this routine.
IERR1	Error indicator. If FALSE, then an error hasn't occurred up to the present time. If TRUE, then a fatal error has occurred.
II	Index and counter used for various purposes.
IIRPT	REPORT variable storage array where the REPORT variable identifiers are stored.
IN1	When deleting a TABLE from the REPORT variable storage array, this indicates the first TABLE element identifier to remove.
IN2	When deleting a TABLE from the REPORT variable storage array, this indicates the last TABLE element identifier to remove.

ISUB	Indicator. If less than or equal to zero, then current command is NOREPORT. If equal to one, then current command is REPORT.
J	Index used to point to various locations in the REPORT variable storage array (IIRPT).
LLXWRK	Counter of the number of locations in current use in the NNWRK array. Also, the last position in the NNWRK array used.
MMRPT	Maximum number of REPORT variables that can be stored in the REPORT variable storage array (dimension of the IIRPT array, set in USRDAT).
NNPTB	Array that contains in the first two words the TABLE names, and the third word points to the location in the TTABL array where the TABLE element values are located.
NNRPT	Number of defined REPORT variables.
NNWRK	Work storage array. Contains the current command parsed into the various fields.
NNXTB	Number of defined TABLEs in a NETWORK.
NTNUL	Null field indicator.
NTTBL	TABLE field indicator.
NTVARD	Derivative variable field indicator.
NTVARO	Independent variable field indicator.
NTVAR1	Primary variable field indicator.
NTVAR2	Secondary variable field indicator.
5. I/O VARIABLES:	
A. INPUT	LOCATION
IERR	F.P.

IIRPT	/I01/
ISUB	F.P.
LLXWRK	/I03/
MMRPT	/I04/
NNPTB	/I02/
NNRPT	/I03/
NNWRK	/I03/
NNXTB	/I03/
NTNUL	F.P.
NTTBL	F.P.
NTVARD	F.P.
NTVAR0	F.P.
NTVAR1	F.P.
NTVAR2	F.P.
B. OUTPUT	LOCATION
IERR	F.P.
IERR1	F.P.
IIRPT	/I01/
NNRPT	/I03/
NNWRK	/I03/

6. CALLING ROUTINE: PROCES

7. CALLED ROUTINES: ARAYCK
EEROR

1. NAME: RREPLC - 527
2. PURPOSE: This routine executes a REPLACE command during an IOSYM simulation.
3. METHOD: This routine stores the replacement NODE number in the fourth word of the replaced NODE in array NNPTND, the NODE storage and pointing array.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
IERR1	Error indicator returned from routine EEROR but not used in this routine.
NNNODE	Number of defined NODEs in the NETWORK.
NNPTND	Array that contains in the first two words the NODE name; the third word contains the pointer to where the NODE information is stored in the NSET array, and the fourth word contains an indicator to show whether or not the NODE has been replaced. If the fourth word is zero, the NODE hasn't been replaced. If a positive value, that indicates what NODE number has replaced it.
NODIN	NODE number of the NODE to be the replacement.
NODOUT	NODE number of the NODE to be replaced.

5. I/O VARIABLES:

A. INPUT	LOCATION
IERR1	F.P.
NNNODE	/I03/
NNPTND	/I02/
NODIN	F.P.
NODOUT	F.P.

B. OUTPUT

LOCATION

NNPTND

/IO3/

6. **CALLING ROUTINE: PPFCMD**

7. **CALLED ROUTINE: EEROR**

1. NAME: RREPORT - 547
2. PURPOSE: This routine collects REPORT statistics during an IOSYM simulation.
3. METHOD: If an initialization of REPORT storage is being done, routine ZZSET is called to do initialization. If the statistics are to be stored, the current value of all the REPORT variables is found, and the variable value is added to the previous values for that variable. The squares of the variable are also added to the past squares for each variable.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Index and pointer to the array containing the REPORT variables.
IDV	Variable identifier of the REPORT variable being processed.
IFUNC	Indicator: 1 = doing REPORT initialization 2 = storing REPORT data values
IIRPT	Array containing the variable identifiers of the REPORT variables.
IISTAT	Indicator: 0 = don't do IOSYM statistics 1 = do IOSYM statistics
L	Pointer to the first location in the WWRK array used to store the REPORT values.
LLRPT	Location in the WWRK array where the REPORT data is stored.
NNRPT	Number of REPORT variables.
VAL	Value of the given REPORT variable.
WWRK	Array that contains the sum of the individual REPORT variable values and the sum of the squares of the individual variables.

5. I/O VARIABLES:

A.	INPUT	LOCATION
	IFUNC	F.P.
	IIRPT	/I01/
	IISTAT	/I03/
	LLRPT	/I03/
	NNRPT	/I03/
	WRK (NNWRK)	/I03/
B.	OUTPUT	LOCATION
	WRK (NNWRK)	/I03/

6. CALLING ROUTINES: DDOSIM
OUTPUT

7. CALLED ROUTINES: VVARBL
ZZSET

1. NAME: RREWRK - 526
2. PURPOSE: This routine is the space manager for the NNWRK/WWRK array. Unused words in this array are eliminated and the pointers to the locations redefined.
3. METHOD: This routine will free up locations in the NNWRK starting at the location LSTRT. If NNL is greater than zero, then it frees up that many words. If NNL is less than or equal to zero, then it frees up words until a zero word is found. Once the words have been freed, then the pointers are reset to point to the new locations. Control then returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
LL	Pointer to the current location when checking for a zero word.
LLCROS	Location in the NNWRK array that holds the pointer to the uniform command sequence for the first CROSS command on the current ARC.
LLEVNT	Location in the NNWRK array that holds the first parameter for an EVENT command.
LLIST	Location in the NNWRK array where the first pointer to the list of commands is located.
LLRPT	Location in the NNWRK array where the first word of REPORT storage is located.
LLTABU	Location in the NNWRK array where the first word of TABULATION data is located.
LLWITH	Location in the NNWRK array where the first word of WITH or MINIMIZE information is located.
LLWRK	Next available location in the NNWRK array.
LSTR	First word of the NNWRK array to be freed up.

LSTRT First word of the NNWRK array to be freed up.

MMWRK Maximum number of locations in the work storage array (dimension of the NNWRK array, set in USRDAT).

NN Number of words to free up or number of words freed up. If less than or equal to zero, then it indicates that words are to be freed up until a zero word is found.

NNL Indicator:

0 = number of words to free up

0 = free up words until a zero word is found

NNWRK Work storage array.

5. I/O VARIABLES:

A. INPUT	LOCATION
LLCROS	/I03/
LLEVNT	/I03/
LLIST	/I03/
LLRPT	/I03/
LLTABU	/I03/
LLWITH	/I03/
LLWRK	/I03/
LSTRT	F.P.
MMWRK	/I04/
NNL	F.P.
NNWRK	/I03/

B.	OUTPUT	LOCATION
	LLCROS	/I03/
	LLEVNT	/I03/
	LLIST	/I03/
	LLRPT	/I03/
	LLTABU	/I03/
	LLWITH	/I03/
	LLWRK	/I03/

6. CALLING ROUTINES: BBLIST
DDOEVN
EVNTS

7. CALLED ROUTINE: SSHIFT

1. NAME: RSAVIN - 79
2. PURPOSE: This routine executes the IOSYM command RESTORE. All the IOSYM and GASP common blocks are read off the given tape.
3. METHOD: This routine reads the given NETWORK information from the given tape and stores the information in the IOSYM and GASP common blocks. Routine TTAPES is called to store the GASP information read on the GASP tape. Control then returns to the calling routine. Only the variables specifically used in this routine are discussed below. Those variables that are just read in are not discussed.

4. INTERNAL VARIABLES:

VARIABLE	DEFINTION
EOFF	End of file indicator. If equal to zero, then there is more information on the tape. If it is any other value, then have reached the end of the file.
IERR	Error indicator. Returned from routine EEROR but not used in this routine.
ITP	Tape number of the tape that contains the NETWORK information.
LLXSET	Number of defined locations in the NSET array.
MMSET	Maximum number of locations available in the NSET array.
MSET	Maximum number of locations read into the NSET array.
N	Number of NSET values to read from the file.

5. I/O VARIABLES:

A. INPUT	LOCATION
ITP	F.P.
B. OUTPUT	LOCATION
EOFF	F.P.

6. CALLING ROUTINE: SAVE

7. CALLED ROUTINES: EEOF (machine specific)
TTAPES

1. NAME: SAVE - 74
2. PURPOSE: This routine processes the SAVE and RESTORE commands.
3. METHOD: If processing a SAVE command, the scratch tape and the SAVE tape are identified. Routine WSAVIN is called to SAVE the current NETWORK information on the scratch tape. Routine RSAVIN is called to read NETWORK information from the SAVE tape. If the NETWORK name agrees with the one already saved, the next NETWORK is read from the SAVE tape. This continues until all the NETWORKs have been stored on the scratch tape. Both the scratch tape and the SAVE tapes are rewound, and the information on the scratch tape is copied to the SAVE tape. The original NETWORK is then read in. Control then returns to the calling routine.

If processing a RESTORE command, the SAVE tape is identified and the NETWORKs read off one at a time until the one requested is found. Control then returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
EOFF	End of file indicator. If equal to zero, then there is more information on the tape. If it is any other value, then the end of the file has been reached.
IERR1	Error indicator returned by routine EEROR but not used in this routine.
IITPS	Array that contains the device code numbers for the various tapes. IITPS(1) is the NETWORK SAVE device code number. IITPS(5) is the scratch device code number.
INAME	Array that contains the NETWORK name of the NETWORK to SAVE or RESTORE.
IRTP	Device code number for the tape that contains the SAVE information.
ISUB	Indicator. If equal to zero, then the current command is SAVE. If less than zero, then current command is RESTORE.

IWRITE

Indicator. If equal to one, then a NETWORK summary is done after doing the SAVE or RESTORE command. Any other value and the NETWORK summary isn't done.

IWTP

Device code number for the scratch tape.

NNMNET

Array that contains in the first two words the NETWORK name, and the third word indicates whether or not a NETWORK summary is to be done.

5. I/O VARIABLES:

A. INPUT

LOCATION

E0FF

F.P.

IERR1

F.P.

IITPS

/I01/

ISUB

F.P.

NNMNET

/I03/

B. OUTPUT

LOCATION

(None)

6. CALLING ROUTINE: PROCES

7. CALLED ROUTINES: EEROR
NETSUM
RSAVIN
WSAVIN

1. NAME: SCAN - 19
2. PURPOSE: This routine controls the reading of the input cards and calls the appropriate routines to build the different fields.
3. METHOD: If the last input record processed by this routine was GASP or GO, then a new record is needed so one is read. If any other command was processed, an input record is ready to be processed. If the input cards are to be ECHOed, then the input record is written to the output file. Each column from left to right in the input record is checked for a non-blank character. Depending on what is found, different things occur.

If an alpha field is found, routine BLDALP is called to build an alphanumeric field.

If a number was found, routine BLDNUM is called to build an integer, floating point, or exponential number.

If an operator is found, routine BLDOPR is called to identify which operator was found.

If a delimiter is found, routine BLDDLM is called to identify which delimiter was found.

Each of the columns is investigated until the end of the record is reached.

If the current card isn't a GASP or GO command card, the next input record is read. If the last field in the previous record is an operator or delimiter, or the first field in this record is a delimiter, then this record is processed. If they aren't, then this record is saved for processing next time through. Control then returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
ECHO	Indicator to show whether input cards read while defining a NETWORK are printed out or ECHOed. If TRUE, then they are printed out. If FALSE, they are not.
FIRST	Indicator. If TRUE, then haven't finished processing the first IOSYM input record yet. If FALSE, then the first input record has already been processed.

IALPHA	Contains the integer representation of four characters of an alphanumeric string.
ICHAR	Array containing the internal IOSYM representation of all of the characters (A-Z, 0-9, ., **, *, /, +, -, =,), (, ,, \$).
ICLEVL	An array that contains the current command hierarchy from level one to level LEVEL.
IERRF	Premature end of file indicator. If TRUE, then the end of the input file has reached prematurely.
IERR1	Error indicator. If FALSE, then no error has occurred up to the present time. If TRUE, then a fatal error has occurred.
IERR2	Error indicator. If FALSE, then no error has occurred in the SCAN routines. If TRUE, then a fatal error has occurred in the SCAN routines.
KARDD	The array containing the characters of the input data record currently being worked on.
KBLANK	Internal IOSYM representation of a blank character.
KHAR	Current input character being investigated.
LITMAX	Maximum number of fields that can be processed (the dimension of the LITSTR, NCODE, and NVAL arrays, set in USRDAT).
N	Index pointer and counter used for various purposes.
NCARD	Number of input records that have been processed.
NCODE	Array containing the type of fields that were processed from the input records.

NCRDR	The input device code number for reading input (set in USRDAT).
NDLM	Delimiter value of the delimiter identified in routine BLDDLM.
NDX	Pointer to the current character under investigation in the input record read.
NDXVAL	Counter to show how many fields have been processed.
NDX1	Current command number.
NOPR	Operator value of the operator identified in routine BLDDPR.
NPRNT	The output device code number for writing output (set in USRDAT).
NTAB	Current column in the input data record under investigation.
NTALP	Alpha field indicator.
NTALPC	Alpha field indicator.
NTDLM	Delimiter field indicator.
NTERR	Error field indicator.
NTFLD	Contains the type of number field built in BLDNUM (integer or floating point).
NTFLP	Floating point number field indicator.
NTINT	Integer number field indicator.
NTNUL	Null field indicator.
NTOPR	Operator field indicator.
NUM	Number built by routine BLDNUM.
NVAL	Array containing the values of the fields that were processed from the input record.

5. I/O VARIABLES:

A. INPUT	LOCATION
ECHO	/ME1/
FIRST	/ME1/
IALPHA	F.P.
ICHAR	/ME1/
ICLEVL	/ME1/
IERRF	F.P.
IERR2	F.P.
KARDD	/ME1/
LITMAX	F.P.
NCARD	/ME1/
NCRDR	/GCOM1/
NDLM	F.P.
NDXVAL	/ME1/
NOPR	F.P.
NPRNT	/GCOM1/
NTAB	F.P.
NTALP	F.P.
NTALPC	F.P.
NTDLM	F.P.
NTERR	F.P.
NTFLD	F.P.
NTFLP	F.P.
NTINT	F.P.

	NTNUL	F.P.
	NUM	F.P.
	NVAL	F.P.
B.	OUTPUT	LOCATION
	FIRST	/ME1/
	IERR1	F.P.
	KARDD	/ME1/
	NCARD	/ME1/
	NCODE	F.P.
	NDX1	/ME1/
	NTAB	F.P.
	NVAL	F.P.

6. CALLING ROUTINE: PARSE

7. CALLED ROUTINES: ARAYCK
 BLDALP
 BLDDLML
 BLDNUM
 BLDOPR
 EEOF (Machine specific)
 EEROR

1. NAME: SCOND - 522
2. PURPOSE: This routine is called by GASP to determine a CROSS event or process any user state events.
3. METHOD: If there are current CROSS commands, they are evaluated by calling routine KROSS and the result stored in the LFLAG array to be used by GASP. Routine USCOND, a user written routine, is called to process any user state conditions. Control then returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
IFLAG	Next available LFLAG array location that can be used.
ITF	Truth condition indicator: <ul style="list-style-type: none"> 1 = Less than or equal to zero 2 = Less than zero 3 = Equal to zero 4 = Greater than zero 5 = Greater than or equal to zero
L	Location in the NNWRK array where the pointer to the next CROSS command is located.
LFLAG	State condition flags (see GASP book for more description).
LFNC	Location in the NSET array where the CROSS command information is stored.
LLCROS	Location in the NNWRK array that contains the pointer to the uniform command sequence for the first CROSS command on the current ARC.
LSS	Index of the crossing variable. The crossing variable is a: <ul style="list-style-type: none"> primary variable if $LSS > 0$, derivative variable if $LSS < 0$, none if $LSS = 0$.
NFLAG	Number of state condition flags.

NFLG	Number of user state condition flags.
NNCROS	Number of CROSS commands on the current ARC.
NNEQT	Number of primary, derivative, and current CROSS variables.
NNWRK	Work storage array.
NSET	Storage array containing the link listed NETWORK.
QSET	Equivalenced to the NSET array used to retrieve floating point numbers from the NSET array.

5. I/O VARIABLES:

A. INPUT	LOCATION
LLCROS	/I03/
NNCROS	/I03/
NNEQT	/GCOM2/
NNWRK	/I03/
NSET	/ /
QSET	/ /
B. OUTPUT	LOCATION
LFLAG	/GCOM2/
NFLAG	/GCOM2/

6. CALLING ROUTINE: GASP (main routine in the GASP program)

7. CALLED ROUTINES: KROSS (GASP routine)
USCOND (user written routine)

1. NAME: SETJ
2. PURPOSE: This routine initializes the ALLJ common block.
3. METHOD: Each of the variables in the ALLJ common block is set to an initial value.
4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
II72, I72	Indicator. If equal to zero,, then there are only 72 columns per computer page. If equal to one, there are more.
JPRNT, JPNT	Device code number to print output.
JRD, JRDD	Device code number to read input.
NSI, NSIG	Number of significant digits.
PREC	Machine precision.

5. I/O VARIABLES:

A. INPUT	LOCATION
II72	F. P.
JPRNT	F. P.
JRDD	F. P.
B. OUTPUT	LOCATION
I72	/ALLJ/
JPRT	/ALLJ/
JRD	/ALLJ/
NSIG	/ALLJ/
PREC	/ALLJ/

6. CALLING ROUTINE: IOSYM
7. CALLED ROUTINES: (None)

1. NAME: SETSTR - 30
2. PURPOSE: This routine stores the uniform command sequence (UCS) for the SET command (assignment statement with an equal sign). It stores the variable to be set, pointer to the expression, and the truth condition.
3. METHOD: The command number is stored in the NSET array. If another command points to this one, a pointer is stored to point to where this command begins. The assignment variable is stored in the UCS. A location for a pointer to the expression is set aside. The truth condition is stored in the UCS, and a location for the pointer to the next command is set aside. Routine FUNSTR is called to process and store the expression itself. Control then returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
IERR	Error indicator. If FALSE, then an error hasn't occurred in this routine. If TRUE, then a fatal error has occurred in this routine.
IERR1	Error indicator. If FALSE, then an error hasn't occurred up to the present time. If TRUE, then a fatal error has occurred.
INX	Location in the work array where the truth condition is stored.
INX1	Position in the NSET array where the pointer to the location in the NSET array, where the expression starts, is located.
IPTCMD	Pointer to the next command. If IPTCMD is zero, then no command uniform sequence points to the current command.
LLXSET	Counter of the number of locations in current use in the NSET array. Also, the last position in the NSET array used.
LSET	Temporary counter of the number of locations in the NSET array needed.

MMSET	Maximum number of locations in the NSET array (dimension of the NSET array set in routine USRDAT).
NNWRK	Work storage array. Contains the current command parsed into the various elements.
NSET	Storage array where the link listed NETWORK is stored.
NTVARD	Derivative variable field indicator.
NTVAR0	Independent variable field indicator.
NTVAR1	Primary variable field indicator.
NTVAR2	Secondary variable field indicator.

5. I/O VARIABLES:

A. INPUT	LOCATION
IERR	F.P.
IERR1	F.P.
IPTCMD	F.P.
LLXSET	/I03/
LSET	F.P.
MMSET	/I04/
NNWRK	/I03/
NTVARD	F.P.
NTVAR0	F.P.
NTVAR1	F.P.
NTVAR2	F.P.
B. OUTPUT	LOCATION
IERR	F.P.
IERR1	F.P.

IPTCMD	F.P.
LLXSET	/I03/
LSET	F.P.
NNWRK	/I03/
NSET	/ /

6. CALLING ROUTINE: PROCES
7. CALLED ROUTINES: ARAYCK
FUNSTR

1. NAME: SORTJ
2. PURPOSE: This routine will sort array X, from lowest to highest and maintain the correspondences of X(I) to Y(I).

3. METHOD: A common shell sort is done.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I J	Do loop indices.
N NN	Lengths of the X and Y arrays to process.
X	Array to sort from lowest to highest.
Y	Array corresponding to the X array.
Z	Temporary variable.

5. I/O VARIABLES:

A. INPUT	LOCATION
NN	F.P.
X	F.P.
Y	F.P.
B. OUTPUT	LOCATION
X	F.P.
Y	F.P.

6. CALLING ROUTINE: (None - not used at this time)

7. CALLED ROUTINE: (None)

1. NAME: SRHARC - 31
2. PURPOSE: This routine processes the ACTIVATE, DEACTIVATE, and SWITCH commands during a NETWORK. It stores the command number and the internal IOSYM ARC number. This routine also processes these commands under a CHANGE command. In this case, the initial status of the ARC is changed.
3. METHOD: If an ACTIVATE, DEACTIVATE, or SWITCH command is being processed while reading in a NETWORK, the command number is stored in the NSET array, and if another command points to this one, the start location of this command is stored. The internal IOSYM ARC number is also stored, and a pointer to the next command is also set aside. Control then returns to the calling routine.

If an ACTIVATE, DEACTIVATE, or SWITCH command is being processed under the CHANGE mode, the ARC is identified, and the location in its uniform command sequence (UCS), where the initial status is located, is found. The appropriate information, depending on what command is being processed, is stored in the ARC UCS. Control then returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
ICLEVL	Array that contains the current command hierarchy from level one to level LEVEL.
IERR	Error indicator. If FALSE, then no error has occurred in this routine. If TRUE, then a fatal error has occurred in this routine.
IERR1	Error indicator. If FALSE, then no error has occurred up to the present time. If TRUE, then a fatal error has occurred.
IN1	Pointer to various items in the ARC UCS.
IN2	Current command number.
IPTCMD	Pointer to the next command. If IPTCMD is zero, then no command uniform command sequence points to the current command.

LEVEL	Indicates what level of command hierarchy is being processed. There are five levels: 0th, 1st, 2nd, 3rd, and 4th.
LLXSET	Counter of the number of locations in current use in the NSET array. Also, the last position in the NSET array used.
LSET	Temporary counter of the number of locations in the NSET array needed.
MMSET	Maximum number of locations in the NSET array (dimension of the NSET array set in routine USRDAT).
NNPTAC	Array containing in the first two words the internal IOSYM representation of the ARC name, and the third word is the pointer to the location in the NSET array where the ARC information is located.
NNWRK	Work storage array. Contains the current command parsed into the various elements.
NSET	Storage array where the link listed NETWORK is stored.

5. I/O VARIABLES:

A. INPUT	LOCATION
ICLEVL	/ME1/
IERR	F.P.
IPTCMD	F.P.
LEVEL	/ME1/
LLXSET	/I03/
LSET	F.P.
MMSET	/I04/
NNPTAC	/I02/

NNWRK	/I03/
NSET	/ /
B. OUTPUT	LOCATION
IERR	F.P.
IERR1	F.P.
IPTCMD	F.P.
LLXSET	/I03/
LSET	F.P.
NSET	/ /

6. CALLING ROUTINE: PROCES

7. CALLED ROUTINE: ARAYCK

1. NAME: SSAVE - 551
2. PURPOSE: This routine stores the IOSYM plot information.
3. METHOD: If statistics aren't being collected or there are no IOSYM plots to do, then control returns to the calling routine. The plot storage array is searched for active plots. The current values of the independent and dependent variables are identified, and routine WWRIT is called to store the values on the plot tape. After all the active plots have been processed, control returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Index to count number of plots processed.
IERR1	Error indicator returned from routine EEROR but not used in this routine.
IISTAT	Indicator. If equal to zero, then no IOSYM statistics are to be processed. If equal to one, then IOSYM statistics are to be collected.
INUM	Number of plots processed.
J	Pointer to the current dependent variable being processed.
MMPLOT	Maximum number of PLOTS on a NETWORK (dimension of NNPLT, set in routine USRDAT).
NNPLOT	Array that contains the PLOT number, the independent variable identifier, and the five dependent variable identifiers for all the plots.
NNPLTS	Number of active PLOTS in the current NETWORK.
NV	Number of dependent variables in a PLOT.
V	Array containing the current values of the five dependent variables.
XIND	Current value of the independent variable.

5. I/O VARIABLES:

A. INPUT	LOCATION
IERR1	F.P.
IISTAT	/I03/
MMPL0T	/I04/
NNPLOT	/I01/
NNPLTS	/I03/
B. OUTPUT	LOCATION
INUM	F.P.
NV	F.P.
V	F.P.
XIND	F.P.

6. CALLING ROUTINE: GASP (main routine in the GASP program)

7. CALLED ROUTINES: EEROR
USSAVE (user written routine)
VVARBL
WWRIT

1. NAME: SSHIFT - 520
2. PURPOSE: This routine will perform an end around or end off shift, up or down NWORDS words, in the array X or IX or both.
3. METHOD: If an end off shift up is to be done, then NX number of words are shifted NX number of words closer to the first word, writing over the old values. If an end around shift up is to be done, then all the words in the array are shifted NX words closer to the first word, with those falling off being brought back on at the end of the array. If an end off shift down is to be done, then NX number of words are shifted NX number of words further from the first words, writing over the old values. If an end around shift down is to be done, then all the words in the array are shifted NX words further from the first word, with those falling off the end being brought back on at the beginning of the array.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Index to current word being manipulated.
ISAVE	Array containing the current values of the IX array being manipulated.
IX	Array containing integer values to be shifted around.
JOFF	1 = Shift end off 2 = Shift end around.
JOFFB	1 = Shift end off 2 = Shift end around.
JUPD	1 = Shift up (toward the first word) 2 = Shift down (toward the last word).
JUPDN	1 = Shift up (toward the first word) 2 = Shift down (toward the last word).
JWCH	1 = Shift the IX array 2 = Shift the X array 3 = Shift both the IX and X arrays.
JWHCH	1 = Shift the IX array 2 = Shift the X array 3 = Shift both the IX and X arrays.
LL	Index to determine which group of ten words is currently being processed.

LOOPS	Number of groups of ten words to process.
LSAV	Number of words in each group.
LX	Location in the X or IX arrays where the next word to process is located.
L1	Location in the X or IX arrays to store the current value in the X or IX arrays at location L2.
L2	Location in the X or IX array that is to be stored in the L1 location in the X or IX arrays.
N	Length of the X and IX arrays.
NNWS	Number of words currently shifting.
NW	Number of words currently processing.
NWO	Number of words to shift.
NWORDS	Number of words to shift.
NX	Length of the X and IX arrays.
SAVE	Array containing the current values of the X array being manipulated.
X	Array containing the floating point values to be shifted around.

5. I/O VARIABLES:

A.	INPUT	LOCATION
	ISAVE	F.P.
	IX	F.P.
	JOFF	F.P.
	JUPDN	F.P.
	JWHCH	F.P.
	NWORDS	F.P.
	NX	F.P.

SAVE	F.P.
X	F.P.
B. OUTPUT	LOCATION
ISAVE	F.P.
IX	F.P.
NW	F.P.
SAVE	F.P.
X	F.P.

6. CALLING ROUTINES:

COLSTR
PROCES
RESCC
RREWRK
TBLFND
WTHSTR

7. CALLED ROUTINES:

IIMOV
VVMOV

1. NAME: SSTWRK - 555
2. PURPOSE: This routine initializes the pointer variables LLIST, LLCROS, LLEVNT, LLTABU, LLRPT, LLWITH, LLWRK, and LLCOL.
3. METHOD: This routine checks to see if there are any active TABULATE commands and if so, the number of locations needed in the work storage array is calculated, and LLTABU is set. If there are REPORT variables, the amount of work storage is calculated, and LLRPT is set. If there are COLLECT OPERATIONS and COLLECT variables, then the amount of storage is calculated, and LLCOL is set. Pointer LLWRK is set to the next available location in the work storage array, and all the other pointers are set to zero.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Index used to determine which TABULATE is being processed.
ICNT	Number of TABULATE variables.
IDV	Variable identifier of the current TABULATE variable being processed.
IERR1	Error indicator returned from routine EEROR but not used in this routine.
J	Pointer to which variable on a TABULATE command is being processed.
LLCOL	Location in the NSET array where the first word of COLLECT statistics are stored.
LLCROS	Location in the NNWRK array that holds the pointer to the first CROSS command uniform command sequence on the current ARC.
LLEVNT	Location in the NNWRK array that holds the first parameter for an EVENT command.
LLIST	Location in the NNWRK array that holds the pointer to the first command in a list of commands.

LLRPT	Location in the NNWRK array where the first word of the REPORT statistics are stored.
LLTABU	Location in the NNWRK array of the first word of TABULATE data.
LLWRK	Next available location in the work storage array (NNWRK).
LLXSET	Last location in the NSET array used.
MMSET	Maximum number of elements in the NSET array (dimension of the NSET array, set in routine USRDAT).
MMTABU	Maximum number of TABULATE commands in a NETWORK (dimension of the NNTABU array, set in USRDAT).
MMWRK	Maximum number of elements in the work storage array (dimension of the NNWRK array, set in routine USRDAT).
NNCOL	Number of COLLECT variables in a NETWORK.
NNCOP	Number of COLLECT OPERATIONS in a NETWORK.
NNRPT	Number of REPORT variables in a NETWORK.
NNTABU	Array containing the TABULATE information.
NNTBS	Number of TABULATE variables.

5. I/O VARIABLES:

A. INPUT	LOCATION
IERR1	F.P.
LLWRK	/I03/
LLXSET	/I03/
MMSET	/I04/

MMTABU	/I04/
MMWRK	/I04/
NNCOL	/I03/
NNCOP	/I03/
NNRPT	/I03/
NNTABU	/I03/
B. OUTPUT	LOCATION
LLCOL	/I03/
LLCROS	/I03/
LLEVNT	/I03/
LLIST	/I03/
LLRPT	/I03/
LLTABU	/I03/
LLWRK	/I03/

6. CALLING ROUTINE: DDOSIM
7. CALLED ROUTINE: EEROR

1. NAME: SSWARC - 505
2. PURPOSE: This routine changes the current status of an ARC.
3. METHOD: The location in the NSET array where the ARC uniform command sequence is located is identified. The location of the current ARC status indicator is calculated. Then, depending on what the current ARC status is to be set to, a one or minus one, or the sign is changed on the current ARC status indicator.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
IARC	ARC number of the ARC that is to have its current status changed.
IDO	Status change indicator: 1 = ACTIVATE 2 = DEACTIVATE 3 = SWITCH
IERR1	Error indicator returned from routine EEROR but not used in this routine.
L	Location in the NSET array where the current ARC status indicator is located.
NNARC	Number of defined ARCs in the NETWORK.
NNPTAC	Array where the first two words contain the ARC names, and the third word points to the location in the NSET array of the ARC uniform command sequence.
NSET	Storage array where the link listed NETWORK is stored.

5. I/O VARIABLES:

A. INPUT	LOCATION
IARC	F.P.
IDO	F.P.
IERR1	F.P.

NNARC /I03/

NNPTAC /I02/

NSET / /

B. OUTPUT LOCATION

NSET / /

6. CALLING ROUTINE: PPFCMD

7. CALLED ROUTINE: EEROR

1. NAME: STASTR - 85
2. PURPOSE: This routine processes STAT commands and stores the appropriate information needed to execute the STAT command during an IOSYM simulation.
3. METHOD: If another command points to this command, it is set to indicate where this command starts. The current command number is stored in the NSET array. The statistics type is identified and stored in the NSET array along with the statistic number. A pointer to the expression is set aside. The T variable identifier is stored, and a position in the NSET array is set aside to point to the next command.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
CCONS	Array that contains all of the constants in a NETWORK.
IERR	Error indicator. If FALSE, then an error hasn't occurred in this routine. If TRUE, then a fatal error has occurred in this routine.
IERR1	Error indicator. If FALSE, then an error hasn't occurred up to the present time. If TRUE, then a fatal error has occurred.
IN1	Index and pointer used for various purposes.
IN2	Statistic number of the current statistic type.
IN3	Location in the CCONS array where the statistic number is stored.
IPTCMD	Pointer to the next command. Location in the NSET array where the pointer to the next command is located. If IPTCMD is zero, then no command uniform command sequence points to the current command.
LLXSET	Counter of the number of locations in current use in the NSET array. Also, the last position in the NSET array used.

LXSET	Temporary storage for the number of NSET elements needed.
MMSET	Maximum number of elements in the NSET array (dimension of the NSET array set in routine USRDAT).
NNCLT	Number of sets of statistics to be collected by GASP routine COLCT.
NNCNS	Number of constants stored in the constants array. Also, the last position in the NNCNS array used.
NNHIS	Number of histograms to be done by GASP.
NNSTA	Number of sets of statistics that can be collected by the GASP routines TIMST and TIMSA.
NNWRK	Work storage array. Contains the current command parsed into the various elements.
NSET	Storage array that contains the link listed IOSYM NETWORK.

5. I/O VARIABLES:

A. INPUT	LOCATION
CCONS	/I02/
IERR	F.P.
IPTCMD	F.P.
LLXSET	/I03/
LXSET	F.P.
MMSET	/I04/
NNCLT	/GCOM4/
NNCNS	/I03/
NNHIS	/GCOM4/

NNSTA	/GCOM4/
NNWRK	/I03/
B. OUTPUT	LOCATION
IERR	F.P.
IERR1	F.P.
IPTCMD	F.P.
LLXSET	/I03/
LXSET	F.P.
NNWRK	/I03/
NSET	/ /

6. CALLING ROUTINE: PROCES

7. CALLED ROUTINES: ARAYCK
EEROR
FUNSTR
RESCC
RESEQ

1. NAME: STATE - 523
2. PURPOSE: This routine defines the state variables.
3. METHOD: This routine initializes all the derivative variables to zero. Routine DDOLST is called to do the commands in the list. Routine USTATE, a user written routine, is called to do any user state definitions. All the IOSYM CROSS commands are evaluated and control returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
DD	Array containing the derivative variable values at the time TNOW.
L	Location in the NNWRK that contains the pointer to the uniform command sequence for the next CROSS command.
LFNC	Location in the NSET array where the first word of the uniform command sequence for the CROSS command is located.
LLCROS	Number of CROSS commands and derivative variables.
LSS	Pointer to the SS array where the value of the CROSS variable is located.
NNCROS	Number of CROSS commands on the current ARC.
NNEQD	Number of primary variables in the NETWORK.
NNEQT	Number of derivative, state, and CROSS variables.
NNWRK	Work storage array.
NSET	Storage array where the link listed NETWORK is stored.
SS	Array containing the value of the primary variables at the time TNOW.

5. I/O VARIABLES:

A. INPUT	LOCATION
DD	/GCOM2/
LLCROS	/I03/
NNCROS	/I03/
NNEQD	/GCOM2/
NNEQT	/GCOM2/
NNWRK	/I03/
NSET	/ /
SS	/GCOM2/
B. OUTPUT	LOCATION

(None)

6. CALLING ROUTINES:

AARCST
GASP (GASP routine)
XXINTL

7. CALLED ROUTINES:

DDOLST
EEVAL
USTATE (user written)
ZZSET

INDA	Number of TABLE elements to be stored in the TTABL array.
IN1	Index and counter used for various purposes.
JJTAB	Number of defined TABLEs plus one.
JTAB	Pointer to locations in the NNPTB array.
LENGTH	Number of TTABL elements needed to move to fill in or open up for a TABLE.
LEVEL	Indicates what level of command hierarchy is being processed. There are five levels: 0th, 1st, 2nd, 3rd, and 4th.
LITMAX	Maximum number of items that can be stored in the literal storage array (dimension of the LITSTR array set in USRDAT).
LITNDX	Index pointing to the current position in the literal storage array being processed.
LITSTR	Literal storage array where the first word is the literal field type, and the second and third words contain the field value.
LLTAB	Last position in the TTABL array used to store the elements in the current TABLE.
LLXWRK	Counter of the number of locations in current use in the NNWRK (work storage) array. Also, the last position in the NNWRK array used.
LOL	Indicator. If TRUE, then a list of literals is desired. If FALSE, only one literal is desired.
LTAB	Location in the TTABL array where the last element of the last TABLE is stored.

1. NAME: TBLFND - 38
2. PURPOSE: This routine processes TABLE commands and stores the initial values and the TABLE name in the appropriate locations.
3. METHOD: Routine NAMFND is called to process the TABLE name. Routine NUMFND is called to determine the TABLE dimension. If this TABLE was already defined, and the current dimension isn't the same as the old dimension, and this processing is occurring under the CHANGE mode, an error is processed. If not in CHANGE mode, the number of TABLE elements is adjusted and shifted depending on the new dimension. The TABLE values are stored by calling routine NUMFND to get the number, then storing it in the appropriate position in the TTABL array. Repetitive numbers can be put in like 6*1.0, and six of the TABLE elements will be stored as ones. After all the elements are stored, control returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
CCONS	Array containing all of the constants in a NETWORK.
I	Index and pointer used for various purposes.
ICLEVL	An array containing the current command hierarchy from level one to level LEVEL.
IERR	Error indicator. If FALSE, then an error hasn't occurred in this routine. If TRUE, then a fatal error has occurred in this routine.
IERR1	Error indicator. If FALSE, then an error hasn't occurred up to the present time. If TRUE, then a fatal error has occurred.
II	Pointer to location in the TTABL array where a TABLE element value is being stored.
IMOVE	Number of new TABLE element places needed or number of old element places no longer needed in the TTABL array.

MMTAB	Maximum number of TABLE elements that can be stored in the TTABL array (dimension of the TTABL array set in USRDAT).
MMXTB	Maximum number of TABLEs that can be stored in the NNPTB array (dimension of the NNPTB array set in USRDAT).
NDXLIT	Pointer to the current position in the literal storage array LITSTR being investigated.
NNCNS	Number of defined locations in the constant storage array (also the last position used in that array).
NNPTB	Array containing in the first two words the TABLE name, and the third word is the pointer to the location in the TTABL array where the TABLE element values are stored.
NNWRK	Work storage array.
NNXTB	Number of defined TABLEs in a NETWORK. Also, the last position in the NNPTB array used.
NOADD	Indicator. If FALSE, then an alpha field was found. If TRUE, then an alpha field wasn't found.
NOELE	Indicator. If TRUE, then there are no defined elements in a TABLE so a default of one is used. If FALSE, then there are defined elements for the TABLE.
NTALPC	Alpha field indicator.
NTDLM	Delimiter field indicator.
NTFLP	Floating point number field indicator.
NTINT	Integer number field indicator.
NTNUL	Null field indicator.
NTOPR	Operator field indicator.

PPREC

Precision of the computer machine,
set in USRDAT.

TTABL

Array containing all the TABLE values.

5. I/O VARIABLES:

A. INPUT

LOCATION

CCONS

/I02/

ICLEVL

/ME1/

IERR

F.P.

LEVEL

/ME1/

LITMAX

F.P.

LITNDX

F.P.

LITSTR

F.P.

LLXWRK

/I03/

MMTAB

/I04/

MMXTB

/I04/

NDXLIT

F.P.

NNCNS

/I03/

NNPTB

/I02/

NNWRK

/I03/

NNXTB

/I03/

NOADD

F.P.

NTALPC

F.P.

NTDLM

F.P.

NTFLP

F.P.

NTINT

F.P.

NTNUL

F.P.

	NTOPR	F.P.
	PPREC	/I04/
B.	OUTPUT	LOCATION
	CCONS	/I02/
	IERR	F.P.
	IERR1	F.P.
	LOL	F.P.
	NDXLIT	F.P.
	NNCNS	/I03/
	NNPTB	/I02/
	NNWRK	/I03/
	NNXTB	/I03/
	NOADD	F.P.

6. CALLING ROUTINE; PARSE

7. CALLED ROUTINE: ARAYCK
EEROR
NAMFND
NUMFND
SShift

1. NAME: TBTSTR - 37
2. PURPOSE: This subroutine processes TABULATE and NOTABULATE commands and stores the information to do tabulations during an IOSYM simulation.
3. METHOD: If a TABULATE command is being processed, the TABULATE number is identified. The time option is identified and stored in the NNTABU array. The variable identifiers and the report type for each variable are identified and stored in the NNTABU array. Control then returns to the calling routine.

If a NOTABULATE command is being processed, the TABULATE number is identified, and the location in the NNTABU array where that TABULATE number information is stored is initialized to minus ones. Control then returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
CCONS	This array contains all the constants in a NETWORK.
I	Index used to point to the location in the NNTABU array to initialize.
IERR	Error indicator. If FALSE, then an error hasn't occurred in this routine. If TRUE, then a fatal error has occurred in this routine.
IERR1	Error indicator. If FALSE, then an error hasn't occurred up to the present time. If TRUE, then a fatal error has occurred.
IINDX	Pointer to the NNWRK array where the current field being processed is located.
INDA	Indicator 1 = Time option off 2 = Time option on
INDX16	Location in the NNTABU array where the next TABULATE variable is to be stored.

IN1	TABULATE number of the TABULATE or NOTABULATE command being processed.
ISSET	Indicator of the variable report type. 1 = Start 2 = End 3 = Change
ISUB	Indicator 0 = Process NOTABULATE command 1 = Process TABULATE command
LLTABU	Tabulation number.
LLXWRK	Last position in the work storage array used.
MMTABU	Maximum number of TABULATEs in a NETWORK.
NNCNS	Number of constants in a NETWORK.
NNTABU	Array containing the TABULATE information. The first word indicates whether the time option is on or off; the second, fourth, sixth and eighth words are the TABULATE variable identifiers, with the third, fifth, seventh and ninth words being the report types for the four variables, respectively.
NNWRK	Work storage array. Contains the current command parsed into the various fields.
NTNUL	Null field indicator.
NTTBL	Table field indicator.
NTVARD	Derivative variable field indicator.
NTVARO	Independent variable field indicator.
NTVAR1	Primary variable field indicator.
NTVAR2	Secondary variable field indicator.

5. I/O VARIABLES:

A. INPUT	LOCATION
CCONS	/I02/
IERR	F.P.
IERR1	F.P.
ISUB	F.P.
LLXWRK	/I03/
MMTABU	/I04/
NNCNS	/I03/
NNTABU	/I01/
NNWRK	/I03/
NTNUL	F.P.
NTTBL	F.P.
NTVARD	F.P.
NTVAR0	F.P.
NTVAR1	F.P.
NTVAR2	F.P.
B. OUTPUT	LOCATION
CCONS	/I02/
IERR1	F.P.
LLTABU	/I03/
NNCNS	/I03/
NNTABU	/I01/

6. CALLING ROUTINE: PROCES

7. CALLED ROUTINES: ARAYCK
EEROR

1. NAME: TOFSTR - 66
2. PURPOSE: This routine processes the TRUE and FALSE commands and stores the necessary information to execute the command during an IOSYM simulation.
3. METHOD: If a TRUE command is being processed, the IF pointer to the TRUE command is set to point to the next available location in the NSET array where the commands under the TRUE command will be stored. Control then returns to the calling routine.

If a FALSE command is being processed, the IF pointer to the FALSE command is set to point to the next available location in the NSET array where the commands under the FALSE command will be stored. Control then returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
IERR	Error indicator. If FALSE, then an error hasn't occurred in this routine. If TRUE, then a fatal error has occurred in this routine.
IERR1	Error indicator. If FALSE, then an error hasn't occurred up to the present time. If TRUE, then a fatal error has occurred.
IIFPT	Array that contains the IF command pointers. Word one contains the negative of the LABEL number to show this information is for an IF. Word two contains the location in the NSET array to store the pointer to the TRUE part of the IF command. Word three contains the location in the NSET array to store the pointer to the FALSE part of the IF command. Word four contains the location in the NSET array to store the pointer to the IF LABEL.
IN1	Location in the NSET array where the pointer to where the TRUE or FALSE information starts.
IN2	Indicator. If less than zero, then a FALSE command is being processed. If equal or greater than zero, then a TRUE command is being processed.

IPTCMD	Location in the NSET array where the pointer to the next command is to be stored. If zero, the last command doesn't point to any command.
LLXSET	Last position in the NSET array that was used.
LXSET	Number of NSET elements needed to process the command.
MMSET	Maximum number of elements in the NSET array (dimension of the NSET array, set in routine USRDAT).
NFPT	Current IF level being processed.
NNFPT	Number of stacked IF and LOOP commands.
NSET	Storage array where the link listed NETWORK is stored.

5. I/O VARIABLES:

A. INPUT	LOCATION
IERR	F.P.
IERR1	F.P.
IIFPT	/I02/
IN2	F.P.
LLXSET	/I03/
LXSET	F.P.
MMSET	/I04/
NNFPT	/I03/
B. OUTPUT	LOCATION
IERR1	F.P.
IIFPT	/I02/
IPTCMD	F.P.

LXSET

F.P.

NSET

/ /

6. CALLING ROUTINE: PROCES

7. CALLED ROUTINES: ARAYCK
EEROR

1. NAME: TRCSTR - 91
2. PURPOSE: This routine processes the TRACE and NOTRACE commands and stores the appropriate information needed to perform TRACE during an IOSYM simulation.
3. METHOD: If a TRACE command is being processed, a check is done to see if TRACE information has already been stored; if not, five words in the NSET array are set aside to store the TRACE information. The pointer LLTRAC is set to point to the first word of TRACE information. The locations used to store the TRACE information are initialized to null fields. The TRACE variables, if any, are identified and stored in the NSET array. Control then returns to the calling routine.

If a NOTRACE command is being processed, a check is done to see if there are any variables specified on the NOTRACE command. If there aren't, then the TRACE pointer is set to a negative number to indicate not to do TRACE. If there are variables to delete, then the variables on the NOTRACE command are deleted from the TRACE information. Control then returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Index and counter used for various purposes.
IERR	Error indicator. If FALSE, then an error in this routine hasn't occurred. If TRUE, then a fatal error has occurred in this routine.
IERR1	Error indicator. If FALSE, then an error hasn't occurred up to the present time. If TRUE, then a fatal error has occurred.
IIR	Do loop counter.
IIT	Position in the NSET array where the last TRACE variable is stored.
J	Do loop index and counter.
K	Current variable number being processed.
LLT	Pointer to location in the NSET array where the variable currently being processed is located.

LLTRAC

Location in the NSET array where the TRACE information is located.

0 - then the TRACE information is inactive
=0 - then there is no TRACE information
0 - then the TRACE is active

LLXSET

Last position in the NSET array used.

LXSET

Number of NSET elements needed to process the TRACE command.

MMSET

Maximum number of elements in the NSET array (dimension of the NSET array, set in routine USRDAT).

NNWRK

Work storage array where the current command parsed into the various fields is located.

NSET

Storage array where the link listed NETWORK is stored.

5. I/O VARIABLES:

A. INPUT

LOCATION

IERR

F.P.

IERR1

F.P.

LLTRAC

/I03/

LLXSET

/I03/

LXSET

F.P.

MMSET

/I04/

NNWRK

/I03/

NSET

/ /

B. OUTPUT

LOCATION

IERR1

F.P.

LLTRAC

/I03/

LLXSET

/I03/

LXSET

F.P.

NSET

/ /

6. CALLING ROUTINE: PROCES

7. CALLED ROUTINES: ARAYCK
EEROR

1. NAME: TRMSTR - 32
2. PURPOSE: This routine processes the TERM and CROSS commands and stores the appropriate information to use during an IOSYM simulation.
3. METHOD: This routine processes the TERM or CROSS commands on an ARC. If a PICK command hasn't been processed on the current ARC, routine PCKCK is called to do some initialization. If an OPR command hasn't been processed on the current ARC, then some initialization is done. If this is the first TERM or CROSS command on the current ARC, a location is set aside in the NSET array to store the number of TERMS and CROSSES processed. It is initially set to zero. A location in the NSET array is set aside to point to the expression on the TERM or CROSS. The number of TERMS or CROSSES on the ARC is incremented by one and stored in the NSET array. The TERM or CROSS command number is stored in the NSET array. If this isn't the first TERM or CROSS on this ARC, point the last TERM or CROSS to this one. The TERM or CROSS number is stored in the NSET array.

If the command is a CROSS command, then the CROSS tolerance is stored in the NSET array. The truth condition indicator is identified and stored in the NSET array. Then for both commands a location in the NSET array is set aside to point to the other commands under the TERM or CROSS command. A location is also set aside to point to the next TERM or CROSS, and a pointer to the next location is stored as the location of the expression on the TERM or CROSS. Routine FUNSTR is called to process the expression and store it in the NSET array. Control then returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
AN2	CROSS command tolerance.
CCONS	Array containing all the constants in a NETWORK.
IERR	Error indicator. If FALSE, then an error hasn't occurred in this routine. If TRUE, then a fatal error has occurred in this routine.
IERR1	Error indicator. If FALSE, then an error hasn't occurred up to the present time. If TRUE, then a fatal error has occurred.

IFIR	Indicator. If FALSE, then an OPR command might not have been processed. If TRUE, then an OPR command has been processed on the current ARC.
INDX4	Location in the NSET array where the pointer to the OPR command on an ARC is to be stored.
INDX5	Location in the NSET array where the pointer to the first TERM or CROSS command is to be stored.
INDX9	Location in the NSET array where the number of TERMS and CROSSES on the current ARC is to be stored.
IND10	Location in the NNWRK array where the first element to process in the expression is located.
INX	Location in the NSET array where the pointer to the expression is to be stored.
IN1	Pointer to the location in the CCONS array where the CROSS command tolerance is located.
IN2	TERM or CROSS number.
IN8	Location in the CCONS array where the TERM or CROSS number is located.
IN9	Number of constants to remove from the CCONS array because they are no longer needed.
IPICK	Location of the pointer in a PICK command for an ARC, leaving a NODE, to the next ARC leaving the same NODE.
IPTCMD	Location in the NSET array where the pointer to the next command is to be stored.
IPTCM2	Location in the NSET array where the pointer to the next TERM or CROSS is to be stored.

ISNODE	Current start NODE.
LLXSET	Location in the NSET array of the last position used.
LSET	Number of elements in the NSET array needed to process a command.
MMSET	Maximum number of elements in the NSET array (dimension of the NSET array, set in USRDAT).
NDX1	Current command number.
NNWRK	Work storage array that contains the current command parsed into the various fields.
NSET	Storage array where the link listed NETWORK is stored.
OPR	Indicator. If FALSE, an OPR command on the current ARC hasn't been processed yet. If TRUE, an OPR command has been processed.
PICK	Indicator. If FALSE, a PICK command on the current ARC hasn't been processed yet. If TRUE, PICK command has been processed.
QSET	Array equivalenced to the NSET array used to store the real value of the CROSS tolerance.
TERM	Indicator. If FALSE, a TERM or CROSS command on the current ARC hasn't been processed yet. If TRUE, at least one TERM or CROSS on the current ARC has been processed.

5. I/O VARIABLES:

A. INPUT	LOCATION
CCONS	/I02/
ID	F.P.
IERR	F.P.

IERR1	F.P.
IFIR	F.P.
INDX4	F.P.
INDX5	F.P.
IPICK	F.P.
ISNODE	F.P.
LLXSET	/I03/
LSET	F.P.
MMSET	/I04/
NDX1	F.P.
NNWRK	/I03/
NSET	/ /
OPR	F.P.
PICK	F.P.
TERM	F.P.
B. OUTPUT	LOCATION
CCONS	/I03/
IERR1	F.P.
IPTCMD	F.P.
LLXSET	/I03/
LSET	F.P.
NNWRK	/I03/
NSET	/ /
QSET	/ /
TERM	F.P.

6. CALLING ROUTINE: PROCES

7. CALLED ROUTINES: ARAYCK
EEROR
FUNSTR
PCKCK
RESCC

1. NAME: TRUTH - 20
2. PURPOSE: This routine determines which truth condition was processed and assigns the correct code number.
3. METHOD: A check is done to see what the alpha field being processed is. If it is GE, then the code number 5 is returned to the calling routine. If it is a GT, then 4 is returned. If EQ, then 3; if LT, then 2, and if LE, then 1 is returned. If the alpha string is not a GE, GT, EQ, LT, or LE, then an error is processed, and a zero code number is returned.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
IERR1	Error indicator. If FALSE, then an error hasn't occurred up to the present time. If TRUE, then a fatal error has occurred.
ILIT	When entering this routine, ILIT is the internal IOSYM representation of the alpha string that is the truth condition. When leaving this routine, ILIT is the code number corresponding to the correct truth condition.

5. I/O VARIABLES:

A. INPUT	LOCATION
ILIT	F.P.
B. OUTPUT	LOCATION
IERR1	F.P.
ILIT	F.P.

6. CALLING ROUTINES: ALPSRH
FUNC5

7. CALLED ROUTINE: EEROR

1. NAME: TTABU - 536
2. PURPOSE: This routine collects the TABULATE statistics.
3. METHOD: If statistics aren't being processed, there are no active TABULATE commands, or there are no OPERATIONS. Control returns to the calling routine. If collecting statistics at the beginning of an OPERATION, the type S and C variables are identified and their current values processed and stored in the work storage array. Control then returns to the calling routine.

If collecting statistics at the end of an OPERATION, then all three types must be collected: type S, E, and C. For type S, the start value of the variable is passed to routine WWRT along with other information to write the information on the TABULATE tape. For type E, the end value is identified and passed to routine WWRT. For type C, the initial value is subtracted from the current value and passed to routine WWRT. This is done for each variable on each active TABULATE. Control then returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Index to determine which TABULATE command is being processed.
IDV	Variable identifier of the TABULATE variable being processed.
IFUNC	Indicator determines where the TTABU function is being processed. <ul style="list-style-type: none"> 1 = TTABU is being processed at the beginning of an OPERATION for type S and C commands. 2 = TTABU is being processed at the end of an OPERATION to store type S, C, or I information.
IIARC	ARC number for current ARC being processed.
IIOP	OPERATION for current OPERATION being processed.

IISTAT	Indicator. If IISTAT is zero, then IOSYM statistics aren't being processed. If IISTAT is equal to one, they are.
INUM	Current TABULATE number. If greater than zero, then the time option is off; if less than zero, then the time option is on.
IOPAR	Array that contains in the first word the current OPERATION number, and the second word is the current ARC number.
ITOP	Indicator. If equal to one, then the time option for the current TABULATE command is off. If equal to two, then the time option is on.
ITYP	Type of report type of the current variable being processed, either S, E, or C.
J	Index to determine which variable on the current TABULATE command is being processed.
LLT	Location in the work array where the current variable's value is stored.
LLTABU	Pointer to the work array where the storage for the TABULATE statistic information starts.
LN	Location in the TAB array to store the TABULATE value to write on the TABULATE tape with the S, E, or C information.
MMTABU	Maximum number of TABULATE commands that can be in a NETWORK (the dimension of the NNTABU array, set in routine USRDAT).
NNTABU	Array that contains in the first word the indicator to show whether this TABULATE is active or not and if the time option is on or not. The second, fourth, sixth, and eighth words are the TABULATE variable identifiers. The third, fifth, seventh, and ninth words indicate what type of TABULATE variables are processed, respectively.

NNWRK

The WWRK array is equivalenced to this array and is used to store intermediate TABULATE results.

T

Array that contains the OPERATION start time in the first word and the OPERATION end time in the second.

TAB

This array contains the current report values for the given TABULATE command to write on the TABULATE tape.

TNOW

Current simulation time.

WWRK

This array is equivalenced to the NNWRK array and is used to store temporary TABULATE values.

5. I/O VARIABLES:

A. INPUT

LOCATION

IFUNC

F.P.

IIARC

/I03/

IIOP

/I03/

IISTAT

/I03/

LLTABU

/I03/

MMTABU

/I04/

NNTABU

/I01/

NNWRK

/I03/

TNOW

/GCOM1/

WWRK

/I03/

B. OUTPUT

LOCATION

INUM

F.P.

IOPAR

F.P.

LN

F.P.

NNWRK

/I03/

T

F.P.

TAB

F.P.

WWRK

/I03/

6. CALLING ROUTINES: AARCST
EVNTS
7. CALLED ROUTINES: VVARBL
WWRIT

1. NAME: TTAPES - 552
2. PURPOSE: This routine performs several functions on auxiliary files. It writes all GASP storage to tape, reads all GASP storage from tape, and rewinds the TABULATE and PLOT tapes.
3. METHOD: If it is to write all the GASP storage to tape, the correct tape number is identified and all the GASP information is written out.

If it is to read all the GASP storage from tape, the current tape number is identified and all the GASP information is read in.

If it is to rewind the TABULATE and PLOT tapes, the tape numbers for the tapes are identified and rewound.

After each of these, control is returned to the calling routine. Only the variables actually used in this routine are discussed below. Those variables just read in or written out are not discussed.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
IITPS	This array contains the tape device code numbers for the various tapes. IITPS(2) is the GASP information tape, IITPS(3) is the PLOT tape, IITPS(4) is the TABULATE tape.
ITP	Device code number currently processing.
IZ	Indicates what type of processing to do: <ul style="list-style-type: none"> 1 = Write GASP storage to tape. 2 = Read GASP storage from tape. 3 = Rewind TABULATE and PLOT tapes.

5. I/O VARIABLES:

A. INPUT	LOCATION
IITPS	/I01/
IZ	F.P.

B. OUTPUT

LOCATION

(None)

6. CALLING ROUTINES: DDOSIM
ENDNET
INTLC
RSAVIN
WSAVIN

7. CALLED ROUTINES: (None)

1. NAME: TTRAC - 92
2. PURPOSE: This routine executes a TRACE command during an IOSYM simulation.
3. METHOD: The TRACE variable values, current time, current ARC, and OPERATION or NODE are identified, and routine PPPRNT is called to print the TRACE information.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Index that counts the number of variables processed.
II	Index that points to the location in the NSET array where the variable identifier is located.
IIARC	ARC number of the ARC being processed. If IIARC is less than one, then a NODE is being processed.
IINXND	NODE number of the NODE being processed, or next to be processed.
IIOP	OPERATION number of the current OPERATION being processed.
IOPNO	This array contains in the first word the current OPERATION number, and the second word contains the current ARC number or the negative of the current NODE number.
LLTRAC	Location in the NSET array where the TRACE command information is located.
NSET	Storage array where the link listed NETWORK is stored.
NV	Number of TRACE variables processed.
T	Array that contains in the first word the location in the NSET array where the TRACE variable identifiers are located. The second word contains the current time.

TNOW

The current simulation time.

VAL

Array that contains the current values of the TRACE variables.

5. I/O VARIABLES:

A. INPUT

LOCATION

IIARC

/I03/

IINXND

/I03/

IIOP

/I03/

LLTRAC

/I03/

NSET

/ /

TNOW

/GCOM1/

B. OUTPUT

LOCATION

IOPNO

F.P.

NV

F.P.

T

F.P.

VAL

F.P.

6. CALLING ROUTINES: AARCST
NNODST

7. CALLED ROUTINES: PPPRNT
VVARBL

1. NAME: TYPSTR - 64
2. PURPOSE: This routine processes the TYPE commands.
3. METHOD: If the TYPE command occurs on an ARC, a check is done to make sure PICK and OPR commands are processed. If processing the first type command on the ARC, then need to setup pointers to all the TYPE commands before processing the TYPE command. The TYPE pointer is set to point to the next location in the NSET array where the first command on the TYPE will be located. Control then returns to the calling routine.

If the TYPE command occurs on a GLOBAL command, then the type of TYPE command is identified, and the appropriate location in the NNPTGB array is set to point to the next location in the NSET array where the TYPE commands will start. Control then returns to the calling routine.

If the TYPE command occurs on an OPERATION command, processing proceeds just like in GLOBAL except the NNPTOP array is used instead of the NNPTGB array.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
ICLEVL	This array contains the current command hierarchy from level one to level LEVEL.
ID	Dimension of the IPICK array.
IERR	Error indicator. If FALSE, then an error hasn't occurred in this routine. If TRUE, then a fatal error has occurred in this routine.
IERR1	Error indicator. If FALSE, then an error hasn't occurred up to the present time. If TRUE, then a fatal error has occurred.
IFIR	Indicator. If TRUE, then an OPR or a TYPE command has already been processed on the current ARC. If FALSE, an OPR nor a TYPE command have been processed.
IIOP	OPERATION number of the current OPERATION.

NNPTGB

Array that contains in the first two words the internal IOSYM representation of the word GLOBAL. The third, fourth, and fifth words point to where the TYPE S, I, and E commands start in the NSET array, respectively.

NNPTOP

Array that contains in the first two words the internal IOSYM representation of the OPERATION names. The third, fourth, and fifth words point to where the TYPE S, I, and E commands start in the NSET array, respectively. The sixth word indicates whether the OPERATION is a COLLECT OPERATION or not.

NNWRK

Work storage array. Contains the current command parsed into the various fields.

NSET

Storage array where the link listed NETOWRK is stored.

OPR

Indicator. If TRUE, then the OPR information for the current ARC has been processed. If FALSE, it hasn't.

PICK

Indicator. If TRUE, then the PICK information for the current ARC has been processed. If FALSE, it hasn't.

5. I/O VARIABLES:

A. INPUT	LOCATION
ICLEVL	/ME1/
ID	F.P.
IERR	F.P.
IFIR	F.P.
IIOP	/I03/
INDX4	F.P.
IPICK	F.P.
ISNODE	F.P.

ITYP1	F.P.
LEVEL	/ME1/
LLXSET	/I03/
LSET	F.P.
MMSET	/I04/
NNPTGB	/I02/
NNPTOP	/I02/
NNWRK	/I03/
NSET	/ /
OPR	F.P.
PICK	F.P.
B. OUTPUT	LOCATION
IERR1	F.P.
IFIR	F.P.
IPTCMD	F.P.
ITYP1	F.P.
LLXSET	/I03/
LSET	F.P.
NNPTGB	/I02/
NNPTOP	/I02/
NSET	/ /

6. CALLING ROUTINE: PROCES

7. CALLED ROUTINES: ARAYCK
EEROR
PCKCK

1. NAME: U
2. PURPOSE: This is a user written routine to process any IOSYM user functions. Currently this is a dummy routine that does no processing. The value of U returned is zero. This routine should be replaced by the user written U routine if required.

3. METHOD: Not applicable.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
IAC	Array containing the entities in a user function.
IFF	User function number.
IPAR	Array containing the parameter sets specified in a user function.
ITAB	Array containing the TABLEs specified in a user function.
NA	Number of accessible entities in user function.
NP	Number of parameter sets in user function.
NT	Number of TABLEs in user function.
NX	Number of arguments in user function.
U	FUNCTION name contains the value of the function returned to the calling routine.
X	Array containing the arguments specified in a user function.

5. I/O VARIABLES:

A. INPUT	LOCATION
IAC	F.P.
IFF	F.P.
IPAR	F.P.

ITAB

F.P.

NA

F.P.

NP

F.P.

NT

F.P.

NX

F.P.

X

F.P.

B. OUTPUT

LOCATION

U

FUNCTION

6. CALLING ROUTINE: EEVAL

7. CALLED ROUTINES: (None)

1. NAME: UEEROR
2. PURPOSE: This is a user-written routine to process any user errors. Currently this is a dummy routine that does no processing. This routine should be replaced by the user-written UERROR routine if required.
3. METHOD: Not Applicable
4. INTERNAL VARIABLES

VARIABLE	DEFINITION
I	Dummy variable
IERO	Error number to process.
5. I/O VARIABLES:

A. INPUT	LOCATION
IERO	F.P.
B. OUTPUT	LOCATION
(None)	
6. CALLING ROUTINE: EEROR
7. CALLED ROUTINE: (None)

1. NAME: UEVENT
2. PURPOSE: This is a user-written routine to process any user EVENT commands codes. Currently this is a dummy routine that does no processing. This routine should be replaced by the user-written UEVENT routine if required.

3. METHOD: Not Applicable

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Dummy variable.
IV	EVENT code number to process.
NP	Number of parameters in EVENT to process.
PAR	Array that contains the EVENT parameters.

5. I/O VARIABLES:

A. INPUT	LOCATION
IV	F.P.
NP	F.P.
PAR	F.P.
B. OUTPUT	LOCATION
(None)	

6. CALLING ROUTINE: DDOEVT

7. CALLED ROUTINE: (None)

1. NAME: UEVNTS
2. PURPOSE: This is a user written routine to process any user defined time events. Currently this is a dummy routine that does no processing. This routine should be replaced by the user written UEVNTS routine if required.
3. METHOD: Not applicable
4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
IEND	Indicator. If equal to zero, then the OPERATION is to continue. If any other value, the OPERATION is ended.
IFLG	Number of state events.
IX	Event code number
5. I/O VARIABLES:

A. INPUT	LOCATION
IFLG	F.P.
IX	F.P.
B. OUTPUT	LOCATION
IEND	F.P.
6. CALLING ROUTINE: EVNTS
7. CALLED ROUTINE: (None)

1. NAME: UINTLC
2. PURPOSE: This is a user-written routine to process any user initialization. Currently this is a dummy routine that does no processing. This routine should be replaced by the user-written UINTLC routine if required.
3. METHOD: Not Applicable
4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Dummy variable.
5. I/O VARIABLES:

A. INPUT	LOCATION
(None)	
B. OUTPUT	LOCATION
(None)	
6. CALLING ROUTINE: INTLC
7. CALLED ROUTINE: (None)

1. NAME: UNCODE - 63
2. PURPOSE: This routine deciphers an IOSYM identifier or the internal representation of a name into an array with the characters of the name as one character per word.
3. METHOD: It is first determined whether an internal representation of a name is to be decoded or an IOSYM identifier (IIDV) is to be decoded. If an IIDV is to be decoded into its name, the type of variable is first identified, either independent variable, secondary variable, primary variable, TABLE, derivative variable, substitutable parameter, constant, or operator. For the variables, the variable number is also identified and with that information the internal IOSYM representation of the name is identified and is processed like the name was originally set to be processed.

If the internal representation of a name is to be deciphered, each character is processed and decoded one at a time and stored in the IOUT array. Once all the characters have been processed, control returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Index and counter used for various purposes.
ICHAR	Array containing all the IOSYM recognized characters.
ICODE	Array containing in the first word the first four characters of the name in internal IOSYM representation. The second word contains the next three characters and a blank.
ICODE1	Array containing in the first word the first four characters of the name in internal IOSYM representation. The second word contains the next three characters and blank.
IDEM	Indicates which power of ten is being processed.
IDV	Temporary storage used for various purposes.
IDV1	Current digit in a number being processed.

IERR1	Error indicator returned from routine EEROR but not used in this routine.
II	TABLE number of the TABLE being processed.
IIDV	Variable identifier of the variable being processed.
III	Index counter.
IIOPR	Array that contains the substitutable parameters.
INT1	Indicator to show whether the first digit in a number has been processed or not. 0 = First digit hasn't been processed. 1 = First digit has been processed.
IN1	IOSYM code number of the current character being processed.
IN2	Internal representation of the characters left to decode in a name.
IOUT	Array that contains the decoded name as one character per word.
IRET	Indicator used for various purposes.
J	Index to show which of the four characters in a name is being processed.
LIBFUN	Array that contains the FORTRAN library function names.
MMFUNF	Number of menu functions.
MODE	Indicator. If equal to two, then an internal name representation is to be decoded. If equal to one, then a variable identifier is to be decoded.
NNPTB	Array that contains in the first two words the TABLE names and the third word points to the location in the TTABL array where the TABLE values are stored.

NNPT0	Array containing the independent variable name in internal IOSYM representation.
NNPT1	Array containing the primary variable names in internal IOSYM representation.
NNPT2	Array containing the secondary variable names in internal IOSYM representation.
NNXTB	Number of defined TABLES in a NETWORK.
NUM	Variable index value at TABLE.
NWT	Number of decoded characters in the name.
NXTB	Number of defined TABLES plus one.
VVAR2	Array containing the initial and current values of the secondary variables.

5. I/O VARIABLES:

A. INPUT	LOCATION
ICHR	/ME1/
ICODE1	F.P.
IERR1	F.P.
IIOPR	/I02/
LIBFUN	/ME1/
MMFUNF	/I04/
MODE	F.P.
NNPTB	/I02/
NNPT0	/I02/
NNPT1	/I02/
NNPT2	/I02/

NNXTB	/I03/
VVAR2	/I02/
B. OUTPUT	LOCATION
IOUT	F.P.
NWT	F.P.

6. CALLING ROUTINES:

- NETSUM
- PPRNT
- PPRCOL
- PPRPLT
- PPRRPT
- PPRTAB
- UNCO1
- UPLOT (user written)
- WSAVIN
- WTHMN

7. CALLED ROUTINE: EEROR

1. NAME: UNCO1 - 77
2. PURPOSE: This routine is a specialized call to the UNCODE routine to process variable names.
3. METHOD: Routine UNCODE is called to decipher the variable name and is blank filled to NWT number of characters. The resultant array is returned to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
IARRAY	Array containing the variable name in internal IOSYM representation.
ICHAR	Array containing all the characters recognized by IOSYM.
IRAY	Array containing the deciphered characters in a variable name, blank filled, one character per word.
IRAY1	Array containing the deciphered characters in a variable name, blank filled, one character per word.
IWT	Number of blanks needed to fill name.
J	Number of characters in the IRAY array.
K	Index and counter.
NUM	Number of characters a name is supposed to have.
NWT	Number of characters in a variable name.

5. I/O VARIABLES:

A. INPUT	LOCATION
IARRAY	F.P.
ICHAR	/ME1/
IRAY1	F.P.

NUM	F.P.
NWT	F.P.
B. OUTPUT	LOCATION
IRAY	F.P.

6. CALLING ROUTINES: DDOSIM
NETSUM
PPRCOL
PPRPLT
PPRRPT
PPRTAB

7. CALLED ROUTINE: UNCODE

1. NAME: UOTPUT
2. PURPOSE: This is a user-written routine to process any user output. Currently this is a dummy routine that does no processing. This routine should be replaced by the user-written UOTPUT routine if required.
3. METHOD: Not Applicable
4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Dummy variable.
5. I/O VARIABLES:

A. INPUT	LOCATION
(None)	
B. OUTPUT	LOCATION
(None)	
6. CALLING ROUTINE: OTPUT
7. CALLED ROUTINE: (None)

1. NAME: UPL0T
2. PURPOSE: This is a user-written routine to process any user type PLOTs. This routine should be replaced by the user-written UPL0T routine if required.
3. METHOD: Not Applicable.
4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Current PLOT number.
ITP	File number that the PLOT information is stored on.
NNVAR	Array containing the IOSYM variable identifiers for the independent and dependent variables for the current PLOT.
NUMPLT	Number of dependent variables on the current PLOT.
5. I/O VARIABLES:

A.	INPUT	LOCATION
	I	F.P.
	ITP	F.P.
	NNVAR	F.P.
	NUMPLT	F.P.
B.	OUTPUT	LOCATION
	(None)	
6. CALLING ROUTINE: PPRPLT
7. CALLED ROUTINE: (None)

1. NAME: UREAD
2. PURPOSE: This is a user-written routine to process any user input reading needed. Currently this is a dummy routine that does no processing. This routine should be replaced by the user-written UREAD routine if needed.
3. METHOD: Not Applicable
4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Dummy variable.
5. I/O VARIABLES:

A. INPUT	LOCATION
(None)	
B. OUTPUT	LOCATION
(None)	
6. CALLING ROUTINE: DDOSIM
7. CALLED ROUTINE: (None)

1. NAME: USCOND
2. PURPOSE: This is a user-written routine to process any user defined state conditions. Currently this is a dummy routine that does no processing. This routine should be replaced by the user-written USCOND routine if needed.
3. METHOD: Not Applicable
4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Dummy variable.
5. I/O VARIABLES:

A. INPUT	LOCATION
(None)	
B. OUTPUT	LOCATION
(None)	
6. CALLING ROUTINE: SCND
7. CALLED ROUTINE: (None)

1. NAME: USRDAT - 83
2. PURPOSE: This is the routine where all the machine specific variables are set and where the variables that indicate the size of various arrays are set by the user.
3. METHOD: Not applicable.
4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
MAXEXP	Maximum size of the exponent that a number can have on this computer.
MAXNUM	Maximum number of digits that a number can have on this computer.
MMARC	Dimension of the NNPTAC array. Maximum number of ARCs a NETWORK can have.
MMARY	Dimension of the LLARY array. Maximum number of LABELs one section of a NETWORK can have.
MMCNS	Dimension of the CCONS array. Maximum number of constants a NETWORK can have.
MMCOLC	Dimension of the NNCOLC array. Maximum number of COLLECT variables a NETWORK can have.
MMFPT	Dimension of the IIFPT array. Maximum number of stacked IF and LOOP commands a NETWORK can have.
MMLINE	Maximum number of lines that will fit on a computer page.
MMNODE	Dimension of the NNPTND array. Maximum number of NODEs a NETWORK can have.
MMOP	Dimension of the NNPTOP array. Maximum number of OPERATIONS a NETWORK can have.

MMOPR	Dimension of the IIOPR array. Twice the number of substitutable parameters a NETWORK can have.
MMPLLOT	Dimension of the NNPLLOT array. Maximum number of PLOTs a NETWORK can have.
MMRPT	Dimension of the IIRPT array. Maximum number of REPORT variables allowed in a NETWORK.
MMSET	Dimension of the NSET array. Maximum number of elements that can be used to store the link listed NETWORK.
MMTAB	Dimension of the TTABL array. Maximum number of TABLE elements a NETWORK can have.
MMTABU	Dimension of the NNTABU array. Maximum number of TABULATEs that a NETWORK can have.
MMV2	Dimension of the NNPT2 and VVAR2 arrays. Maximum number of secondary variables a NETWORK can have.
MMWRK	Dimension of the NNWRK array.
MMXTB	Dimension of the NNPTB array. Maximum number of TABLEs a NETWORK can have.
NCRDR	This is the device code number where input is read from.
NPRNT	This is the device code number where output is put on.
PPREC	This is the computer precision of the machine.

5. I/O VARIABLES:

A. INPUT	LOCATION
(None)	

B.	OUTPUT	LOCATION
	MAXEXP	/MCHDEP/
	MAXNUM	/MCHDEP/
	MMARC	/I04/
	MMARY	/I04/
	MMCNS	/I04/
	MMCOLC	/I04/
	MMFPT	/I04/
	MMLINE	/I04/
	MMNODE	/I04/
	MMOP	/I04/
	MMOPR	/I03/
	MMPL0T	/I04/
	MMRPT	/I04/
	MMSET	/I04/
	MMTAB	/I04/
	MMTABU	/I04/
	MMV2	/I04/
	MMWRK	/I04/
	MMXTB	/I04/
	NCRDR	/GCOM1/
	NPRNT	/GCOM1/
	PPREC	/I04/

6. CALLING ROUTINE: IOSYM

7. CALLED ROUTINE: None.

1. NAME: USSAVE
2. PURPOSE: This is a user-written routine to process any user plot save information. Currently this is a dummy routine that does no processing. This routine should be replaced by the user-written USSAVE routine if required.
3. METHOD: Not Applicable
4. INTERNAL VARIABLES

VARIABLE	DEFINITION
I	Dummy variable.
5. I/O VARIABLES:

A. INPUT	LOCATION
(None)	
B. OUTPUT	LOCATION
(None)	
6. CALLING ROUTINE: SSAVE
7. CALLED ROUTINE: None

1. NAME: USTAT
2. PURPOSE: This is a user-written routine to process any user defined statistics. Currently this is a dummy routine that does no processing. This routine should be replaced by the user-written USTAT routine.
3. METHOD: Not Applicable
4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Dummy variable.
IS	Statistics time number.
5. I/O VARIABLES:

A. INPUT	LOCATION
IS	F.P.
B. OUTPUT	LOCATION
(None)	
6. CALLING ROUTINES: AARCST
EVNTS
OTPUT
7. CALLED ROUTINE: (None)

1. NAME: USTATE
2. PURPOSE: This is a user written routine to process any user state definitions. Currently this is a dummy routine that does no processing. This routine should be replaced by the user written USTATE routine if required.
3. METHOD: Not applicable.
4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Dummy variable
5. I/O VARIABLES:

A. INPUT	LOCATION
(None)	
B. OUTPUT	LOCATION
(None)	
6. CALLING ROUTINE: STATE
7. CALLED ROUTINE: None

1. NAME: UTERM
2. PURPOSE: This is a user written routine to process any user TERM events. Currently this is a dummy routine that does no processing. This routine should be replaced by the user written UTERM routine if required.

3. METHOD: Not applicable

4. INTERNAL VARIABLES

VARIABLE

DEFINITION

I

Dummy variable.

IEND

OPERATION end indicator. If equal to zero, then the OPERATION is not ended; if any other value, the OPERATION is ended.

5. I/O VARIABLES:

A. INPUT

LOCATION

IEND

F.P.

B. OUTPUT

LOCATION

(None)

6. CALLING ROUTINE: EVNTS

7. CALLED ROUTINE: (None)

1. NAME: UUPRNT - 93
2. PURPOSE: This routine is for use by user-written routines to perform a PRINT command.
3. METHOD: The current time and the current value of the PRINT variables are identified. The current OPERATION and ARC or NODE is identified and stored in the IOPNO array. If the PRINT command is to be written to the TABULATE tape, routine WWRT is called. If PRINT is to be done immediately, routine PPPRNT is called to PRINT the information.
4. INTERNAL VARIABLES:

VARIABLES

DEFINITION

I	Pointer to the current variable being processed.
IIARC	ARC number of the current ARC being processed. If IIARC is less than zero, then an ARC isn't being processed. A NODE is.
IINXND	Current NODE number of the NODE being processed.
IIOP	OPERATION number of the OPERATION currently being processed.
IOPNO	If processing an ARC command, the first word contains the current OPERATION number and the second word contains the ARC number. If processing a NODE, the first word is not used, and the second word is the negative of the NODE number.
ITAB	Indicator. If equal to zero, then PRINT is to be done immediately. If not equal to zero, then the PRINT information is written on the TABULATE tape.
IVAL	Array that contains the IOSYM variable identifiers of the variables to PRINT.
NV	Number of variables to process.

T Array that has a pointer to the variable identifier in the first word and the current time in the second word.

TNOW Current simulation time.

VAL Array that contains the current values of the PRINT variables.

5. I/O VARIABLES:

A. INPUT LOCATION

IIARC /I03/

IINXND /I03/

IIOP /I03/

ITAB F.P.

IVAL F.P.

NV F.P.

TNOW /GCOM1/

B. OUTPUT LOCATION

VAL F.P.

6. CALLING ROUTINES: (None. Not used at this time. Called from user-written routines.)

7. CALLED ROUTINES: PPPRNT
VVARBL
WWRIT

1. NAME: VADDJ
2. PURPOSE: This routine computes $Y = A1 * X1 + A2 * X2$ for all of Y, X1, and X2.
3. METHOD: The X1 and X2 arrays and the A1 and A2 variables are passed to this routine and each array element in X1 is multiplied by A1 and added to the corresponding array element in X2 multiplied by A2 and stored in Y.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
AA1	Constant to multiply the X1 array by.
AA2	Constant to multiply the X2 array by.
A1	Constant to multiply the X1 array by.
A2	Constant to multiply the X2 array by.
I	Do loop index.
N, NI	Number of array elements to compute.
X1	Array to multiply by the constant A1.
X2	Array to multiply by the constant A2.
Y	Array containing the results of this routine.

5. I/O VARIABLES:

A. INPUT	LOCATION
A1	F. P.
A2	F. P.
N	F. P.
X1	F. P.
X2	F. P.

B OUTPUT

LOCATION

Y

F. P.

6. CALLING ROUTINE: (None. Not used at this time)
7. CALLED ROUTINE: (None)

1. NAME: VARFND - 49
2. PURPOSE: This routine processes independent, primary and secondary variables.
3. METHOD: Routine ALPFND is called to process the variable name. Routine NUMFND is called to find the variable type number. Routine NUMFND is called to process the initial value. If there is no initial value specified, the initial value is set to zero.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
AN2	Initial value of the variable being processed.
CCONS	Array containing the constants in the NETWORK. It is also used as temporary variable storage.
I	Index pointer and counter used for various purposes.
ICLEVL	Array containing the current command heirarchy from level one to level LEVEL.
IERR	Error indicator. If FALSE, then an error hasn't occurred in this routine. If TRUE, then a fatal error has occurred in this routine.
IERR1	Error indicator. If FALSE, then an error hasn't occurred up the present time. If TRUE, then a fatal error has occurred.
IN1	Variable type of the variable to process.
IN2	Pointer used for various purposes.
IN3	Secondary variable number.
IRAY1	First part of variable name in IOSYM notation.
IRAY2	Second part of variable name in IOSYM notation.

IVAR	Variable identifier of the current variable number.
LEVEL	Command level pointer.
LITMAX	Dimension of the LITSTR array.
LITNDX	Position of the last defined location in the literal storage array LITSTR.
LITSTR	Array where the literals are stored after being scanned and prior to being parsed.
LOL	Indicator. If TRUE, then a list of literals is wanted. If FALSE, only one literal is desired.
MMV1	Maximum number of primary variables a NETWORK can have (the dimension of the NNPT1 array, set in routine USRDAT).
MMV2	Maximum number of secondary variables a NETWORK can have (the dimension of the NNPT2 array, set in routine USRDAT).
NCNS	Actual number of permanent constants in the CCONS array.
NDXLIT	Pointer to the current position in the literal storage array LITSTR.
NNCNS	Number of constants in the CCONS array.
NNEQD	Number of derivative variables in a NETWORK.
NNEQT	Number of primary variables in a NETWORK.
NNPTB	Array that contains the TABLE names in the first two words and a pointer to the location in the TTABL array where the TABLE elements are stored.
NNPTO	Array containing the independent variable names.

NNPT1	Array containing the primary variable names.
NNPT2	Array containing the secondary variable names.
NNV1	Number of defined primary variables.
NNV2	Number of defined secondary variables.
NNWRK	Work storage array containing temporary storage.
NNXTB	Number of TABLEs processed.
NOADD	Indication. If FALSE, an alpha field was found. If TRUE, then an alpha field wasn't found.
NTALPC	Alpha field indicator.
NTDLM	Delimiter field indicator.
NTFLP	Floating point number field indicator.
NTINT	Integer number field indicator.
NTNUL	Null field indicator.
NTOPR	Operator field indicator.
NV2	Counter used for various purposes.
SSIN	Array that contains the initial values of the primary variables.
TTIME1	Initial value of the independent variable.
VVAR2	Array containing the current and initial values of the secondary variables.

5. I/O VARIABLES:

A. INPUT	LOCATION
CCONS	/I02/
ICLEVL	/ME1/

IERR	F.P.
LEVEL	/ME1/
LITMAX	F.P.
LITNDX	F.P.
LITSTR	F.P.
LOL	F.P.
MMV1	/I04/
MMV2	/I04/
NDXLIT	F.P.
NNCNS	/I03/
NNEQD	/GCOM2/
NNEQT	/GCOM2/
NNPTB	/I02/
NNPT0	/I02/
NNPT1	/I02/
NNPT2	/I02/
NNV1	/I03/
NNV2	/I03/
NNWRK	/I03/
NNXTB	/I03/
NOADD	F.P.
NTALPC	F.P.
NTDLM	F.P.
NTFLP	F.P.
NTINT	F.P.

	NTNUL	F. P.
	NTOPR	F. P.
B.	OUTPUT	LOCATION
	CCONS	/I02/
	IERR	F. P.
	IERR1	F. P.
	NDXLIT	F. P.
	NNCNS	/I03/
	NNEQD	/GCOM2/
	NNEQT	/GCOM2/
	NNPTB	/I02/
	NNPT0	/I02/
	NNPT1	/I02/
	NNPT2	/I02/
	SSIN	/I02/
	TTIME1	/I02/
	VVAR2	/I02/

6. CALLING ROUTINE: PARSE

7. CALLED ROUTINES: ALPFND
 ARAYCK
 EEROR
 NUMFND

1. NAME: VMOVJ
2. PURPOSE: This routine calculates $XNEW = XMULT * XOLD$, where XNEW and XOLD are arrays, for each XOLD and XNEW element.
3. METHOD: Each XOLD element is multiplied by XMUL and stored in the corresponding location in the XNEW vector.
4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Do loop index.
N, NN	Number of elements to process in the arrays.
XMUL, XMULT	Multiplication factor.
XNEW	Array to store the calculations in.
XOLD	Array to multiply XMULT by.
5. I/O VARIABLES:

A. INPUT	LOCATION
N	F. P.
XMULT	F. P.
XOLD	F. P.
B. OUTPUT	LOCATION
XNEW	F. P.
6. CALLING ROUTINE: (None. Not used at this time)
7. CALLED ROUTINE: (None)

1. NAME: VMULJ
2. PURPOSE: This routine calculates $X = XMULT * X$, where X is an array, for each X element.
3. METHOD: Each X element is multiplied by XMULT and stored in the X element.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	DO loop index.
N, NN	Number of X array elements to process.
X	Array to process.
XMUL, XMULT	Multiplication coefficient.

5. I/O VARIABLES:

A. INPUT	LOCATION
N	F.P.
X	F.P.
XMULT	F.P.
B. OUTPUT	LOCATION
X	F.P.

6. CALLING ROUTINE: (None. Not used at this time)
7. CALLED ROUTINES: (None)

1. NAME: VVARBL - 514
2. PURPOSE: This routine returns the value of the given variable identifier to the calling routine.
3. METHOD: The type of variable is identified either TABLE, independent, primary, secondary, derivative, substitutable parameter or constant. Then depending on whether the current or initial value is desired, the appropriate value is found and returned to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
CCONS	This array contains the constants in a NETWORK.
I	Index and pointer used for various purposes.
ID IDV IDVV	Variable identifiers.
IERR1	Error indicator returned from routine EEROR but not used in this routine.
IIOPR	Array containing the substitutable parameters and their substitutes.
IM	Indicator. If equal to one, VVARBL returns the current value. If equal to two, VVARBL returns the initial value.
ITBL	TABLE number of TABLE being processed.
I12	Indicator. If equal to one, then VVARBL returns the current value of the given variable. If equal to two, the initial value is returned.
LMENT	Value of the variable TABLE index.
NNDUM	Number of substitutable parameters in a NETWORK.
TTABL	Array containing the values of all the TABLE elements.

VVARBL

Function name contains the variable value upon returning to the calling routine.

5. I/O VARIABLES:

A. INPUT	LOCATION
CCONS	/IO2/
IDV	F.P.
IERR1	F.P.
IIOPR	/IO2/
I12	F.P.
NNDUM	/GCOM2/
TTABL	/IO2/
B. OUTPUT	LOCATION
VVARBL	FUNCTION

6. CALLING ROUTINES:

CCOLC
DDOEV
DDPRNT
EEVAL
GGGET
GGSTAT
NETSUM
PPPUT
RREPRT
SSAVE
TTABU
TTRACE
UUPRNT

7. CALLED ROUTINES:

EEROR
GGTAB
GGVAR
VWVTMN

1. NAME: VVMOV - 518
2. PURPOSE: This routine will set each element in X1 equal to the elements in array X2 times a multiplier XMUL.
3. METHOD: Two arrays of length N and a multiplier XMUL are passed to this routine. N elements in the second array are multiplied by XMUL and stored in the first array. Control then returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Index and counter used to determine which array element is being processed.
N	Number of elements in each array.
NN	Number of elements in each array.
XM	Multiplier to multiply the second array elements by.
XMUL	Multiplier to multiply the second array elements by.
X1	First array. This is the array the values are stored in.
X2	This is the second array. These are the values that are multiplied by XM.

5. I/O VARIABLES:

A. INPUT	LOCATION
N	F.P.
XMUL	F.P.
X2	F.P.
B. OUTPUT	LOCATION
X1	F.P.

6. CALLING ROUTINES:

AARCST
INTLC
SSHIFT

7. CALLED ROUTINE: (None)

1. NAME: VVWTMW - 534
2. PURPOSE: This routine checks to see if a variable identifier, is a WITH or MINIMIZE variable. If it is, it returns the variable's initial value, if not; control just returns to the calling routine.
3. METHOD: If there are no WITH variables, control returns to the calling routine. If the number of WITH variables is minus, then there are MINIMIZE variables; they are identified and their initial values are calculated, and control returns to the calling routine. If there are WITH variables, then the initial value is calculated from the upper and lower bounds as specified on a WITH command or as identified by a READ command. Control then returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Indicator to determine if the variable is a READ variable or not. If I equals zero, then the variable is a READ variable.
IDV IDVV	Variable identifier of the variable being investigated.
IISIM	Current simulation number.
L	Index used to point to locations in the NNWRK where processing variable information.
LEND	Location in the NNWRK where the last WITH or MINIMIZE variable is located.
LLWITH	Location in the NNWRK where the first word of WITH or MINIMIZE variable information is located.
NNSIM	Number of simulations to be done.
NNWITH	Number of WITH variables if positive. If negative, it is the negative of the number of MINIMIZE variables.
NNWRK	Work storage array that contains the WITH or MINIMIZE variable information.

VAL

Value of the variable being processed.
If the variable is a WITH or MINIMIZE
variable, then the initial value of
the variable is returned.

WWRK

Work storage array equivalent to the
NNWRK array. Used to retrieve real
values from the NNWRK array.

5. I/O VARIABLES:

A. INPUT	LOCATION
IDV	F.P.
IISIM	/I03/
LLWITH	/I03/
NNSIM	/I03/
NNWITH	/I03/
NNWRK	/I03/
VAL	F.P.
WWRK	/I03/
B. OUTPUT	LOCATION
VAL	F.P.

6. CALLING ROUTINE: VVARBL

7. CALLED ROUTINE: (none)

1. NAME: WSAVIN - 80
2. PURPOSE: This routine writes all of the IOSYM and GASP common blocks to the given tape to use later to RESTORE a NETWORK.
3. METHOD: Routine TTAPES is called to read the GASP information from its tape. All the NETWORK information from the GASP and IOSYM common blocks are written to the tape. Control then returns to the calling routine. Those variables that are just written out are not discussed below.
4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Index used when initializing an array.
ICHAR	Array containing all the IOSYM recognized characters.
IRAY1	Array containing the NETWORK name as one character per word.
ITP	Tape number of the tape to write the information to.
LLXSET	Next usable position in the NSET array.
N	Number of defined locations in the NSET array.
NNMNET	Array containing the NETWORK name in the first two words and an indicator to determine if a NETWORK summary is to be printed in the third word.
NPRNT	Device code number for writing output.
NWT	Number of characters in NETWORK name returned from routine UNCODE but not used in this routine.

5. I/O VARIABLES:

A. INPUT	LOCATION
ICHAR	/ME1/
ITP	F.P.
LLXSET	/I03/

NNMNET

/I03/

NPRNT

/GCOM1/

NWT

F.P.

B. OUTPUT

LOCATION

(none)

6. CALLING ROUTINE: SAVE

7. CALLED ROUTINES: TTAPES
UNCODE

1. NAME: WTHSTR - 95
2. PURPOSE: This routine processes the WITH commands.
3. METHOD: The number of WITH variables is increased by one. The variable identifier is stored in the work storage array. If the next field is a number, it is stored as the variable's lower value. If not a number, then a zero is stored to indicate that the variable is a READ variable. If there are two number fields in a row, then the first is the lower bounds, and the second is the upper bounds. If only one, then the upper and lower bounds are the same. This is continued until all the variables have been processed. Then control returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
CCONS	Array that contains the constants in the NETWORK and storage for temporary constants.
I	Index used to point to the various fields in the NNWRK array.
IERR	Error indicator. If FALSE, an error hasn't occurred in this routine. If TRUE, a fatal error has occurred in this routine.
IERR1	Error indicator. If FALSE, a fatal error hasn't occurred up to the present time. If TRUE, a fatal error has occurred.
IN1	Pointer to the CCONS array where the upper or lower bounds of the variable is stored.
LLWITH	Used as an index to where the fields are to be stored in the work array. At the end of this routine, LLWITH is set to point to the start of the WITH information in the NNWRK array.
LLWRK	Next location in the NNWRK array available.
LLXWRK	Location in the NNWRK where the last WITH field information was stored prior to being parsed.

LWITH	Number of locations in the NMWRK array needed to process the next WITH variable.
MMWRK	Maximum number of fields available in the work storage array (the dimension of the NNWRK array, set in USRDAT).
NNREAD	Number of READ variables processed.
NNWITH	Number of WITH variables processed including READ variables.
NNWRK	Work storage array that contains the current command parsed into the different fields. Also contains the WITH variable information upon leaving the WTHSTR array.
WWRK	Equivalenced to the NNWRK array used to store real values in the NNWRK array.

5. I/O VARIABLES:

A. INPUT	LOCATION
CCONS	/I02/
IERR	F.P.
IERR1	F.P.
LLXWRK	/I03/
MMWRK	/I04/
NNWRK	/I03/
B. OUTPUT	LOCATION
IERR	F.P.
IERR1	F.P.
LLWITH	/I03/
LLWRK	/I03/
NNREAD	/I03/
NNWITH	/I03/

NNWRK

/I03/

WWRK

/I03/

6. CALLING ROUTINE: PROCES

7. CALLED ROUTINES: ARAYCK
EEROR
SSHIFT

1. NAME: WWIT - 537
2. PURPOSE: This routine writes the appropriate information to the PLOT and TABULATE tapes.
3. METHOD: If TABULATE information is to be written to the TABULATE tape, the TABULATE tape number is identified and the passed information written on the tape.

If PLOT information is to be written to the PLOT tape, the PLOT tape number is identified and the passed information written on the tape.

If PRINT information is to be written to the TABULATE tapes, the TABULATE type number is identified and the passed information written on the tape.

After writing the information to the appropriate tape, control returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Index to variable value being written out to tape.
ICODE	Indicator 1=TABULATION information to process. 2=PLOT information to process. 3=PRINT information to process.
IITPS	Array containing the device code number for the different tapes. IITPS (4) is the TABULATE tape device code number. IITPS (3) is the PLOT tape device code number.
INUM	PLOT or TABULATE number.
IOPNO	When processing TABULATE information, the first word is the current OPERATION number and the second word is the current ARC number. If processing PRINT information and if the second word is positive, then the first word is the current OPERATION number and the second word is the current ARC number.

If the second word is negative, then the first word isn't used and the second word is the negative of the current NODE NUMBER. If processing PLOT information, this array isn't used.

IPNT

Type of PRINT to be done.

1=PRINT
2=ERROR
3=TRACE
4=USER

ITP

Device code number to write on.

IVAL

Variable identifiers of the PRINT variables.

NINE

Indicator to show that the information written on the TABULATE tape is a PRINT.

NV

Number of variables processed.

T

Array that contains for PRINT information in the first word the pointer to the location in the IVAL array where the variable identifiers start and the second word is the current time. For TABULATE information, the first word is the start time and the second word is the end time of the OPERATION. For PLOT information, the first word is the value of the independent variable.

VAL

Values of the variables to save.

5. I/O VARIABLES:

A. INPUT

LOCATION

ICODE

F.P.

IITPS

/I01/

INUM

F.P.

IOPNO

F.P.

IPNT

F.P.

IVAL

F.P.

NV

F.P.

T

F.P.

VAL

F.P.

B. OUTPUT

LOCATION

(none)

6. CALLING ROUTINES: DDPRNT
SSAVE
TTABU
UUPRNT

7. CALLED ROUTINE: (none)

1. NAME: WTHMN - 543
2. PURPOSE: This routine changes the initial values of variables in accordance with the WITH or MINIMIZE command.
3. METHOD: If there are no WITH or MINIMIZE variables, control returns to the calling routine. If there are WITH or MINIMIZE variables, then their new current values are calculated and control returns to the calling routine.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Indicator to determine if the variable is a READ variable or not. If I equals zero, then the variable is a READ variable.
ICHAR	Array containing the IOSYM recognized characters.
IDV	Variable identifier of the variable being processed.
IISIM	Current simulation number.
IRAY	Array containing name information, one character per word.
IWT	Number of blanks needed to fill a name.
JJ	Pointer to the IRAY array where a blank should be stored.
JK	Index used to process blank filling the name.
L	Pointer and index to the NNWRK array where the WITH or MINIMIZE variable is located.
LEND	Location in the NNWRK array where the last WITH or MINIMIZE variable is located.
LLWITH	Location in the NNWRK array where the WITH or MINIMIZE information starts.

NNSIM	Number of simulations in the computer run.
NNWITH	Number of WITH variables if positive. Negative of the number of MINIMIZE variables if negative.
NNWRK	Work storage array where the WITH and MINIMIZE information is stored.
NPRNT	Device code used to print output.
NWT	Number of characters in a name.
TTBEG	Beginning time of the simulation.
VAL	Initial value of the variable.
WWRK	Array equivalenced to the NNWRK array used to retrieve real values from the NNWRK array.

5. I/O VARIABLES:

A. INPUT	LOCATION
ICHAR	/ME1/
IISIM	/I03/
IRAY	F.P.
LLWITH	/I03/
NNSIM	/I03/
NNWITH	/I03/
NNWRK	/I03/
NPRNT	/GCOM1/
NWT	F.P.
WWRK	/I03/
B. OUTPUT	LOCATION
IDV	F.P.

TTBEG

/GCOM1/

VAL

F.P.

6. CALLING ROUTINE: INTLC
7. CALLED ROUTINES: PPIDF
UNCODE

1. NAME: XMAGJ
2. PURPOSE: This routine determines the characteristic of Y.
3. METHOD: The log of Y is taken to find the power of ten that Y is. The result is taken to the power of ten and the result is returned.
4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
PREC	Machine precision.
XMAGJ	Upon returning from this routine, this contains the result of this function.
Y	Variable to find the characteristic of.
Z	Intermediate result.

5. I/O VARIABLES:

A. INPUT	LOCATION
PREC	/ALLJ/
Y	F.P.
B. OUTPUT	LOCATION
XMAGJ	FUNCTION

6. CALLING ROUTINE: PVARJ
7. CALLED ROUTINE: EVALJ

1. NAME: XPMNMX
2. PURPOSE: This routine calculates the range for use by routine XPLOT. This routine was obtained from the SANDIA MATHEMATICAL PROGRAM LIBRARY/DIVISION 2642.
3. METHOD: Given an array with values, this routine will find the maximum and minimum value by a common shell sort.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Index used to determine which array element is being investigated.
INIT	Indicator. If greater than one, then the initial maximum and minimum values have been set.
N	Dimension of the X array, and number of elements to check.
X	Array to calculate minimum and maximum from.
XMAX	Maximum value in the X array.
XMIN	Minimum value in the X array.

5. I/O VARIABLES:

A. INPUT	LOCATION
INIT	F.P.
N	F.P.
X	F.P.
B. OUTPUT	LOCATION
INIT	F.P.
XMAX	F.P.
XMIN	F.P.

6. CALLING ROUTINE: XPLOT
7. CALLED ROUTINE: (None)

1. NAME: XPLOT
2. PURPOSE: This routine does line printer plots. This routine was obtained from the SANDIA MATHEMATICAL PROGRAM LIBRARY. For documentation on this routine, see SANDIA document SAND75-0545. The only modification to this routine was to allow it to do five curves on a single plot instead of four.
3. METHOD: See SAND75-0545 document.
4. INTERNAL VARIABLES: Not applicable.
5. I/O VARIABLES: Not applicable.
6. CALLING ROUTINE: PPRPLT
7. CALLED ROUTINES: XPMNMX
XPSYMB

1. NAME: XPSYMB
2. PURPOSE: This routine inserts symbols for a curve into the line image. This routine was obtained from the SANDIA MATHEMATICAL PROGRAM LIBRARY and is discussed in the Sandia document SAND75-0545.
3. METHOD: See the SAND75-0545 document.
4. INTERNAL VARIABLES: See above.
5. I/O VARIABLES: See above.
6. CALLING ROUTINE: XPLOT
7. CALLED ROUTINES: (None)

1. NAME: XTAJ
2. PURPOSE: This routine computes $B(J) = \text{Sum over } I \text{ of } A(I, J) * X(I)$ Where A is a matrix and B and X are arrays.
3. METHOD: The A(I, J) matrix is multiplied by array X (I) for all I and in B(J) array. This is done for each I, control then returns stored to the calling matrix.

4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
A	Matrix to be multiplied.
B	Array that is returned with the calculated values.
I	Index for pointing to the array rows.
J.	Index for pointing to the array columns.
M, MI	First dimension of the A matrix and the dimension of the X array.
N	Second dimension of the A matrix and the dimension of the B array.
X	Array to multiply the A matrix by.
Z	Temporary storage for the calculations of A(I, J) times X(I).

5. I/O VARIABLES:

A. INPUT	LOCATION
A	F. P.
M	F. P.
N	F. P.
X	F. P.
B. OUTPUT	LOCATION
B.	F. P.

6. CALLING ROUTINE: (None. Not used at this time)
7. CALLED ROUTINE: (None)

1. NAME: XXDATN - 545
2. PURPOSE: This routine contains the major portion of the GASP-IV routine DATIN. This routine is used to process GASP input for IOSYM. Only minor changes have been made, and these are clearly indicated by comments in the routine. For a detailed description of this routine, see the GASP book description of routine DATIN.
3. METHOD: See GASP-IV book description of routine DATIN.
4. INTERNAL VARIABLES: See above.
5. I/O VARIABLES: See above.
6. CALLING ROUTINES: BXDATN
CXDATN
7. CALLED ROUTINES: BUILD (GASP routine)
DFAUT (GASP routine)
ERRIN (GASP routine)
FILEM (GASP routine)
IMAP (GASP routine)
IPACK (GASP routine)
SET (GASP routine)

1. NAME: XXINTL - 544
2. PURPOSE: This routine contains the initialization portions of the GASP-IV routine DATIN. Changes are noted by comments in the routine. See the GASP-IV book for a detailed description of the DATIN routine.
3. METHOD: See the GASP-IV book.
4. INTERNAL VARIABLES: See above.
5. I/O VARIABLES: See above.
6. CALLING ROUTINE: INTLC
7. CALLED ROUTINES: CLEAR (GASP routine)
DRAND (GASP routine)
PRNTQ (GASP routine)
PRNTS (GASP routine)
SET (GASP routine)
STATE

1. NAME: ZERO - 33
2. PURPOSE: This routine is used to initialize all the IOSYM arrays and variables before processing a new NETWORK.
3. METHOD: The variables are assigned default values and routines IISSET and ZZSET are called to initialize the arrays. Routine DFAUT is called to initialize GASP variables.
4. INTERNAL VARIABLES: A description of all the internal variables can be found in the data descriptions in volume one of the IOSYM programmers manual.
5. I/O VARIABLES: Not applicable.
6. CALLING ROUTINE: CHOIC2
7. CALLED ROUTINES: ARAYCK
DFAUT
IISSET
ZZSET

1. NAME: ZVECJ
2. PURPOSE: This routine zeros out the input vector.
3. METHOD: Each element in the input vector is set to zero.
4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	Do loop index.
N, NN	Number of elements to zero out.
X	Array to zero out..

5. I/O VARIABLES:

A. INPUT	LOCATION
N	F. P.
B. OUTPUT	LOCATION
X	F. P.

6. CALLING ROUTINE: (None. Not used at this time)
7. CALLED ROUTINE: (None)

1. NAME: ZZSET - 516
2. PURPOSE: This routine will set the elements in array Z to the value in VAL.
3. METHOD: The elements one at a time are set to VAL using a DO loop.
4. INTERNAL VARIABLES:

VARIABLE	DEFINITION
I	DO loop index and counter.
N	Dimension of the Z array and the number of elements to set to the value of VAL.
NN	The number of elements in the Z array to set to VAL.
V	Value to set the Z array elements to.
VAL	Value to set the Z array elements to.
Z	Array to be set.

5. I/O VARIABLES:

A. INPUT	LOCATION
N	F.P.
VAL	F.P.
B. OUTPUT	LOCATION
Z	F.P.

6. CALLING ROUTINES: AARCST
CCOLC
INTLC
NNODST
RREPRT
STATE
ZERO

7. CALLED ROUTINES: (None)

DISTRIBUTION:

Amoco Production Company
Research Center
P.O. Box 591
Tulsa, Oklahoma 74102
Attn: B. Wilder

Department of Energy (10)
Division of Geothermal Energy
Mail Station 3344
12th & Penn, NW
Federal Building
Washington, DC 20461
Attn: C. McFarland
C. Carwile (2)
M. Skalka
A. Jelacic
R. Toms
R. R. Reeber
J. Salisbury
R. Black
M. Scheve

Department of Energy (5)
Office of Oil, Gas, and Shale
Technology
Mail Station D107
Washington, DC 20545
Attn: J. Smith
M. Adams
B. Dibona
D. Duke
A. Sikri

BDM Corporation
1801 Randolph Rd SE
Albuquerque, NM 87106
Attn: A. J. Mansure

Dresser Industries, Inc.
P.O. Box 24647
Dallas, TX 75224
Attn: J. W. Langford

Dyna-Drill
P.O. Box C-19576
Irvine, CA 92713
Attn: Larry Diamond

Livesay Consultants
129 Liverpool
Cardiff, CA 92007
Attn: B. J. Livesay

Halliburton
Drawer 1413
Duncan, OK 73533
Attn: Dwight Smith

Loffland Brothers Company
P.O. Box 2847
Tulsa, OK 74101
Attn: H. E. Mallory

Los Alamos National Scientific
Laboratory
Mail Stop 570
Los Alamos, NM 87545
Attn: John C. Rowley

Mobil Research and Development
Corporation
Field Research Laboratory
P.O. Box 900
Dallas, TX 75221
Attn: W. Gravley

NL Baroid Petroleum Services
City Centre Building, Suite 365W
6400 Uptown Boulevard
Albuquerque, NM 87110
Attn: Gene Polk

NL Petroleum Services
P.O. Box 1473
Houston, TX 77001
Attn: J. Fontenot

Otis
P.O. Box 34380
Dallas, TX 75243
Attn: W. D. Rumbaugh

Phillips Petroleum Company
P.O. Box 239
Salt Lake City, UT 84110
Attn: Earl Hoff

Pritsker and Associates, Inc.
P.O. Box 8345
Albuquerque, NM 87198
Attn: J. Polito

Scientific Simulation Inc.
P.O. Box 9331
Albuquerque, NM 87119
Attn: D. M. Smith (5)

Smith Tool Company
P.O. Box C-19511
Irvine, CA 92713
Attn: J. Vincent

Texas A&M University
College Station, TX 77843
Attn: M. Friedman
Dept. of Geology

Union Geothermal Division
Union Oil Company of California
Union Oil Center
Los Angeles, CA 90017
Attn: D. E. Pyle

Shell Oil Co.
Two Shell Plaza
P.O. Box 2099
Houston, TX 77001
Attn: W. E. Bingman

4700 J. H. Scott
4740 R. K. Traeger
4741 J. R. Kelsey (20)
4743 H. C. Hardee
4746 B. Granoff
4748 B. E. Bader
4750 V. L. Dugan
4751 J. R. Tillerson
4752 H. M. Dodd
4756 Y. T. Lin (5)
3141 L. J. Erickson (5)
3151 W. L. Garner (3)
For DOE/TIC (Unlimited Release)
DOE/TIC (25)
(J. Hernandez, 3154-4)