SLAC – 234(REV.)
UC – 32
(M)

HANDYPAK
A HISTOGRAM AND DISPLAY PACKAGE

(RELEASE 6.5)

A. Boyarski

Stanford Linear Accelerator Center
Stanford University
Stanford, California 94309

November 1980
Revised June 1986
Revised September 1988

Prepared for the Department of Energy
under contract number DE-AC03-76SF00515

Printed in the United States of America. Available from the National Technical Information Service, U.S. Department of Commerce, 5285 Port Royal Road, Springfield, Virginia 22161. Price: Printed Copy A07, Microfiche A01.

ABSTRACT

HANDYPAK is a set of Fortran subroutines which a user may call for defining, accumulating, and outputting histograms or scatter plots. Sliced projections and statistical means and variances can be generated. Smooth curves and user specified functions or contours may also be drawn. User blocks may also be defined and manipulated. Histograms and user blocks can be saved on disk, and can be retrieved from disk at a later time. Output can be directed to a line printer, or to a variety of graphic devices. Plotting formats may be changed by the user. Windowing features allow multiple plots per page on graphic devices. Handypak also runs interactively with a conversational command language which allows histograms to be manipulated or displayed at a terminal. This code works on IBM VM/CMS and VAX/VMS operating systems.

Acknowledgements

This package has evolved over the years to its present form, and has had many contributors along the way. The display portion was done first in 1967 on a SDS 9300 computer at SLAC. In 1971, A. J. Cook added the histogramming portion when the package was installed on the Sigma 5 computer at SPEAR. In 1972 the package was installed on the IBM machine, and many contributions were made by R. L. A. Cottrell and C. A. Logg in the scatterplot, statistics, and option setting areas. M. Fisherkeller helped in recoding parts of the package for the VAX computer. Contributions were also made by various users who offered new suggestions and tested existing features.

ABOUT THIS MANUAL

This manual serves both as a user guide and a reference manual. The first three sections give an introduction and a functional description of Handypak. Section 4 gives the descriptions of each subprogram. Examples of displays are found in section 5. Section 6 describes Conversational Handypak.

REVISION HISTORY

Changes since the previous release are marked with a | in the margin.

Release 6.5 (1988) incorporates the following changes:

- DPWDOW added to produce multiple windows on graphic plots.

- 'GETN' option added to HBLOCK to retrieve size of the block and the title contents.

- HANDYPAK was internally recoded in Fortran77. However, user callable routines still retain Hollerith arguments (not character) for backward compatibility. Two new routines (DOPTC and HOPTC) use CHARACTER arguments.

- DPUG2 is no longer needed, since the default is UG77. The old FORTHX version of Unified Graphics can not be used.

- Added a YALT=3 option for the A*4 mode of storage, giving the variance and the error in the variance for each bin.

- Fixed errors for EFF and ASM modes of storage. Errors were too large when the success (or failure) count was less than 10 for trials between 12 and 500.

- Some improvements were made in waiting after a plot on interactive devices.

- Added continuation ability to CHP2 command lines.

- Additions and changes in DOPT:

NAME	sets name field
DATE	sets date field
SYMSIZE	sets size of plotting symbol.
LINDEN	can be 1 (faint) to 5 (dark)
PNTDEN	no longer coupled to LINDEN.
FULSCR	can be changed any time, not just at start.

- Small fixes were made in many places.

Release 6.3 (1986) incorporated the following changes:

- Improvements to 1-D line printer plots were made:
  - smooth curve, analytic function, and multidata values are printed as well as drawn on the graph.
  - YERR, YCUM fields can be turned off/on.
  - bin over/under flows are marked with '>' or '<'.

- Improvements to 1-D graphic plots were made:
  - larger titles now possible.
  - margins increase as needed.

- Improvements to 2-D graphic plots were made:
  - 2-D histogram (sometimes called leggo plots)
  - 2-D mesh plot
  - histogram slices along X or Y
  - line style slices along X or Y
  - hidden lines may be on or off in above.

- TITLE strings may now be up to 256 characters long, and a new notation allows duplex strings on graphic devices.

- Scaling is done in engineering notation (powers of 1000).

- The following new DOPT options were added

MARKER	- specifies UG marker symbol for 1-D plots
PNTSYM	- specifies UG marker symbol for scatterplots
ISOMETRIC	- scatterplot is plotted as an isometric plot
TISIZE	- controls size of titles and labels, and margins are increased for large sizes.

JBFONT	- controls plotting of jobname and Handypak logo.
CHFMT	- specifies character fonts
DEVICE	- allows (re)defining of graphic devices.
2DLINE	- specifies type of isometric plot (as mesh plot, 2-D histogram, line slices, or histogram slices).
HIDE	- controls hidden line removal.
ESCT	- set control string for flipping to text screen.
ESCG	- " " " " " " graphic " .
OMIT	- is now set automatically according to HOMIT.

- Some of the defaults in DOPT have been changed:

Option	Was	New default	
XTIC	8	0	(do auto tic)
YTIC	8	0	(do auto tic)
XAUTO	1	2	(do full data scaling)
Y2AUTO	1	2	(do full data scaling)
L2CAUTO	0	1	(auto columns in 2-D LP plots)
L2ZAUTO	0	1	(auto Z scale in 2-D LP plots)
OVPRT	1	0	(does not overprint)

- Mode of storage changes:
  - WEV has been redone
  - LPT has been added (keeps last N points from HCUM1)

- HWRITE can MOD (APPEND) to an existing file.

- Conversational Handypak (CHP2) was added.

- Extra options for UGOPEN are allowed in device arguments.

- DINIT has been added to allow the UG buffer to be other than the common block /SCPBUF/.

- Subroutine NWPAGE has been replaced by DWRHDR.

- DMGRUT no longer suppresses the accumulation of scatterpoints.

- Object file DPUG2 makes Handypak use UG77 rather than old UG.

- Former HPAK and DPAK routines have been removed.

CONTENTS

Section 1

HANDYPAK

1.1 INTRODUCTION

HANDYPAK, a histogram and display package, is a set of Fortran subroutines which are useful for generating and displaying histograms having one or two dimensions (1-D or 2-D). This package was written with the intent of being easy to use with a minimum number of subroutines and common blocks, but yet allow the user to change various options in the histogramming and plotting phases. A user's program would include calls to subroutines in this package for defining histograms, for accumulating into these histograms, and for outputting the results. A wide choice of options is available for both the defining and outputting stages. It is also organized to be useful both in a batch as well as in a real-time interactive environment.

The histogram package uses one common array of memory to store all the histograms in a dynamic fashion so that new histograms may be added or old ones deleted. Each histogram in this array contains not only the binned data but also its complete description such as bin width, number of bins, title, means and variances (optional), normalization factor, display options (optional), etc. The memory size for each bin may be either 1-byte, 2-byte integer, 4-byte integer, 4-byte real, or other special cases which use 8- or 12-bytes (2 or 3 words). Section 2 describes the histogramming routines.

Displays of histograms, or any graphic information independent of the histogram package, can be plotted by the display portion in HANDYPAK. Displays can output to a line printer or graphic device such as a calcomp, versatec plotter, microfilm plotter, graphic terminal, and others. The Unified Graphics Fortran77 subroutines by R. Beach [1] are used to provide graphic output. Default settings of options are provided for such items as size of plot, scale factors, number of tic marks, linear/log scales, error bars, etc. which the user may change.

1.2 NOTATION AND CONVENTIONS

HANDYPAK uses the following naming convention for subprograms and common blocks, with only a few exceptions:

HISTOGRAM routines begin with the letter H
DISPLAY routines begin with the letter D

The Fortran convention is used in this manual for variable names. Variables starting with I, J, K, L, M, or N are INTEGER and the rest are REAL unless otherwise specified. Use is made of LOGICAL*1, INTEGER*2, and REAL*8 in addition to INTEGER and REAL.

Some of the routines can be called with a variable number of arguments. If the optional arguments, denoted by square brackets, are not supplied then default values are provided. For example,

```
CALL HINIT(NH [,NHASH [,NSCAT]])
```

means that HINIT can be called with 1, 2, or 3 arguments, as

```
CALL HINIT(NH)
CALL HINIT(NH,NHASH)
CALL HINIT(NH,NHASH,NSCAT)
```

Arguments which supply text strings are passed as Hollerith for backward compatibility with the previous (FORTHX) version. Fortunately, both IBM and VAX Fortran compilers allow literal string arguments enclosed in quotes to be used in place of Hollerith. For example, both of the following will work:

```
CALL HOUT(4HALL )      !Formally correct
CALL HOUT('ALL ')      !OK also
```

but the following

```
CALL HOUT('ALL')       !Incorrect
```

will not work correctly because 4 characters are not supplied. Similarly, some arguments need 8 characters, such as 'TEK4010 ', the name of a graphic device. This manual uses the quoted string format, but it must be remembered that such arguments are really HOLLERITH unless expressly stated otherwise.

1.3 SIMPLE EXAMPLE

The following example illustrates the ease with which histograms can by generated. A plot is made of the frequency distribution of a random variable having a normal distribution of zero mean and unit variance as given by a function GAUSS(X). A histogram with an identifier 1 is defined with 32 bins of INTEGER*2 storage, a low edge bin value of -4.0, a bin width of 0.25, a main title 'EXPT-A', an x-subtitle 'DEVIATION', and a y-subtitle 'FREQ'. The entire program is:

```
      CALL HDEF1(1,'I*2 ',32,-4.0,0.25,'EXPT-A;DEVIATION;FREQ@')
      DO 100 I=1,500
  100 CALL HCUM1(1,GAUSS(X),1.0)
      CALL HOUT(1)
```

and the resulting line printer plot is shown in figure 1.1

To generate a plot on a graphic device such as a Versatec plotter, only one additional call is needed,

```
      CALL HOUT(1,'VEP12FF ')
```

(together with appropriate JCL or device assignment) to give the plot shown in figure 1.2. More examples are shown in section 5.

*

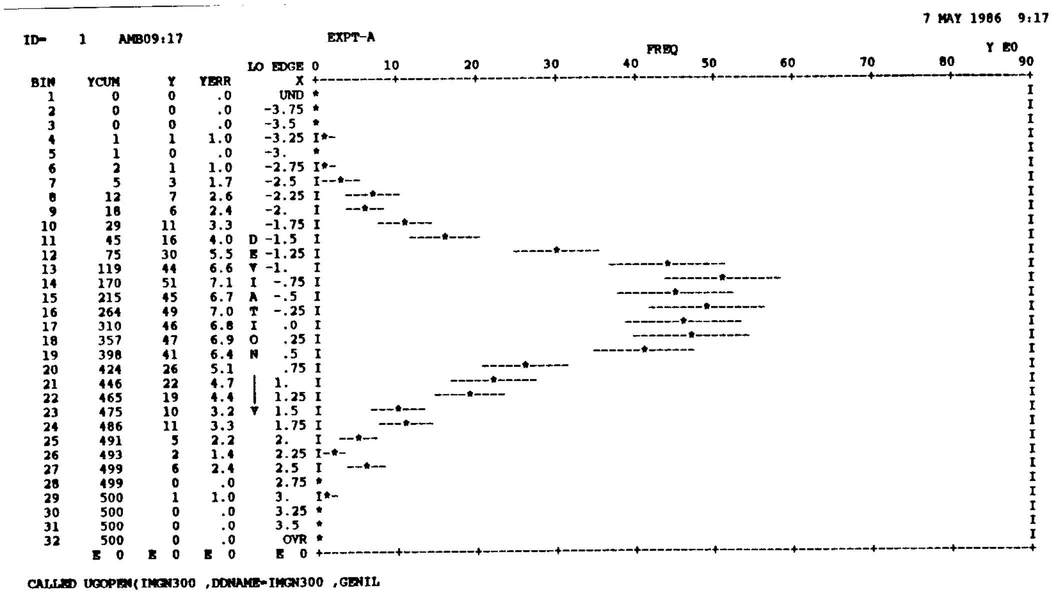


Figure 1.1 1-D Histogram Plot, on line printer, default options

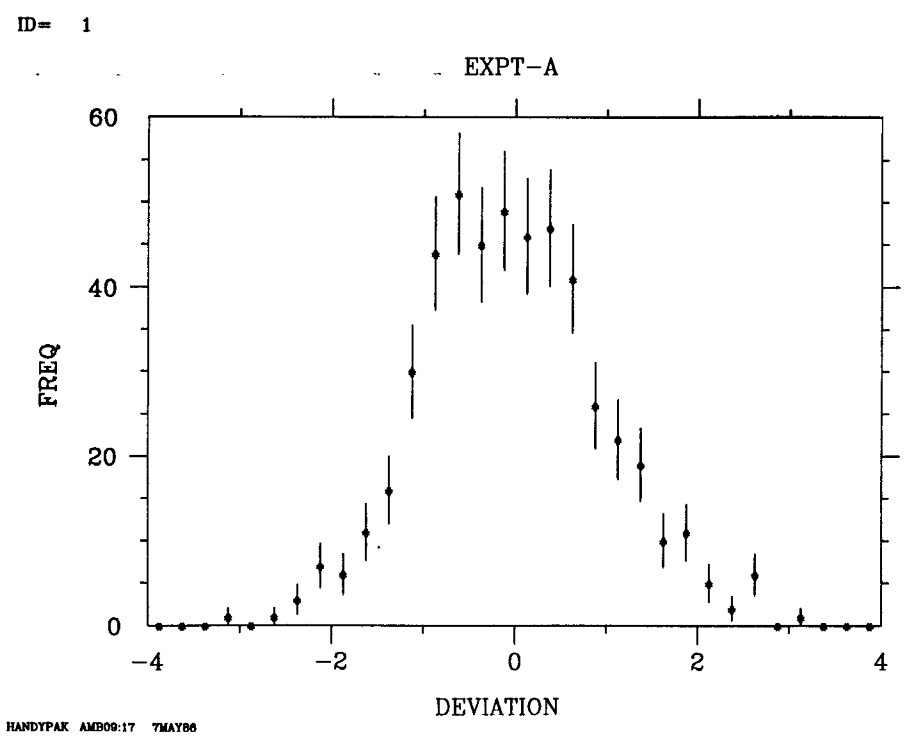


Figure 1.2 1-D Histogram Plot, on graphic device, default options

Section 2

FUNCTIONAL DESCRIPTION OF HISTOGRAM PACKAGE

2.1 OVERVIEW

Only a few routines need be learned to use the histogram package. The default settings of the various options are generally adequate for new users. The most common sequence of operations and the subroutines involved is:

- Define storage size (optional)	HINIT
- Define the histograms	HDEF1, HDEF2, HOPTN
- Accumulate	HCUM1, HCUM2
- Output the histograms	HOUT, DOPT, HOPTN
- Clear the histograms	HCLR

Each of the steps in the above sequence is described in this section from a functional point of view. The subroutines themselves are described more fully in section 4. The heart of the histogram package is the pooled buffer area which all histogram routines interrogate or modify, as outlined below.

All histograms are stored in the common block /HCOM/, whose default size is 2000 words. If a larger size is needed, then the user should declare this common block with the larger size and also include a call to HINIT (described later) indicating this size. The histograms are stored one after the other in a dynamic fashion in HCOM. Each histogram, or node, has a unique identifier (ID) which is used by the various routines to find the location of the histogram. ID may be an integer in the range 1 to 9999, or it may be a word of hollerith characters, as 'ABCD'.

In most cases the user does not need to be concerned with the structure of HCOM. Subroutines are provided which manage this area. A description of /HCOM/ is given in Appendix A.

2.2 INITIALIZING BUFFERS

A user should declare a larger size for the histogram buffer in HCOM (if necessary) before defining any histograms. The size of HCOM is internally defined to be 2000 words. This size is generally adequate for the new user, but will need to be increased, for example to 5000, or 10000, or more words as the user's program expands. Increasing the size requires that the user declare HCOM with the larger size and also call HINIT with that size. For example,

```
COMMON / HCOM / M(5000)

CALL HINIT(5000)
```

defines the histogram storage area to be 5000 words.

HINIT can be called with additional arguments as described in section 4 to define the sizes of the hash tables and scatterplot buffer, and the unit number for the scatterplot scratch file. If the user does not explicitly call HINIT, then the first time call to any other histogram routine (e.g. HDEF1) will implicitly call HINIT to assure that /HCOM/ is initialized. There is no block data program for /HCOM/. HINIT prints the release number, and the size of HCOM which is to be used. The user may call HSPACE after having defined all histograms to print out the amount of free space still available in HCOM.

The default unit number for the scatterplot scratch file is set to 19.

2.3 DEFINING HISTOGRAMS

One and two dimensional histograms are defined or allocated in /HCOM/ by the HDEF1 or HDEF2 subroutines, respectively. Arguments to these routines specify the histogram identifier (ID), the mode of storage (MS), the number of bins (NX,NY), the low edge bin value (XL,YL), the width of a bin (XW,YW), and a title (TITLE), as follows,

```
1-D:    CALL HDEF1(ID,MS,NX,XL,XW,TITLE)
2-D:    CALL HDEF2(ID,MS,NX,NY,XL,YL,XW,YW,TITLE)
```

The MS and TITLE variables are described later in sections 2.4 and 3.4 respectively. Other histogram options, such as whether overflows are to be kept or lost, whether statistical summations are to be made, the normalization constant used in making the plot, etc. are taken from default values. The default values of the options used by HDEF are described in section 4.18. The default settings of the options may be changed by HOPTN at any time. Histograms already defined are unaffected, while subsequent HDEF's will use the new default values. HOPTN can also overwrite the options for selected histograms (or all histograms) with specified values.

Certain options should only be changed prior to any accumulations being made for that histogram, otherwise surprising results will be found at the output stage. These options are ones that relate to the statistics block and the histogram bins, namely,

'STAT', 'SLIMIT', and 'SOMIT' for statistics, and
'HOMIT', 'HBINS', 'HOMIT', 'HLOW', and 'HWIDTH' for the bins.

Examples:

Define a 1-D histogram with identifier 1, I*2 storage, 20 bins, low edge value of 25.0, bin width 5.0, and title 'TEST 1',

```
CALL HDEF1(1,'I*2 ',20,25.0,5.0,'TEST 1@')
```

Define a 2-D histogram with identifier 'ABCD', 'E*4' mode of storage, 30 X bins by 20 Y bins, low edge X of 0., low edge Y of 15., X bin width of 1.0, Y bin width of 2., and a title 'TEST 2',

```
CALL HDEF2('ABCD','E*4 ',30,20, 0.,15., 1.,2., 'TEST 2@')
```

A scatterplot would be included in the previous example by,

```
CALL HDEF2('ABCD','E*4S',30,20, 0.,15., 1.,2., 'TEST 2@')
```

2.4 MODE OF STORAGE

The mode of storage variable (MS) provides a very important feature in Handypak. This variable specifies not only the number of bytes to be used for each accumulation bin, but also the kind of accumulation that is to be done, as well as specifying whether a point-by-point scatterplot is to be made.

In most cases, only a simple accumulation of counts is desired, with an error value being the square root of the accumulated counts per bin. In this case MS can be 'L*1 ', 'I*2 ', or 'I*4 ' for 1-, 2-, or 4-byte storage per bin. However, if accumulations are done with both positive and negative values, then correct errors can be supplied by using the 'E*4 ' mode of storage. Other more specialized cases can accumulate means and variances for each bin, rather than counts, by using the 'A*4 ' mode of storage. It is also possible to accumulate trials and successes, and then have the ratio (efficiency) of successes over trials printed out with the (asymmetric) 95% confidence levels for each bin.

A summary of all modes of storage is shown below, together with the index IMS that is stored in the node.

IMS	MS	BYTES	TYPE	PRINCIPAL USE
1	'L*1 '	1	integer	counts
2	'I*2 '	2	integer	counts
3	'I*4 '	4	integer	counts
4	'R*4 '	4	real	weighted counts
5	'E*4 '	8	2 real words	value and error
6	'M*4 '	8	2 real words	mean and error
7	'WEV '	8	2 real words	weighted events
8	'A*4 '	12	3 real words	avg and variance
9	'EFF '	8	2 real words	efficiency (ratio)
10	'EF2 '	4	2 halfwords	" "
11	'EF1 '	2	2 bytes	" "
12	'ASM '	8	2 real words	asymmetry
13	'AS2 '	4	2 halfwords	"
14	'AS1 '	2	2 bytes	"
15	'LPT '	8	2 real words	graph of points

For 2-D histograms, an additional 'S' character may be included in the 4'th position to indicate that a point by point scatterplot file is to be generated in addition to the histogram bin storage. For example

'I*2S'

specifies a storage mode of 2-byte integer bins, as well as point storage into a scatterplot file. In the output phase, the binned data is used for line printer plots, while the scatterplot file is read back to make point by point scatterplots on graphic devices.

In the following descriptions for the various modes of storage, W(1) is the value (weight) accumulated by HCUM1 or HCUM2 and W(2) is the error (for the storage modes that need it), while B(1), B(2), and B(3) designate the storage cell(s) for a bin. F refers to the output result for a bin in the plot stage (as provided by the functions HY and HYE, or by HZ and HZE). In some of the cases, it is desirable to output alternate forms of data by setting the YALT option. For the 'EFF' mode of storage for example, it is possible to output either the efficiency (=number of successes over number of trials), or the number of trials, or the number of successes. The value of 'YALT' (set by HOPTN) selects alternate output for a histogram.

1. L*1 Mode of Storage

Accumulates value into 1-byte cells, and outputs accumulated value for each cell. The error is taken as the square root of the cell content. The maximum storage count is 255 on IBM, and -128 to 127 on the VAX.

W(1) is the value

B(1) contains sum of the W(1)

F = B(1) +/- SQRT(B(1)) (YALT not used)

2. I*2 Mode of Storage

Accumulates value into 2-byte cells, and outputs accumulated value for each cell. The error is taken as the square root of the cell content.

W(1) is the value

B(1) contains sum of the W(1)

F = B(1) +/- SQRT(B(1)) (YALT not used)

3. I*4 Mode of Storage

Accumulates value into INTEGER cells, and outputs accumulated value for each cell. The error is taken as the square root of the cell content.

W(1) is the value

B(1) contains sum of the W(1)

F = B(1) +/- SQRT(B(1)) (YALT not used)

4. R*4 Mode of Storage

Accumulates value into REAL cells, and outputs accumulated value for each cell. The error is taken as the square root of the cell content.

W(1) is the value

B(1) contains sum of the W(1)

F = B(1) +/- SQRT(B(1)) (YALT not used)

5. E*4 Mode of Storage

Accumulates value and squared error, and outputs accumulated value and square root of accumulated squared error.

W(1) is the value
W(2) is the square of the error

B(1) contains sum of the W(1)
B(2) contains sum of the W(2)

F = B(1) +/- SQRT(B(2)) (YALT not used)

6. M*4 Mode of Storage

Accumulates weighted means and variances from the input values and errors, and outputs the weighted mean and error for each bin.

W(1) is the value
W(2) is the square of the error

B(1) contains {sum(W(1)/W(2))/sum(1./W(2))}
(i.e. the weighted mean)
B(2) contains {1./sum(1./W(2))}
(i.e. the error in the weighted mean)

F = B(1) +/- SQRT(B(2)) (YALT not used)

7. WEV Mode of Storage

Accumulates weighted (or corrected) counts and errors for each bin, and outputs the accumulated value and error.

W(1) is the weighted count (e.g. 1.0/efficiency)

B(1) contains sum of weighted counts W(1)
B(2) contains sum of W(1)**2 (i.e. error = weight)

F = B(1) +/- SQRT(B(2) (YALT not used)

8. A*4 Mode of Storage

Accumulates the average value, the variance, and number of entries (calls) for each bin, and outputs the average and the standard deviation of the values at each bin. B(1), B(2), and B(3) are updated for each HCUM1 or HCUM2 call. The previous mean and variance are used to compute the new values.

W(1) is the value

B(1) contains the average value (=sum(W(1))/CALLS)
B(2) contains the variance
(=sum((W(1)-B(1))**2/(CALLS-1.))
B(3) contains sum of 1.0 (=CALLS to HCUM)

```
F = B(1) +- SQRT(B(2))                 (YALT = 0, gives
                                         avg +- std dev)
F = B(1) +- SQRT(B(2)/B(3))            (YALT = 1, gives
                                         avg +- err in avg)
F = B(3) +- SQRT(B(3))                 (YALT = 2, gives
                                         num of calls)
F = B(2) +- SQRT(2*B(2)**2/B(3))       (YALT = 3, gives
                                       variance +- err in var)
```

9. EFF, EF2, and EF1 Modes of Storage

Accumulates the number of trials and success, and outputs the ratio, which can be used as an efficiency or a proportion, for each bin. The number of trials, or the number of successes or failures can also be displayed. EFF, EF2 and EF1 operate similarly, except EF2 and EF1 use less storage and store integer (2-byte or 1-byte) values only.

W(1) is 1.0 for success, or 0.0 for failure.

B(1) contains sum of successes
B(2) contains sum of trials (=calls)

```
F = B(1)/B(2) +hi err -low err         (YALT = 0, ratio)
F = B(1) +- SQRT(B(1))                 (YALT = 1, successes)
F = (B(2)-B(1)) +- SQRT(B(2)-B(1))     (YALT = 2, failures)
F = B(2) +- SQRT(B(2))                 (YALT = 3, trials)
```

12. ASM, AS2, and AS1 Modes of Storage

Accumulates the number of trials and successes, and outputs the asymmetry, given by

$$A = \frac{\#successes - \#failures}{\#successes + \#failures}$$

for each bin. The number of successes, failures, or trials may also be displayed. ASM, AS2, and AS1 operate similarly, except AS2 and AS1 use less storage (2-byte and 1-byte) and store integer values only.

```
W(1) is the success value (1.0 for success,
                           0.0 for failure)

B(1) contains sum of successes
B(2) contains sum of trials

F = 2.*(B(1)/B(2))-1.0 +hi -lo err    (YALT = 0, asymmetry)
F = B(1) +- SQRT(B(1))                (YALT = 1, successes)
F = (B(2)-B(1)) +-SQRT(B(2)-B(1))     (YALT = 2, failures)
F = B(2) +- SQRT(B(2))                (YALT = 3, trials)
```

15. LPT Mode of Storage

Stores and displays the X,Y coordinate pairs provided by HCUM1(ID,X,Y) for the last N points, where N is the value supplied in HDEF1. This mode of storage is particularly useful for monitoring online devices by displaying only the recent history of the devices. This mode is only possible for 1-D histograms. Note that the XL and XW arguments in HDEF1 are not used, but must be supplied as 0.0, 0.0.

```
B(1) contains X coordinate
B(2) contains Y coordinate

Y(I) vs X(I)                          (YALT=0, default)
Y(I) vs X(I)-XFIRST                   (YALT=1)
Y(I) vs X(I)-XLAST                    (YALT=2)
```

2.5 ACCUMULATION

Accumulations into 1-D or 2-D histograms are made by HCUM1 or HCUM2 respectively. The arguments to these routines are the histogram identifier (ID) the bin coordinate (X), or coordinates (X,Y) for 2-D, and the value or weight (W) to be accumulated into that bin, as follows

```
1-D:    CALL HCUM1(ID,X,W)
2-D:    CALL HCUM2(ID,X,Y,W)
```

If no histogram exists with the specified ID, then these routines simply return with no error message.

The bin number from the X value is calculated by

```
IBIN=(X - low edge X)/(X bin width) + 1
```

where the low edge X, and the X bin width, are those defined by HDEF1 or HDEF2 (or HOPTN) for that histogram.

When an integer storage mode is used for a histogram, the value accumulated is the rounded value for W. i.e. 0.5 LE W LT 1.5 is accumulated as 1, and -1.5 LE W LT -0.5 is taken as -1.

If a bin contents overflows its storage mode, then further accumulation into that bin is lost - the maximum value remains in that bin.

If the 'STAT' option in HOPTN was set true for a histogram, then HCUM1 or HCUM2 also builds the summations (i.e sum(X), sum(X*X), etc.) that are needed for calculating the means and variances in the output stage.

Examples:

```
CALL HCUM1(1,26.3,1.0)
```

will accumulate 1.0 into the bin subtending the coordinate value 26.3 in histogram 1.

```
CALL HCUM2('ABCD',11.1,13.3,1.0)
```

accumulates 1.0 into the bin subtending the X coordinate 11.1 and Y coordinate 13.3, in histogram 'ABCD'.

OVERFLOWS AND UNDERFLOWS DURING ACCUMULATION

A. Abscissa coordinate(s) out of range.

When the value of X in an HCUM1 argument (or X or Y in HCUM2) is outside the limits of the histogram bins, then an underflow or overflow condition occurs in the abscissa coordinate. Such overflows and underflows are handled in one of two ways, depending on the setting of the HOMIT option (by HOPTN) for the histogram. If HOMIT is false (the default), then underflows and overflows are simply accumulated into the first and last (edge) bins of the histogram. For a 2-D histogram, the whole perimeter will contain the overflows and underflows. In the output stage, these edge bins are displayed but are not used by the scale routine when auto scale factors are being calculated, nor when calculating statistics from the bin values.

The alternate method of handling overflows requires that the HOMIT option be set true (by HOPTN) for the histogram(s). In this case, overflows and underflows are omitted from the edge bins so that only the true histogram contents are stored in these bins. The edge bins are used for scaling and statistics calculations in this case. The number of times that underflows or overflows occurred can still be displayed by HOUT if the NCALLS option in DOPT is set true, but these numbers will be printed separately after the histogram.

B. Bin content overflow or underflow

When accumulation into a bin results in that bin reaching an underflow or overflow, then that bin is set to the minimum or maximum value (i.e. latched) and is no longer modified. In the output stage, such bins are flagged to indicate the overflow (1-D output only). This latching is only done for 'L*1', 'I*2', and 'I*4' modes of storage.

2.6 OUTPUTTING

Histograms and scatterplots, as well as smooth curves, functional curves and contours are output by the routine HOUT. Arguments specify which histogram (ID) is to be output and what device (OPT) is to be used, as follows

```
CALL HOUT(ID,OPT)
```

If the second argument is not supplied, then the default device is used (initially the line printer). ID may have the value 'ALL ' to output all histograms.

The default unit number for the line printer is 6. Error messages are also output on unit 6. These may be changed with the 'OUNIT' and 'EUNIT' options by the DOPT routine.

HOUT is a driver to the display routines DUT1 or DUT2, depending on the dimensionality of the histogram. It is the DUT routine that actually generates the histogram plot, the smooth curves, the functional curves and/or the contours. Hence, the various options for the DUT routines (as set by the DOPT routine), also apply to the HOUT routine.

For graphic scatterplots, HOUT uses DUT1 to draw the frame, axes, and contours (if any), and then calls HSCPLT to do the point by point plot. HSCPLT was adapted from the KIOWA package [2]. Scatter points are read off a scratch file, are ordered for efficient Calcomp use, and are then plotted.

There are a large variety of options that may be changed by DOPT prior to calling the output routines. Some examples of options, and their defaults are:

'L2COL'	4	number of columns used per bin in 2-D line printer plots.
'ERRORS'	1	error bars for 1-D plots.
'FRAME'	.TRUE.	frame is drawn on graphic plots.
'YLOG'	.FALSE.	log scale for Y axis.
'SMOOTH'	.TRUE.	draw smooth curve for the data.
'XTIC'	0	number of major tics for X axis in 1-D plots.

The full set of options with their defaults are described in DOPT, section 4.3.

Alternately, these options may be set within each histogram by calling HOPTN with that option. This allows all histograms to be output with a single call, namely HOUT('ALL '..), and yet each histogram may have any of its own display options which over-ride the default values.

In addition to DOPT options, there are parameters related to histograms, which may be different for each histogram, that may be changed by the HOPTN routine prior to outputting the histogram by HOUT or HSLICE. These HOPTN options are:

'SOUT'	for enabling the output of statistics
'HNORM'	for setting a normalization constant
'HMSCAL'	for specifying manual scale factors
'TITLE'	for specifying a new title
'MTERR'	for specifying the error for an empty bin.
'CONTOUR'	supplies contour function for scatterplots
'NCONT'	for specifying number of contours
'FUNCTION'	supplies analytical function(s) to be plotted
'NFUNC'	for specifying number of analytical functions.

HOPTN is described in section 4.19.

Overflows and underflows are stored in the edge bins (the default). However, they may be stored separately by setting the HOMIT option in HOPTN for the histogram, and displayed by HOUT if the NCALLS options is set in DOPT. If NCALLS is true, but HOMIT is false, then HOUT issues only one item, namely

NCALLS = <number of calls>

If HOMIT is true also, then a 1-D printer output gives

NCALLS = <under> / <hist> / <over>

and a 2-D printer output gives

```
           <ux,uy>     <ux>      <ux,oy>
NCALLS =    <uy>      <hist>      <ov>
           <ux,ov>     <ox>      <ox,oy>
```

where ux, uy, ox, oy mean under x, under y, over x, and over y respectively.

Examples:

Output histogram 1 onto the line printer,

```
CALL HOUT(1,'PRINTER ')
```

Output all histograms to the printer,

```
CALL HOUT('ALL ','PRINTER ')
```

Since the line printer is the default device, then the second argument in the above examples may be omitted if no other device is ever used in the job, e.g.,

```
CALL HOUT('ALL ')
```

2.7 SLICING

A 2-D histogram may be sliced and displayed as a 1-D histogram by the HSLICE routine. The slice direction (XORY) may be either along the X coordinate or the Y coordinate. The width (IWDTH) of the slice (i.e. transverse to the slice direction) can be 1 or more bins wide, starting with bin IBEG. If IWDTH is 0, then all bins beyond IBEG are used. HSLICE sums along the width of the slice. Statistics may also be generated for a slice (calculated from the bin contents within the slice). The output device (OPT) may also be specified as

```
CALL HSLICE(ID,XORY,IBEG,IWDTH,OPT)
```

If OPT is not specified, then the default device is used. If IBEG or IWDTH are not specified, then the values 1 and 0 respectively, are used. If either 'STAT' or 'SOUT' option has been set true by HOPTN for the ID'th histogram, then HSLICE also outputs the statistics for the slice for that histogram.

HSLICE first sets up the slicing variables in /DPMODE/ and then calls HOUT to generate the output. Hence, all the options mentioned in HOUT (and HOPTN and DOPT) also apply to HSLICE.

Examples:

Make a slice along the X coordinate for Y bins from 3 to 7 inclusive (5 bins) for the 2-D histogram with identity 'ABCD',

```
CALL HSLICE('ABCD','X',3,5)
```

2.8 STATISTICS

Statistics provided by HANDYPAK consist of means and standard deviations for 1-D and 2-D histograms, and also correlation coefficients and the error ellipse for 2-D histograms. These values are meaningful only for histograms which have a well defined peak (or cluster) and little or no background in the periphery of the distribution.

The summations needed (i.e. sum(X), sum(X*X), etc.) for calculating the means and variances can be done in one of two ways in HANDYPAK, depending on whether the 'STAT' option was set true of false by HOPTN for that histogram.

1. 'STAT' true - The summations are made during the accumulation phase by HCUM1 or HCUM2, using the argument values supplied by the HCUM calls.

2. 'STAT' false - The summations are made using only the histogram bin contents and bin coordinates. This method does not give as accurate an answer as the first because of the bin resolution, but it saves computer time since HCUM does not have to make the sums on each call.

There are two ways of outputting the statistical values - either with the histogram bin plot or separately. HOUT or HSLICE can output both the histogram contents and the statistics (if 'STAT' and/or 'SOUT' were set true in HOPTN). If only statistics are desired, then the routine HSTAT should be called instead (described in section 4.25). HSTAT also has two optional arguments which may be supplied to restrict the range of bins over which the statistics is calculated.

The HOPTN routine provides several options for controlling the statistics for a histogram. These options are:

'STAT'	for allocating or deleting the statistics block.
'SOUT'	for enabling the statistics output by HOUT.
'SLIMIT'	for over-riding the default limits over which statistics are made.
'SOMIT'	for controlling whether limits are to be used.

The options 'STAT', 'SOMIT', and 'SLIMIT' must be issued after the HDEF and before the HCUM calls are made for that histogram. 'SOUT' may be issued anytime prior to the HOUT but after the HDEF calls are made for that histogram.

Examples:

1. Output both the histogram bin contents and statistics for histogram 1,

```
CALL HOPTN('SOUT',.TRUE.,1)
CALL HOUT(1,'PRINT')
```

2. Print only the statistics for histogram 1,

```
CALL HSTAT(1,'PRINT')
```

3. Print statistics for a slice along X axis for histogram 'ABCD' for Y bins 3 to 7 inclusive (5 bins),

```
CALL HSTAT('ABCD','XPRINT',DUM,3,5)
```

Note that a dummy argument DUM must be supplied in this case.

4. Print statistics for histogram 1, using X-bins 3 through 11 inclusive,

```
CALL HSTAT(1,'PRINT',DUM,3,11)
```

2.9 FETCHING

In most cases the simple sequence of routines HDEF-HCUM-HOUT is sufficient for obtaining results with HANDYPAK. However, as the user's analysis routines become more sophisticated, it may be necessary to obtain information from the histogram so that further specialized processing can be done. Such items as the bin coordinates, the bin contents, the statistical values, etc. may be needed by the user's program. The following routines are provided for this:

1. HGET can print or return the number of bins, the mode of storage, the low edge value, the bin width, and the title.

2. HOPTN with the 'GET,....' option can return the parameters or status, such as whether statistics are defined, whether histogram bins are defined, what limits are used for the statistics, the values of manual scale factors, etc. for a histogram.

3. HX, HY, HYE, H2Y, HZ, HZE, and HN - These functions provide the coordinate values, the bin contents, the error values, and the number of calls made for the histogram last selected by a call to HPNTR(ID).

4. HSTAT with the 'GET,....' option returns statistical values for a histogram.

5. HPNTR(ID,ITEM) returns the location in HCOM for ITEM (e.g. 'STATS', 'HIST', 'PARMS', etc.) in the ID'th histogram.

Examples:

Print the definitions for all histograms,

```
CALL HGET('ALL ').
```

Return values of the dimensionality (ND), the number of bins (NB), the mode of storage (MS), the low bin edge (XL), the bin width (XW) for the ID'th histogram (NB, XL, and XW dimensioned 2),

```
CALL HGET(ID,'GET ',ND,NB,MS,XL,XW).
```

Fill SL(2) with limits used for making statistics,

```
CALL HOPTN('GET,SLIMIT',SL,ID).
```

Obtain the value in the 2'nd bin for the ID'th histogram (1-D),

```
IF(HPNTR(ID).EQ.0)  -- return, no such histogram
VAL=HY(2)
```

2.10 CLEARING

A histogram may be cleared by the call

```
CALL HCLR(ID)
```

where ID is the histogram identifier, or 'ALL ' if all histograms are to be cleared. Both the bin contents and the statistics (if defined) are set to zero.

HCLR is automatically called by HDEF1 or HDEF2 when a histogram is initially created.

Example:

Clear the contents of histogram 'ABCD',

```
CALL HCLR('ABCD')
```

Clear all histograms,

```
CALL HCLR('ALL ')
```

2.11 USER BLOCKS

2.11.1 User Block Within Histogram

In special cases it is convenient to store miscellaneous information with each histogram in HCOM. Examples might include a list of cuts used for each histogram, a description of each histogram, parameters used in generating each histogram, etc. HANDYPAK provides a block that the user may define for each histogram to store this information. The user first defines the size of the block by

```
CALL HOPTN('USIZE',NWD,ID)
```

where NWD is the number of words to be allocated in the ID'th histogram, or in all histograms if ID has the value 'ALL '. The entire block of data can then be stored or retrieved by the calls

```
CALL HOPTN('UDATA',A,ID)
CALL HOPTN('GET,UDATA',A,ID)
```

where A is an array dimensioned NWD words long. If only parts of the user block are to be updated, then the following method can be used:

```
(define /HCOM/ and /HNODE/ as in Appendix A)

 INTEGER HPNTR
 MUSER=HPNTR(ID,'USER')
 IF(MUSER.EQ.0) --> error, no histogram or no user block
 M(MUSER+J) --> gives contents of J'th word in user block.
```

2.11.2 User Block All to Itself

It is possible to define a user block with its own ID and title. Each such block has its own control area, ID, title and a single data array. The routine HBLOCK can define such blocks and store data into or retrieve data from these blocks. Such blocks are useful particularly when it is desired to write out histograms onto disk (by HWRITE), together with non-histogram data such as tallies or constants. By defining a user block for the tallies, another for the constants and filling these, they may also be written out with the histograms. Subsequent jobs or cooperating tasks in a multi-processing system may read back the histograms and/or the user blocks (by HREAD).

2.12 WRITING TO DISK

Histograms and user blocks may be written to disk by the routine HWRITE, and read into memory by the routine HREAD. There are arguments for each of these routines to allow either the entire /HCOM/ common block to be written/read, or to transfer only selected histograms, or to write out all histograms but be able to read only selected ones at a later time. Each time that HWRITE is called, an internal record count is bumped and written out with the data so that a subsequent HREAD can access that particular record. This allows multiple writes to be made for the same histogram(s). For example, an analysis job could fill histograms for one set of data, write out the histograms, clear them, then analyse another set of data and write the histograms again. A subsequent job calling HREAD could select the histograms for the first or second set of data.

Section 4.26 describes the calling sequences for HREAD and HWRITE.

Examples:

1. Write out all the histograms, user blocks, and the Handypak control area in HCOM to a single (first) record on logical unit 20,

```
CALL HWRITE
```

At a later time, this file can be read into HCOM (completely replacing all histograms there) from logical unit 21 by,

```
CALL HREAD
```

Note that all definitions made by HDEF1, HDEF2, or HBLOCK are wiped out and replaced by those off the file when HREAD is called in this manner.

2. Write out HCOM as in example 1, but write to the next record on logical unit 20, leaving the previous record(s) intact,

```
CALL HWRITE(0,20,0)
```

Read in all of HCOM from the second record on logical unit 21,

```
CALL HREAD(0,21,2)
```

The two examples above read and write all of HCOM, and it is not possible to read or write only some of the histograms in this mode. If the latter is desired, it may be done by specifying an identifier, as follows:

3. Write out all histograms and user blocks (but not the control area in HCOM) to the next record on logical unit 20,

```
CALL HWRITE('ALL ',20,0)
```

Read back only one histogram having identifier 'XYZ1' from record 3 on logical unit 21, and add the contents to that in HCOM if 'XYZ1' is already defined in HCOM,

```
CALL HREAD('XYZ1',21,3)
```

If 'XYZ1' is to be replaced by that from the file, then the call is

```
CALL HREAD('XYZ1',21,3,'REPL')
```

4. HREAD has provisions for inquiring about record numbers for specific nodes, positioning to a desired record number, or skipping over records. For example, the record number on logical unit 21 for the first node having an identifier 'ABCD' is returned into IREC by

```
CALL HREAD('ABCD',21,IREC,'GET ')
```

Section 3

FUNCTIONAL DESCRIPTION OF DISPLAY PACKAGE

3.1 OVERVIEW

A set of routines (DUT1 and DUT2) are available in HANDYPAK for generating displays of graphs on a line printer or on a variety of graphic devices. These routines are called by the histogram output routines (HOUT, HSLICE) but may be called directly for other purposes also. The default option settings, stored in the common block /DPMODE/, control the various characteristics of the displays, such as whether a frame is to be drawn, what kind of scales are to be used, are error bars to be drawn, etc. These option settings may be changed by the subroutine DOPT. Most options can be set any time prior to the output calls. However, the options for setting the screen size and (Versatec) line density must be made by DOPT prior to calling HOUT, HSLICE, or DPSLCT.

Given a set of coordinate pairs {x.,y.}, a graph of y. vs x. can be plotted by stepping the index i from 1 to N. Similarly, the set of coordinates {x.,y.,z..} can specify the surface of z.. vs x. and y. as i steps from 1 to NX and j steps from 1 to NY. The first case is called a 1-D display in Handypak, denoting the single index i, while the latter is called 2-D for the double index case. The routines DUT1 and DUT2 generate the 1-D and 2-D displays.

Smooth curves may be drawn for the data points if the SMOOTH option is set by DOPT. Analytical functions may also be drawn by supplying appropriate arguments to DUT1 or DUT2.

The set of coordinates for a graph may be supplied in the form of functions for DUT1 or DUT2, or by a set of arrays for DUT1A and DUT2A. By supplying multi-dimensioned arrays or multi-dimensioned functions, several graphs may be superimposed on the same plot by one call to DUT1 or DUT1A. Alternately, any number of calls to the graphic routines may be made in an overplot format to superimpose several graphs on a single plot on a graphic devices (but not on the line printer).

3.2 INITIALIZING GRAPHIC DEVICES

Display options and buffers used by the graphic routines have to be initialized before graphic output may be generated. The Unified Graphics System is used to generate the graphic output. There are three ways of doing the initialization, either by calling DINIT or by calling DPINIT, or neither. By using DINIT, the user supplies the address of the element buffer to be used by the graphic element routines within handypak. This method allows the user to have more than one element buffer. If DINIT is not explicitly called by the user, then DPINIT will be used instead, in which case the single buffer in the /SCPBUF/ common block is used. It has a default size of 500 words, and may be increased in size by declaring a larger common block /SCPBUF/ and then calling DPINIT. For example,

```
COMMON / SCPBUF / NBUF,BUF(800)

CALL DPINIT(800)
```

defines the graphic element buffer to be 800 words. If the user does not call DINIT or DPINIT explicitly, then DPINIT is called implicitly by the first display routine (e.g. DUTl, HOUT) or by HINIT, whichever is called first.

The 500 word default is generally sufficient, except for certain interactive devices which can overflow the device output buffer defined within the unified graphics routines. If the element buffer overflows (during calls to UGTEXT, UGLINE, etc.) then UGXERR is called. The UGXERR routine in turn outputs the element buffer to the device to make room for more elements, but in this output process the device buffer may overflow. If this happens, UGXERR is called again, but this makes it a recursive call, which is not allowed. To avoid this possibility, the element buffer should be made large enough to hold the largest plot without overflowing. The devices with this problem are ones with a hardware buffer, such as 'IBM2250' and 'SLACXSS'.

Graphic devices must first be opened in the Unified Graphics system before they are used. DPSLCT opens a graphic device the first time it is called for that device. Also, HOUT calls DPSLCT, so if HOUT is called with a graphic device, that device is assured to be opened.

3.3 BASIC DISPLAY DRIVERS

There are only two display driver routines (DUT1 and DUT2) within HANDYPAK for generating all the 1-D and 2-D plots on either the line printer or on graphic devices. Other output routines, such as DUT1A, DUT2A, HOUT, HEXEC etc., are interface routines that in turn call DUT1 or DUT2. The graph to be drawn is specified by a set of function coordinates supplied in the arguments of the calling sequence, and the format of the plot is controlled by a set of default parameters or options. The subroutine DOPT may be used to fetch or modify the settings of these options.

A 1-D display is generated by

```
CALL DUT1(FX,FY,FYE,N,TITLE,SMAN)
```

where FX(I), FY(I), FYE(I) are functions which provide the X and Y coordinates, and the Y and/or X errors for the N points (I=1,N) to be plotted. TITLE and SMAN supply the title string and the manual scale factors as described in sections 3.4 and 3.5 respectively. SMAN is used by DUT1 only if the manual scales option ('AUTO', 'XAUTO', or 'YAUTO') was selected by DOPT. Error values for the lower and upper bars may be equal (the default) or they may be unequal. In the latter case, FYE must be supplied with additional indices as described in DUT1, and the 'ERROR' option must be set by DOPT for asymmetric errors.

Similarly, a 2-D display is generated by

```
CALL DUT2(FX,FY,FZ,FZE,NX,NY,TITLE,SMAN)
```

where FX(I), FY(J), FZ(I,J), and FZE(I,J) provide the X, Y, and Z coordinates, and the Z error for the NX points along the X axis (I=1,NX), and the NY points along the Y axis (J=1,NY). TITLE and SMAN are the title and manual scale factors as in DUT1. A 2-D line printer plot consists of an array of numbers, with one number per (I,J) bin. On graphic devices, the 2-D plot is drawn as an isometric projection.

Additional arguments may be supplied for both DUT1 and DUT2 to provide analytic functions and/or contours that are to be drawn with the data points. These arguments are described in DUT1 and DUT2 in section 4.

Smooth curves may be drawn for the data points in 1-D plots if the 'SMOOTH' or 'XSMOOTH' options are selected by DOPT prior to the call to DUT1.

DUT2 will make a sliced projection onto a 1-D plot if the SLICE option is set in DOPT.

3.4 TITLE FORMAT

The display routines DUT1 and DUT2, as well as others such as HDEF, HOPTN, DUT1A, etc., have an argument for specifying a title string that (eventually) is to be drawn on the display. This TITLE variable is declared as a LOGICAL*1 array of length 256 in HANDYPAK. If a user supplies a title of more than 256 characters, then only the first 256 are used. If less than 256 are specified, then an '@' character should be added at the end to indicate the end of the string.

Substrings for the x-, y-, and z-axis may be specified by using semi-colon (';') separators. The general format for the TITLE is

'Main title; xstring; ystring; zstring@'

where blanks are treated as significant characters. Missing substrings need the corresponding ';' if another substring follows, e.g.

'Main string;;ystring@'

specifies only the main title and the y subtitle. Trailing semi-colons may be omitted, e.g.

'main title@'

The title is plotted on top of the plot, the substrings are plotted along the corresponding axes, where possible, as shown in figures 1.1 and 1.2.

On line printer plots, subtitles are marked with the directional arrow made from a 'V' or '--->' characters. If a line printer supports over-printing ('+' in column 1), then a better downward arrow is made if the OVPRT option in DOPT is set true.

Duplex characters in title strings

The Unified Graphics system has a duplex character format which allows special mathematical symbols, plotting symbols, and a variety of character fonts to be plotted on graphic devices. In this mode, two input characters define one plotted character. Usually, the second character of the pair defines the font for the first. Such duplex characters may also be specified in Handypak's title strings. Note- the CHFMT option in DOPT must be set to 1 or 2 to select the duplex character set. Also note that some of the font control characters differ between the two versions of Unified Graphics. There are two ways to specify duplex characters in title substrings.

1) '$' format

If the first character in a substring is '$', then all following characters up to but not including the terminating ';' or '@' are paired, where each pair specifies one plot character.

Example, the string:

'$S ALMLPLLLEL T ILTLLLEL@'

produces

'Sample Title'

2) '{' format

If the first character in a substring is '{', then this specifies that a font control character follows, and subsequent characters in the substring are to be output in that font until either a new '{' opens another font or a terminating ';' or '@' is found. A '}' control character closes a font and restores the previous font. The above example for 'Sample Title@' is entered as:

'{ S{LAMPLE } T{LITLE}@'

where the ' ' and 'L' font control characters specify upper and lower case respectively. There are a few special font control characters which do subscripts, superscripts, and change size of characters, as follows:

'{^CCC... }'	plots CCC... in superscript mode
'{_CCC... }'	plots CCC... in subscript mode
'{>CCC... }'	increases size of CCC...
'{<CCC... }'	decreases size of CCC...
'{+CCC... }'	saves location, plots CCC..., and restores location for next character.

Nesting of '{' control fonts may be done to any level, and trailing '}' control characters may be omitted before a ';' or '@' string termination.

Figure 5.2 shows an example of this style of title string.

3.5 SCALE FACTORS

Scale factors for each of the axes can be computed automatically from the data being plotted, or they can be supplied by the user. There are two types of auto-scale factors, namely full tic and full data. Full tic scale factors generate tic marks at each end of the axis, and the data will generally not extend over the full range of the axis. Full data scale factors are such that the data completely spans the range of the axis, and the tic marks fall as needed (not necessarily having tics at the ends). The default is automatic scaling, with full tic for the ordinate axis, and full data for the abscissa axis.

Manual scales are selected by one or more of the options 'XAUTO', 'YAUTO', 'AUTO', or 'G2ANGS' to the DOPT routine. Once selected, the manual scales remain in effect for all subsequent displays made by the DUT's (or HOUT, HSLICE, etc.) until reset by new calls to DOPT. The plotting routines use the scale factors for each axis, as follows:

Word	Contents (for linear scales)	
1	VMIN	- value at origin of axis (default 0.)
2	DV	- value between major tic marks (default 1.0)
3	VTICS	- number of major tic intervals (default 8.)
4	VSUBS	- number of minor tic intervals (default 0.)

Word	Contents (for logarithmic scales)	
1	VMIN	- value of exponent at origin.
2	DV	- exponent increment between decades (=1.0)
3	VTICS	- number of decades
4	VSUBS	- number of minor tics within a decade (0. to 9.)

Full tic mode has a VMIN that is an integral number of DV units and VTICS is an integral (REAL) number with no fraction. In full data mode, both VMIN or VTICS may contain fractional parts.

If manual scales are specified for an axis, and the corresponding DV is zero, then auto scales will be used instead for that axis. Major tics have labels plotted with the tics, while minor tics do not. A 1-D display needs an 8-word array to specify manual scale factors for the X and Y axes, while a 2-D display needs 14 words for the X, Y, and Z axes, as follows:

```
1-D -  XMIN,DX,XTICS,XSUBS,YMIN,DY,YTICS,YSUBS
2-D -  XMIN,DX,XTICS,XSUBS,YMIN,DY,YTICS,YSUBS,ZMIN,DZ,
       ZTICS,ZSUBS,ROT,ELV
```

where the last two words for the 2-D case are the rotation and elevation angles (in degrees) to be used for isometric plots on graphic devices. The default values are 40. degrees for both of these angles.

Neither manual scales nor log scales are allowed for scatter-plots on graphic devices.

If manual scale factors are to be used for an HOUT call, then the scale factors must first be stored in the histogram in /HCOM/ by issuing a call to HOPTN with the 'HMSCAL' option, and also one or more of the manual scale options must be set by DOPT. Subsequently, when HOUT calls the DUT routine, the scales saved in HCOM are used as the argument in the DUT call.

Examples:

Use manual scales in a call to DUT1,

```
REAL A(8)
DATA A/0., 1., 5., 4.,  10., 2., 4., 0./
EXTERNAL FX,FY,FYE

CALL DOPT('AUTO',0)
CALL DUT1(FX,FY,FYE,20,'DUT1 WITH MANUAL SCALES@',A)
END
```

Use the same manual scales as above for a 1-D histogram display having ID=5,

```
(same data statement for A as above)
CALL HOPTN('HMSCAL',A,5)
CALL DOPT('AUTO',0)
CALL HOUT(5,'PRINTER')
```

3.6 OVERPLOTTING

Overplotting provides the means for plotting more than one graph on the same plot. There are two ways of doing the overplotting in HANDYPAK.

1. The multi-plot mode: NF graphs (NF > 0) each having NP points can be plotted by issuing a single call to the DUT1 routine. The argument N should contain the value

   N = NP + NF*1000

   and the functions FX, FY, and FYE should be doubly indexed. That is, FX(I,J), FY(I,J), and FYE(I,J) supply the X-coordinate, the Y value and the Y error respectively for the I'th point and the J'th graph. Then the call

   CALL DUT1(FX,FY,FYE,N,TITLE)

   will produce a multi-plot output on either a line printer or a graphic device, and by default, the points will carry the labels '*' for the first graph, 'A' for the second, 'B' for the third, etc. One common set of automatic scale factors are computed so that all graphs fit in the plot.

2. Over-plot mode: Here, distinct calls can be made to HOUT or the DUT routines, where each call adds a graph to the plot but does not eject the plot. This mode is possible only on graphic devices, not on the line printer. The call

   CALL DOPT('OVER',.TRUE.)

   turns on the overplot mode. After this call, one or more graphs may be generated by HOUT or the DUT routines. Each graph may change any of the options, such as 'FRAME', 'ERRORS', etc. described in DOPT. Scale factors are not re-computed between plots (by DUT1 or DUT2) while in an overplot mode unless the 'RESCALE' option in DOPT is set true. After all the overplot calls have been done, then the call

   CALL DOPT('OVER',.FALSE.)

   turns off this mode and completes (ejects) the plot.

Example:

```
CALL DOPT('OVER',.TRUE.)      !turn on overplot mode
CALL HOUT(1)                  !plot 1
CALL DOPT('FRAME',0)          !no frame for next plot
CALL HOUT(2)                  !plot 2 on top of 1
CALL DOPT('OVER',.FALSE.)     !complete and eject page.
```

3.7 WINDOWING

The subroutine DPWDOW controls windowing on graphic devices (line printer plots can not do windowing). The number of windows for the horizontal and vertical axes may be specified, as well as the window number for the next plot.

Windows are numbered from left to right along the X axis, and from top to bottom in Y, with window 1 being the top left most plot on the page, for example:

```
      Y

     -----------------
    | window | window |
    |   1    |   2    |
    |--------|--------|     This is the default (2 by 2)
    | window | window |
    |   3    |   4    |
     -----------------    ---> X
```

Once windowing is turned on, then subsequent calls to HOUT, HSLICE, DUT1, or DUT2 will automatically generate plots to the next window in sequence. When the last window is filled, the page is completed and ejected, and the next plot goes to a new page. A call to HOUT('ALL ') will automatically interlace 4 plots per page until all the plots are done. It is also possible to overplot while in window mode. In this case the overplotting is done to one window until DOPT('OVER',.FALSE.) is done. The next plot goes to the next window in sequence. While in overplot mode, new calls to DPWDOW may be made to redefine the next window on the current page.

Examples:

CALL DPWDOW(1)	Turns windowing ON. (2 by 2 per page).
CALL DPWDOW(0)	Turns windowing OFF.
CALL DPWDOW(1,3,2)	Do windowing, 3 along X and 2 in Y)
CALL DPWDOW(1,3,2,4)	As above, but next plot goes to window 4.

3.8 INTERACTIVE VS BATCH MODE

The DUT routines are capable of generating graphic displays in either a batch environment or an interactive environment. In batch mode, the graph is made first and then the plot is ejected (by calling UGPICT). This assures that the last plot is complete when the job stops. For an interactive device, the reverse order is done - the screen is first cleared by UGPICT and then the graph is generated. This leaves the display up for viewing until the next one is made. These modes can be set by the INTER argument in DPSLCT.

INTER	result
0	plot first, then eject/erase screen (batch mode)
1	eject/erase screen, then plot. Wait for carriage return, if device has a keyboard and the WAIT option (in DOPT) is set to 1.
2	no eject/erase/wait is done.

The default case is 0 (batch mode).

For the last mode (2), Handypak only issues graphic primitives for drawing lines, points, or text. It is up to the user to initialize, erase, or eject the page.

Section 4

PROGRAM DESCRIPTIONS

This section describes each routine that may be called by the user. Internal HANDYPAK routines are not described.

The routines are listed in alphabetical order.

4. PROGRAM DESCRIPTION

4.1 DINIT

Initialize display parameters and define the buffer to be used by the graphic routines.

```
+------------------------------------------+
|                                          |
|    INTEGER BUF(NNN)                      |
|    CALL DINIT(BUF,NNN)                   |
|                                          |
+------------------------------------------+
```

where NNN is the size of BUF in words.

Note - if DINIT is not explicitly called, then DPINIT is called internally and the buffer in the /SCPBUF/ common block is used instead. If DINIT is called, a dummy version of DPINIT is loaded.

4.2 DMGRUT, DMSCAT

A saving in core storage can be made if only the line printer plots are to be used in a job. The subprograms DMGRUT and DMSCAT are provided for this. By including DMGRUT in the LOAD or LINK step, dummy routines are loaded instead of the Handy-pak's graphic routines, and the Unified Graphics routines will not be referenced by Handypak. Similarly, DMSCAT loads in dummy routines for the scatterplot references. There are no arguments to these routines.

4.3 DOPT

All the options which control the display generation by DUT1 or DUT2 are stored in the /DPMODE/ common block. The routine DOPT provides a convenient interface for changing the options without referencing DPMODE directly.

```
+-------------------------------------------+
|                                           |
|     CALL DOPT ( OPT,PARM1 [,PARM2])       |
|     CALL DOPTC(COPT,PARM1 [,PARM2])       |
|                                           |
+-------------------------------------------+
```

where OPT is a 4-character HOLLERITH word containing an option string, and PARM1 and PARM2 are parameters to be saved in DPMODE for that option. COPT is a CHARACTER argument of length 4 supplying the option string.

It is also possible to retrieve the values for the options, or to reset them to their original (default) values. If the option string is preceded by 'GET,option' then PARM1 [and PARM2] receive the value for that option. If the option string is preceded by 'RES,option' then the option value will be reset to its original value (i.e. its value at the time DPINIT was first called). The call,

```
CALL DOPT('RES,ALL ')
```

will reset all options to their original values.

The possible option strings for OPT are listed below. The variable type for PARM (INTEGER, REAL, or LOGICAL) is the same as that shown for the default column. Numbers with decimal points specify that PARM is REAL, otherwise INTEGER. Logical is specified by T and F for .TRUE. and .FALSE. respectively. The column labeled 'WHO' shows which plots use the option, where L1, L2, G1, and G2 refer to the 1-D and 2-D line printer, 1-D and 2-D graphic plots respectively, and 11 means both L1 and G1, and 22 means both L2 and G2.

A '*' at the end of the WHO field signifies that this option may also be called by HOPTN. In this case, the option value in PARM1 is saved with the histogram, and will override the default in /DPMODE/ when the histogram is output by HOUT.

OPT	DEF-AULT	WHO		Description
				AXES, FRAME, and SCALES
FRAME	T	G1 G2	*	Draw frame and axes.
AUTO	1	11 22		Auto scales for all axes.
XAUTO	2	G1 G2	*	Auto scales for X axis.
YAUTO	1	11 G2	*	Auto scales for Y axis (1-D).
Y2AUTO	2	G2	*	Auto scales for Y axis (2-D).
ZAUTO	1	L2 G2	*	Auto scales for Z axis (2-D). where: 0 - no auto (user supplied) 1 - auto scales, full tic mode 2 - auto scales, full data mode (ZAUTO and YAUTO are equivalent)
XLOG	F	G1 G2	*	Log scales for X axis (1-D and 2-D).
YLOG	F	11 G2	*	Log scales for Y axis (1-D).
Y2LOG	F	G2	*	Log scales for Y axis (2-D).
ZLOG	F	L2 G2	*	Log scales for Z axis (2-D). (ZLOG and YLOG are equivalent).
XTIC	0	G1	*	Number of major X tic intervals (1-D).
YTIC	0	11	*	Number of major Y tic intervals (1-D).
X2TIC	5	G2	*	Number of major X tic intervals (2-D).
Y2TIC	5	G2	*	Number of major Y tic intervals (2-D).
ZTIC	5	G2	*	Number of major Z tic intervals (2-D). where: 0 means use an internal (best) value.
XSUBTIC	0	G1	*	Number of X sub-tic intervals (1-D).
YSUBTIC	0	11	*	Number of Y sub-tic intervals (1-D).
X2SUB	0	G2	*	Number of X sub-tic intervals (2-D).
Y2SUB	0	G2	*	Number of Y sub-tic intervals (2-D).
ZSUBTIC	1	G2	*	Number of Z sub-tic intervals (2-D). where: 0 means use an internal (best) value.
XZERO	F	G1 G2	*	Force a zero on X axis when auto-scaling X.
YZERO	F	11 G2	*	Force a zero on Y axis when auto-scaling Y.
Y2ZERO	F	G2	*	Force a zero on Y axis when auto-scaling Y.
ZZERO	F	G2	*	Force a zero on Z axis when auto-scaling Z. (ZZERO and YZERO are equivalent)

```
OMIT     T    11 22   omit edge bins in both X and Y, or
XOMIT    T    11 22 * omit edge X bins, or
YOMIT    T    22    * omit edge Y bins,
                      will omit first and last bins (in X for 1-D,
                      X and/or Y in 2-D) when auto-scaling, and
                      when calculating statistics from bin values.
                      These values are automatically set by HOUT
                      according to value of HOMIT.

TICDIR   F    G1    * Tic direction - tic marks drawn inward if
                      PARM is true.

MARGIN 0.1, G1 G2     Margin sizes in 1-D graphic plot. PARM1
                      must be a 4-word REAL array containing the
                      margin spacing for the four sides, in the
                      following order (defaults shown in parenthe-
                      sis):
                        PARM1(1)  XLO or left side (0.20)
                        PARM1(2)  YLO or bottom side (0.15)
                        PARM1(3)  XHI or right side (-0.10)
                        PARM1(4)  YHI or top side (-0.15)

                        If any margin is 0.0, then no tics, labels
                      or sub-title is drawn for that side on
                      graphic plots.

LMARG    .20 G1     * The left margin size is in PARM1.
RMARG   -.10 G1     * The right margin size is in PARM1.
BMARG    .15 G1     * The bottom margin size is in PARM1.
TMARG   -.15 G1     * The top margin size is in PARM1.

NUMFMT   1    11 22 * Number format used in labels and lists,
                        0 - no decimals, only integers
                        1 - decimals used where needed, in
                            engineering notation.
```

GRAPH FORMAT

ERROR 1 11 G2 * Controls the type of error bars to be used, i.e. symmetric or asymmetric in X and/or Y. (see DUT1 for specifying error values).

PARM1	X error	Y error	
0	-	-	(none)
1	-	sym	(+-Y err)
2	-	asym	
4	sym	-	(+-X err)
5	sym	sym	
6	sym	asym	
8	asym	-	
9	asym	sym	
10	asym	asym	

(Note- default for EFF or ASM histograms is 2, and can only be changed by HOPTN)

MARKER 0 G1 Marker number used in 1-D graphic plots.
0 - use character instead (see CHAR).
1 - 1'st UG marker (plus)
2 - 2'nd UG marker (X)
3 - 3'rd UG marker (diamond)
4-10 for box, fancy diamond, etc.
The default is 0.

CHAR '* ' 11 * Character used in 1-D plots (if MARKER is 0). May also specify the color, error, and line mode in the 2'nd, 3'rd and 4'th character position. If specified, the latter take precedence over COLOR, ERROR and LINE option values, for example,
'*210' means CHAR='*', COLOR=2, ERROR=1, LINE=0
PARM2 specifies the graph number (1 to 10) and if not supplied, graph 1 is assumed.
Note - 4 characters must be supplied.
A blank character (' ') gets plotted as '*' on the line printer and a point on graphic plot. If CHAR=0, then no character is plotted.

LINE 0 G1 * controls line drawing between data points,
0 - no lines, only points
1 - connect points with lines
2 - draw histogram plot

COLOR	0	G1		Specifies color of the line, point, character, or error bar for the data plot. Axes and labels always white. PARM color 0 - white 1 - white 2 - red 3 - green 4 - blue 5 - yellow 6 - magenta 7 - cyan
LOWIN	F	11 22	*	Specifies whether the input function(s) to DUT1 or DUT2 give coordinate values at the low edge of histogram bins (TRUE) or not (FALSE). HOUT and HSLICE automatically set LOWIN to TRUE before calling the DUT routines, and reset it to the original (entry) value on exit. If LOWIN is true, then the histogram boundary lines are drawn at the input coordinate values supplied to DUT1, otherwise they will be shifted down by half a bin.
CENTER	T	11 22	*	Specifies whether the location of the point (or character or error bar) is to be shifted upward by half a bin or not. This option also depends on the setting of LOWIN. If LOWIN is false, the CENTER option is ignored - the points are plotted as given. If LOWIN is true, then the points are (are not) shifted upward by half a bin if CENTER is true (false).
2DLINE	6	G2	*	Specifies type of isometric plot, 0 - lines along X axis 1 - lines along Y axis. 2 - 2D mesh plot 4 - hists along X axis. 5 - hists along Y axis 6 - 2D histogram plot (hidden lines are controlled by HIDE option)
HIDE	1	G2		Specify hidden line removal, 1 - remove hidden lines (default), 0 - display all lines, hidden or not.
G2ANGS	0	G2		Specify which isometric angles to use, 0 - use 'ROTANG' and 'ELVANG' in DOPT, 1 - use angles supplied by manual scales.
ROTANG	40.	G2		Default isometric rotation angle (deg.).

ELVANG	40.	G2	Default isometric elevation angle (deg.).
TISIZE	1	G1 G2	Character size for titles and labels. 0 = small .015" (normal text size) 1 = large .025" - default >1 = .015+.010*TISIZE. use 3 or 4 for publications or slides.
SYMSIZE	-1	G1 G2	Size of symbols or markers. -1 = scale with TISIZE. 0 = small .015" (normal text size) 1 = large .025" - default >1 = .015+.010*SYMSIZE. use 2 or 3 for publications or slides.
CHFMT	0	G1 G2	Character format of titles and labels in graphics plots. 0 = basic character set, using hardware generator when possible. 1 = basic character set, but no hardware generated characters. 2 = fancy (duplex) character set.
LINDEN	4	G1 G2	Line density on graphic devices such as the Versatec or Imagen. PARM may range from 1 to 5, (faint to dark). However, if LINDEN is 0 when UGOPEN is called, lines will be faint and cannot be changed later.
PNTDEN	4	G2	Point density in scatter plots on graphic devices such as Versatec or Imagen. PARM can range from 1 to 5 (faint to dark).
PNTSYM	0	G2	* Symbol used in graphic scatter plots, 0 - point 1 - 'x' 2 - '+' 3 to 10 are 'diamond', 'square', and more complicated variations of these.

GLOBAL CONTROL OPTIONS

OUNIT	6		Unit number for line printer output.
EUNIT	6		Unit number for error messages.
DEVICE			Name and index for a device in the DEVTYP table. PARM1 is the name (8 characters) and PARM2 is the index value (3 to 20). Once a device is entered in the DEVTYP table, then DPSLCT may be used to select that device.
IDEV	2		Device index of active device. (Useful for interoggating current device).
EJECT	T	L1 L2 *	Eject a new page for each plot by DUT1 or DUT2.
HEADER	' '	L1 L2	PARM is a hollerith string of up to 96 characters supplying a global header line to be printed subsequently with each new plot on the line printer, or whenever DWRHDR is called. If less than 96 characters are supplied, the last character should be an '@'. The header is stored in 24 words in the common block /DPHEAD/.
ITRACE	1	11 22	Controls whether a traceback is to be generated when certain errors (illegal arguments) are detected. 0 - don't, 1 - do provide trace
WAIT	1	G1 G2	Controls whether a 'wait' is to be done after a plot on a graphic device having a keyboard. 0 - don't, 1 - do wait for a CR.
ESCT		G1 G2	Sets the escape string needed to flip a graphic terminal into text mode after a plot is done. The string may contain control characters, or it may be in Z (or HEX) format, e.g. DOPT('ESCT','Z277FF087',9) sets the text escape string to HEX(277FF087).
ESCG		G1 G2	Sets the escape string needed to flip a graphic terminal into graphic mode before a plot is started.

DISPLAY SIZE (GRAPHIC DEVICES)

FULSCR	T	G1 G2		If true, then the full (rectangular) screen will be used on devices that have rectangular surfaces, otherwise a square screen is used.
SIZE	1.,1.	G1 G2		PARM1 and PARM2 are size (fraction) of display in X and Y coordinates, respectively,
XSIZE	1.	G1 G2	*	Size (fraction) of display in X coordinate,
YSIZE	1.	G1 G2	*	Size (fraction) of display in Y coordinate, Size values must be between 0.0 and 1.0, a value of 1.0 means the total screen size (as set by FULSCR above) is to be used.
OFFSET	0.,0.	G1 G2		PARM1 and PARM2 are offset (fraction) of display in X and Y coordinates, respectively,
XOFF	0.	G1 G2	*	Offset (fraction) of display in X,
YOFF	0.	G1 G2	*	Offset (fraction) of display in Y, Offset values must be between 0.0 and 1.0, although 1.0 would mean the whole plot is off-screen. Full size plots have an offset of (0.,0.) while an upper left quadrant plot would have an offset of (0.,0.5), and a SIZE of (0.5,0.5).
QUADRANT	0	G1 G2		Quadrant in which the plot is generated. PARM1 is a character string which specifies the quadrant. Two characters are needed to specify a quadrant, and one character is needed to specify two quadrants (half screen). These characters may be any valid combination of 'L', 'R', 'B', or 'T' for the left, right, bottom, or top side respectively. The order is not important, 'TL' gives same result as 'LT'. If an illegal character is given, the default (full size) is used. The values of PARM1 and the resultant regions are as follows:

PARM1	region
'TL'	top left quadrant
'TR'	top right quadrant
'BL'	bottom left quadrant
'BR'	bottom right quadrant
'T '	top half
'B '	bottom half
'L '	left half
'R '	right half
0	full screen (the default)

CHSCALE	1	G1 G2		Scaling mode for character size when XSIZE is less than 1.0, 0 - use fixed character size (fig. 5.25a) 1 - use character size that scales with XSIZE and YSIZE (fig. 5.25b).

OVERPLOTTING

OVER	F	G1 G2	*	Over plot mode - the DUT's don't eject or erase screen while OVER is true. The plot is completed when OVER is set false by DOPT.
RESCAL	F	G1 G2	*	Rescale between plots, if auto-scales are being used while in overplot mode.

LINE PRINTER OPTIONS

LCOLS	133	L1 L2	*	Total number of columns used in plot. Max is 256, if the printer supports it.
LLINE	60	L2	*	Number of lines/page in line printer plot.
HIST	F	L1	*	Histogram plot for line printer (fills plot with XXX's).
GRAT	F	L1	*	Put graticule every 10 columns in 1-D line printer plot.
OVPRT	F	L1 L2		Does overprinting ('+' in column 1) when outputting subtitles, etc. If false, some overprinting is omitted.
L2SYM	10	L2	*	Number of symbols (contours) in 2-D line printer plot when only one or two columns per bin are used. Plotting symbols 'A', 'B', ... are used for plotting values 11, 12, ... The maximum L2SYM is 36 (symbol 'Z').
L2COL	4	L2	*	Number of columns used per bin of a 2-D plot when L2CAUT is false.
L2CAUT	T	L2	*	Compute number of columns per bin automatically, by using as many as necessary to represent all values, but yet fit the plot on one page. If there are more than 50 Y bins, then 1 column per bin is used (if more than 100 Y bins, then continuation pages are used).

L2ZAUT	T	L2	* Compute scale factor automatically for Z coordinate in 2-D line printer plot.
LSUMS	T	L1 L2	* Print out YCUM (1-D) or TOTALS (2-D) in line printer plot.

IDENTIFIER

IDFONT	T	G1 G2	If true (false), then the ID font is (not) plotted on graphic plots.
A4ID	' '	L1 L2	Character string of identifier, to be printed or plotted by the DUT1 or DUT2 display routines (4 characters).
ID	0	L1 L2	Value of identifier in numeric or hollerith format (1 word).
JBFONT	3	G1 G2	Controls plotting of job name/date and 'HANDYPAK' logo in each picture. 0 = neither plotted 1 = plot job name/date 2 = plot logo 3 = plot both job name/date and logo.
NAME	' '	11 22	Sets job name (8 hollerith characters in PARM1).
DATE	' '	11 22	Sets job date (8 hollerith characters in PARM1).

NUMBER OF CALLS & OVERFLOWS

NCALLS	F	11 22	* If TRUE, then the number of calls and the number of overflows and underflows in HCUM1/2 are displayed by the DUT routines.

SLICING

SLICE	F	L2 G2	*	Do slicing in DUT2. See following options also.
SLDIR	0	L2 G2	*	Direction of slice is along X (0) or Y (1). Slice width is along Y (0) or X (1).
SLBEG	1	L2 G2	*	Beginning bin number in width of slice.
SLWIDTH	0	L2 G2	*	Number of bins across width of slice. A 0 value indicates all bins beyond SLBEG.

SMOOTHING

SMOOTH	0	11 22	*	Controls whether a smooth curve is to be drawn with the data or not. 0 - no smoothing 1 - plot smooth curve using a gaussian kernel with a sigma given by value of XSMOOTH. If XSMOOTH is zero, then a sigma is automatically calculated from the (first) data interval multiplied by value of WSMOOTH (i.e. sigma = 2.5 bin widths by default).
XSMOOTH	0.	11 22		Sigma of gaussian kernel for the smoothing. If non zero, then smoothing will be done with this width. If zero, then no smoothing is done unless SMOOTH is true.
YSMOOTH	0.	22		Sigma in Y of gaussian kernel for a 2-D case. This value only used when slicing a 2-D histogram along Y direction and also doing the smoothing on the slice.
WSMOOTH	2.5	11 22	*	Default number of bins for the sigma of the gaussian kernel used in the smoothing.

4.4 DPCLOS

Close the active graphic device, or all devices.

```
+-------------------------------------------+
|                                           |
|           CALL DPCLOS[(ARG)]              |
|                                           |
+-------------------------------------------+
```

If no argument is given, or ARG is '   ', then the current graphic device is closed. If ARG is 'SCAT', then the scatter-plot file is closed. If ARG is 'ALL ', then all graphic devices and the scatterplot file are closed. DPCLOS should be called at the end of a job to properly terminate the last plot on some devices (e.g. Versatec).

4.5 DPINIT

Initialize display parameters and define the size of the graphic element buffer in /SCPBUF/.

```
+-------------------------------------------+
|                                           |
|    COMMON / SCPBUF/ NBUF,BUF(NB)          |
|    CALL DPINIT( [NB])                     |
|                                           |
+-------------------------------------------+
```

where NB is the size, in words, of the element buffer BUF. If NB is not supplied, or if neither DINIT nor DPINIT is explicitly called, then 500 words are used. See DINIT for an alternate way of supplying BUF.

4.6 DPSLCT

Selects the device to which all future output is to be directed. When a graphic device is selected for the first time, it is opened. DPSLCT sets the device number in IDEV (in /DPMODE/, initially 2 for the line printer). This default setting may be changed to a new device by

```
+------------------------------------------+
|                                          |
|       CALL DPSLCT(OPT [,INTER])          |
|                                          |
+------------------------------------------+
```

where OPT is either a numeric value in the range 1 to 20 to be saved in IDEV, or an 8-byte hollerith string which specifies the new device name. In the latter case, a search is made over the list of known devices (stored in DEVTYP in DPMODE) and IDEV is then set for that name. The second argument sets the INTER flag, as described in section 3.7. The default list of device names at SLAC are loaded by a block data, contained in subroutine DBLOAD, as shown by DEVTYP in Appendix B.

New device names may be entered at execution time by DOPT as follows:

```
        CALL DOPT('DEVICE',NAME,IDEV)
```

where name is an 8-character Hollerith variable, and IDEV is the index to DEVTYP with a value between 3 and 20. This change should be made before DPSLCT, HOUT, DUT1, or DUT2 is called. The name must be identical to that used by Unified Graphics for that device.

It is also possible to transmit additional information to UGOPEN by using a compound device name, made by adding a comma immediately after name and appending options after the comma (no blanks). A trailing blank or '@' must be supplied to terminate the string. For example

```
      CALL DPSLCT('IMGN300,ROTAXIS ',0)
```

will open the IMGN300 device with the ROTAXIS option.

The user may also call UGOPEN directly, instead of calling DPSLCT, provided this call is done prior to HOUT, DUT1, or DUT2, and the IDENT argument in UGOPEN has the value of IDEV corresponding to the named device.

4.7 DPWDOW

Sets up windowing to provide multiple plots per page on a graphic device.

```
+----------------------------------------------------+
|                                                    |
|        CALL DPWDOW(IC [,NWX,NWY [,IW ] ] )         |
|                                                    |
+----------------------------------------------------+
```

where:

IC may be 1 to turn windowing on, and 0 to turn it off. NWX and NWY specify the number of equally sized windows in the X and Y directions respectively, and IW gives the number of the next window to be used. See Windowing in Section 3 for the numbering scheme. The default values for the optional arguments NWX, NWY, and IW are 2, 2, and 1 respectively.

Once windowing is on, then all subsequent calls to DUT1, DUT2, HOUT, or HSLICE will automatically increment to the next window for each plot. When the last window is filled, the next plot starts on a new page at window number 1. When overplotting is turned on, window incrementing is suppressed, and plots go to the same window until overplotting is turned off. Windowing is done by setting the SIZE and OFFSET options (see DOPT).

4.8 DUT1, DUT1A

Basic routine for generating 1-D displays of data points, smooth curve, and/or analytic functions complete with frame, axes, and title, on a line printer or a graphic device.

```
+----------------------------------------------------+
|                                                    |
|   CALL DUT1(FX,FY,FYE,N,TITLE [,SCAL [,F [,NF]]])  |
|                                                    |
|                         or                         |
|                                                    |
|   CALL DUT1A(AX,AY,AYE,N,TITLE [,SCAL [,F [,NF]]]) |
|                                                    |
+----------------------------------------------------+
```

where:

FX, FY, and FYE are user supplied functions (declared EXTERNAL) which give the data point coordinates, and F is a user supplied function (declared EXTERNAL) which defines the analytic function(s).

FX(I) gives the X coordinate value for the I'th data point I=1,N. For histogram graphs, the upper edge of the N'th bin must also be supplied in FX(N+1).

FY(I) gives the Y coordinate value for the I'th data point

FYE(I) gives the error in Y and/or X for the I'th data point

N is the number of points (maximum 999). If N is zero, FX, FY, and FYE are not used.

TITLE supplies a title string as described in section 3.4.

SCAL supplies a set of manual scale factors (8 words) as described in section 3.5. SCAL is only used if one of the manual scales options in DOPT has been selected (XAUTO, YAUTO, or AUTO), or if F is supplied but there are no data points (N=0).

F(X) gives the Y ordinate value at the coordinate X for the analytic function when NF=1, or when NF is not supplied.

NF is the number of analytic functions to be drawn. If NF is greater than 1, then F is called by DUT1 with a second argument K (i.e. F(X,K) where K=1 to NF), which the user may use to determine which function value to return.

Multiple sets of data points, each with NP points, may also be plotted on either a line printer or graphic device. This is accomplished by specifying the number of sets (NS) in the high order digits of N, as

$$N = NS*1000 + NP$$

In this case the user's data functions are called with a second argument which the user may use to determine which data set to supply, namely FX(I,M), FY(I,M), and FYE(I,M) should provide the X, Y, and error value at point I and data set M, where I=1 to NP, and M=1 to NS. The default labels for the data points are '*' for the first set, 'A' for the second, 'B' for the third, etc. on both the line printer and graphic device. These may be changed by the 'CHAR' option in DOPT. If more than 1000 points are to be drawn, then the factor 1000 above may be increased by setting IMULT in /DPMODE/.

Error bars may be drawn that are equal for the upper and lower bars, or they may be unequal (asymmetric), and errors may be drawn along either the Y or X axes or both. The 'ERROR' option in DOPT controls which error bars are to be used. When other than the symmetric Y errors are to be drawn, then one or two additional arguments must be supplied in the FYE function, as follows:

```
     FYE(I,K,L)        when NS=0
     FYE(I,M,K,L)      when NS>0 (multi-data plot)
```

where L=1 for error in Y, L=2 for error in X, and K=1 for upper error, and K=2 for lower error. (symmetric errors use K=1 only).

The user may specify whether FX supplies the low edge value of a data bin or the central value, by setting the LOWIN option in DOPT prior to calling DUT1. The LOWIN option is used in several places. If LOWIN is TRUE (FALSE), then

1) the X boundary in graphic histogram plots is not (is) shifted down by half bin from FX,
2) the X data values in line printer plots are labeled 'LO EDGE' ('MIDBIN'),
3) in line printer plots, F(X) is calculated for X at the bin center, (low edge),
4) the statistical mean of X is calculated from the bin values FX+half a bin (FX) at each bin.

When a line printer plot is made with both data points with error sand an analytic function, then the chisquare is also printed out. If multiple data sets or multiple analytic functions are supplied, then the chisquare is given for the first data set and the first analytic function only.

The format of the display as well as the destination device is controlled by the settings of the options in the /DPMODE/ block. The settings of these options may be changed by the user prior to issuing the DUT calls. The default output device is the line printer initially, but it may be changed to a new default device by the DPSLCT (or HOUT) routine. See section 4.3 for a list of all the options.

DUT1A has the same functional description as DUT1, except the AX, AY, and AYE are REAL arrays rather than functions, with the same number of subscripts as FX, FY, and FYE. DUT1A sets up arguments and then calls DUT1 to perform the plot. Note that the array dimension must be the same as the number of points in the N2 argument.

4.9 DUT2, DUT2A

Basic routine for generating 2-D displays of data points and/or analytic functions, complete with frame, axes, and title on a line printer or a graphic device. These are represented as isometric projections on graphic devices, and as tabular arrays on a line printer output. Contour curves may also be drawn on the line printer (contours on graphic scatterplots are done by HOUT).

```
+-----------------------------------------------------+
|                                                     |
|     CALL DUT2(FX,FY,FZ,FZE,NX,NY,TITLE [,SCAL]      |
|                  [,F,NF [,C,NC]]] )                 |
|                                                     |
|                         or                          |
|                                                     |
|    CALL DUT2A(AX,AY,AZ,AZE,NX,NY,TITLE [,SCAL]      |
|                  [,F,NF [,C,NC]]] )                 |
|                                                     |
+-----------------------------------------------------+
```

where:

FX, FY, FZ, and FZE are user supplied functions (declared EXTERNAL) which give the data point coordinates, F is a user supplied function (declared EXTERNAL) which defines the analytic function(s), and C is a user defined function (declared EXTERNAL) which defines the contour curve(s).

FX(I)	gives the X coordinate value for the I'th data point I=1,NX. For histogram graphs, the upper edge of the last I bin must also be supplied in FX(NX+1).
FY(J)	gives the Y coordinate value for the J'th data point J=1,NY. For histogram graphs, the upper edge of the last J bin must also be supplied in FY(NY+1).
FZ(I,J)	gives the Z coordinate value for the I,J'th data point
FZE(I,J)	gives the error in Z for the I,J'th data point
NX	is the number of points in the X coordinate
NY	is the number of points in the Y coordinate
TITLE	supplies a title string as described in section 3.4.
SCAL	supplies a set of manual scale factors (14 words) as described in section 3.5. SCAL is only used if one of the manual scales options in DOPT has been selected (XAUTO, YAUTO, AUTO, G2ANGS).

F(X,Y) gives the Z ordinate value at the coordinates X,Y for the analytic function when NF=1, or when NF is not supplied.

NF is the number of analytic functions to be drawn. If NF is greater than 1, then F is called by DUT2 with a third argument K (i.e. F(X,Y,K) where K=1 to NF), which the user may use to determine which function value to return.

C(X,Y) gives the contour curve to be overprinted on line printer plots along the locus of points where C=0.

NC is the number of contours to be drawn. If NC is greater than 1, then C is called by DUT2 with a third argument K (i.e. C(X,Y,K) where K=1 to NC), which the user may use to determine which contour is being drawn.

Error bars and functions are not displayed by DUT2, since otherwise the plots would be too cluttered. These are used however in slices and in calculation of the chi square for the function. Equal upper and lower error values are the default, but if the 'ERROR' option in DOPT specifies asymmetric errors in Y, and a slice is being made, then an additional index K should be supplied in FZE, namely

FZE(I,J,K)

to provide the upper error if K=1 and lower if K=2.

The user may specify whether FX and FY supply the low edge values of a data bin, or the central value, by setting the LOWIN option in DOPT. This option performs similar results as described in DUT1, except that both the X and Y coordinates are affected.

When DUT2 is called with both data points with errors and an analytic function, then a chisquare is also calculated and printed out. If multiple analytic functions are specified, only the first is used in the chisquare. When a slice is made of a 2-D display which has an analytic function, then the curve of the slice of the analytic function is also made on the 1-D slice. The analytic function is evaluated at the coordinates given by FX and FY if the LOWIN option (in DOPT) is false, otherwise the mid-bin values are used by adding half a bin width to each of the FX and FY values.

The format of the display as well as the destination device is controlled by the settings of the options in the /DPMODE/ block. The settings of these options may be changed by the user prior to issuing the DUT calls. The default output device is the line printer initially, but it may be changed to a new default device by the DPSLCT (or HOUT) routine. See section 4.3 for a list of all the options.

DUT2A has the same functional description as DUT2, except the AX, AY, AZ, and AZE are REAL arrays rather than functions, with the same number of subscripts as FX, FY, FZ, and FZE. DUT2A sets up arguments and then calls DUT2. See the note in DUT1A.

4.10 DWRHDR

This routine is called by DUT1 or DUT2 at the beginning of each new line printer plot for (optionally) ejecting a new page and printing out the global header line. This routine may be also be called by the user for printing the same header line on a new page by

```
CALL DWRHDR(IUNIT,SKIP)
```

where

IUNIT is the logical unit number for the line printer

SKIP is a LOGICAL variable which is true (false) if a new page eject is (is not) to be made.

The contents of the header line can be changed by using the HEADER option in DOPT.

4.11 HBLOCK

Generates a user block and manipulates contents in such a block.

```
+----------------------------------------+
|                                        |
|   CALL HBLOCK(ID,IC [,N [,A [,I1]]] )  |
|                                        |
+----------------------------------------+
```

Where:

ID is the identifier of the block

IC is a control word specifying the operation to be done,

```
IC = 'DEF ' defines the block (HCLR wont clear it)
IC = 'DEF0' defines the block (HCLR will clear it)
IC = 'SET ' stores values into the block
IC = 'GET ' retrieves values from the block
IC = 'ADD ' adds to the contents of the block
IC = 'IADD' adds to the contents of the block
            in integer mode
IC = 'CLR ' clears the contents of the block
IC = 'CLE '    "     "     "     "   "    "
IC = 'GETN' returns size of block in N, and
            the title in A (if A is supplied).
```

N number of items being defined, or added, etc. If not supplied, then the number is taken from the value used in the DEF operation for this ID. N must be supplied in a 'DEF ' operation.

A is a variable or array which is either stored in the block if IC='SET ', filled from the block if IC='GET ', added to the block contents if IC='ADD ' or 'IADD'. In a DEF operation, A supplies the title string (20 words)

I1 is the first item in the block for the set, add, etc. If not supplied, the value 1 is used for I1.

The block is stored in /HCOM/ and is linked to all other histogram nodes just like a histogram. A block may be deleted by HDEL, and it may be written to secondary storage by HWRITE and read by HREAD. The structure of a block differs from a histogram in that it only has the node control area, a title, and the data block itself. Blocks defined as 'DEF0' may be cleared by HCLR. ('DEF ' blocks cannot be cleared by HCLR).

4.12 HCLR

Clear (zero) a histogram or a user block.

```
+------------------------------------------+
|                                          |
|              CALL HCLR(ID)               |
|                                          |
+------------------------------------------+
```

HCLR clears (zeros) the bin contents and statistics for the ID'th histogram, or all histograms if ID has the value 'ALL '. HCLR will also clear the contents of blocks that have been defined with 'DEF0' in HBLOCK.

4.13 HCOMB

The bin contents of one histogram may be combined (added, subtracted, multiplied, or divided) with the corresponding bin contents of a second histogram, and the results stored in a third histogram by:

```
+----------------------------------------------------+
|                                                    |
|     CALL HCOMB(C1,ID1,OPER,C2,ID2,ID3,TITLE)       |
|                                                    |
+----------------------------------------------------+
```

where

```
   C1     coefficient factor for first histogram
   ID1    ID of first histogram
   OPER   specifies operation:
            'PLUS'   add
            'MINUS'  subtract
            'MULT'   multiply
            'DIVIDE' divide
   C2     coefficient factor for second histogram
   ID2    ID of second histogram
   ID3    ID of third (resultant) histogram
   TITLE  title for the third histogram
```

Symbolically, the resultant histogram is

```
   ID3 = C1*ID1  OPER  C2*ID2
```

on a bin by bin basis. ID1 and ID2 must be histograms of the same dimensions. ID3 will be created by HCOMB, with the same number of bins and with an 'E*4' mode of storage. If ID3 already exists, HCOMB returns with an error message.

4.14 HCUM1, HCUM2

Accumulate a value into a histogram bin, a scatterplot file, and/or the histogram statistics,

```
+------------------------------------------+
|                                          |
|           CALL HCUM1(ID,XV,WT)           |
|                                          |
|                    or                    |
|                                          |
|          CALL HCUM2(ID,XV,YV,WT)         |
|                                          |
+------------------------------------------+
```

where:

ID is the histogram identifier.

XV is the X coordinate value to be binned.

YV is the Y coordinate value to be binned.

WT is the weight to be added to the bin, and/or statistics.

HCUM1 adds WT to the 1-D bin subtending the value XV and HCUM2 adds WT to the 2-D bin subtended by the values of XV and YV. Overflows are added to the first or last bin in the overflowing coordinate, unless the histogram has the 'HOMIT' option set true (see HOPTN), in which case overflows are stored separately, and are output by HOUT if the NCALLS option is true (see DOPT). Some modes of storage require that WT be a 2-word array (see section 2.4).

If the histogram was defined with a scatterplot, then HCUM2 also adds the XV,YV point into the scatterplot file, provided the point lies within the X and Y bounds of the histogram as defined by HDEF2. If WT is negative, such points are plotted as a '~' (NOT) sign in scatterplots on graphic devices.

If the histogram has a statistics block (STAT option in HOPTN), and the coordinate value(s) lie within the limits set by the SLIMIT option in HOPTN, then the values of XV or XV,YV are used to generate statistical summations. These summations are used in the output stage by HOUT or HSTAT for calculating the mean(s) and variance(s) of the distribution of the data points. If the statistics block is not present, these summations are not made (HOUT and HSTAT use the bin contents instead).

4.15 HDEF1, HDEF2

Add a new histogram node to /HCOM/.

```
+------------------------------------------------------+
|                                                      |
|       CALL HDEF1(ID,MS,NX,XL,XW,TITLE [,SUB])        |
|                          or                          |
|    CALL HDEF2(ID,MS,NX,NY,XL,YL,XW,YW,TITLE [,SUB])  |
|                                                      |
+------------------------------------------------------+
```

where:

ID	is the histogram identifier which may be an integer in the range 1 to 9999 inclusive, or a 4-character text string. However, 'ALL ' is not allowed for an ID.
MS	is the mode of storage, described in section 2.4.
NX	is the number of bins in X.
NY	is the number of bins in Y.
XL	is the low edge value of the first X bin.
YL	is the low edge value of the first Y bin.
XW	is the width of one X bin.
YW	is the width of one Y bin.
TITLE	is a title string, described in section 3.4.
SUB	(optional) is a subroutine name, declared EXTERNAL, which is used in interactive applications as described in section 3.7.

HDEF1 defines a 1-D histogram and HDEF2 defines a 2-D histogram. If a histogram with identifier ID already exists, or there is not enough room in /HCOM/, the new definition is not made and an error message is issued.

4.16 HDEL

Delete a histogram or a user block from /HCOM/.

```
+------------------------------------------+
|                                          |
|             CALL HDEL(ID)                |
|                                          |
+------------------------------------------+
```

HDEL deletes the ID'th node from HCOM, or all nodes if ID has the value 'ALL '. The contents of HCOM are scrunched when a node is deleted.

4.17 HGET

Various parameters of a histogram (as set by HDEF or HOPTN) may be obtained by:

```
+------------------------------------------+
|                                          |
|   CALL HGET(ID, [OPT [,ND,NB,MS,XL,XW    |
|          [,IH [,TITLE]]]])               |
|                                          |
+------------------------------------------+
```

If HGET is called with only one argument, or OPT has the value 'PRINT', then HGET prints the parameters of the ID'th histogram, or all histograms if ID has the value 'ALL '. If OPT has the value 'GET ', then HGET returns the parameters into the remaining arguments as follows:

```
ND      dimension (1 or 2) of the histogram
NB      INTEGER NB(2).  NB(1)  gives number of X bins (1-D or
        2-D) and NB(2) give number of Y bins (2-D)
MS      mode of storage ('L*1', I*2', etc.)
XL      REAL XL(2). XL(1) gives low edge X value (1-D or 2-D)
        and XL(2) give low edge Y value (2-D)
XW      REAL XW(2).  XW(1) gives X bin width (1-D or 2-D) and
        XW(2) gives Y bin width (2-D)
IH      gives the pointer for histogram bins within HCOM.
          M(IH+I)      gives I'th 'I*4' bin,
          H(2*IH+I)    gives I'th 'I*2' bin,
          R(IH+2*I-1)  gives I'th value for 'E*4'
          R(IH+2*I)    gives I'th (error)**2 for 'E*4'
            (see Appendix A)
TITLE   returns the title of the histogram (80 bytes max).
```

4.18 HINIT

This routine initializes the control area of /HCOM/. Any defined histograms are erased by this call. If the user never directly issues a call to HINIT, it will be called internally with default values by the first call to any H-routine.

```
+------------------------------------------------+
|                                                |
|   COMMON / HCOM / M(NH)                        |
|                                                |
|   CALL HINIT(NH [,NHASH [,NSCAT [,LUNIT ]]])   |
|                                                |
+------------------------------------------------+
```

where the arguments and their default values are:

```
NH     (2000) is the size of HCOM
NHASH   (137) is the size of the hash tables
NSCAT   (100) is the size of the scatterplot buffer
LUNIT    (19) is the logical unit for the scatterplot
              scratch file.
```

The value of NHASH should be at least 3 times the number of defined histograms in order to have an efficient histogram search (by HPNTR). The scatterplot buffer accumulates scatter points, and is dumped when full to a scratch file. If many scatter points are to be made, it may be necessary to increase NSCAT in order to decrease to I/O rate for the scatterplot scratch file.

4.19 HMAP

This routine prints out the location and linkage of nodes and values of nodal parameters in HCOM.

```
+-----------------------------------------+
|                                         |
|               CALL HMAP                 |
|                                         |
+-----------------------------------------+
```

There are no arguments for this routine.

4.20 HOPTN

HOPTN provides a means of modifying already defined histograms, and also sets defaults that will be used by subsequent (new) histogram definitions.

```
+-----------------------------------------+
|                                         |
|      CALL HOPTN( OPT,PARM [,ID])        |
|                  or                     |
|      CALL HOPTC(COPT,PARM [,ID])        |
|                                         |
+-----------------------------------------+
```

HOPTN updates (or fetches) parameter values for a histogram, or it can add or delete various sub-sections of a histogram. OPT (4-character Hollerith argument) specifies the option being changed. PARM is a parameter word or array (LOGICAL, INTEGER, or REAL), and ID is the histogram to be modified. If ID has the value 'ALL ' or the ID argument is not supplied, then all histograms are modified, including the default for the option. If ID=0, then only the default value is changed for the option, so that future histogram definitions will use the new default. HOPTC is identical to HOPTN, except that COPT is a CHARACTER*4 argument in place of OPT.

If the option string begins with a 'GET,' followed by the option, then HOPTN returns the value(s) to the user in PARM. e.g. 'GET,STAT' would return PARM true (false) if the statistics block were (not) defined for the histogram specified by ID.

The options generally fall into one of two groups, namely those that affect the histogram contents itself, and those that affect display options when the histogram is plotted. The former options are described in detail here, while the latter options are the same as (some of) those in DOPT and are described there instead. (in DOPT, those options that are also valid for HOPTN are marked by a '*').

If a display option has been set by HOPTN, then when HOUT (or HEXEC) is called later, this option value is used for the histogram instead of the default value stored in /DPMODE/. If the option so stored in a histogram is no longer desired there (i.e. the default in /DPMODE/ should be used), it may be reset by HOPTN by preceding the option with 'RES,'. For example, HOPTN('RES,YLOG',0,ID) would restore the YLOG option for the ID'th histogram. (The second argument is not used, but must be supplied).

Following is a list of histogram options that are recognized. The default column shows the value(s) as set by HDEF1 or HDEF2. The default value shown also indicates the variable type

(INTEGER, REAL, or LOGICAL) for PARM. Numbers with decimals are REAL, otherwise INTEGER, and T and F are LOGICAL true and false respectively. A '*' in the default field indicates that the default may be changed for that option.

OPT	DEF-AULT	PARM	Description
STAT	F	*	if PARM is TRUE (FALSE), then a statistics block is defined (deleted) in the node.
SLIMIT	--	*	PARM is a REAL array specifying the limits within which statistical sums are to be made. PARM must be 2 words for 1-D histograms, and 4 words for 2-D. Note that calls to HDEF1 or HDEF2, or a call to HOPTN('STAT',...), will set these limits to default values which are the range of the histogram bins. PARM(1) lowest X for 1-D or 2-D PARM(2) highest X for 1-D or 2-D PARM(3) lowest Y for 2-D PARM(4) highest Y for 2-D Note - PARM must be 4 words when ID is 'ALL ' or 0, or when OPT is 'GET,SLIMIT'.
SOMIT	F	*	if PARM is TRUE, then omit overflows from the statistical sums for values of X or Y beyond the limits given by 'SLIMIT'.
SOUT	F	*	if PARM is TRUE (FALSE), then future calls to HOUT will (not) output statistics. If statistical summations are available (i.e. 'STAT' was true), then these sums are used to calculate the means and variances, otherwise the histogram bin contents are used instead. The default is false in HDEF, but this default is changed to the value supplied in CALL HOPTN('STAT',value,'ALL ').
HBINS	T	*	if PARM is TRUE (FALSE), then the histogram bins are defined (deleted) in the node. The scatter-plot or statistics remain unchanged however.
HOMIT	F	*	if PARM is FALSE, then overflows are added to the first or last (edge) bin of the histogram, otherwise overflows are omitted from the edge bins.

OPT	DEF-AULT		PARM Description
HLOW	--		PARM is a one- or two-word REAL array containing the low edge bin value XL or XL,YL for 1-D or 2-D histograms respectively.
HWIDTH	--		PARM is a one- or two-word REAL array containing the bin width value XW or XW,YW for 1-D or 2-D histograms respectively.
HNORM	1.0	*	PARM is a normalization constant such that future calls to HOUT, HY, or HZ, give (bin content)/PARM.
HMSCAL	--	*	PARM is a REAL array that supplies the manual scale factors beginning with XMIN as described in section 3.5 (4 words for 1-D, 14 for 2-D) to be used in HOUT when manual scale factors have been selected. The default is 0., 1., 8., and 0. (4 words), for each axis.
TITLE	--	*	the string in PARM is stored in the title block. See section 3.4 for a description of a valid title.
USIZE	0	*	PARM supplies the size (in words) of the user block.
UDATA	0		the contents of the PARM array (USIZE words) are saved in the user block.
MTERR	0	*	if PARM is 0 (1), then the bin content for an empty bin is taken as 0.0 (1.0) by HYE and HZE when calculating error.

OPT	DEF-AULT		PARM Description
FUNCTION	-	*	PARM supplies the address of a user function (declared EXTERNAL) to be used in the output stage by HOUT for generating analytical curves. This option sets the NFUNC parameter to 1. HOUT passes the function address and the value of NFUNC to the DUT routines as the F and NF arguments (see DUT1 or DUT2). The user function should be defined as F(X) or F(X,K) for 1-D histograms, and F(X,Y) or F(X,Y,K) for 2-D histograms. In line printer plots, the function is calculated at the bin center, since LOWIN is set true by HOUT or HEXEC.
NFUNC	0	*	PARM is the number of analytic functions.
CONTOUR	-	*	PARM supplies the address of a user function (declared EXTERNAL) to be used in the output stage by HOUT for drawing contours in 2-D line printer plots or graphic scatterplots. This option sets the NCONT parameter to 1. HOUT passes the function address and the value of NCONT to DUT2 as the C and NC arguments described in DUT2.
NCONT	0	*	PARM is the number of contours.
YALT	0	*	Alternate output for certain modes-of-storage (see section 2.4).
ISOMETRIC	0	*	Directs HOUT to do an isometric plot instead of scatterplot on graphic devices for histograms defined as scatterplots ('S' used in MS arg in HDEF2). 0 = do scatterplot 1 = do isometric plot

4.21 HOUT

Output a histogram, scatterplot, analytic function, contour, and/or statistics.

```
+-------------------------------------------+
|                                           |
|          CALL HOUT(ID [,OPT])             |
|                                           |
+-------------------------------------------+
```

HOUT outputs the ID'th histogram (or all histograms if ID has the value 'ALL ') to the device given in the text string in OPT. If OPT is not supplied, or it doesn't contain a valid device then the default device is used. The initial default device is the line printer, but whenever HOUT or DPSLCT specifies another valid device, this device becomes the new default. HOUT calls DPSLCT(OPT) to set the device name.

HOUT calls DUT1 or DUT2 in order to generate the plot. Hence the description of DUT1 and DUT2 also applies to HOUT, and in particular, the options set by DOPT also apply to HOUT. Some options set by HOPTN also affect the output.

4.22 HPNTR

There is one common routine (HPNTR) used by all histogram routines for locating a histogram within the HCOM block. This routine may also be called by users as follows:

```
INTEGER HPNTR

IF (HPNTR(ID).EQ.0) --> not found
(found)
```

HPNTR returns back the value 1 if the histogram with identifier ID is found in HCOM, and 0 if it is not found. If found, then the pointer for the histogram is stored in the variable NODE (see Appendix A).

HPNTR can also return the pointer in HCOM for items within a histogram node, as follows:

```
ISUB = HPNTR(ID,ITEM)
```

```
where ITEM = 'STAT' for the statistics block
             'PARM' for the parameter block
             'HIST' for the histogram block
             'TITL' for the title block
             'DISP' for the display parameters
             'USER' for the user block within the histogram
```

Appendix A describes the contents of these items.

4.23 HSET1, HSET2

The contents of a specific bin of a histogram may be set by,

```
+------------------------------------------+
|                                          |
|          CALL HSET1(ID,XV,WT)            |
|                                          |
|                  or                      |
|                                          |
|          CALL HSET2(ID,XV,YV,WT)         |
|                                          |
+------------------------------------------+
```

These routines have the same calling sequence as HCUM1 and HCUM2, except that the value in WT replaces the bin contents. The statistics values are not changed.

4.24 HSLICE

A 2-D histogram may be sliced and projected into a 1-D output by:

```
+------------------------------------------+
|                                          |
|    CALL HSLICE(ID,XORY [,IBEG,IWDTH      |
|                  [,OPT]])                |
|                                          |
+------------------------------------------+
```

A slice of the ID'th histogram is made onto the X (Y) axis beginning with the Y (X) bin number IBEG and summing over IWDTH bins in the Y (X) direction when XORY contains the string 'X' ('Y'). If IWIDTH is 0, then all bins beyond IBEG are summed. If the optional arguments are not supplied, the default values are 1 for IBEG, 0 for IWDTH, and the default device for OPT. OPT may be a device string, or 'NOUNIT', in which case the output is suppressed. The latter is useful when the slice functions HFSLC and HERSLC are to be used to access the contents of bins for the slice.

4.25 HSPACE

Prints out the amount of free space still available in /HCOM/.

```
+------------------------------------------+
|                                          |
|              CALL HSPACE                 |
|                                          |
+------------------------------------------+
```

4.26 HSTAT

The routine HSTAT outputs the statistical means and standard deviations for a 1-D or 2-D histogram. The histogram itself is not plotted by this routine. If both the histogram and statistics are desired, then HOUT or HSLICE should be called instead with the 'SOUT' option set true (see HOPTN).

There are two ways in which statistics are calculated, depending on whether statistical summations are made or not by the HCUM routines (HCUM1 or HCUM2). If the histogram was defined with the 'STAT' option set true (see HOPTN), then sums are made during the HCUM calls, and these sums are used by HSTAT and HOUT to calculate the means and variances. Otherwise, the histogram bin contents are used to calculate the means and variances. The latter (bin) method is used for sliced histograms (by HSLICE), and also when the trailing arguments in HSTAT are supplied for the purpose of limiting the bin range over which the statistics are calculated.

```
+----------------------------------------------+
|                                              |
|       CALL HSTAT(ID,OPT [,DAT [,I1,I2        |
|                 [,J1,J2]]])                  |
|                                              |
|                     or                       |
|                                              |
|     CALL HSTAT(ID,OPT [,DAT [,IBEG,IWDTH     |
|                 [,I1,I2]]])                  |
|                                              |
+----------------------------------------------+
```

where the first case is used for un-sliced 1-D or 2-D histograms, and the second case is for sliced output.

Statistical values are printed or returned for the ID'th histogram, depending on the option stored in OPT, as follows:

```
   OPT      RESULT
 'PRINT'    print out statistics on line printer
 'GET '     return statistics into the DAT array, as follows:
             DAT( 1) = number of calls made to HCUM1 (or HCUM2).
                       If stats are being made from the bin
                       values, then this is the number of
                       (non-empty) bins.
             DAT( 2) = sum of weights
             DAT( 3) = mean value of X
             DAT( 4) = standard deviation in X
             DAT( 5) = lowest X found by HCUM, or lowest
                       bin value if stats are calculated from
                       bins.
             DAT( 6) = highest X found by HCUM, or highest
                       bin value if stats are calculated from
                       bins.
             DAT( 7-10) = not used
      End of list for 1-D statistics, 2-D statistics continue
             DAT( 9) = mean value of Y
             DAT(10) = standard deviation in Y
             DAT(11) = lowest Y found in HCUM2, or lowest
                       Y bin if stats are calculated from bins.
             DAT(12) = highest Y found in HCUM2, or
                       highest Y bin if stats are calculated
                       from bins.
             DAT(13) = not used
             DAT(14) = not used
           DAT(15-18)= covariance (x,x), (x,y), (x,y), (y,y)
             DAT(19) = correlation coefficient (rho)
             DAT(20) = not used
             DAT(21) = error ellipse major axis
             DAT(22) =   "     "     minor   "
             DAT(23) =   "     "     angle (rad)

 'XPRINT'   print stats for a slice along X axis
 'YPRINT'     "      "    "  "   "     "  Y   "
 'XGET'     return   "    "  "   "     "  X   "
 'YGET'       "      "    "  "   "     "  Y   "
```

I1 and I2 specify the (inclusive) range of X bins to be used for 1-D or 2-D histograms, while J1 and J2 give the range of Y bins. A zero value in any of these cases means all bins for that edge are used. If these arguments are not supplied, zero values are used instead.

IBEG and IWDTH are the first bin and number of bins in the slice as described in HSLICE, and if not supplied the values 1 and 0 respectively are used. When slicing, I1 and I2 refer to the axes along the slice.

4.27 HWRITE, HREAD

Histograms, or nodes, may be written out to a mass storage device (e.g. disk), and later retrieved from mass storage. It is possible to transfer only one particular node, or all nodes, or all nodes including the HCOM control area by issuing only one call to the HWRITE or HREAD routines. Each call to HWRITE also bumps a record count which is written out with the nodes, so that later HREAD can read the node(s) from a specified record.

```
+------------------------------------------------------+
|                                                      |
|     CALL HWRITE( [ID [,LUN [,IREW [,IREC]]]] )       |
|                                                      |
|                        and                           |
|                                                      |
|  CALL HREAD( [ID [,LUN [,IREC [,ICTL [,ID2]]]]] )    |
|                                                      |
+------------------------------------------------------+
```

where:

ID — is the identifier of a node (1-D, 2-D, or BLOCK), or
'ALL ' means all nodes are transferred, one by one.
0 means all of /HCOM/ is to be transferred.
(0 is the default)

LUN — is the logical unit number for the read or write.
The default is 20 for HWRITE and 21 for HREAD.

IREW — specifies rewinding prior to writing,
0 = no rewind
1 = rewind is done (the default).
-1 = append (write next record after last one on the file).

IREC — record number.

ICTL — is a control word for HREAD:
'ADD ' - add contents from disk to that in memory (ADD is the default).
'REPL' - replace node in memory with that from disk.
'SKIP' - skip (position) over this node on disk.
'GET ' - return ID and IREC of the next record on disk, and leave the file positioned at its beginning.

ID2 — is the (optional) new identifier for the node after it is read from disk, i.e the identifier is replaced by ID2. If not supplied, the default for ID2 is ID.

output

Each time HWRITE is called, an internal record count is bumped by 1. HWRITE returns this record number into IREC (if this argument is provided). If IREW is 1, then HWRITE writes to record 1.

If ID=0 in a HWRITE call, it is not possible to add such files with HREAD. ID='ALL ' should be used if histograms are later to be added by HREAD.

input

If ICTL is 'GET ', then the record number is returned in IREC.

If ICTL is not 'GET ', then:
IREC specifies the record number, or
IREC=0 means reading is to start at the beginning of the file (this is the default).

IREC='NEXT' means the next record is used.

IREC=0 and ID='ALL ', then all nodes are read off all records.

Records written out with an ID=0 can only be read back in the same way with an ID=0. If there is an incompatibility, HREAD issues an error message and aborts the request. A record written out with ID='ALL ' can be read back either with an ID='ALL ' or with individual identifiers within that record.

If ICTL is either 'ADD ' or 'REPL' but there is no node in memory having an identifier ID, then a new node with this ID is created and filled from the one on disk.

4.28 HX, HY, HZ,...

A set of functions are provided which return the coordinate values of the bin(s) and the normalized bin contents for 1-D or 2-D histograms. HPNTR must have been called just prior to the use of these functions, to set up the NODE pointer to the desired histogram. These functions return REAL values as follows:

HX(I) returns the X-coordinate value of the I'th bin (1-D or 2-D)

HY(I) returns the normalized value of the I'th bin (1-D)

HYE(I [,K [,L]]) returns the normalized error of the I'th bin of a 1-D histogram. K and L arguments are optional arguments for upper (K=1) or lower (K=2) error bar values, and for errors in X (L=2) or Y (L=1). The defaults are K=1 and L=1.

H2Y(J) returns the Y coordinate value of the J'th bin (2-D)

HZ(I,J) returns the normalized value of the (I,J)'th bin (2-D)

HZE(I,J [,K [,L]]) returns the normalized error of the (I,J)'th bin of a 2-D histogram. K and L are optional arguments for upper (K=1) or lower (K=2) error bar values, and for errors in X (L=2) or Y (L=1). The defaults are K=1 and L=1.

HN(I) (INTEGER) returns the number of HCUM1 calls that were under, within, or over the histogram range for I = -1, 0, or 1 respectively. For 2-D histograms, HN(I,J) should be used, where J refers to the Y coordinate just as I does for X.

To access the contents of bins in a slice of a 2-D histogram, a call to HSLICE with the 'NOUNIT' option should first be made, and then the following REAL functions may be used:

HFSLC(I) returns the normalized contents for the I'th bin in the slice

HERSLC(I) returns the normalized error for the I'th bin in the slice.

The normalizing factor for the above functions is the value set by the 'HNORM' option in HOPTN (default 1.0), and is used as follows:

e.g. HY(I) = (contents of bin I)/(normalizing factor)

A set of INTEGER functions HBINX and HBINY are provided which give the bin number that corresponds to a coordinate value in the ID'th histogram.

IBIN = HBINX(XV)	returns the integer bin number for the value XV in the X coordinate for a 1-D or 2-D histogram,
JBIN = HBINY(YV)	returns the integer bin number for the value YV in the Y coordinate for a 2-D histogram.

Section 5

EXAMPLES

Following is the full program for generating all the display figures shown in this manual.

```
      COMMON /HCOM/ HCOM(3000)
C
      REAL MASS2(3)
      DATA MASS2/0.02,0.25,0.88/
C
      REAL AX(25,2),AY(25,2),AYE(25,2),S(8),SF1(8)
      DATA S/4*0.,0.,10.,4.,5./
      DATA SF1/0.,.5,6.,5.,0.,10.,5.,2./
C
C       DATA FOR BINARY PROPORTIONS
      INTEGER TRIALS(20),SUCCES(20)
      DATA TRIALS/2*4,3*15,8*60,7*400/
      DATA SUCCES/0,1,1,2,3,2,5,15,30,45,55,58,60,390,380,350,
     >            200,50,20,10/
C
C       DATA FOR PLOT HAVING ASYMMETRIC ERRORS IN X AND Y
      REAL X2(10),Y2(10),E2(10,2,2),E2SCAL(8)
      DATA X2/2.2, 3.,3.5, 4.,4.5, 5., 6., 8.,11.,16./
      DATA Y2/6.5,8.3, 8., 7.,5.5, 5., 4., 3.,2.5,1.8/
      DATA E2/1., .6, .5, .4, .4, .3, .3, .2, .3, .4,
     >        3., .7, .5, .4, .4, .4, .4, .3, .3, 1.,
     >        .5, .2, .2, .2, .2, .2, .6, .8, 2., 5.,
     >        .8, .2, .2, .2, .2, .2, .5, .8,1.5, 2./
      DATA E2SCAL/-.5,5.,4.6,5.,0.,2.,5.5,2./
C
      EXTERNAL FX,FY,FYE,VY,GX,GY,GZ,GZE,F1,F1A,F2
C
      GAUSS(X)=SQRT(-2.*ALOG(RAN1(X)))*COS(2.*3.14159*RAN1(X))
C   GUASS IS A NORMALLY DISTRIBUTED RANDOM VARIABLE WITH ZERO
C   MEAN AND UNIT VARIANCE, AND RAN1 IS A RANDOM NUMBER IN
C    RANGE 0. TO 1.0
C
C   SET DENSE OPTION FOR IMAGEN
      CALL DOPT('LINDEN',5)
C   SET DUPLEX GRAPHIC CHARACTERS
      CALL DOPT('CHFMT',2)
C
C   INITIIALIZE HISTOGRAM BUFFER
      CALL HINIT(3000)
C
```

```
C
C  FIGURE 1.1 AND 1.2
C  -------------------------
C
C  1-D HISTOGRAM PLOTS TO LINE PRINTER AND GRAPHIC DEVICES,
C  WITH DEFAULT SETTINGS OF OPTIONS
C
      CALL HDEF1(1,'I*2 ',32,-4.0,0.25,
     >           'EXPT-A;DEVIATION;FREQ@')
C  (ALSO GENERATE STATISTICAL SUMS, OUTPUTTED LATER)
      CALL HOPTN('STATS',.TRUE.,1)
      CALL HOPTN('SOUT',.FALSE.,1)
      DO 10 I=1,500
   10 CALL HCUM1(1,GAUSS(X),1.0)
      CALL HOUT(1)
C  MAKE A PLOT ON THE IMGN300
      CALL HOUT(1,'IMGN300 ')
C  AND ALSO A SEQ FILE FOR TEKTRONIX PLOT
CCCCC      CALL HOUT(1,'SEQ4010 ')
C
C  FIGURE 5.1 AND 5.2
C  ------------------
C
C  ADD ANALYTIC CURVE TO ABOVE 1-D HISTOGRAMS
      CALL HOPTN('FUNC',F1,1)
C  USE 5 TIC INTERVALS ALONG X AXIS
      CALL DOPT('XTIC',5)
      CALL HOUT(1,'PRINTER')
C  USE DIAMOND MARKER SYMBOLS (3'RD UG MARKER)
      CALL DOPT('MARKER',3)
C  AND MAKE LARGER TITLE SIZES
      CALL DOPT('TISIZE',3)
C  AND MAKE FANCY TITLES
      CALL HOPTN('TITLE',
     >'{ D{LISTRIBUTION OF} 1/({M2}{+{DOO}}2{GPS})
     {LE}{^-(X/{GS}){^2}  (goes on previous line)
     > ;{ X/{GS};FREQUENCY@',1)
      CALL HOUT(1,'IMGN300 ')
C
C
C  FIGURE 5.3 AND 5.4
C  ------------------
C
C  USE LOG SCALE FOR Y AXIS FOR ABOVE 1-D HISTOGRAM
      CALL DOPT('YLOG',.TRUE.)
C  USE FULL DATA SCALING FOR X AXIS IN GRAPHIC PLOT
      CALL DOPT('XAUTO',2)
      CALL HOUT(1,'IMGN300 ')
C  RESTORE TITLE SO THAT LINE PRINTER CAN HANDLE IT
      CALL HOPTN('TITLE','EXPT-A;DEVIATION;FREQ@',1)
C  ALSO TURN ON THE NUMBER OF CALLS OUTPUT
      CALL DOPT('NCALLS',.TRUE.)
      CALL HOUT(1,'PRINTER ')
      CALL DOPT('YLOG',.FALSE.)
```

```
C
C
C  FIGURE 5.5 AND 5.6
C  ------------------
C
C  USE MANUAL SCALE FACTORS FOR ABOVE 1-D HISTOGRAM
C  (STORE THEM SO HOUT CAN GET THEM)
      CALL HOPTN('HMSCAL',S,1)
C  THEN SELECT THE MANUAL SCALES OPTION FOR Y
      CALL DOPT('YAUTO',0)
C  ALSO TURN OFF THE 'ID=' FONT
      CALL DOPT('IDFONT',0)
C  AND USE INWARD TIC MARKS
      CALL DOPT('TICDIR',1)
      CALL HOUT(1,'PRINTER ')
      CALL HOUT(1,'IMGN300 ')
      CALL DOPT('YAUTO',1)
      CALL DOPT('IDFONT',1)
C
C
C  FIGURE 5.7 THROUGH 5.10
C  -----------------------
C
C  1-D UNDERFLOWS AND OVERFLOWS - TWO METHODS
C  GENERATE TWO 1-D HISTS, ONE WITH OVERFLOWS IN EDGE BINS,
C  AND ANOTHER WITH OVERFLOWS OMITTED FROM EDGE BINS
C
      CALL HDEF1('1A  ','R*4 ',32,-4.0,0.25,
     > 'DEFAULT OVERFLOWS;DEVIATION;FREQ@')
      CALL HDEF1('1B  ','R*4 ',32,-4.0,0.25,
     > 'OVERFLOWS OMITTED;DEVIATION;FREQ@')
      CALL HOPTN('HOMIT',.TRUE.,'1B  ')
C  ALSO TURN ON STATISTICS FOR ONE CASE
      CALL HOPTN('STAT',.TRUE.,'1A  ')
      CALL HOPTN('SOMIT',.TRUE.,'1A  ')
C  OTHER CASE WILL USE STATS FROM BINS
      CALL HOPTN('SOUT',.TRUE.,'1B  ')
C
C  GENERATE IDENTICAL GAUSSIAN DISTRIBUTIONS WITH
C  BACKGROUND IN EACH HISTOGRAM
      DO 20 I=1,500
      RX=GAUSS(X)
      BX=100.*(RAN1(X)-.2)
      CALL HCUM1('1A  ',BX,0.5)
      CALL HCUM1('1B  ',BX,0.5)
      CALL HCUM1('1A  ',RX,1.0)
   20 CALL HCUM1('1B  ',RX,1.0)
C  TURN ON NUMBER OF CALLS
      CALL DOPT('NCALLS',.TRUE.)
C  OUTPUT THE DEFAULT CASE (OVERFLOWS IN EDGE BINS)
      CALL HOUT('1A  ','PRINTER ')
      CALL HOUT('1A  ','IMGN300 ')
C  OUTPUT THE ALTERNATE CASE
      CALL HOUT('1B  ','PRINTER ')
```

```
      CALL HOUT('1B  ','IMGN300 ')
C
C
C  FIGURE 5.11 AND 5.12
C  -----------------
C
C  DUT1 EXAMPLE, USING NORMAL AUTO SCALES
C  (ALSO TURN TICS INWARD IN 1-D GRAPHIC PLOTS)
      CALL DOPT('TICDIR',.TRUE.)
C
      CALL DPSLCT('IMGN300 ')
      CALL DUT1(FX,FY,FYE,30,'DUT1 PLOT@')
C
C  DUT1 EXAMPLE, USING A FORCED ZERO ON Y AXIS
C
      CALL DOPT('YZERO',.TRUE.)
      CALL DUT1(FX,FY,FYE,30,'DUT1 WITH YZERO@')
      CALL DOPT('YZERO',.FALSE.)
      CALL DOPT('TICDIR',.FALSE.)
C
C
C  FIGURE 5.13 AND 5.14
C  -------------------------
C
C  2-D PLOTS TO LINE PRINTER AND GRAPHIC DEVICE,
C  WITH DEFAULT SETTINGS OF OPTIONS
C
      CALL DPSLCT('PRINTER ')
      CALL DUT2(GX,GY,GZ,GZE,24,12,
     >        'DUT2 PLOT@')
      CALL DPSLCT('IMGN300 ')
      CALL DUT2(GX,GY,GZ,GZE,24,12,
     >       'DUT2 PLOT@')
C
C
C  FIGURE 5.15 AND 5.16
C  --------------------------
C
C  2-D SCATTERPLOTS AND 2-D HISTOGRAMS
C   1. DEFINE THEM
C
      CALL HDEF2('SCT1','I*2S',30,40,-.25,0.,0.05,0.05,
     > 'SCATTER PLOT;SQUARED MASS;MOMENTUM@')
      CALL HDEF2('SCT2','I*2S',1,1,0.,0.,1.2,1.2,
     > 'BETA VS P;MOMENTUM;BETA@')
C
C   2. FILL THEM UP
C
      DO 100 I=1,2500
      E=-ALOG(RAN1(X))*.50
      R=RAN1(X)
      M=1
      IF(R.GT.0.5) M=2
      IF(R.GT.0.8) M=3
```

```
      IF(E*E.LE.MASS2(M)) GO TO 100
      P=SQRT(E*E-MASS2(M))
      T=5.*E/P + .30*GAUSS(X)
      P=P + .01*P*SQRT(P*P+2.)*GAUSS(X)
      IF(P.LT.0.1) GO TO 100
      E=P*T/5.
      AM2=E*E-P*P
      CALL HCUM2('SCT1',AM2,P,1.0)
      BETA=P/E
      CALL HCUM2('SCT2',P,BETA,1.0)
  100 CONTINUE
C
C   3. OUTPUT THE 'SCT1' SCATTERPLOT ('SCT2' DONE LATER)
C
      CALL HOUT('SCT1','PRINTER ')
C  ADD A CONTOUR CURVE
      CALL HOPTN('CONTOUR',F2,'SCT1')
      CALL HOUT('SCT1','IMGN300 ')
C
C
C  FIGURE 5.17
C  -----------
C
C  SAME SCATTER PLOT, BUT USING MANUAL Z AND C SCALING
C  (ONE COLUMN WIDE PER BIN, 16 SYMBOLS IN Z, AND
C   INCLUDE A CONTOUR PLOT GIVEN BY FUNTION F2)
C
      CALL DOPT('L2ZAUT',.FALSE.)
      CALL DOPT('L2CAUT',.FALSE.)
      CALL DOPT('L2COL',1)
      CALL DOPT('L2SYM',16)
      CALL HOUT('SCT1','PRINTER ')
C         RESTORE OPTIONS TO THEIR DEFAULTS
      CALL DOPT('RES,L2ZAUT')
      CALL DOPT('RES,L2CAUT')
      CALL DOPT('RES,L2SYM')
      CALL HOPTN('NCONT',0,'SCT1')
C
C
C  FIGURE 5.18
C  -----------
C
C  SAME SCATTER PLOT, BUT USING LOGSCALING FOR Z AXIS
C
      CALL DOPT('YLOG',.TRUE.)
      CALL HOUT('SCT1','PRINTER ')
      CALL DOPT('YLOG',.FALSE.)
C
C
C  FIGURE 5.19 TO 5.22
C  -------------------
C
C  2-D HISTOGRAM - WITH HIDDEN LINES REMOVED.
C
```

```
      CALL HOPTN('ISOMETRIC',1,'SCT1')
      CALL HOUT('SCT1','IMGN300 ')
C
C  ROTATE THE 2-D HISTOGRAM
      CALL DOPT('ROTANGLE',-40.)
C  RESTORE DEFAULT TITLE SIZE
      CALL DOPT('RES,TISIZE')
      CALL HOUT('SCT1')
      CALL DOPT('RES,ROTANGLE')
C
C  2-D MESH PLOT - WITH HIDDEN LINES REMOVED.
      CALL DOPT('2DLINE',2)
      CALL HOUT('SCT1')
C
C  2-D LINE SLICES - WITH HIDDEN LINES REMOVED.
      CALL DOPT('2DLINE',1)
      CALL HOUT('SCT1')
C
C
C  FIGURE 5.23 AND 5.24
C  ---------------------------
C
C  2-D UNDERFLOWS AND OVERFLOWS - TWO METHODS
C  GENERATE TWO 2-D HISTS, ONE WITH OVERFLOWS IN EDGE BINS,
C  AND ANOTHER WITH OVERFLOWS OMITTED FROM EDGE BINS
C
      CALL HDEF2('2A  ','I*2 ',25,40,-5.0,-4.0,.4,.2,
     >                    '2-D HIST, DEFAULT OVERFLOWS@')
      CALL HDEF2('2B  ','I*2 ',25,40,-5.0,-4.0,.4,.2,
     >                    '2-D HIST, OVERFLOWS OMITTED@')
C  ALSO TURN ON STATS IN ONE CASE, USE BINS IN OTHER
      CALL HOPTN('SOUT',.TRUE.,'2A  ')
      CALL HOPTN('STAT',.TRUE.,'2B  ')
      CALL HOPTN('SOMIT',.TRUE.,'2B  ')
      CALL HOPTN('HOMIT',.TRUE.,'2B  ')
      DO 50 I=1,1000
      R1=GAUSS(X)
      R2=GAUSS(X)
      RX=R1+.5*R2
      RY=R2
C  ADD A FLAT BACKGROUND TO THE GAUSSIAN
      BX=100.*(RAN1(X)-.5)
      BY=100.*(RAN1(X)-.5)
      CALL HCUM2('2A  ',BX,BY,1.0)
      CALL HCUM2('2B  ',BX,BY,1.0)
      CALL HCUM2('2A  ',RX,RY,1.0)
   50 CALL HCUM2('2B  ',RX,RY,1.0)
C
C  2-D PRINTER PLOT WITH NUMBER OF CALLS
      CALL HOUT('2A  ','PRINTER')
      CALL HOUT('2B  ')
      CALL DOPT('NCALLS',.FALSE.)
C
C
```

```
C  FIGURE 5.25 AND 5.26
C  ---------------------------
C
C  USE SLICES FROM SCT1 TO FILL AX,AY,AYE FOR MULTI-PLOT
C
      CALL HSLICE('SCT1','Y',5,5,'NOUNIT  ')
      DO 120 I=1,25
      AX(I,1)=H2Y(I)
      AY(I,1)=HFSLC(I)
  120 AYE(I,1)=HERSLC(I)
      CALL HSLICE('SCT1','Y',15,15,'NOUNIT  ')
      DO 130 I=1,25
      AX(I,2)=H2Y(I)
      AY(I,2)=HFSLC(I)
  130 AYE(I,2)=HERSLC(I)
      N2=2025
      CALL DOPT('LINE',1)
      CALL DPSLCT('PRINTER ')
      CALL DUT1A(AX,AY,AYE,N2,
     >'DUT1A MULTI-PLOT PLUS A FUNCTION;ENERGY;COUNTS@',S,F1A)
      CALL DPSLCT('IMGN300 ')
      CALL DUT1A(AX,AY,AYE,N2,
     >'DUT1A MULTI-PLOT PLUS A FUNCTION;ENERGY;COUNTS@',S,F1A)
C
C
C  FIGURE 5.27
C  -----------
C
C  USE DUT1 WITH ONLY A FUNCTION (NO DATA POINTS)
C
      CALL DUT1(0,0,0,0,
     >'DUT1 WITH FUNCTION ONLY, NO DATA POINTS@',SF1,F1)
C
C
C  FIGURE 5.28
C  -----------
C
C  SLICE A 2-D INTO A 1-D LINE PRINTER PLOT
C  (BAR GRAPH, NO ERROR BARS, AND 4 Y TIC INTERVALS)
C
      CALL DPSLCT('PRINTER ')
      CALL DOPT('HIST',.TRUE.)
      CALL DOPT('ERROR',0)
      CALL DOPT('YTIC',4)
      CALL HSLICE('SCT1','X',10,9)
C
C
C  FIGURE 5.29A AND FIGURE 5.29B
C  -----------------------------
C
C  OVERPLOT FOUR PLOTS (ONE PER QUADRANT)  ON GRAPHIC DEVICE
C       1) USING FIXED MARGIN AND CHARACTER SIZES
C       2) SCALING MARGINS AND CHARACTERS WITH THE PLOT SIZE
C          (THIS IS THE DEFAULT)
```

```
C
      DO 350 I=1,2
      CALL DOPT('CHSCALE',I-1)
C    (TOP LEFT QUADRANT)
C
      CALL DPSLCT('IMGN300 ')
      CALL DOPT('QUAD','TL')
      CALL DOPT('RESCALE',.TRUE.)
      CALL DOPT('TICDIR',.TRUE.)
      CALL DOPT('OVER',.TRUE.)
      CALL DOPT('XTIC',4)
      CALL DOPT('YTIC',4)
      CALL DOPT('LINE',1)
      CALL HSLICE('SCT1','X',10,9)
C
C    (TOP RIGHT QUADRANT)
      CALL DOPT('QUAD','TR')
      CALL DOPT('LINE',2)
      CALL DOPT('CHAR','    ')
      CALL HSLICE('SCT1','X',10,9)
C
C    (BOTTOM LEFT QUADRANT)
      CALL DOPT('QUAD','BL')
      CALL HOUT('SCT2')
C
C    (BOTTOM RIGHT QUADRANT)
      CALL DOPT('QUAD','BR')
      CALL DOPT('XLOG',.TRUE.)
      CALL DOPT('YLOG',.TRUE.)
      CALL DOPT('LINE',0)
      CALL DOPT('CHAR','+   ')
      CALL DUT1(FX,VY,FYE,30,'DUT1 LOG-LOG PLOT@')
      CALL DOPT('OVER',.FALSE.)
      CALL DOPT('XLOG',.FALSE.)
      CALL DOPT('YLOG',.FALSE.)
  350 CONTINUE
      CALL DOPT('RES,CHSCALE')
C
C
C  FIGURE 5.30
C  -----------
C
C  MAKE TWO BUTTING HALF FRAME PLOTS ONTO GRAPHIC DEVICE
C
      CALL DOPT('QUAD','T ')
      CALL DOPT('BMARG',0.)
      CALL DOPT('OVER',.TRUE.)
      CALL DOPT('LINE',2)
      CALL DOPT('ERROR',1)
      CALL DOPT('MARKER',0)
      CALL DOPT('CHAR','    ')
      CALL DOPT('TISIZE',0)
      CALL DOPT('XTIC',8)
      CALL DOPT('YTIC',4)
```

```
      CALL HSLICE('SCT1','X',10,6)
      CALL DOPT('QUAD','B ')
      CALL DOPT('RES,BMARG')
      CALL DOPT('TMARG',0.)
      CALL HSLICE('SCT1','X',15,30)
      CALL DOPT('RES,TMARG')
      CALL DOPT('OVER',.FALSE.)
C
C
C  FIGURE 5.31
C  -----------
C
C  MAKE TWO DISJOINT HALF FRAME PLOTS ONTO GRAPHIC DEVICE
C
      CALL DOPT('OVER',.TRUE.)
      CALL DOPT('ERROR',0)
      CALL DOPT('QUAD','L ')
      CALL DOPT('XTIC',4)
      CALL DOPT('YTIC',5)
      CALL HSLICE('SCT1','Y',5,5)
      CALL DOPT('QUAD','R ')
      CALL HSLICE('SCT1','Y',10,5)
      CALL DOPT('QUAD',0)
      CALL DOPT('OVER',.FALSE.)
C
C
C  FIGURE 5.32
C  -----------
C
C  GENERATE DIRECTORY OF ALL HISTOGRAMS
C
      CALL HGET('ALL ')
C
C
C  FIGURE 5.33
C  -----------
C
C  GENERATE A SMOOTH CURVE FOR THE DATA POINTS
C
C  FIRST, RESET ALL OPTIONS TO THEIR DEFAULTS
      CALL DOPT('RES,ALL ')
C  USE 4'TH MARKER SYMBOL (SQUARES)
      CALL DOPT('MARKER',4)
      CALL DOPT('TICDIR',1)
      CALL DOPT('TISIZE',1)
      CALL HOPTN('RES,NFUN',0,'SCT1')
C  SET THE SMOOTH OPTION, WITH A 1.1 BIN WIDTH SIGMA
      CALL DOPT('SMOOTH',.TRUE.)
      CALL DOPT('WSMOOTH',1.1)
      CALL DOPT('XTIC',4)
      CALL HSLICE('SCT1','X',11,15,'IMGN300 ')
      CALL DOPT('SMOOTH',.FALSE.)
C
C
```

```
C  FIGURE 5.34
C  -----------
C
C  GENERATE A PLOT OF PROPORTIONS (EFFICIENCIES)
C
      CALL HDEF1('PRP1','EFF ',20,1.0,1.0,
     >'PROPORTIONS, WITH 95% CONFIDENCE LIMITS@')
      DO 400 I=1,20
      N=TRIALS(I)
      IF(N.LE.0) GO TO 400
      DO 390 J=1,N
      W=0.
      IF(SUCCES(I).GE.J) W=1.0
  390 CALL HCUM1('PRP1',I-1.0,W)
  400 CONTINUE
      CALL HOUT('PRP1','IMGN300 ')
C
C
C  FIGURE 5.35
C  -----------
C
C  1-D PLOT WITH ASYMMETRIC ERRORS IN X, ASYMMETRIC IN Y
C
      CALL DOPT('ERROR',10)
      CALL DOPT('AUTO',0)
      CALL DOPT('TICDIR',1)
      CALL DOPT('MARKER',4)
      CALL DOPT('CHFMT',2)
      CALL DPSLCT('IMGN300 ')
      CALL DUT1A(X2,Y2,E2,10,'ERRORS IN X AND Y@',E2SCAL)
C
C  CLOSE ALL GRAPHIC DEVICES
      CALL DPCLOS('ALL ')
      STOP
      END
C
C  FUNCTIONS FOR THE EXAMPLES
C
      FUNCTION FX(I)
      FX=0.1*I
      RETURN
C
      ENTRY FY(I)
      FY=10. + AMOD(I*I/100.,1.0)
      RETURN
C
      ENTRY VY(I)
      VY=10.*EXP(-(I-15)**2/10.) + 1.0/((I-15.)**2+1.0)
      RETURN
C
      ENTRY FYE(I)
      FYE=0.1
      RETURN
C
```

```
      ENTRY GX(I)
      GX=(I-12.0)/5.0
      RETURN
C
      ENTRY GY(J)
      GY=(J-6.0)/2.
      RETURN
C
      ENTRY GZ(I,J)
      GZ=100.*EXP(-(I-12)**2/15. - (J-6)**2/5.
     >          -(I-12)*(J-6)/10.)
      RETURN
C
      ENTRY GZE(I,J)
      GZE=0.
      RETURN
C
      ENTRY F1(X,I)
      F1=500.*0.25*0.39894*EXP(-X*X/2.0)
      RETURN
C
      ENTRY F1A(X,I)
      F1A=125.*EXP(-(X+.025)/.50)
      RETURN
C
      ENTRY F2(X,Y,I)
      F2=((X-.88)/.25)**2 + ((Y-0.6)/.5)**2 - 1.0
      RETURN
      END
```

*

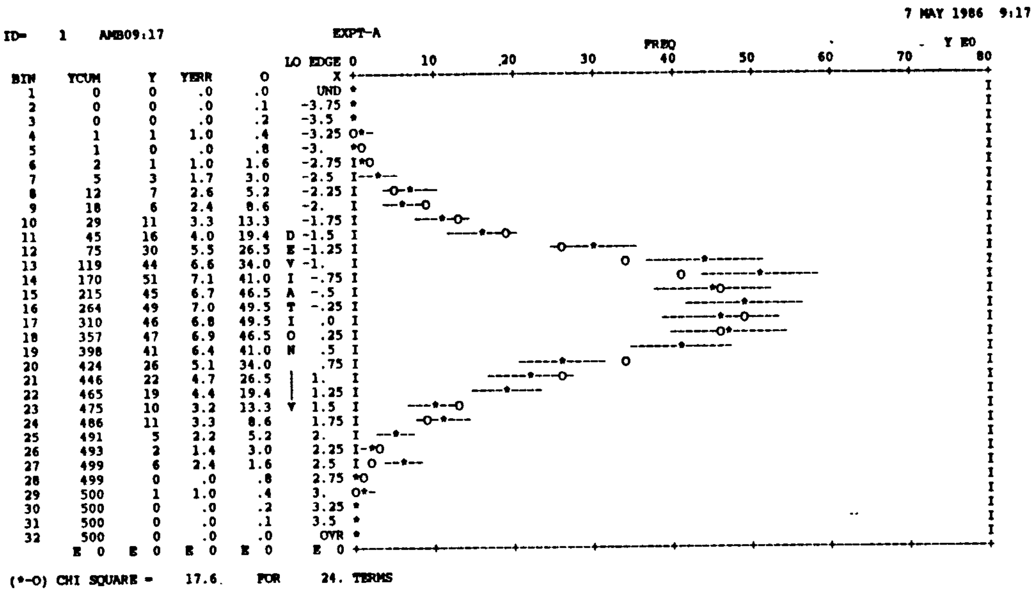


Figure 5.1 1-D Line Printer Histogram, plus analytic function.

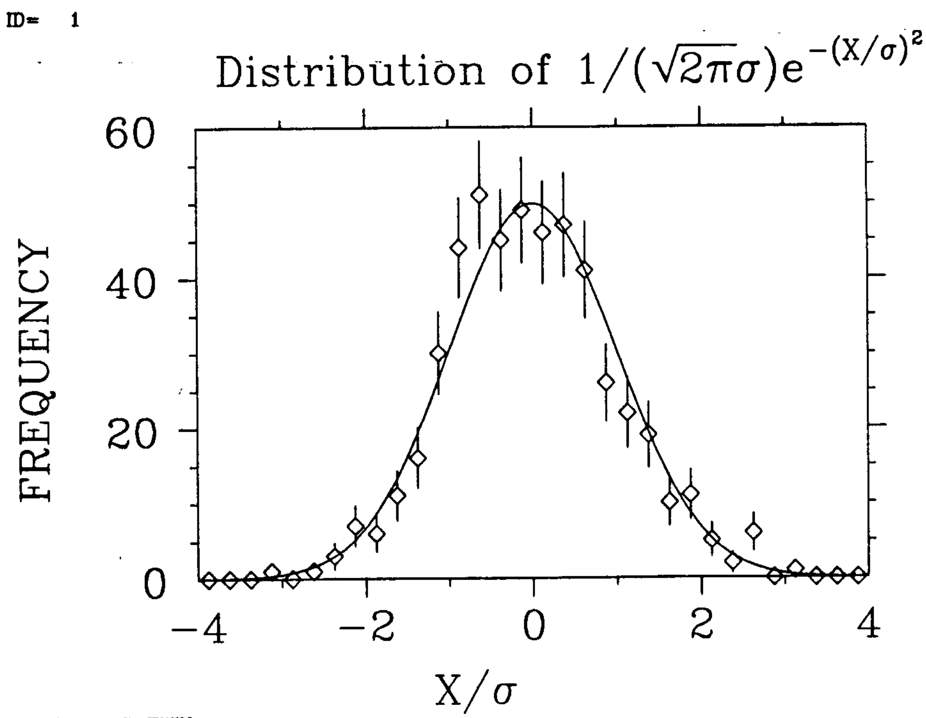
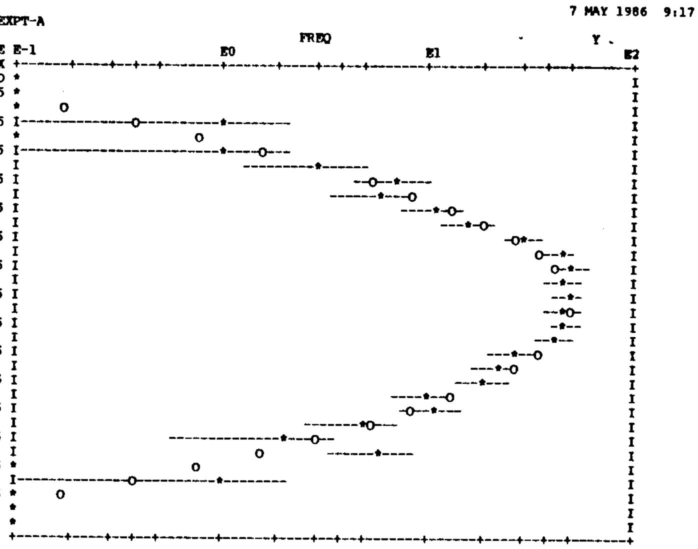


Figure 5.2 1-D Graphic Histogram, MARKER symbols, large duplex titles, plus analytic function.

*

ID= 1 AMB09:17 EXPT-A



BIN	YCUM	Y	YERR	O		LO EDGE E-1 X
1	0	0	.0	.0		UND
2	0	0	.0	.1		-3.75
3	0	0	.0	.2		-3.5
4	1	1	1.0	.4		-3.25
5	1	0	.0	.8		-3.
6	2	1	1.0	1.6		-2.75
7	5	3	1.7	3.0		-2.5
8	12	7	2.6	5.2		-2.25
9	18	6	2.4	8.6		-2.
10	29	11	3.3	13.3		-1.75
11	45	16	4.0	19.4	D	-1.5
12	75	30	5.5	26.5	E	-1.25
13	119	44	6.6	34.0	V	-1.
14	170	51	7.1	41.0	I	-.75
15	215	45	6.7	46.5	A	-.5
16	264	49	7.0	49.5	T	-.25
17	310	46	6.8	49.5	I	.0
18	357	47	6.9	46.5	O	.25
19	398	41	6.4	41.0	N	.5
20	424	26	5.1	34.0		.75
21	446	22	4.7	26.5	|	1.
22	465	19	4.4	19.4	|	1.25
23	475	10	3.2	13.3	▼	1.5
24	486	11	3.3	8.6		1.75
25	491	5	2.2	5.2		2.
26	493	2	1.4	3.0		2.25
27	499	6	2.4	1.6		2.5
28	499	0	.0	.8		2.75
29	500	1	1.0	.4		3.
30	500	0	.0	.2		3.25
31	500	0	.0	.1		3.5
32	500	0	.0	.0		OVR
	E 0	E 0	E 0	E 0		E 0

(*-O) CHI SQUARE = 17.6 FOR 24. TERMS

#CALLS= 500

Figure 5.3 1-D Line Printer Plot, log scales on Y axis, plus analytic function.

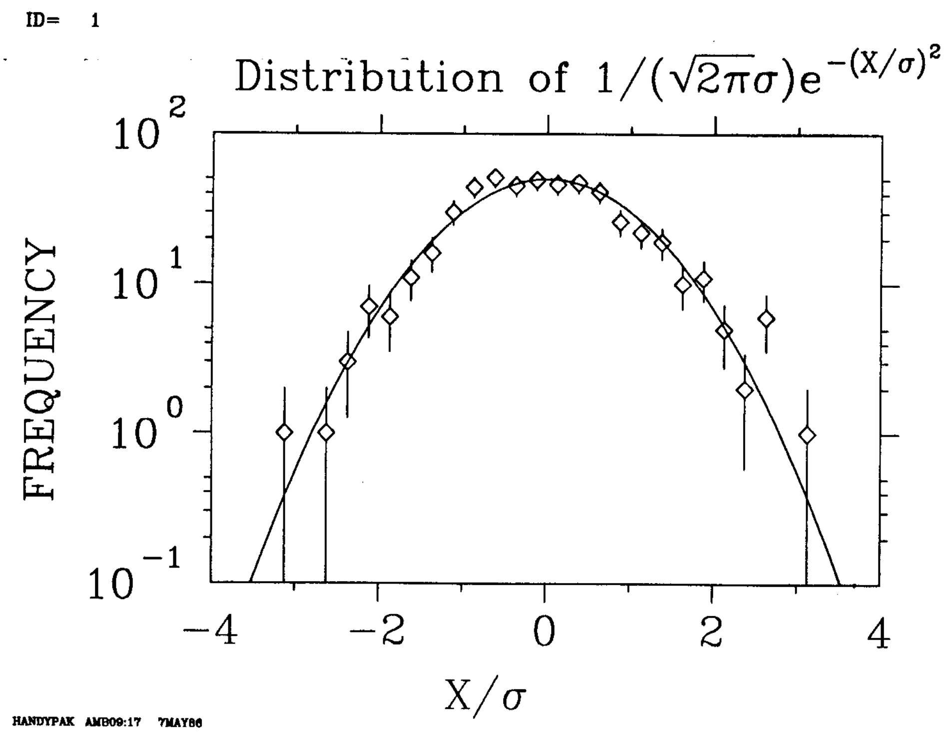
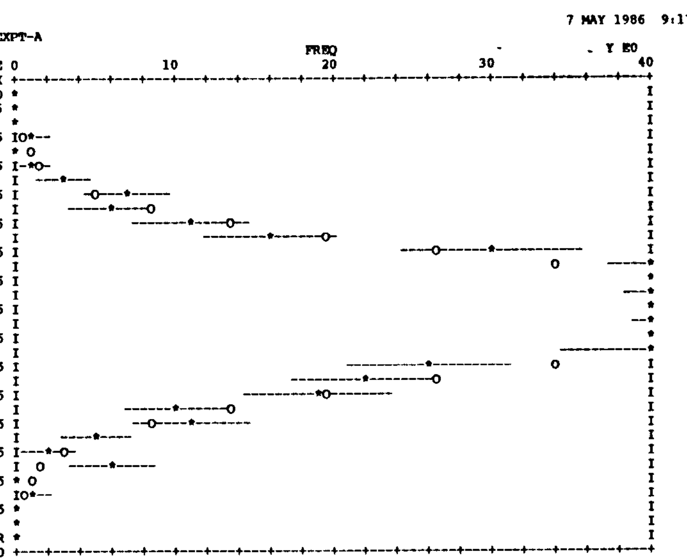


Figure 5.4 1-D Graphic Plot, log scales on Y axis, plus function.

*

ID- 1 AMB09:17 EXPT-A

BIN	YCUM	Y	YERR	O		LO EDGE X
1	0	0	.0	.0		UND
2	0	0	.0	.1		-3.75
3	0	0	.0	.2		-3.5
4	1	1	1.0	.4		-3.25
5	1	0	.0	.8		-3.
6	2	1	1.0	1.6		-2.75
7	5	3	1.7	3.0		-2.5
8	12	7	2.6	5.2		-2.25
9	18	6	2.4	8.6		-2.
10	29	11	3.3	13.3		-1.75
11	45	16	4.0	19.4	D	-1.5
12	75	30	5.5	26.5	E	-1.25
13	119	44	6.6	34.0	V	-1.
14	170	51	7.1	41.0	I	-.75
15	215	45	6.7	46.5	A	-.5
16	264	49	7.0	49.5	T	-.25
17	310	46	6.8	49.5	I	.0
18	357	47	6.9	46.5	O	.25
19	398	41	6.4	41.0	N	.5
20	424	26	5.1	34.0		.75
21	446	22	4.7	26.5	|	1.
22	465	19	4.4	19.4	|	1.25
23	475	10	3.2	13.3	V	1.5
24	486	11	3.3	8.6		1.75
25	491	5	2.2	5.2		2.
26	493	2	1.4	3.0		2.25
27	499	6	2.4	1.6		2.5
28	499	0	.0	.8		2.75
29	500	1	1.0	.4		3.
30	500	0	.0	.2		3.25
31	500	0	.0	.1		3.5
32	500	0	.0	.0		OVR
	E 0	E 0	E 0	E 0		E 0



(*-O) CHI SQUARE = 17.6 FOR 24. TERMS

#CALLS= 500

Figure 5.5 1-D Line Printer Plot, manual scale factors, and analytic function, and NCALLS.

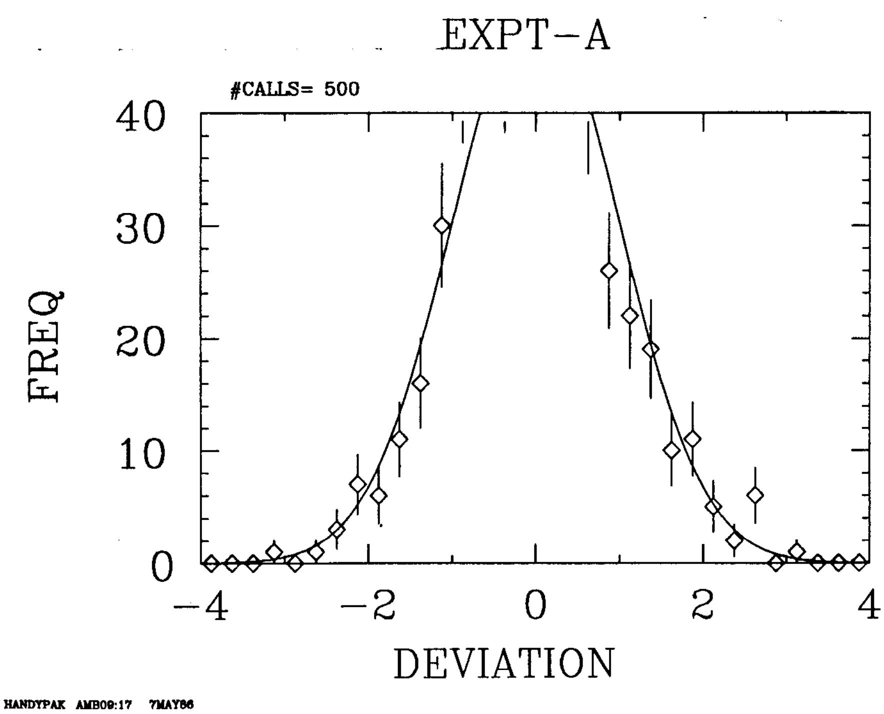


Figure 5.6 1-D Graphic Plot, manual scale factors, and analytic function, and NCALLS.

*

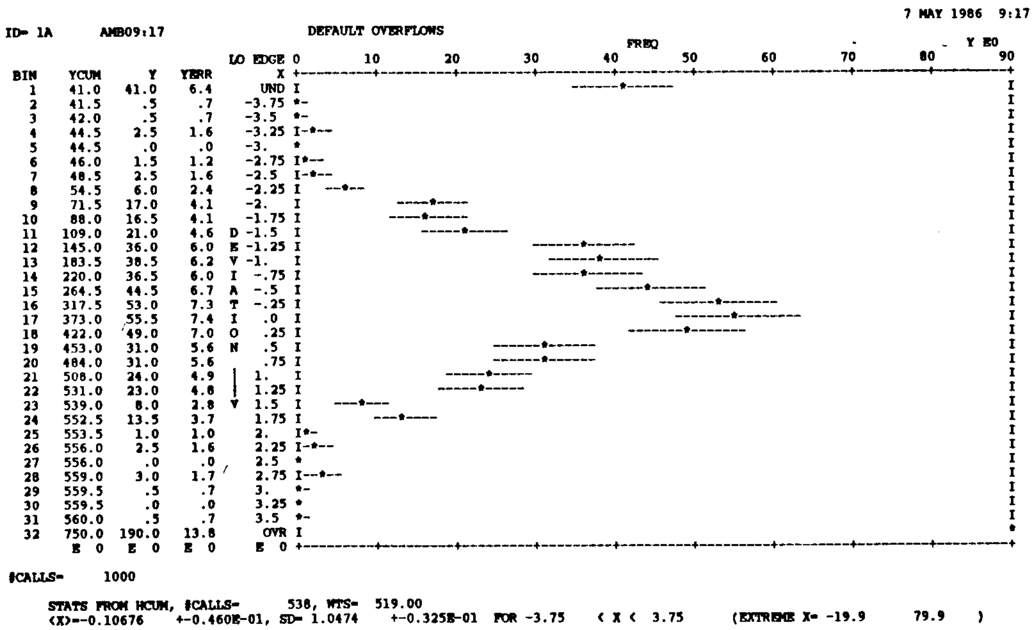


Figure 5.7 1-D Line Printer Hist, with default overflows in edge bins, NCALLS, and statistics (SOUT).

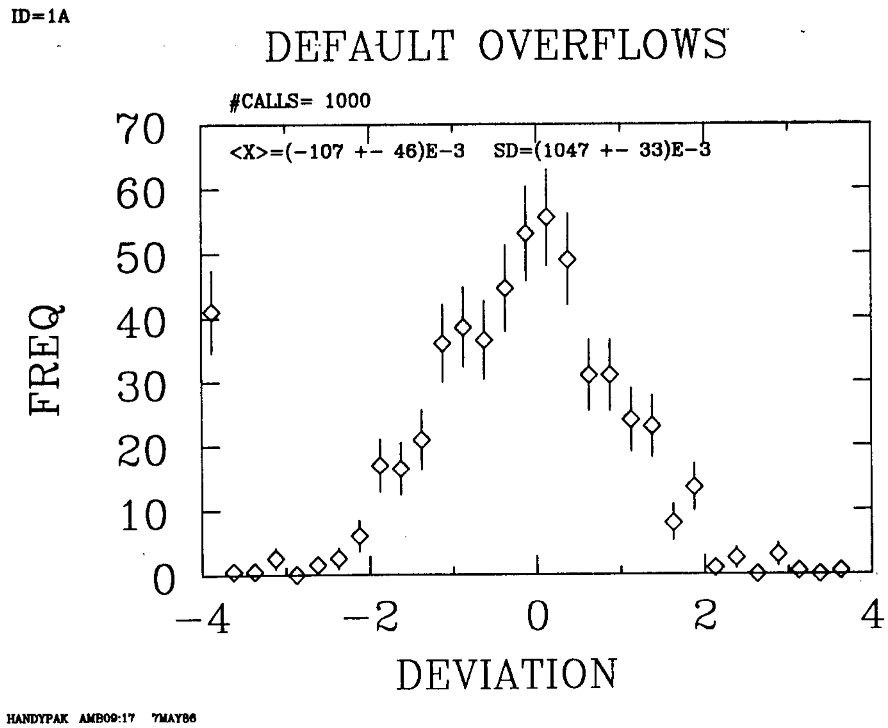


Figure 5.8 1-D Graphic Hist, with default overflows in edge bins, NCALLS, and statistics (SOUT).

*

```
                                                                                                    7 MAY 1986  9:17
 ID= 1B      AMB09:17                    OVERFLOWS OMITTED                       FREQ                          .  Y E0
                                 LO EDGE 0         10        20        30        40        50        60        70        80        90
 BIN   YCUM       Y    YERR            X +---------+---------+---------+---------+---------+---------+---------+---------+---------+
   1     .5      .5      .7      -4.     *-                                                                                        I
   2    1.0      .5      .7      -3.75   *-                                                                                        I
   3    1.5      .5      .7      -3.5    *-                                                                                        I
   4    4.0     2.5     1.6      -3.25   I-*--                                                                                     I
   5    4.0      .0      .0      -3.     *                                                                                         I
   6    5.5     1.5     1.2      -2.75   I*--                                                                                      I
   7    8.0     2.5     1.6      -2.5    I-*--                                                                                     I
   8   14.0     6.0     2.4      -2.25   I   --*--                                                                                 I
   9   31.0    17.0     4.1      -2.     I            ----*----                                                                    I
  10   47.5    16.5     4.1      -1.75   I           ----*----                                                                     I
  11   68.5    21.0     4.6    D -1.5    I               -----*-----                                                               I
  12  104.5    36.0     6.0    E -1.25   I                             ------*------                                               I
  13  143.0    38.5     6.2    V -1.     I                               ------*-------                                            I
  14  179.5    36.5     6.0    I  -.75   I                             ------*-------                                              I
  15  224.0    44.5     6.7    A  -.5    I                                     ------*-------                                      I
  16  277.0    53.0     7.3    T  -.25   I                                             -------*-------                             I
  17  332.5    55.5     7.4    I   .0    I                                               -------*--------                          I
  18  381.5    49.0     7.0    O   .25   I                                         -------*-------                                 I
  19  412.5    31.0     5.6    N   .5    I                        ------*------                                                    I
  20  443.5    31.0     5.6        .75   I                        ------*------                                                    I
  21  467.5    24.0     4.9    |  1.     I                  -----*-----                                                            I
  22  490.5    23.0     4.8    |  1.25   I                 -----*-----                                                             I
  23  498.5     8.0     2.8    V  1.5    I    ---*---                                                                              I
  24  512.0    13.5     3.7       1.75   I         ---*----                                                                        I
  25  513.0     1.0     1.0       2.     I*-                                                                                       I
  26  515.5     2.5     1.6       2.25   I-*--                                                                                     I
  27  515.5      .0      .0       2.5    *                                                                                         I
  28  518.5     3.0     1.7       2.75   I--*--                                                                                    I
  29  519.0      .5      .7       3.     *-                                                                                        I
  30  519.0      .0      .0       3.25   *                                                                                         I
  31  519.5      .5      .7       3.5    *-                                                                                        I
  32  520.0      .5      .7       3.75   *-                                                                                        I
          E  0    E  0    E  0       E  0 +---------+---------+---------+---------+---------+---------+---------+---------+---------+

 #CALLS=       81 /     540 /     379    = UNDER / HIST / OVER

      STATS FROM BINS, #CALLS=       29, WTS=  520.00
      <X>=-0.10192    +-0.462E-01, SD= 1.0544    +-0.327E-01  FOR -4.00    < X <  4.00      (EXTREME X= -3.88       3.88    )
```

Figure 5.9 1-D Line Printer Hist, with overflows omitted from edge bins (HOMIT set true), NCALLS, and statistics.

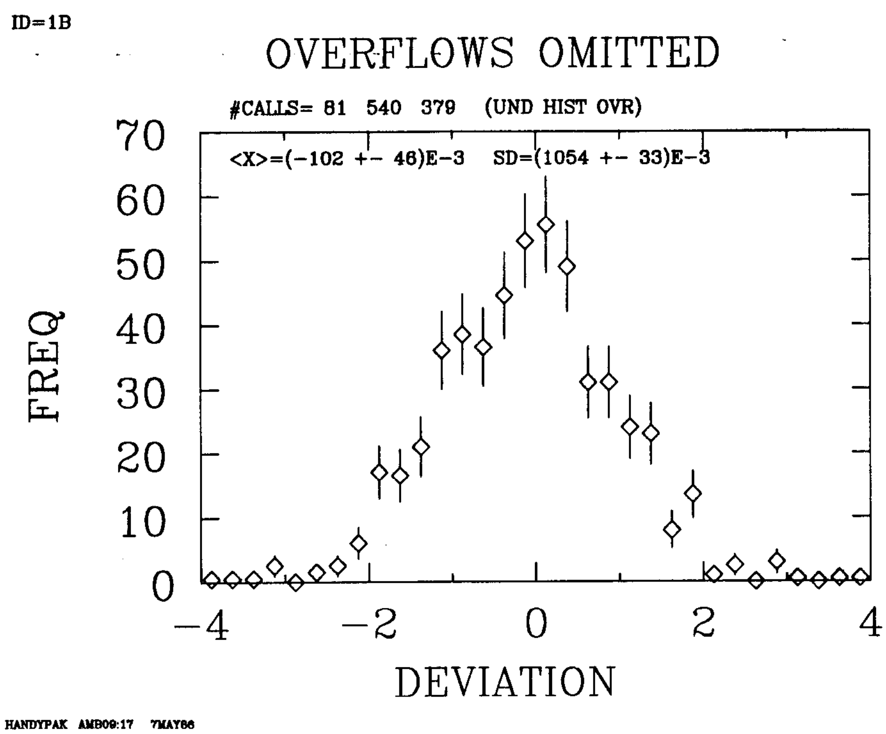


Figure 5.10 1-D Graphic Hist, with overflows omitted from edge bins (HOMIT set true), NCALLS, and statistics.

*

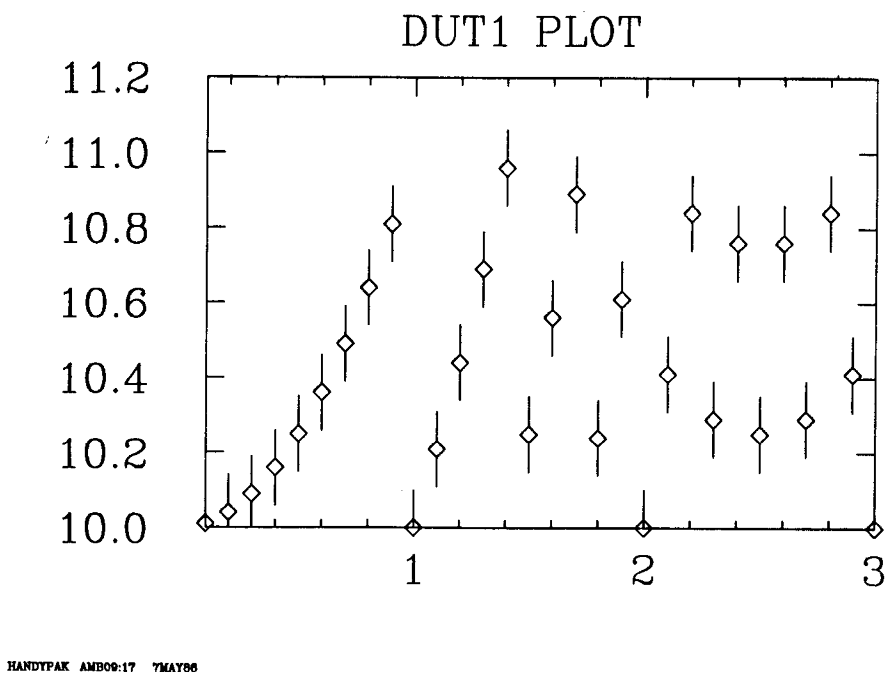


Figure 5.11 DUT1 Graphic Plot, normal auto scale factors.

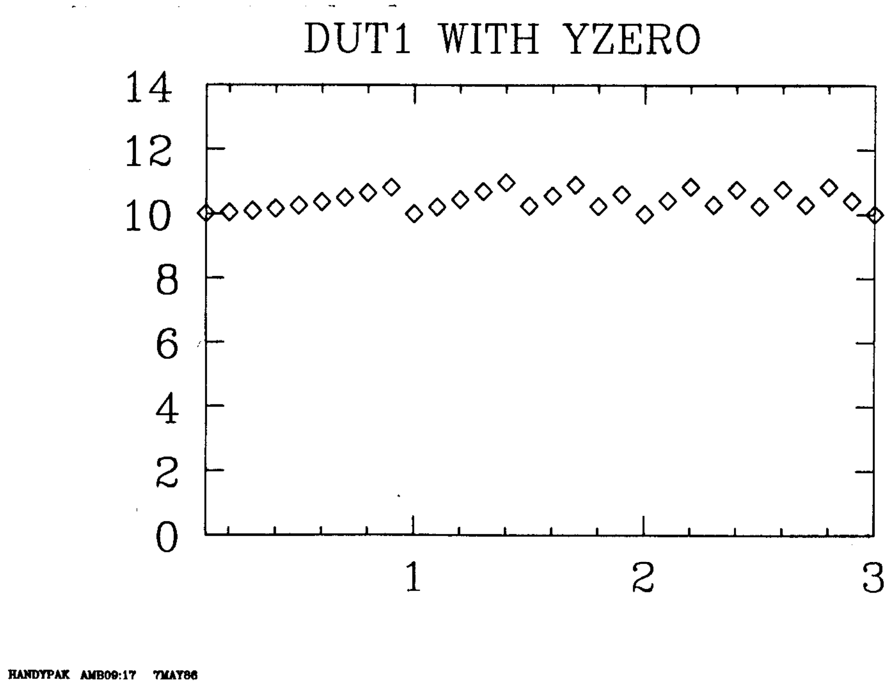


Figure 5.12 DUT1 Graphic Plot as above, with auto scale factors and zero forced on Y axis (YZERO set true).

*

AMB09:17 DUT2 PLOT 7 MAY 1986 9:17

Z.E -3	J	1	2	3	4	5	6	7	8	9	10	11	12	
I	X \ Y	-2.5	-2.0	-1.5	-1.0	-.5	.0	.5	1.0	1.5	2.0	2.5	3.0*E 0	YTOT.E -1
1	-2.2				2	9	31	77	127	141	104	52	17	6
2	-2.			1	8	38	127	283	423	423	283	127	38	18
3	-1.8		1	5	34	150	452	910	1228	1111	674	274	75	49
4	-1.6		2	21	127	516	1403	2556	3122	2556	1403	516	127	124
5	-1.4	1	9	77	423	1550	3813	6287	6948	5147	2556	851	190	279
6	-1.2	3	34	248	1228	4076	9072	13534	13534	9072	4076	1228	248	564
7	-1.	10	104	697	3122	9379	18888	25496	23069	13992	5689	1550	283	1023
8	-.8	31	283	1713	6948	18888	34415	42035	34415	18888	6948	1713	283	1666
9	-.6	83	674	3688	13534	33287	54881	60653	44933	22313	7427	1657	248	2434
10	-.4	190	1403	6948	23069	51342	76593	76593	51342	23069	6948	1403	190	3191
11	-.2	382	2556	11456	34415	69304	93551	84648	51342	20874	5689	1039	127	3754
12	.0	674	4076	16530	44933	81873	100000	81873	44933	16530	4076	674	75	3962
13	.2	1039	5689	20874	51342	84648	93551	69304	34415	11456	2556	382	38	3753
14	.4	1403	6948	23069	51342	76593	76593	51342	23069	6948	1403	190	17	3189
15	.6	1657	7427	22313	44933	60653	54881	33287	13534	3688	674	83	7	2431
16	.8	1713	6948	18888	34415	42035	34415	18888	6948	1713	283	31	2	1663
17	1.	1550	5689	13992	23069	25496	18888	9379	3122	697	104	10	1	1020
18	1.2	1228	4076	9072	13534	13534	9072	4076	1228	248	34	3		561
19	1.4	851	2556	5147	6948	6287	3813	1550	423	77	9	1		277
20	1.6	516	1403	2556	3122	2556	1403	516	127	21	2			122
21	1.8	274	674	1111	1228	910	452	150	34	5	1			48
22	2.	127	283	423	423	283	127	38	8	1				17
23	2.2	52	104	141	127	77	31	9	2					5
24	2.4	18	34	41	34	18	7	2						2
	E 0													
XTOT.E -3		11803	50974	159011	358357	583501	686457	583484	358323	158970	50940	11785	1966	30156

Figure 5.13 2-D Line Printer Plot by DUT2, default options.

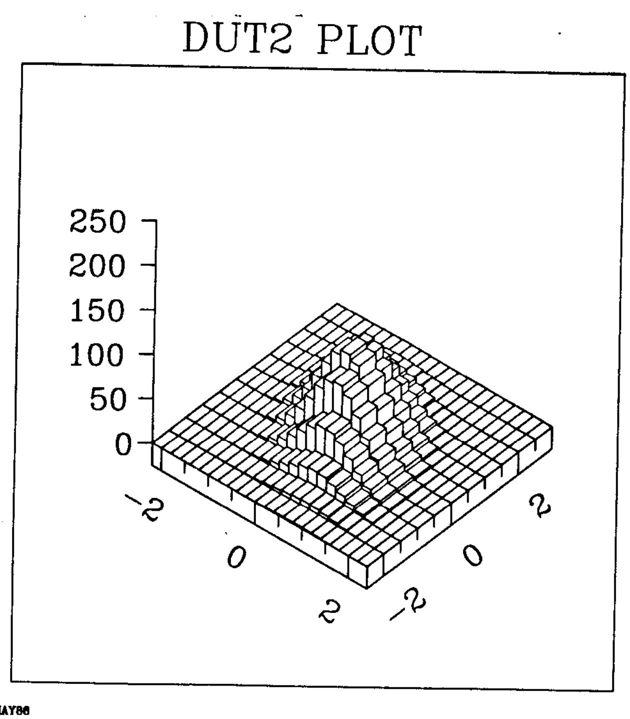


Figure 5.14 2-D Graphic Plot by DUT2, default options.

*

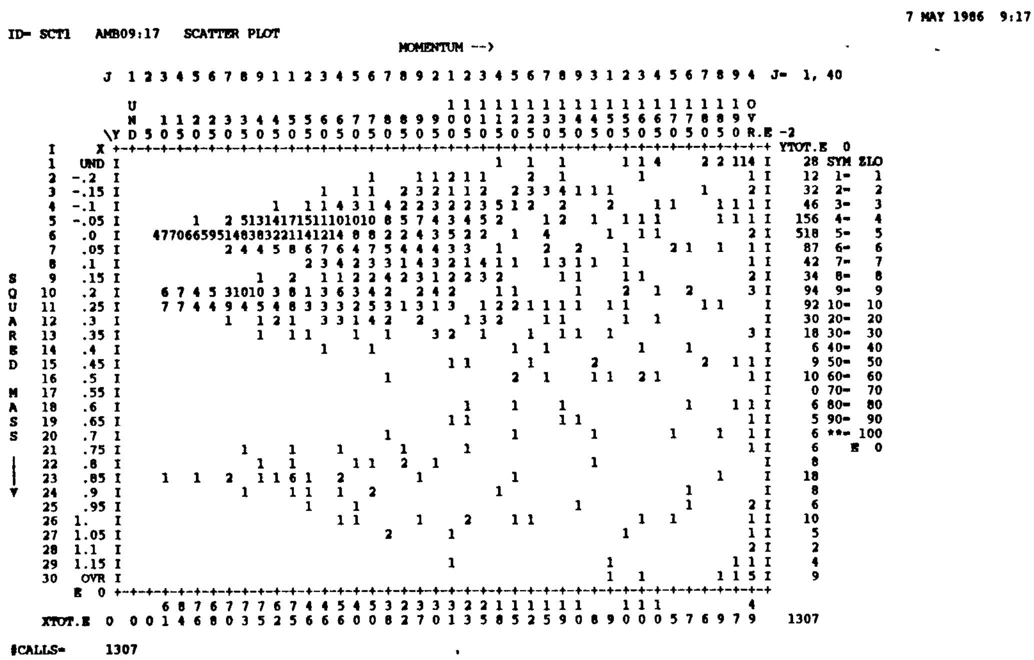


Figure 5.15 2-D Line Printer Scatterplot, default options.

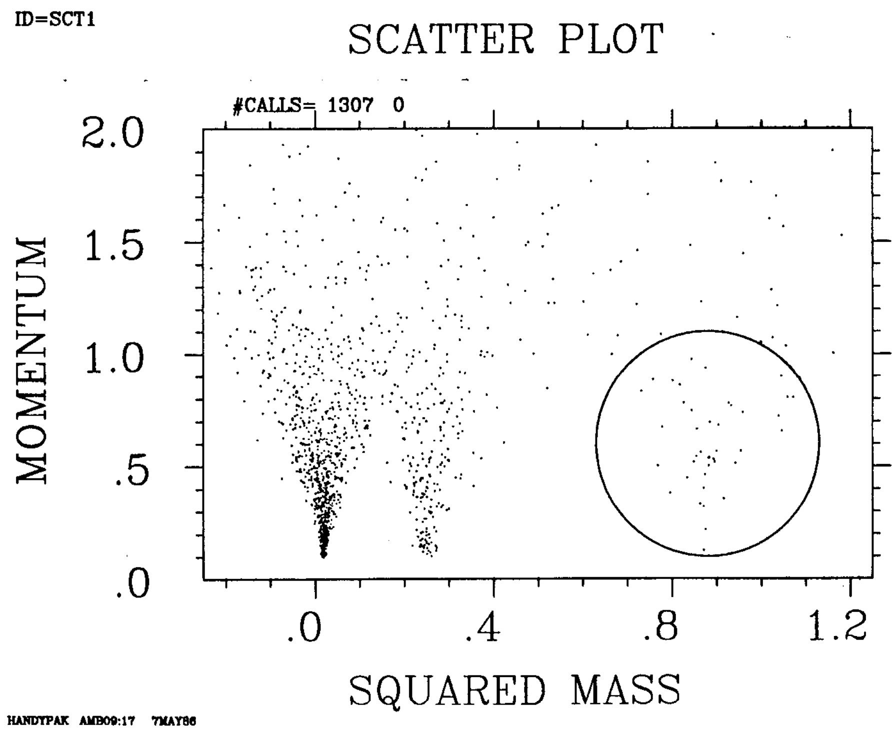


Figure 5.16 Scatterplot on Graphic Device, with a contour curve.

*

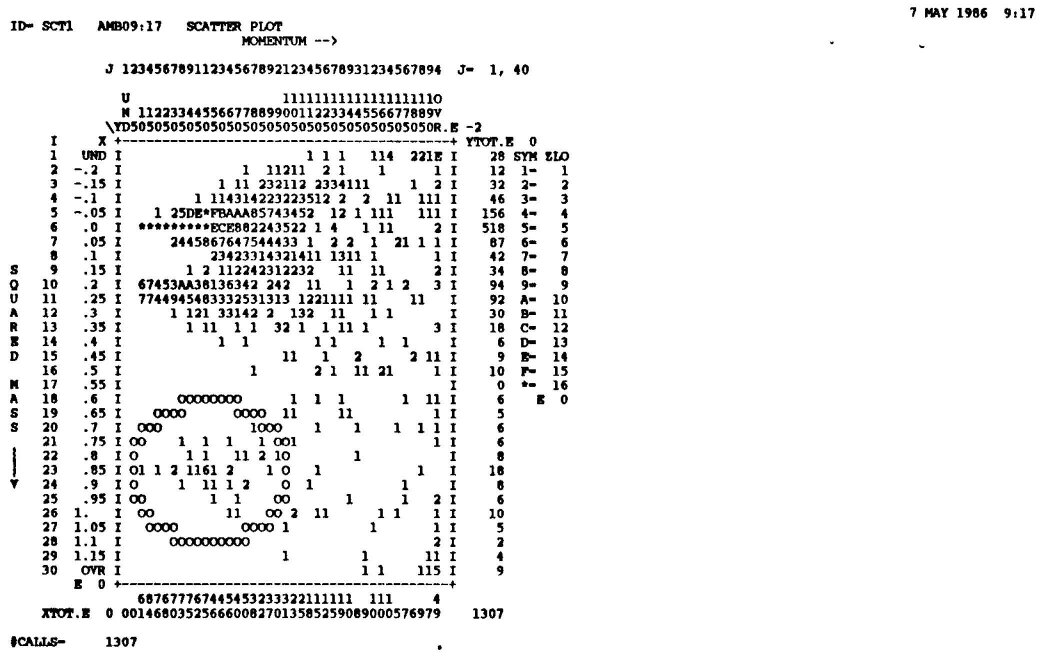


Figure 5.17 LP Scatterplot as in 5.15, but with manual scaling (L2ZAUT and L2CAUT set false), and a contour curve.

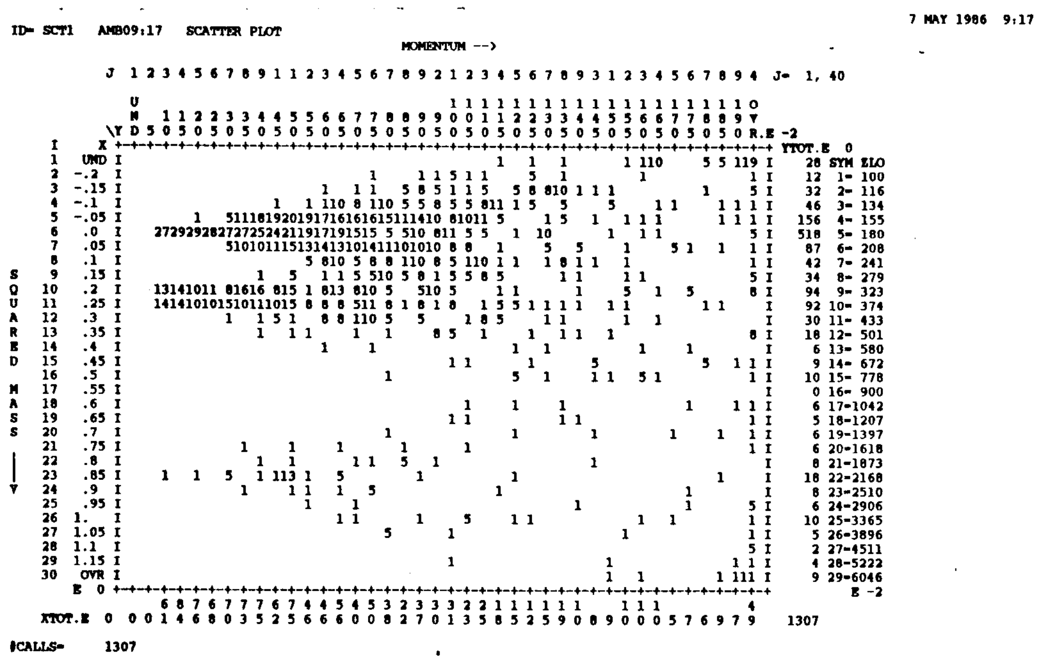


Figure 5.18 LP Scatterplot as in 5.15, but with log scales for Z (ZLOG set true).

*

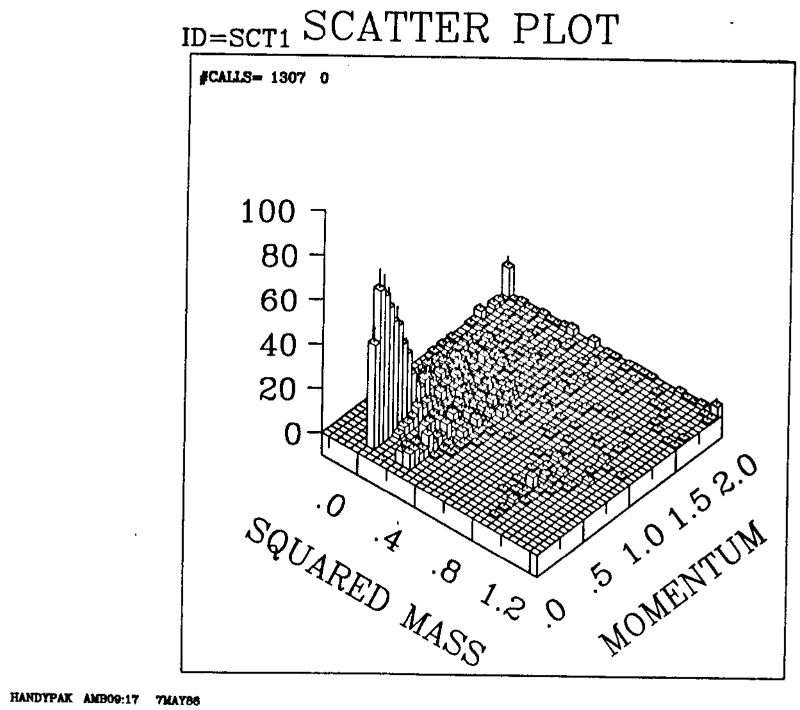


Figure 5.19 2-D Histogram in isometric format (ISOMETRIC set to 1).

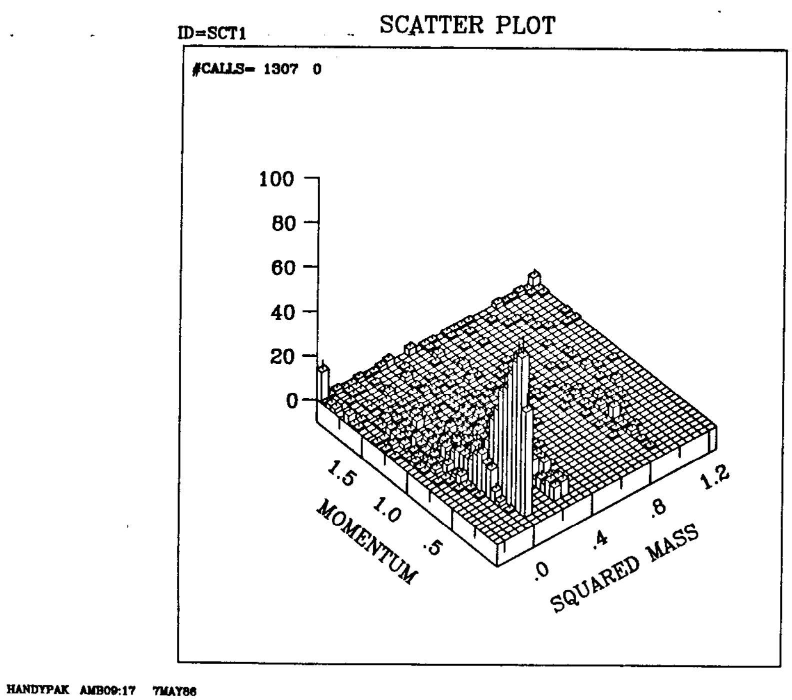


Figure 5.20 2-D Histogram, rotated by -80 degrees (ROTANGLE -40.)

*

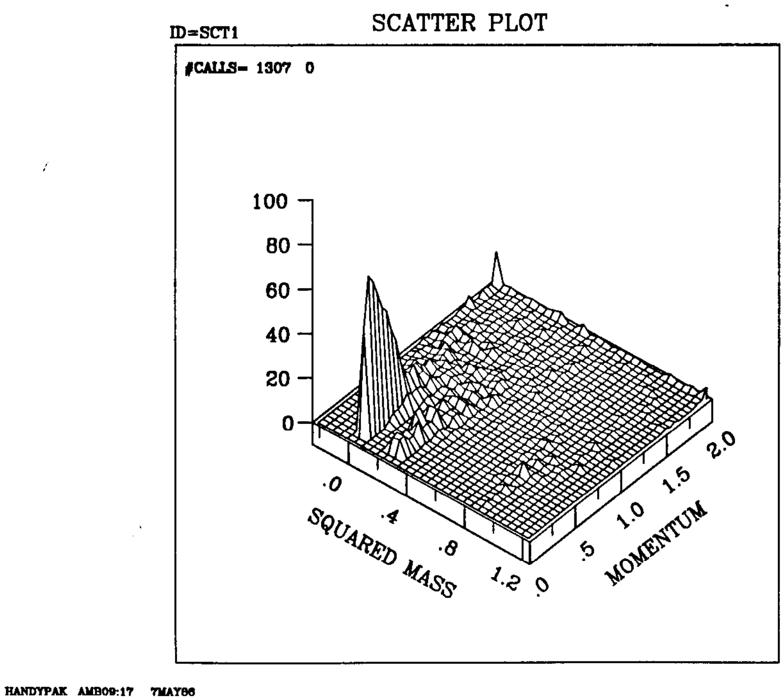


Figure 5.21 2-D Mesh plot (2DLINE set to 2).

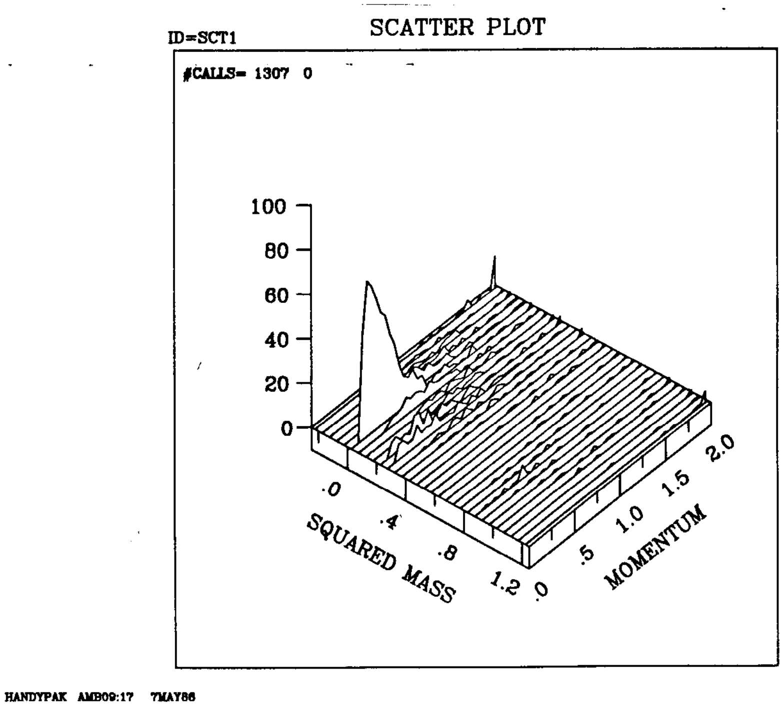


Figure 5.22 2-D Mesh slices along Y axis (2DLINE 1).

*

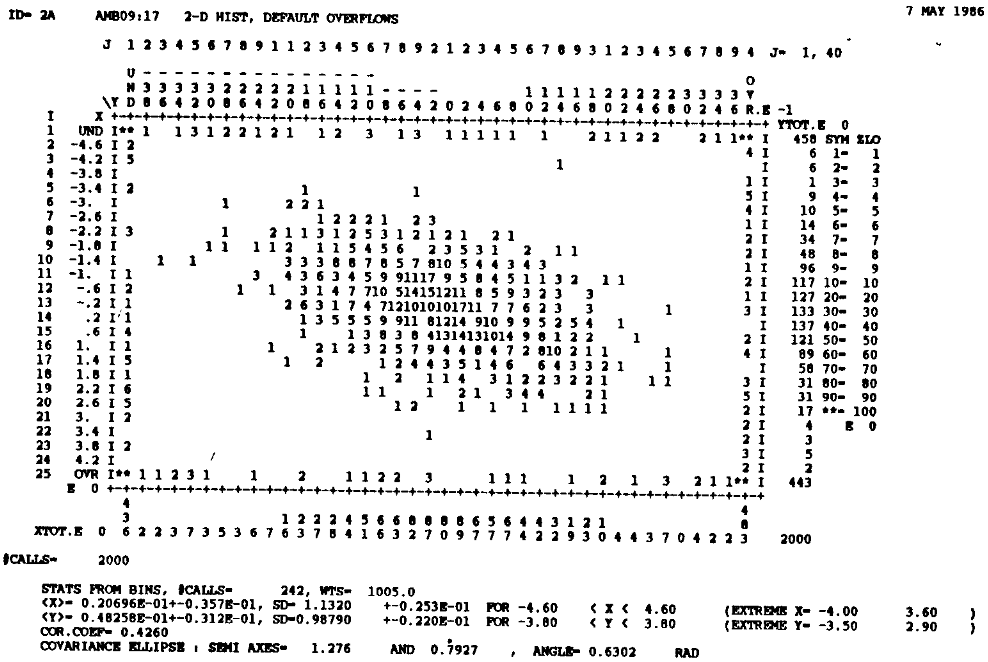


Figure 5.23 2-D Line Printer Hist, with overflows in edge bins, by default.

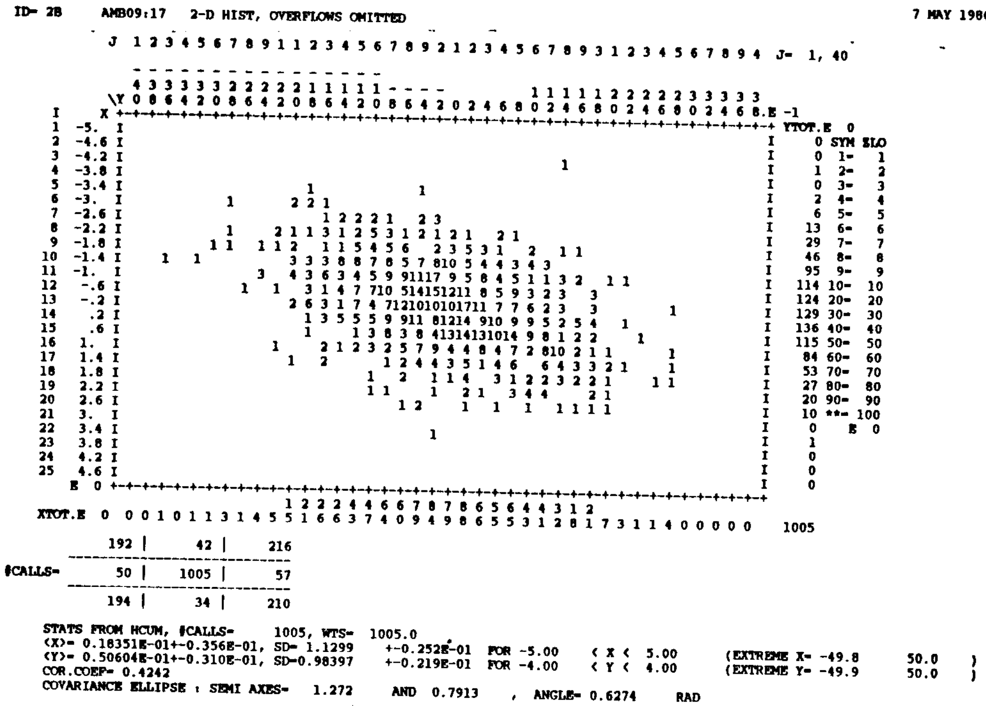
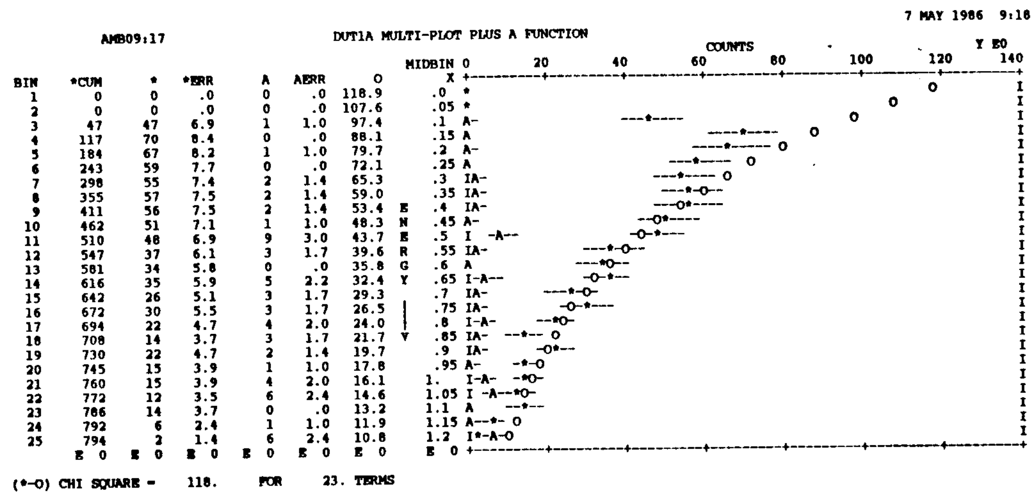


Figure 5.24 2-D Line Printer Hist, with overflows omitted from edge bins (HOMIT set true).

*



BIN	*CUM	*	*ERR	A	AERR	O	MIDBIN X
1	0	0	.0	0	.0	118.9	.0
2	0	0	.0	0	.0	107.6	.05
3	47	47	6.9	1	1.0	97.4	.1
4	117	70	8.4	0	.0	88.1	.15
5	184	67	8.2	1	1.0	79.7	.2
6	243	59	7.7	0	.0	72.1	.25
7	298	55	7.4	2	1.4	65.3	.3
8	355	57	7.5	2	1.4	59.0	.35
9	411	56	7.5	2	1.4	53.4	.4
10	462	51	7.1	1	1.0	48.3	.45
11	510	48	6.9	9	3.0	43.7	.5
12	547	37	6.1	3	1.7	39.6	.55
13	581	34	5.8	0	.0	35.8	.6
14	616	35	5.9	5	2.2	32.4	.65
15	642	26	5.1	3	1.7	29.3	.7
16	672	30	5.5	3	1.7	26.5	.75
17	694	22	4.7	4	2.0	24.0	.8
18	708	14	3.7	3	1.7	21.7	.85
19	730	22	4.7	2	1.4	19.7	.9
20	745	15	3.9	1	1.0	17.8	.95
21	760	15	3.9	4	2.0	16.1	1.
22	772	12	3.5	6	2.4	14.6	1.05
23	786	14	3.7	0	.0	13.2	1.1
24	792	6	2.4	1	1.0	11.9	1.15
25	794	2	1.4	6	2.4	10.8	1.2
	E 0	E 0	E 0	E 0	E 0	E 0	E 0

Figure 5.25 DUTlA Multi-Data Plot, with an analytic function.

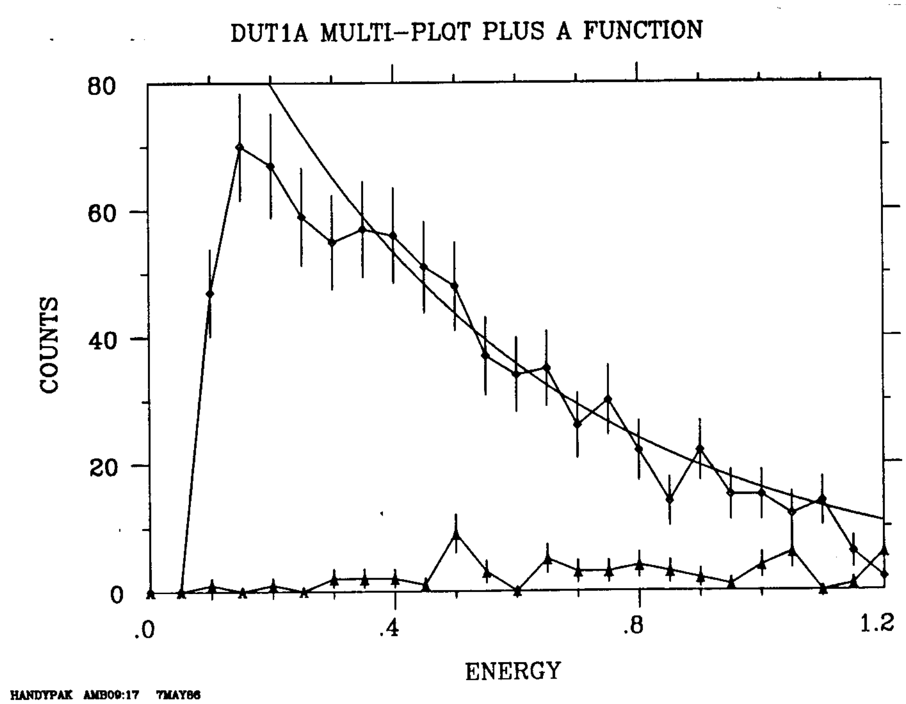


Figure 5.26 DUTlA Multi-Data Plot, on graphic device, LINE option, and an analytic function.

*

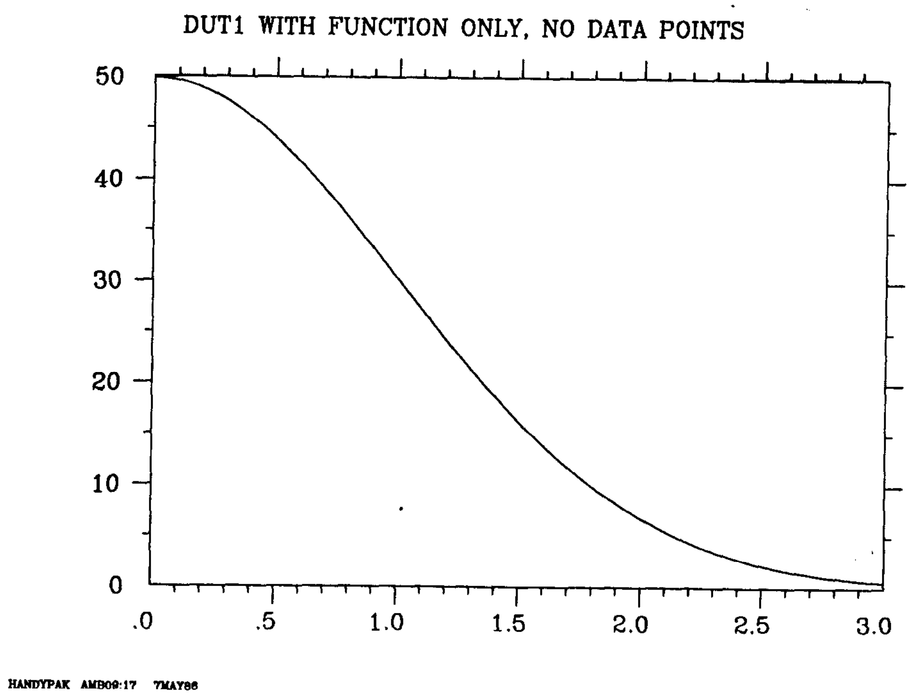


Figure 5.27 DUT1, with analytic function only, no data points.

```
ID= SCT1     AMB09:17          SLICE -- SCATTER PLOT                                              7 MAY 1986  9:18
                                                                          -- OVER Y(10-18)                        Y E0
                         LO EDGE 0         20        40        60        80        100       120       140       160       180      200
BIN    YCUM      Y           X +---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+
  1       0      0         UND *
  2       1      1         -.2 *                                                                                                    I
  3       6      5        -.15 IX*                                                                                                  I
  4      23     17         -.1 IXXXXXXX*                                                                                            I
  5     123    100        -.05 IXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX*                                                  I
  6     236    113          .0 IXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX*                                            I
  7     288     52         .05 IXXXXXXXXXXXXXXXXXXXXXXXXX*                                                                          I
  8     306     18          .1 IXXXXXXXX*                                                                                           I
  9     320     14  S      .15 IXXXXXX*                                                                                             I
 10     350     30  Q       .2 IXXXXXXXXXXXXXX*                                                                                     I
 11     382     32  U      .25 IXXXXXXXXXXXXXXX*                                                                                    I
 12     398     16  A       .3 IXXXXXXX*                                                                                            I
 13     402      4  R      .35 IX*                                                                                                  I
 14     404      2  E       .4 I*                                                                                                   I
 15     404      0  D      .45 *                                                                                                    I
 16     405      1          .5 *                                                                                                    I
 17     405      0  M      .55 *                                                                                                    I
 18     405      0  A       .6 *                                                                                                    I
 19     405      0  S      .65 *                                                                                                    I
 20     406      1  S       .7 *                                                                                                    I
 21     409      3         .75 I*                                                                                                   I
 22     414      5  |       .8 IX*                                                                                                  I
 23     424     10  |      .85 IXXXX*                                                                                               I
 24     429      5  V       .9 IX*                                                                                                  I
 25     431      2         .95 I*                                                                                                   I
 26     433      2        1.   I*                                                                                                   I
 27     435      2        1.05 I*                                                                                                   I
 28     435      0        1.1  *                                                                                                    I
 29     435      0        1.15 *                                                                                                    I
 30     435      0         OVR *                                                                                                    I
      E 0    E 0          E  0 +---------+---------+---------+---------+---------+---------+---------+---------+---------+---------+
```

Figure 5.28 Slice of a 2-D Histogram, on line printer, with HIST option, no error bars, and 4 tic intervals in Y.

*

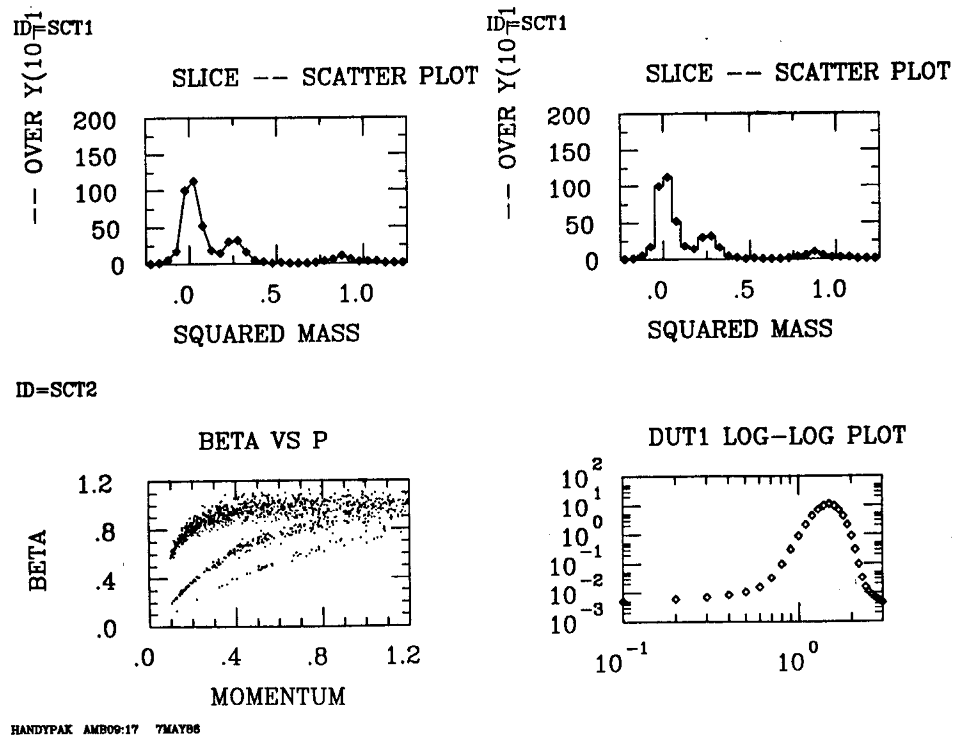


Figure 5.29a Four Quadrant Plot, generated by HSLICE, HOUT, and DUT1, using options OVER, QUADRANT, etc, and fixed character sizes (CHSCALE 0).

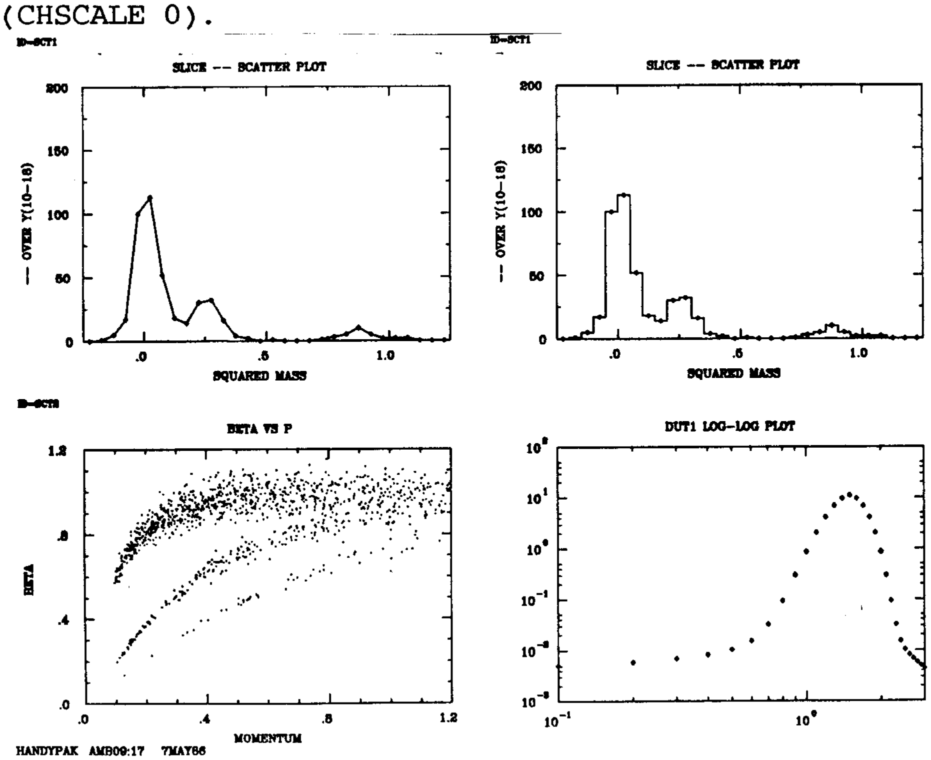


Figure 5.29b As above, but using character sizes scaled with the plot size, the default (CHSCALE set to 1).

*

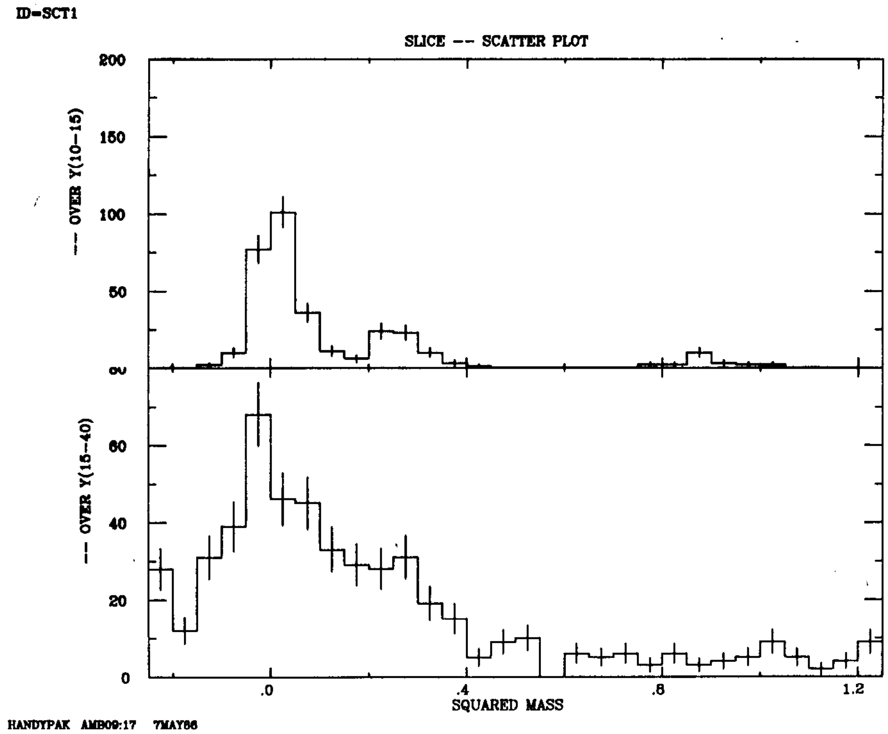


Figure 5.30 Two Half Frame Plots, butted together, generated by HOUT using options OVER, QUADRANT, and MARGIN.

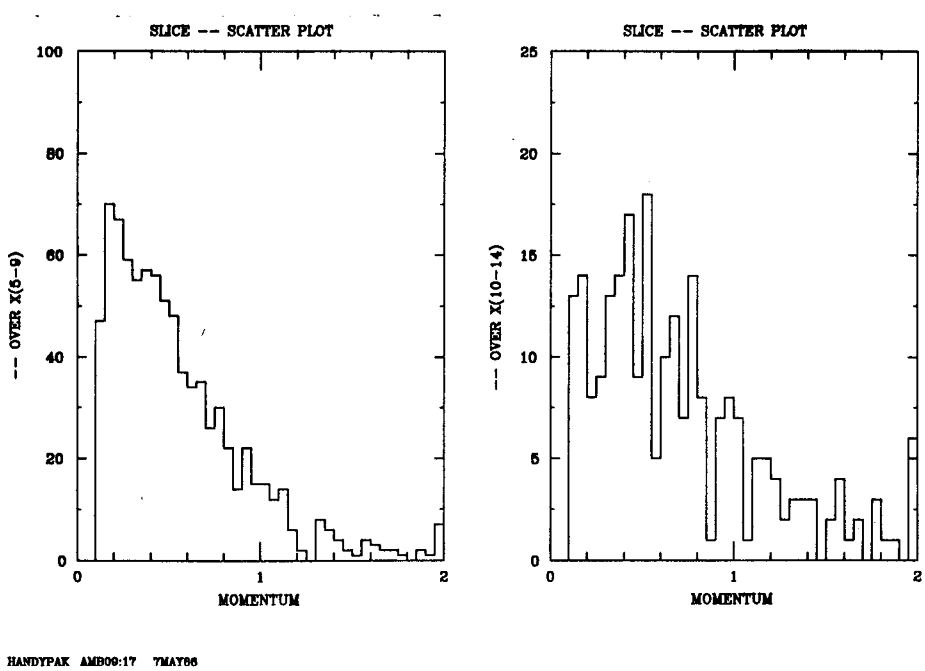


Figure 5.31 Two Half Frame Plots, generated by HOUT using options OVER, and QUADRANT.

*

DIRECTORY OF HISTOGRAMS IN HCOM

ID	LOC	SIZE	MS	SCAT	STATS	X/Y	BINS	LOWEDGE	WIDTH	TITLE
1	392	92	I*2	N	Y	X	32	-4.0000	0.25000	EXPT-A;DEVIATION;FREQ@
1A	484	112	R*4	N	Y	X	32	-4.0000	0.25000	DEFAULT OVERFLOWS;DEVIATION;FREQ@
1B	596	100	R*4	N	N	X	32	-4.0000	0.25000	OVERFLOWS OMITTED;DEVIATION;FREQ@
SCT1	696	682	I*2	Y	N	X	30	-0.25000	0.50000E-01	SCATTER PLOT;SQUARED MASS;MOMENTUM@
						Y	40	0.00000E+00	0.50000E-01	
SCT2	1378	80	I*2	Y	N	X	1	0.00000E+00	1.2000	BETA VS P;MOMENTUM;BETA@
						Y	1	0.00000E+00	1.2000	
2A	1458	580	I*2	N	N	X	25	-5.0000	0.40000	2-D HIST, DEFAULT OVERFLOWS@
						Y	40	-4.0000	0.20000	
2B	2038	614	I*2	N	Y	X	25	-5.0000	0.40000	2-D HIST, OVERFLOWS OMITTED@
						Y	40	-4.0000	0.20000	

348 HDEF WORDS LEFT IN HCOM(3000)

Figure 5.32 Directory List of all histograms as generated by HGET.

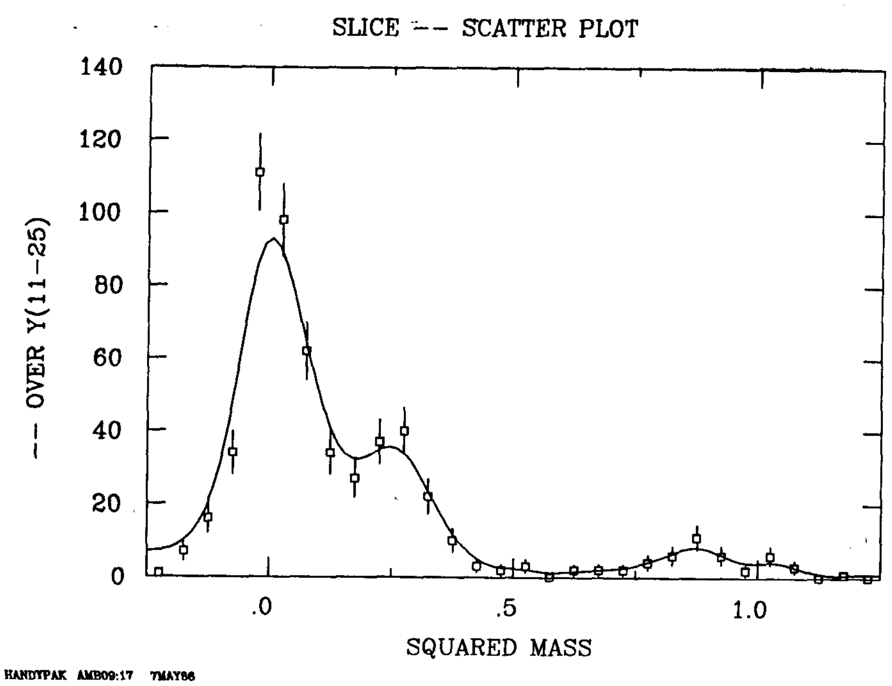


Figure 5.33 Example of a smooth curve drawn for the data points (SMOOTH set true).

* ID=PRP1

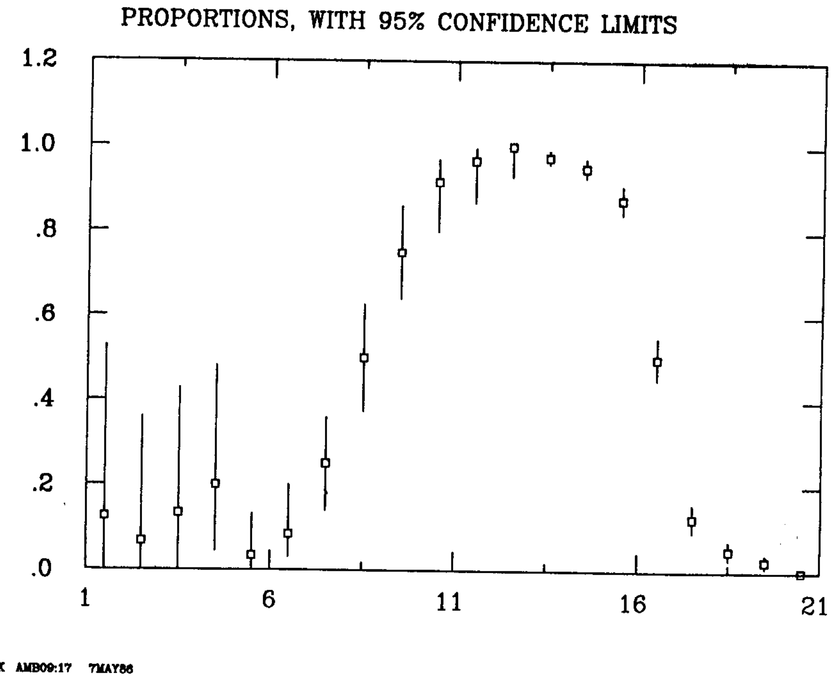


Figure 5.34 Plot of proportions (efficiencies), using the 'EFF' mode of storage. Error bars (asymmetric) are the 95% confidence limits.

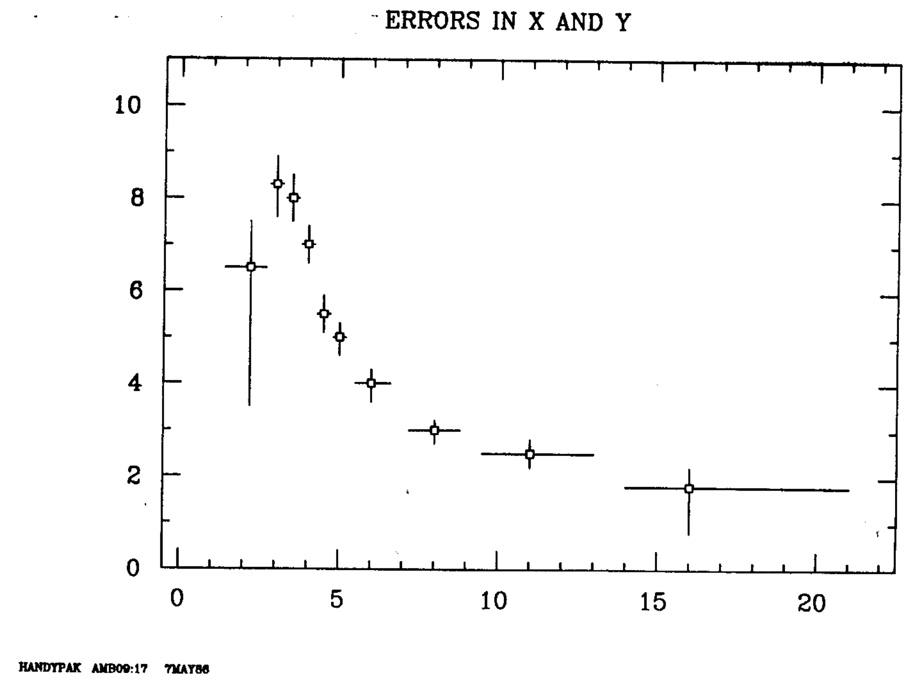


Figure 5.35 Example of a plot with errors in X as well as Y (ERROR set to 10).

5. EXAMPLES

Section 6

CONVERSATIONAL HANDYPAK

6.1 CHP2

Conversational HandyPak (CHP2) is a program module which reads commands from a terminal or a file, and calls the appropriate Handypak routine to execute the commands. An EXEC file on VM or a COM file on the VAX are provided to set up the environment before the load module is entered. A prompt 'CHP>' is issued when the program is ready to read a command from the terminal. CHP2 commands are similar to the subroutine calls in Handypak, except that the arguments should not be enclosed in parenthesis and the commas between the arguments may be omitted. There are few additional commands needed to interact with the operating system, described later. Other commands (SET, SHOW, and RESET) are variations for the DOPT and HOPTN calls.

In a typical application, a batch job defines and accumulates histograms and writes them to a file with HWRITE('ALL '). Then CHP2 is run from a terminal to read and edit the histograms. Assume that the file HTST HCOM A was created in batch, with a filedef as specified in the Appendix C. Note that units 20 and 21 have built RECFM, LRECL, and BLOCK values in CHP2 as per the Appendix. CHP2 would then be run on VM as follows:

```
FILEDEF 21 DISK HTST HCOM A
CHP2
```

Then in response to the CHP> prompt, entering

```
HREAD ALL
```

will read in all the histograms from the default unit (21). Then

```
HOUT ID
```

would output the ID'th histogram onto the terminal (unit 6), and when finished would prompt for the next command. If a new option were desired, say a log scale, then

```
DOPT 'YLOG' .TRUE.
HOUT ID
```

would output the histogram with a log scale onto the terminal. If you were logged onto a 4010 style terminal, then

```
HOUT ID TEK4010
```

would produce the graphic output on your terminal. At the end of the plot, CHP2 waits for a carriage return, then flips the screen back to alphanumeric mode, and waits for another command. Output may also be sent to the Versatec, for example

```
HOUT ALL VEP12FF
```

would output all the histograms to the VM Versatec printer (sent to the printer spool when you exit CHP2).

CMS commands may also be issued from within CHP2 by issuing the operating system command (OPS), for example

```
OPS FI 21 DISK HTST2 HCOM B
```

would issue "FI 21 DISK HTST2 HCOM B" to CMS.

With the above commands, one can access many histogram files, and add them with HREAD, or manipulate them with HCOMB.

CHP2 is terminated by

```
QUIT
```

Command lines may be continued over many input lines, by placing an & character at the end of each line to be continued.

The following summary of CHP2 commands is printed when the HELP or ? command entered.

```
CHP2 (Conversational HandyPak) command summary
    UPPER CASE means mandatory, lower case or [] is optional

DOPT      OPT [parm1 [parm2]]
dpSLCt    OPT [inter]
dpCLOs
hBlock    ID IC [n [a [i]]]
hCLR      ID
hCOMb     C1 ID1 OPER C2 ID2 ID3 TITLE
hCUM1     ID X W [e]      ('X W [e]' line(s) will continue CUM).
hCUM2     ID X Y W [e]    ('X Y W [e]' lines will continue CUM).
hDEF1     ID MS NX XL XW TITLE
hDEF2     ID MS NX NY XL YL XW YW TITLE
hDEL      ID
hGet      ID [opt]
hMap
hOPTN     OPT [parm [id]]
hOUt      ID [device]
hPntr     ID [item]
hSET1     ID X W [e]
hSET2     ID X Y W [e]
hSLIce    ID XORY [ibeg iwdth [device]]
hSPace
hSTat     ID OPT
dpTXt     'text' [x y] [SIZE s] [ANGLe a] [COLOR N]
hWrite    [id [lun [irew]]]
hREAd     [id [lun [irec [ictl [id2]]]]]
SET       OPT PARM1 [parm2] -- same as 'DOPT OPT PARM1 parm2'
SHow      OPT               -- same as 'DOPT GET,OPT'
RES       OPT               -- same as 'DOPT RES,OPT'
SET/ID    OPT PARM          -- same as 'HOPTN OPT PARM ID'
SHow/ID   OPT               -- same as 'HOPTN GET,OPT XX ID'
RES/ID    OPT               -- same as 'HOPTN RES,OPT XX ID'
OPS       CCCC...           -- send 'CCC..' to operating system
CInput    IUNIT             -- unit for input commands (deflt=5)
END       [unit]            -- signals end of input command file
HELp                        -- prints this command menu
Quit                        -- exit from CHP
```

The leading 'H' need not be given in the H-calls, and the 'DP' may also be omitted in the DP-calls. Quotes may be omitted on most character arguments, but are required for TITLE strings or compound device names, described later.

Although CHP2 is best suited for running off an existing histogram file, it is also possible to define and accumulate histograms at the terminal. Once an HCUM1 or HCUM2 is issued, the next command(s) may optionally omit 'HCUM1' and only supply accumulation data, for example

```
DEF1 1 LPT 20 0 0 'TEST INPUT;X;Y@'
CUM1 1 11. 2.3          ! first point
12. 4.1                 ! second point
13. 5.0                 ! etc.
14. 6.3
15. 4.4
16. 8.6
17. 7.4
18. 9.1
19. 7.7
20. 6.0
SET ERROR 0
SET LSUM 0
SET LCOL 60
OUT 1
```

would accumulate points with X=11., through 20., and produce

```
ID=     1                 TEST INPUT
                                    Y                Y E0
               LO EDGE 0         5         10           15
BIN       Y          X +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  1     2.3         11 I    *                          I
  2     4.1         12 I       *                       I
  3     5.0         13 I         *                     I
  4     6.3   X     14 I            *                  I
  5     4.4         15 I        *                      I
  6     8.6   |     16 I                *              I
  7     7.4   |     17 I              *                I
  8     9.1   V     18 I                 *             I
  9     7.7         19 I              *                I
 10     6.0         20 I           *                   I
      E  0      E  0 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Commands are read from unit 5, the terminal. Commands may be read from another unit number, by issuing

```
CINPUT N
```

where N is a logical unit number (0 to 99). Commands are then read from unit N until the end of file is reached, or an 'END' command is encountered. The END command may also optionally specify another unit number from which the next commands are to be read. (If no unit number is given on the END command, then unit 5 is used).

CHP2 may also be invoked with arguments specifying a file from which commands are to be taken, as well as the device to be used for graphic output (or an IMGN location). For example,

```
CHP2 TEST (IMCLA2
```

would read commands from the file "TEST CHP *", and output to the Imagen "IMCLA2". Further online help is displayed by

```
 CHP2 ?          (on VM)
@CHP2 ?          (on VAX)
```

Special options for some devices can be provided by a compound device string, as shown in the following example,

```
SLCT 'IMGN300,ROTAXIS' 0
```

This opens the Imagen device with axis rotated. The quotes are needed here, otherwise the comma is taken as a delimiter.

When output is directed to a graphic terminal, CHP2 uses escape strings for flipping to alpha or to graphic mode. The default strings are those for the Ann Arbor terminals. New escape strings needed for other terminals can be set from CHP2 using the ESCT and ESCG options in DOPT. Some command files for other terminals have been put on the U disk on VM, e.g.

```
AAGXL CHP U  -- for the Ann Arbor GXL
MAC   CHP U  -- for the MacIntosh
```

For example,

```
CHP2 MAC     -- will start CHP2 and load in the
                MacIntosh escapes strings.
```

Appendix A

CONTENTS OF HCOM

The contents of the common block /HCOM/ and also /HNODE/ are described here. A call to HPNTR(ID) sets the value of NODE in the /HNODE/ common block. This NODE pointer can then be used to access the contents of the ID'th histogram.

```
      COMMON / HNODE/ HASHID,NODE
      COMMON / HCOM / NNODES,IHCOM(2),NHCOM,NODEND,NODE1,HSHLOC,
     >                HSHCNT,HSHLEN,SCTLOC,SCTLEN,SCTUNI,SCTUSE,
     >                SCTREC,SCTTOT,SCTMAX,HPNEW
      INTEGER M(2000),HASHID,HSHLOC,HSHCNT,HSHLEN,SCTLOC,SCTLEN,
     >                SCTNUM,SCTUSE,SCTREC,SCTTOT,SCTMAX,HPNEW
      REAL R(1)
      INTEGER*2 H(1)
      LOGICAL*1 L(1)
      REAL*8 D(1)
      EQUIVALENCE (NNODES,M(1),L(1),H(1),R(1),D(1))
```

where:

HASHID	hash code of most recently referenced node
NODE	pointer to the most recently referenced node

CONTENTS OF HCOM

CONTROL AREA

M(1)	NNODES	total number of nodes
M(2)		reserved
M(3)		reserved
M(4)	NHCOM	size of HCOM in wds (2000)
M(5)	NODEND	pointer to free space in HCOM
M(6)	NODE1	pointer to first node in HCOM
M(7)	HSHLOC	pointer to hash table
M(8)	HSHCNT	pointer to hash multiplicity count
M(9)	HSHLEN	size of hash table in wds (137)
M(10)	SCTLOC	pointer to scatterplot buffer pool
M(11)	SCTLEN	size of scatterplot buffer in wds (100)
M(12)	SCTUNI	unit num of scatterplot file (19)
M(13)	SCTUSE	use status of scatterplot file
M(14)	SCTWDS	num of wds filled in scatterplot buffer
M(15)	SCTTOT	number of scatterplots

```
|   M(16) SCTMAX   maximum number of scatterplots                |
|   M(17) HPNEW    initialization flag for HCOM                  |
+----------------------------------------------------------------+
|                         HASH TABLES                            |
|                                                                |
|   M(HSHLOC+0)    pointer to node with hash code      0         |
|           +1     pointer to node with hash code      1         |
|            .                                                   |
|            .     (HSHLEN entries)                              |
|   M(HSHCNT+0)    num of nodes having hash code       0         |
|   M(HSHCNT+1)    num of nodes having hash code       1         |
|            .                                                   |
|            .     (HSHLEN entries)                              |
+----------------------------------------------------------------+
|                      SCATTERPLOT BUFFER                        |
|                                                                |
|   M(SCTLOC +1)   beginning of scatterplot buffer               |
|             .                                                  |
|       +SCTWDS    last filled word                              |
|       +SCTLEN    last word of scatterplot buffer               |
+----------------------------------------------------------------+
|                         NODE AREA *                            |
|                                                                |
|   M(NODE1+1)     first word of first node                      |
|   M(NODE1+2)     second "   "     "    "                       |
|          .       etc.                                          |
|          .         .                                           |
|   M(NODE +1)     first word of node selected by HPNTR(ID)      |
|   M(NODE +2)     second word                                   |
|          .       etc.                                          |
|          .         .                                           |
|                                                                |
|                  nodes follow one after the other              |
|                                                                |
+----------------------------------------------------------------+
|   M(NODEND+1)    backward pointer to last node in HCOM         |
|                  New node gets inserted here.                  |
|                                                                |
|                                                                |
+----------------------------------------------------------------+
```

* see next page for a description of a node.

CONTENTS OF A HISTOGRAM NODE

control block

M(NODE+ 1)	pointer to next node in this hash group
M(NODE+ 2)	hash code for this node
M(NODE+ 3)	pointer to next node in sequence
M(NODE+ 4)	ID
M(NODE+ 5)	node class 0=BLOCK, 1=1D, 2=2D
M(NODE+ 6)	A4ID - ID in A4 format
M(NODE+ 7)	size of node (wds)
M(NODE+ 8)	size of user block in node (wds)
M(NODE+ 9)	number of sub-pointers to follow
M(NODE+10)	offset for parameter block
M(NODE+11)	offset for statistics block, 0 if none
M(NODE+12)	offset for histogram bins, 0 if none
M(NODE+13)	offset for title block
M(NODE+14)	offset for display options block
M(NODE+15)	offset for user block, 0 if none

parameter block
MPARM = NODE+M(NODE+10)

M(MPARM+ 1)	IMS - mode of storage: 1=L*1, etc.
M(MPARM+ 2)	total words reserved for bin storage
M(MPARM+ 3)	HOMIT - 0(1) don't(do) omit overflows from edge bins of histogram
M(MPARM+ 4)	spare
M(MPARM+ 5)	number of X bins
R(MPARM+ 6)	lowest value of X
R(MPARM+ 7)	width of each X bin
R(MPARM+ 8)	highest value of X
M(MPARM+ 9)	number of Y bins
R(MPARM+10)	lowest value of Y
R(MPARM+11)	width of each Y bin
R(MPARM+12)	highest value of Y
M(MPARM+13)	scatter plot number (0 if none)
M(MPARM+14)	factor for scaling X in scatterplots
M(MPARM+15)	factor for scaling Y in scatterplots
M(MPARM+16)	CONTOUR address for drawing contours
M(MPARM+17)	NCONTOUR number of contours
M(MPARM+18)	spare (to keep even number of words)

statistics block
MSTAT = NODE+M(NODE+11)
MSTA2 = MSTAT/2

M(MSTAT+ 1)	0 (1) don't (do) use limits for stats
R(MSTAT+ 2)	minimum limit of X for statistics
R(MSTAT+ 3)	maximum limit of X for statistics
R(MSTAT+ 4)	lowest X used in statistics
R(MSTAT+ 5)	highest X used in statistics

```
R(MSTAT+ 6)   central value X0 for moments
D(MSTA2+ 4)   SUM 1
D(MSTA2+ 5)   SUM W
D(MSTA2+ 6)   SUM W*(X-X0)
D(MSTA2+ 7)   SUM W*(X-X0)**2
M(MSTAT+15)   spare
R(MSTAT+16)   minimum limit of Y for statistics
R(MSTAT+17)   maximum limit of Y for statistics
R(MSTAT+18)   lowest Y used in statistics
R(MSTAT+19)   highest Y used in statistics
R(MSTAT+20)   central value Y0 for moments
D(MSTA2+11)   SUM W*(Y-Y0)
D(MSTAT+12)   SUM W*(Y-Y0)**2
D(MSTAT+13)   SUM W*(X-X0)*(Y-Y0)
---------------------------------------------------------------
                    histogram bins block
                  MBINS = NODE+M(NODE+12)

M(MBINS+ 1)   beginning of histogram bins
          .
          .
              (number of calls follow after bin storage
               in INTEGER*4 storage)
---------------------------------------------------------------
                        title block
                  MTITL = NODE+M(NODE+13)

M(MTITL+ 1)   beginning of title string
          .
          .
---------------------------------------------------------------
                  display parameter block
                  MDISP = NODE+M(NODE+14)

R(MDISP+ 1)   normalization factor (for HY,HZ, etc.)
M(MDISP+ 2)   1 for do, 0 for don't output statistics
M(MDISP+ 3)   FUNCTION address used for analytical curve
M(MDISP+ 4)   NFUNC  number of functions
M(MDISP+ 5)   Error value for an empty bin (0 or 1)
M(MDISP+ 6)   YALT - alternate output mode in HY, HZ
M(MDISP+ 7)   ISOMETRIC plot if 1 (default is 0).
R(MDISP+ 8)   XMIN  (manual scale factors start here)
R(MDISP+ 9)   DX
R(MDISP+10)   XTICS
R(MDISP+11)   XSUBS
R(MDISP+12)   YMIN
R(MDISP+13)   DY
R(MDISP+14)   YTICS
R(MDISP+15)   YSUBS
R(MDISP+16)   ZMIN  (2-D scale factors continue)
R(MDISP+17)   DZ
R(MDISP+18)   ZTICS
R(MDISP+19)   ZSUBS
R(MDISP+20)   ROTATION ANGLE
```

```
|  R(MDISP+21)     ELEVATION ANGLE                              |
|  M(MDISP+17..    packed display options                       |
|  M(MDISP+31/37) last display option word for 1-D/2-D.         |
+---------------------------------------------------------------+
|                       user block                              |
|                 MUSER = NODE+M(NODE+15)                        |
|                                                               |
|  M(MUSER+ 1)     first word of user block                     |
|          .       (M(NODE+8) entries)                          |
|                                                               |
+---------------------------------------------------------------+
```

Appendix B

CONTENTS OF DPMODE

This describes the contents of the COMMON block /DPMODE/, which stores all the options for the HANDYPAK display routines (DUT1, DUT2, DUT1A, etc.). A BLOCK DATA within HANDYPAK initializes the variables to the values shown in the 'default' column. Users may change the values to their own needs. Some items marked by a '*' are filled by the DUT routines for use by the user. The right most column indicates which routines use the particular option, where L1, L2, G1, and G2 refer to the 1-D and 2-D line printer, 1-D and 2-D graphic plots respectively, and 11 means both L1 and G1 and 22 means both L2 and G2. The column marked 'Wrd' gives the word number when the following declaration is used:

```
COMMON / DPMODE / IMODE(122), RMODE(50), DEVTYP(20)
INTEGER IMODE
REAL RMODE
REAL*8 DEVTYP
```

Alternately, a declaration in terms of the names in the 'Name' column can be made. File DPMODE COMMON U on VM contains such a declaration.

CONTENTS OF /DPMODE/

Wrd	Name	Default	Description	Used by
			IMODE - integer variables	
1	IDEV	2	device number (index to DEVTYP)	
2	NDEVS	0	number of opened graphic devices	
3	LPUNIT	6	line printer unit number	
4	ERUNIT	6	error message unit number	
5	IAUTO	1	auto/manual scaling for F axis : FMIN DF FTICS FSUB 0 - man man man man 1 - auto auto deflt deflt 2 - auto auto auto deflt 3 - auto auto auto auto	G1 G2

6	"	2	auto/manual scaling for X axis (same meaning as for F axis)	G1 G2
7	"	2	auto/manual scaling for Y axis : (same meaning as for F axis)	G2
8	ILOG	0	log/linear scales for F axes 0 - linear, 1 - log	G1 G2
9	"	0	log/linear scales for X axes 0 - linear, 1 - log	11 G2
10	"	0	log/linear scales for Y axes 0 - linear, 1 - log	G2
11	IZERO	0	zero point for auto F scales 0 - don't, 1 - include zero	G1 G2
12	"	0	zero point for auto X scales 0 - don't, 1 - include zero	11 G2
13	"	0	zero point for auto Y scales 0 - don't, 1 - include zero	G2
14	spare	0	spare	
15	IEDGE	1	usage of edge X bins in F scaling, and in calculating stats from bins 0 - use, 1 - omit edges	11 G2
16	"	1	usage of edge Y bins in F scaling, and in calculating stats from bins 0 - use, 1 - omit edges	G2
17	G1TIC	0	default # tic marks along X axis	G1
18	"	0	default # subtic marks in X axis	G1
19	"	0	default # tic marks along Y axis	G1
20	"	0	default # subtic marks in Y axis	G1
21	G2TIC	5	default # tic marks along X axis	G2
22	"	0	default # subtic marks in X axis	G2
23	"	5	default # tic marks along Y axis	G2
24	"	0	default # subtic marks in Y axis	G2
25	"	5	default # tic marks along Z axis	G2
26	"	0	default # subtic marks in Z axis	G2
27	ISCALE	*	value of first X tic mark	G1 G2
28	"	*	increment between X tic marks	G1 G2
29	"	*	exponent for X tic mark	G1 G2
30	"	*	value of first Y tic mark	G1 G2
31	"	*	increment between Y tic marks	G1 G2
32	"	*	exponent for Y tic mark	G1 G2
33	"	*	value of first Z tic mark	G1 G2
34	"	*	increment between Z tic marks	G1 G2
35	"	*	exponent for Z tic mark	G1 G2
36	G1LINS	1	type of lines used with points 0 - no lines, points only 1 - lines between points 2 - histogram mode	G1
37	ERRS	1	error bar selection 0 - no error bars 1 - put in Y error bars other cases are described in DOPT under the ERROR option.	11
38	spare	0	spare	
39	LOWIN	0	describes input coords to DUT's 0 - center, 1 - low edge	G1 G2

40	CENTER	1	plotting position for hist bin 0 - low edge, 1 - center	G1 G2
41	FRAME	1	controls drawing frame and axes 0 - don't, 1 - draw frame	G1 G2
42	OVRPLT	0	controls over-plotting 0 - don't overplot 1 - do overplotting	G1
43	NOVRG	*	counts num of graphic overplots	G1
44			spare	
45	PLTCHR	' '	plot character	11
46	"	'A '	plot character on next overplot	11
47	"	'B '	plot character on next overplot	11
48	"	'C '	plot character on next overplot	11
49	"	'D '	plot character on next overplot	11
50	"	'E '	plot character on next overplot	11
51	"	'F '	plot character on next overplot	11
52	"	'G '	plot character on next overplot	11
53	"	'H '	plot character on next overplot	11
54	"	'I '	plot character on next overplot	11
55	RESCAL	0	rescaling between overplots 0 - don't, 1 - do rescale	
56	SLICE	0	controls slicing from 2-D to 1-D 0 - don't, 1 - do slicing	22
57	SLDIR	0	direction of slice 0 - along X axis 1 - " Y "	22
58	SLBEG	0	beginning bin for width of slice	22
59	SLWDTH	0	number of bins in width of slice (0 means all bins)	22
60	G2ANGS	0	manual/default isometric angles 1 - man, 0 - use ROTANG/ELVANG	G2
61	SLFLAG	0	internal slice flag in DUT2/DUT1	
62	G2LINE	0	type of isometric plot 0 - along X, 1 - along Y axis	G2
63	Spare		 0 - don't, 1 - do print sum	
64	L1GRAT	0	controls graticules 0 - don't, 1 - do print grats	L1
65	L1FILL	0	controls filling of plots on LP 0 - don't fill, 1 - fill	L1
66	LCOL	133	max number of columns per line	L1 L2
67	LSUMS	1	controls YCUM and TOTALS 0 - don't, 1 - do print sum	L1 L2
68	L2CAUT	1	manual/auto scaling of col/bin 0 - auto, 1 - manual	L2
69	L2ZAUT	1	scale Z in 2-D LP scatterplot 0 - don't, 1 - do auto scaling	L2
70	L2COL	4	number of columns per bin	L2
71	L2SYM	10	num of symbols (contours) in Z	L2
72	L2XXX	*		
73	"	*		
74	PNTSYM	0	symbol used in scatterplots 0 - point, 1-10 special symbols	G2
75	IDARG	0	ID (either integer or hollerith)	L2

76	IDA		ID (in A4-format)	L2
77	L2PGE	60	lines/page	L2
78	ISTATS	0	controls outputting of stats 0 - don't write stats 1 - write stats from XAVG, etc. 2 - fill stats, but don't write 3 - fill and write stats	L1 L2
79	IPLOT	1	controls outputting of plots 0 - don't, 1 - do the plots	11 22
80	EJECT	1	controls page ejection 0 - don't, 1 - do each hist	L1 L2
81	TICDIR	0	tic direction 0 - outward, 1 - inward	G1
82	INTER	0	batch (0)/interactive (1) mode	G1 G2
83	IMULT	1000	factor used for decoding number of multi-plots from num of points	11
84	STATYP	' '	Statistics type ('HCUM' or 'BINS')	L1 L2
85	ICALLS	0	If 0 (1), number of calls are not (are) printed.	L1 L2
86	NWCALL	0	Gives number of words in NCALL	L1 L2
87-	NCALL	0	Contain number of calls, ovr/under	L1 L2
96		0	spare	
97	JBNAM	' '	Job name (8 characters)	11 22
99	JBDAT	' '	Job date (8 characters)	G1 G2
101	PNTDEN	4	point density in scatterplots 1 to 5 = faint to dark	G2
102	LINDEN	4	line density on versatec plots 1 to 5 = faint to dark (0 removes GENIL option to UG)	G1 G2
103	SYMSIZ	-1	symbol size used for data points -1 uses value of TITLSZ instead. 0,1,2,.. for smallest to larger	G1
104	TITLSZ	2	character size in titles and labels, 0 is small, 1,2,.. for larger	
105	FULSCR	1	full screen, 0 for square size.	G1 G2
106	IDFONT	1	flag for outputting ID font 0 - dont, 1 - do it	G1 G2
107	COLOR	0	color index for data points, errors, and lines.	G1
108	SMOOTH	0	smoothing flag 0 - dont, 1- do smoothing	11 22
109	ITRACE	1	traceback flag 0 - dont, 1 - do give traceback when illegal arguments are found	all
110	OVPRT	0	over-print flag 0 - dont, 1 - do allow over-printing ('+' in col1) in titles	L1 L2
111	ISLIM1	0	1'st X bin for stats	11 22
112	ISLIM2	0	last X bin for stats	11 22
113	ISLIM3	0	1'st Y bin for stats	11 22
114	ISLIM4	0	last Y bin for stats (0 means all bins for that end)	11 22
115	CHSCAL	1	scaling size for characters 0 - fixed, 1 - scale with GSIZE	G1 G2

116	NUMFMT	1	number format for labels, lists 0 - integer only 1 - real, engineering notation.	11 22
117	JBFONT	3	job name/date and HANDYPAK font 0 - neither, 1- job name/date 2 - font, 3- both.	G1 G2
118	CHFMT	0	Char 0=basic, 1=stroke, 2=duplex	G1 G2
119	IHIDE	1	0-don't, 1-do hide hidden lines	G2
120	MARK	0	Marker type, 0=none (use CHAR) 1 thru 10 for UG markers	G1
121	KBWAIT	1	0-don't, 1-do wait after TEK plot	G1 G2
122	DPNEW	0	initialization flag for /DPMODE/ (='DPAK' when initialized)	

RMODE - real variables

1	GSIZE	1.	horizontal programmer size	G1 G2
2	"	1.	vertical programmer size	G1 G2
3	GOFSET	0.	horizontal programmer offset	G1 G2
4	"	0.	vertical programmer offset	G1 G2
5	ASCALE	*	XMIN used in making plot	G1 G2
6	"	*	DX " " " "	G1 G2
7	"	*	XTICS " " " "	G1 G2
8	"	*	XSUBS " " " "	G1 G2
9	"	*	YMIN " " " "	G1 G2
10	"	*	DY " " " "	G1 G2
11	"	*	YTICS " " " "	G1 G2
12	"	*	YSUBS " " " "	G1 G2
13	"	*	ZMIN " " " "	G2
14	"	*	DZ " " " "	G2
15	"	*	ZTICS " " " "	G2
16	"	*	ZSUBS " " " "	G2
17	ROTANG	40.	default isometric rot angle	G2
18	ELVANG	40.	default isometric elev angle	G2
19	G1MARG	0.20	Low X (left) margin space	G1
20	"	0.15	Low Y (bottom) margin space	G1
21	"	-0.1	High X (right) margin space	G1
22	"	-0.15	High Y (top) margin space	G1
23	XSMTH	0.	Width (sigma) for smoothing in X	11 22
24	YSMTH	0.	Width (sigma) for smoothing in Y (if 0., no smoothing is done)	22
25	CSIZX	.015	Size of characters in X	G1
26	CSIZY	.015	Size of characters in Y	G1
27	WSMTH	2.5	Number of bin widths to be used for XSMTH and YSMTH when the latter are 0. but SMOOTH is on (1).	11 22
28	FYMAX	0.0	bin overflow value	L1
29-50			statistics values (XAVG, etc.)	11 22

DEVTYP - names of devices

1	DEVTYP	'CLEAR '	reserved
2	"	'PRINTER '	line printer
3	"	'VEP12FF '	
4	"	'VEP12CR '	
5	"	' '	
6	"	'SEQ4010 '	
7	"	'TEK4010 '	
8	"	'SDD4010 '	
9	"	'SDDXCOW '	
10	"	'SDDSXSS '	
11	"	'SDDGRIN '	
12	"	' '	
13	"	'IMPRT10 '	
14	"	'CALFICH '	
15	"	'PDEVUGS '	
16	"	'IMGN300 '	
17	"	' '	
18	"	' '	
19	"	'IBM5080 '	
20	"	'IMGNIBM '	

Appendix C

IBM VM SYSTEM

Handypak resides on the U or T disk on the IBM VM/CMS system at SLAC. The Handypak library is (HANDYPAK TXTLIB *), and the supporting object files are (DMGRUT TEXT *) and (DMSCAT TEXT *). Updates to Handypak are described in the file (HANDYPAK MEMO *).

Lineprinter Output Only

If only lineprinter output is desired from Handypak, then the file (DMGRUT text *) should be included in the LOAD command to supply dummy entries for the graphic routines. This case uses the least amount of memory and loads the fastest. The LOAD command looks like the following:

```
GLOBAL TXTLIB HANDYPAK FORTSLAC VSF2FORT
GLOBAL LOADLIB VSF2LOAD
LOAD Yourstuff DMGRUT
```

Lineprinter and Graphic Output

The following gives the LOAD example for Handypak output to either the lineprinter or graphic devices such as the Imagen (IMGN300) and interactive Tektronix terminal (TEK4010), with basic, stroke, or fancy DUPLEX character fonts (CHFMT 0, 1, or 2):

```
GIME UGS77
GLOBAL TXTLIB Yourtxtlib HANDYPAK UGOBJLIB FORTSLAC VSF2FORT
GLOBAL LOADLIB VSF2LOAD
LOAD Yourtext SIMPLEX DUPLEX NUCLEUS IMGN300 ATEK4010 TEK4010
```

A UG driver (e.g. IMGN300) must be supplied for each graphic device being used. If graphic plots are to be made interactively on your Tektronix 4010 terminal, it is necessary to set CP TERM LINESIZE OFF before your program starts execution, and your program should CALL DPSLCT('TEK4010 ',1).
(Note - ATEK4010 is only needed when sending interactive graphics to Ambassador terminals operating in full screen mode at SLAC. ATEK4010 must procede TEK4010. The loader will issue sev-

eral warnings about duplicate definitions when ATEK4010 is used).

Filedefs

Filedefs should be included for the graphic devices that create files. Some examples for the Imagen, Versatec, and Tektronix 4010 (sequential file) are:

```
FI IMGN300 DISK FN FT FM
FI VEP12FF DISK FN FT FM
FI SEQ4010 DISK FN FT FM
```

where FN, FT, and FM specify CMS files of your choice. See the Unified Graphics manual for other devices, or for additional filedef information.

If HWRITE or HREAD are to be used, filedef's are needed for their logical units (default = 20 for HWRITE, 21 for HREAD):

```
FI 20 DISK FN FT FM (RECFM VBS
FI 21 DISK FN FT FM (RECFM VBS
```

If scatter points are to be generated, then a filedef is needed for the scatterplot scratch file (unit 19 by default). If no filedef is given, then handypak issues one as follows:

```
FI 19 DISK HANDYPAK SCATFILE A (RECFM VBS
```

CHP2

CHP2 resides on the U or T disk also. Simply entering CHP2 will start execution of Conversational Handypak. It will generate lineprinter output to the console (unit 6), and graphic output to IMGN300, VEP12FF, SEQ4010, TEK4010, or SDD4040 devices.

Appendix D

VAX SYSTEM

The following logical names are used for the Handypak and Unified Graphics directories on the M2 VAX at SLAC,

```
HPK:    -  Handypak
UG:     -  Fortran 77 Unified Graphics (UG77).
```

These logical names are used here in examples that follow (logical names and directory names may vary on other VAXes).

Lineprinter Output Only

If only lineprinter output is desired from Handypak, then the file DMGRUT.OBJ should be included in the LINK command to supply dummy entries for the graphic routines. This case uses the least amount of memory and loads the fastest. The LINK command looks like the following:

```
LINK Yourstuff,HPK:DMGRUT,HANDYPAK/LIB
```

Lineprinter and Graphic Output

The following gives the LINK example for Handypak output to either the lineprinter or graphic devices such as the Imagen (IMGNIBM) and interactive Tektronix terminal (TEK4010), with basic, stroke, or fancy DUPLEX character fonts (CHFMT 0, 1, or 2):

```
LINK Yourstuff,HPK:HANDYPAK/LIB,-
UG:NUCLEUS,SIMPLEX,DUPLEX,IMGNIBM,TEK4010,UGOBJLIB/LIB
```

A UG driver (e.g. TEK4010) must be supplied for each graphic device being used. If graphic plots are to be made interactively on your Tektronix 4010 terminal, your program should CALL DPSLCT('TEK4010 ',1).

Other Considerations

Since Handypak expects numeric arguments, and the VAX passes character arguments by a descriptor address, some special coding is needed when a character variable is to be used in a call. For example, if a TITLE string is stored in a CHARACTER variable (e.g. CHRBUF), then calls to HDEF1, HDEF2, or the DUT routines must pass the title as an address and not by a descriptor, e.g.

```
CALL HDEF1( ...., %REF(CHRBUF), ... )
```

When HANDYPAK is used in a multi-process environment with more than one process referencing the same global HCOM, then each process should have its own image of /HNODE/ to assure that there is no interference between the processes when HPNTR is called to set up NODE. Care must also be taken as there is no provision for locking out other processes while one process is modifying or re-defining the contents of HCOM.

BIBLIOGRAPHY

1. Beach, Robert, Unified Graphics for Fortran 77. CTGM No. 203, Computation Group, SLAC: 1985.

2. Chaffee R. and Friedman J. SLAC KIOWA. CGTM No. 146, Computation Group, SLAC: 1973.

INDEX