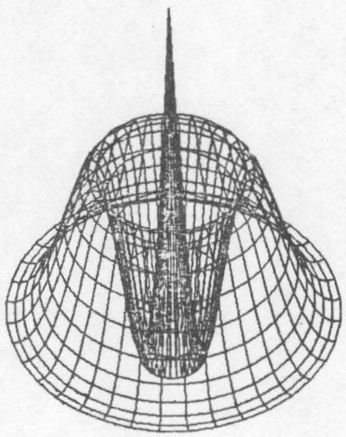
WRUBEL COMPUTING CENTER REPORT

Indiana University, Bloomington, Indiana 47401, 812 337-1911

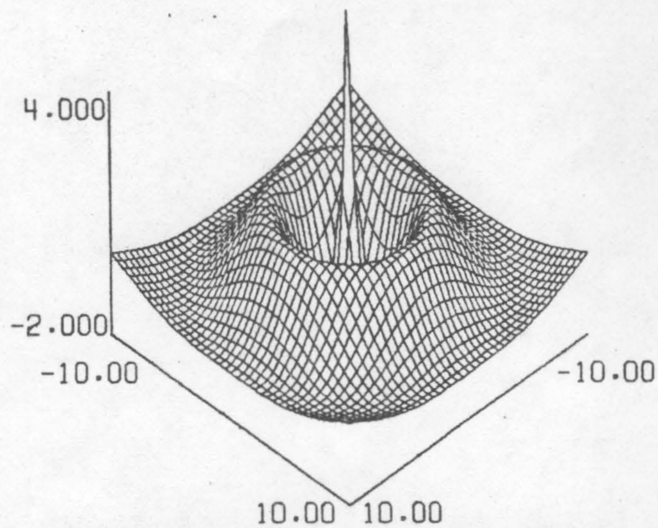
JHVIEW

Programming Supplement #93 (288;5/75)

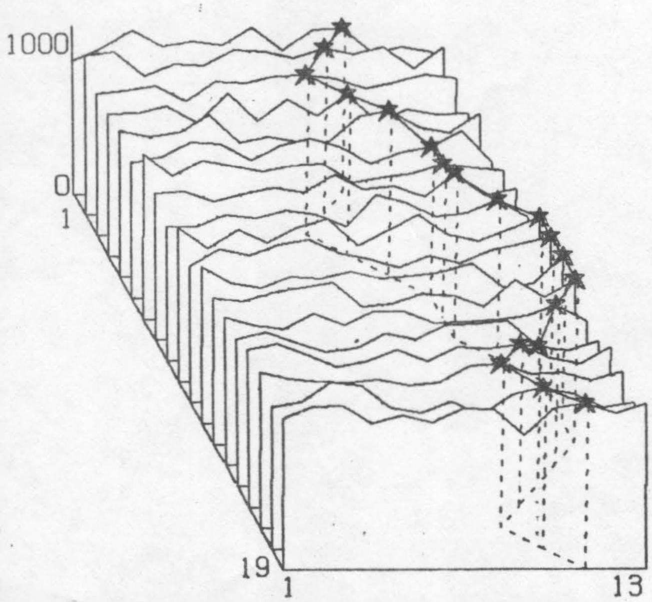
JHVIEW: 3-D PLOTTING PACKAGE



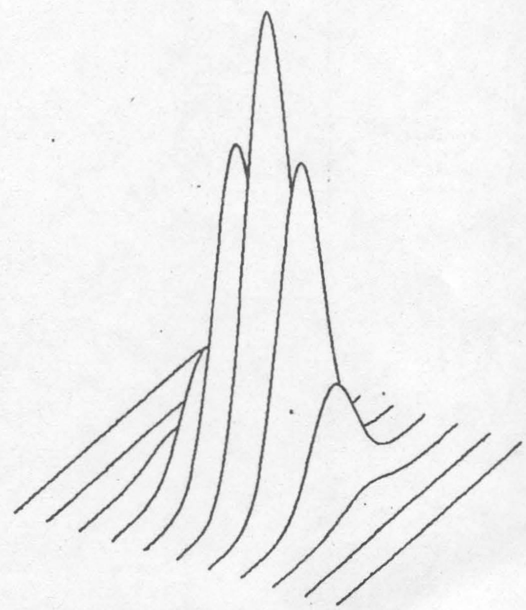
BLIP. 3 DATA CARDS, 11 LINES OF FORTRAN.



SAME, IN RECTANGULAR COORDINATES



NON-STANDARD USE. THE ACROPHILIC PATH.



EXAMPLE OF ISLICE 2 MODE.

JHVIEW is a generalized 3-D plotting package. While some of its options, such as hidden line calculation, are limited to "nearly rectangular" coordinate systems the effective coordinate system is user-defined and arbitrary. Control of the program and setting of parameters is accomplished by a free-format, keyword input system. Required user programming is minimal; the first illustration on page 1 was produced with 11 lines of FORTRAN and two input control cards. In most cases the overall CPU time is between 2 and 4 milliseconds per data point. With very few exceptions only one calculation is made for each data point, so that the time required to plot functions which involve long computation is minimized. The hidden line option seldom increases the total CPU time by more than a second for an entire plot and requires no additional function calculations.

This manual assumes an introductory knowledge of FORTRAN and knowledge of the existence of the WCC plotter manual. Some very basic concepts are discussed with the hope of resolving the confusion many users might otherwise experience. Several examples and a sample program are included. The body of the manual tells what the program does, and the appendices show how to use it.

The potential user must be warned that JHVIEW is designed to deal with orderly data. If your data will not neatly fill a standard two-dimensional array or if your data points are stored in a randomly scattered fashion, some special preparation of the data will be required.

All "standard" applications are described in this manual. A great variety of non-standard combinations are possible for the more adventurous. Additional documentation of a more technical nature should be available in the near future.

TABLE OF CONTENTS

INTRODUCTORY REMARK.

If you have ever successfully converted a collection of data into a multiple-line drawing, whether by machine or by hand, then you will probably have rapid success in using any 3-D plot program. For many people, however, there is some confusion at first, causing much delay and wasted effort, and frequently it turns out that their original idea was impractical or impossible. It is all too common to suppose that the 3-D plot program will somehow figure out what to do with your data and devise an esthetic scheme for displaying it. JHVIEW cannot do this. Therefore, before you start writing programs, get some paper and a small sample of your data and experiment with drawing EXACTLY what you want the computer to draw. The time will be well spent.

INDEPENDENT AND DEPENDENT VARIABLES.

JHVIEW is designed to display the value of a dependent variable, called Z, as a function of two independent variables called X and Y. As an example of such a function, suppose I have made a study of housing unit density in Monroe County. My X and Y are the distances East and North, respectively, from the intersection of College and Fifth Street. If you specify a value of X, such as 7.5 (meaning 7.5 miles East) and 6.5 (miles North) for Y, then according to my data the dependent variable Z is zero dwellings per square mile (because you have specified the middle of Lake Lemon).

The relationship which defines Z may take many forms. It may be an equation, such as $Z = X^2 + Y^2$. It may be an array of data points, such as Z = ARRAY(NX,NY), where NX and NY are the array subscripts. Question: in the latter case, what shall we call the independent variables? Are they NX and NY? The answer is, yes and no. Usually it will be true that there are independent variables X and Y, for which data has been taken at certain values. Then each integer NX corresponds to some particular value of the independent variable X (and the same for NY and Y), so that it is perfectly reasonable to consider X and NX as being two different names for the same thing. In one case we are referring directly to a certain value of X, and in the other we are referring to the NXth value of X.

In order to use JHVIEW it is crucial to understand the correspondence and distinction between NX and X, because JHVIEW will select various pairs of integers NX and NY and you will need to supply an appropriate response.

SUBROUTINE MYSUB.

While your variables X and Y and Z are the quantities of primary interest to you, JHVIEW is unconcerned and in fact may never find out what they are. They are your secret. What you must be able to do is assign a point in an x,y,z coordinate system to every valid pair of integers NX and NY. This requires that you write a FORTRAN subroutine which will be described in detail later, and which calculates these coordinates. JHVIEW will give integer values to NX and NY and call MY SUB once for each data point. It is important to realize that JHVIEW never examines or sorts your data, but relies entirely on the assumption that it can produce a coherent drawing by varying NX and NY in an orderly fashion.

```
SUBROUTINE MY SUB(NX,NY,XC,YC,ZC)
XC = x coordinate corresponding to NX and NY.
YC = y      "           "          "  "   "   ".
ZC = z      "           "          "  "   "   ".
RETURN
END
```

Beware, do not confuse the x, y, and z (lower case) coordinates with the X, Y, and Z (upper case) variables. In most applications they will indeed be the same thing, but this is strictly your choice.

SETTING UP JHVIEW.

When attempting to display a function Z(X,Y), the problems which must be faced are similar to those of a budding architect who is trying to draw a realistic building. Viewpoint, scale, position on the paper, and the size of the drawing itself must all be decided. The first few parameters for JHVIEW deal with these problems.

: parameter SIDE : Consider first the object to be drawn. Even if it is an abstraction, such as scattering cross-section as a function of angle and particle energy, we can imagine it to be represented by a plaster model whose height at any point corresponds to Z. The model just fits inside an imaginary box whose physical size (in centimeters) is given by the three elements of an array called SIDE. This box is referenced to the rectangular xyz coordinate system, with the sides of the box parallel to the three coordinate axes. Varying the SIDEs has the effect of stretching or squashing the plaster model in some direction. So, if we want the peaks to appear taller, we can set SIDE(3) larger than SIDE(1) and SIDE(2), and JHVIEW will stretch the model vertically by that amount.

: parameters XLIMIT, YLIMIT, ZLIMIT : In order to draw the model, it is necessary to establish the location of the box in the coordinate system. The array XLIMIT(2) must be set to the x coordinates of the "x" side of the box. Same for YLIMIT and ZLIMIT. The actual values you assign to these quantities are up to you, so long as all of your calculated XC, YC, and ZC values will fall within those limits.

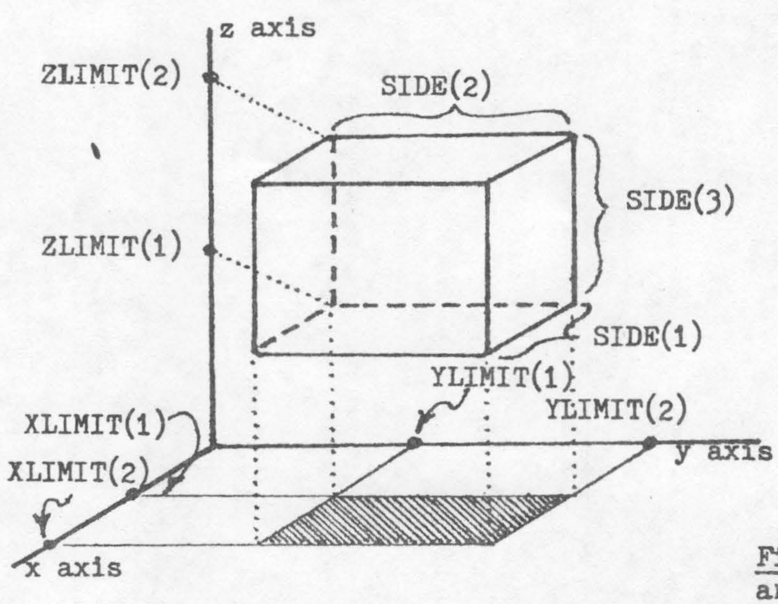


Figure 1. Defining the "box" and its location in the xyz coordinate system with parameters SIDE, XLIMIT, YLIMIT, and ZLIMIT.

parameters R, P, T Next the viewpoint must be defined. This locates the observer (or the observer's eyeball) with respect to the model. The reference point is the corner of the box defined by XLIMIT(1), YLIMIT(1), and ZLIMIT(1). The eyeball is located R cm. from that reference corner at an angle P degrees (measured in the xy plane) from the x axis and T degrees (measured upward) above the xy plane.

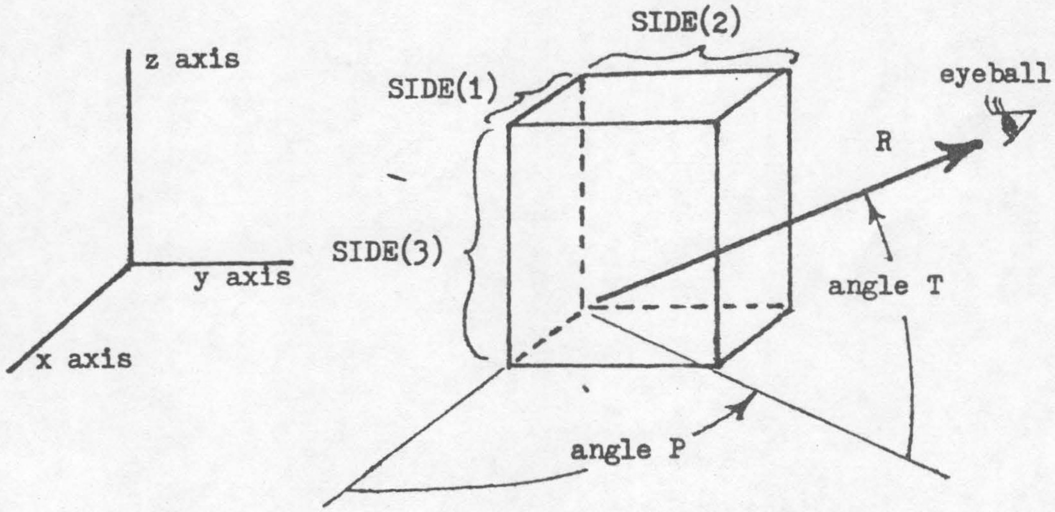


Figure 2. Defining R, P, and T in the xyz coordinate system and relating the parameters R and SIDE.

Notice that R is measured in the same units as SIDE, rather than in units of the x, y, and z axis. This is because the magnitude of R, compared to the SIDEs, determines vanishing point effects. If your eye is close to the box you will see pronounced perspective effects, but if it is very far away (usually preferred for displaying scientific data) the display will become isometric. Because of automatic scaling, the actual numerical values of SIDE and R are of little importance; it is their ratios that matter.

parameters W, H The display produced by JHVIEW is a projection. Imagine a line drawn from each point on the model's surface to the eyeball. If we insert a piece of paper between this surface and the eyeball and make a dot where each line intersects the paper we will obtain a drawing of the surface. Normally one would expect to hold the paper perpendicular to the line of sight (that is, the vector R), but this is not always the most desirable. If the paper is skewed sideways or tilted back, the projection changes. The two additional parameters of JHVIEW which determine these factors are angles W and H. When W = P and H = T, the paper is to be held perpendicular to the line of sight. For other values the effect is sometimes difficult to visualize, so the following rules are given:

(1) Normally, set W = P.
  (a) To force the y axis to be parallel to the bottom edge of the paper, set W = 0. or 180., whichever is closest to P.
  (b) To force the x axis to be parallel to the bottom edge, set W = + or - 90. (whichever is closest to P.)
  (c) To obtain a mirror image add + or - 180. to W.

(2) Normally, set H = T.
  (a) To force the z axis to be straight up and down on the paper, set H = 0.
  (b) To flip the drawing upside down, add + or - 180. to H.

parameters ORIGIN and FRAME The drawing can be scaled to any size by moving the paper closer to the viewer, and positioned by moving it right or left. This is the same as positioning a projected slide image on a screen. The position and size of this final image are specified by parameters ORIGIN and FRAME. They are in real, live inches, and JHVIEW will scale and shift the projection so that a drawing of the imaginary box, on real,live plot paper, would just fit in the FRAME whose lower left hand corner is at ORIGIN.

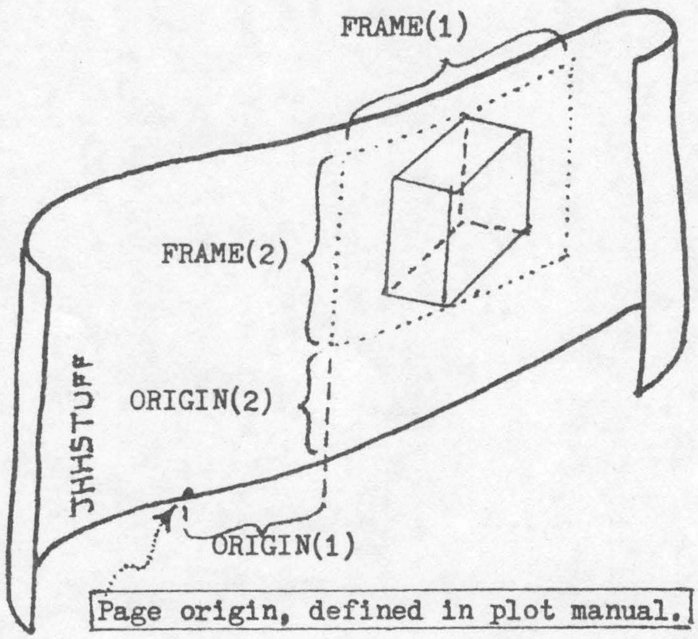


Figure 3. Parameters FRAME and ORIGIN in relation to final plot layout.

DIVERSIONARY EXAMPLE.

Suppose I have been sampling the pollen count, at various altitudes, over a period of a year. I choose to display the time of year (expressed as a day number from 1 to 365) along the x axis and the altitude at which each sample was taken along the y axis. I took samples on 20 days, and the day-numbers are stored in an array called DAYS. I took the samples at the same 5 altitudes ranging from 0 to 10,000 feet on each day, and these altitudes are stored in an array called ALT. This means I have 5 x 20 = 100 pollen counts, and I have read them into an array dimensioned SNEEZE(20,5). I look at my data and find that the counts range from 7. to 805., but for convenience I will scale as if they ranged from 0. to 1000.

```
Having set:  XLIMIT(1) = 1.        XLIMIT(2) = 365.
             YLIMIT(1) = 0.        YLIMIT(2) = 10000.
             ZLIMIT(1) = 0.        ZLIMIT(2) = 10000.
```

the portion of my subroutine relevant to this example is:

```
SUBROUTINE MY SUB(NX,NY,XC,YC,ZC)
DIMENSION DAYS(20), ALT(5), SNEEZE(20,5)
XC = DAYS(NX)
YC = ALT(NY)
ZC = SNEEZE(NX,NY)
RETURN
```

If I choose P = 120. W = 120. T = 20. H = 20. I will be viewing the xyz coordinate system as shown in Fig. 4a. However, I want the x axis, representing the time of year, to be shown as a horizontal line. I can do this by setting W = 90. as in Fig 4b.

But I am still not satisfied. I want the x axis to be increasing from left to right. If I add or subtract 180. degrees from W, I will get a mirror image as in Fig. 4c, and this is my final choice.

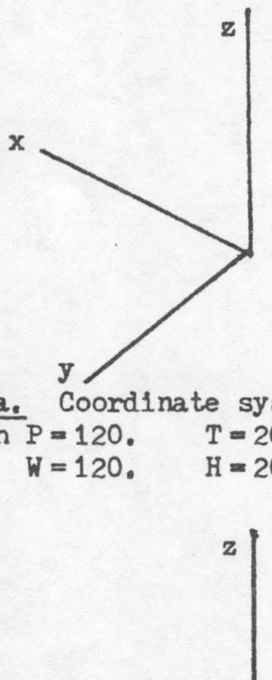


Figure 4a. Coordinate system as seen with P = 120. T = 20. W = 120. H = 20.

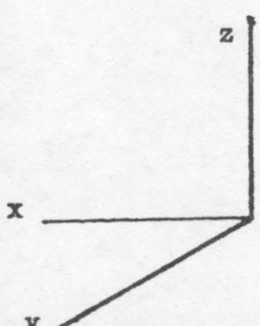


Figure 4b. P = 120., W = 90.

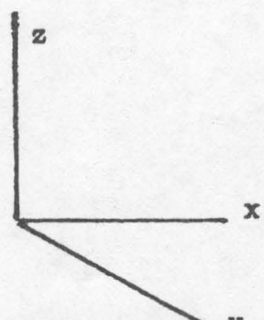


Figure 4c. P = 120., W = -90.

For most applications lines of constant NX will be parallel to the y axis and lines of constant NY parallel to the x axis, and as a result this requirement will usually be satisfied automatically and you don't have to think about it. It is perfectly possible, however, to let the NX define angles A(NX) and the NY define radii RHO(NY), so that MY SUB becomes:

```
...
RC = RHO(NY)
AC = A(NX)
XC = RC * COS(AC)
YC = RC * SIN(AC)
ZC = DATA(NX,NY)
...
```

and the data is displayed in cylindrical coordinates. Lines of constant NY are now circles about the z axis. So, while JHVIEW can easily be made to display data in NEARLY ANY coordinate system, many of them are liable to violate the above requirement and use of the hidden line option will give unpredictable results.

LINE determines the hidden line option:

LINE -1	disables the hidden line option.
LINE 0	hidden lines appear as dotted lines.
LINE 1	hidden lines are not drawn at all (default)
LINE 2	draws small dots along edges of peaks and valleys, but no lines except around boundary of the surface.
LINE 3	draws small dots as for LINE 2 and also at all visible points NX,NY. The result is somewhat like the engraving of George on a dollar bill.

parameter ISLICE The JHVIEW drivers have two primary modes.

In the normal mode, the surface is drawn as a grid of lines, one line for each value of NX and one for each NY specified by NXDO and NYDO. In slice mode the display consists of slices with one slice for each specified value of NY. The pollen count example illustrates slice mode with ISLICE set to 1:

ISLICE 0	no slices; draws intersecting grid lines (default).
ISLICE 1	slices with bottom edges at ZLIMIT(1).
ISLICE 2	Draws only top edge of slices. This is just the grid, with all of the NX lines removed.

parameters IBOX, TITLE, ALIGN, NGRAIN A provision exists for drawing axes of a sort, as illustrated in the pollen example. 3 edges of the box are drawn and labeled with the values XLIMIT, YLIMIT, ZLIMIT.

IBOX 0	no axes (default).
IBOX 1	axes.

JHVIEW will write an optional title of up to 50 characters, as in the pollen example, just below the FRAME.

TITLE * ANYTHING EXCEPT ASTERISKS * specifies the title, which can be changed at any time from one plot to the next.

TITLE ** (two adjacent asterisks, = default) specifies a null title.

ALIGN (default zero) is a special parameter for composite drawings, such as stereoscopic pairs, and may be discussed in a later document.

NGRAIN determines how carefully hidden lines will be calculated. It proves to have little effect on CPU time. Large values (up to 200) provide better resolution. Default is 80.

FREE-FORMAT INPUT.

Although all of these parameters can be set from within your program, JHVIEW is set up to be controlled entirely by a free-format keyword-value control card system. It is simple to use and well worth learning:

Delimiters are any of the following: blank,comma,$,*,),(,/.

Keywords and/or numbers are separated by any number of blanks or commas, or by one of the other delimiters which have special meanings.

Arrays, such as XLIMIT, FRAME, of NYDO are read in by enclosing the array elements in $ signs:

XLIMIT$0.,10.$ YLIMIT $0. 10. $, NYDO $1,10,1$

Character strings are enclosed in asterisks:

TITLE *ELEPHANTS PER CUBIC FURLONG*

Repetition can be specified by parentheses: 3(1, 7.5) is equivalent to 1, 7.5, 1, 7.5, 1, 7.5

Slash / means ignore remainder of card. Thus, comments following a slash will not be read.

Floating point: 1000., +1.E3, 1+3, 1.E+3, 1.+3, 1E3 are equivalent. E3 is illegal.

Invalid numbers such as 1.3A4 and mispunched keywords such as TITTLE produce a detailed diagnostic message and terminate execution.

CONTROL KEYWORDS.

There are several keywords which have no numerical value associated with them. They set internal flags which govern control of the program.

PLOT	temporarily stops reading parameters. Calls subroutine or entry PRESET, calls CLOSEPF if a previous plot has not been closed, and begins drawing next plot. After plot is drawn JHVIEW reads next parameters (for next plot) if any.
END	Similar to PLOT, but causes JHVIEW to plot, call CLOSEPF, and return to main program.
end-of-file	Same as END except no subsequent parameter input is possible.
ADD	Similar to PLOT, but CLOSEPF is not called. This is for stacking plots (by changing origin) or superimposing plots.
RETURN	causes immediate return to main program. All previous keywords will have been processed, but no plot generated.

CLOSEPF	immediately calls CLOSEPF but does nothing else.
DOCUMENT	(default) The contents of common block JHVSET (see below) are printed, except for user variables, prior to each plot. Helpful for documentation.
NODOC	Shuts off DOCUMENT.
PRNT	Initiates a detailed printout of the antics of the input routine. Not generally useful.
UNPRNT	(default) Shuts off PRNT.

STORAGE OF PARAMETERS. USER-DEFINED KEYWORDS.

All parameters with assignable values are stored in a common block which can be used by MY SUB for many purposes:

```
 COMMON/JHVSET/SIDE(3),XLIMIT(2),YLIMIT(2),ZLIMIT(2),
+     R,P,T,W,H,ORIGIN(2),FRAME(2),NXDO(3),NYDO(3),
+     ISLICE,IBOX,ALIGN,LINE,NGRAIN,DATA, TITLE(5),USER(5)
```

This arrangement has two special properties:

(1) Array loading -- the input routine treats all quantities between $ signs as successive array elements. Because of the sequence of variables in block JHVSET it is possible to load several parameters at once. thus:

```
XLIMIT $ 3(0., 10.) $              is equivalent to
XLIMIT $ 0., 10. $  YLIMIT $0., 10.$ ZLIMIT $0., 10. $
```

(2) User variables. USER(1) through USER(5) are reserved for your use. Also you can define up to five keywords of your own. They may be strings of up to 10 characters (no imbedded blanks) which, except for length, would be valid FORTRAN variable names.

For example, DEFINE AXLOTL DEFINE SMOOG2 creates keywords AXLOTL and SMOOG2 which can be used at any time thereafter to set or alter the contents of USER(1) and USER(2) respectively.

(J. HETTMER)

APPENDIX A

SUMMARY OF ALL KEYWORDS AND ASSOCIATED DEFAULTS.

KEYWORD	DEFAULT	ALLOWED VALUES
SIDE	$3(1.)$	any
XLIMIT	$0., 10.$	not both equal
YLIMIT	"	"
ZLIMIT	"	"
ORIGIN	$.5, .5$	any
FRAME	$8., 8.$	any
NXDO	$1, 2, 1$	NXDO(1).LE.NXDO(2), and NXDO(3).GE.1
NYDO	"	"
LINE	1	-1,0,1,2,3
IBOX	0	0,1
ALIGN	0	secret
NGRAIN	80	2.LE.NGRAIN.LE.200
R	1.E10	non-zero ▦▦
P	45.	any ▦▦
T	"	"
W	"	"
H	"	"
DATA	0 (not used by JHVIEW. Useful as a switch in MYSUB)	
TITLE	** (= none)	*up to 50 characters*
FOO (default name of USER(1))	0	any
DEFINE	(FOO)	legal keyword string
DOCUMENT	yes	
NODOC	no	
UNPRNT	yes	
PRNT	no	
PLOT		
ADD		
CLOSEPF		
RETURN		
END		
end-of-file		

▦▦ There is one slightly confusing combined restriction on R,P,T,W,H. With the viewer, paper, and box held as specified, it must be possible to place the projection paper between the viewer and the box without hitting the box. In particular, the viewer must be outside the box. Otherwise, the projection will blow up. If this restriction is violated you will be informed, and an allowed set of values will be used by the program. The cure for exploding projections is usually to increase R.

APPENDIX B

ROUTINES TO BE WRITTEN BY THE USER

Optional statements are in lower case. UPPER CASE MEANS NOT OPTIONAL.

1. MAIN PROGRAM.

PROGRAM name (INPUT,OUTPUT,PLOT,TAPE47 = PLOT, other tapes of your own)
At this point you can read in data or do whatever else you want.

CALL JHVIEW

This has turned control over to the plot program, which can deal with multiple plots depending on your keyword control cards. The "end" and "return" keywords return control to main program, but are optional. An end-of-file also causes a return after the plot is drawn. If you used an "end" or "return" card you are now back in control and can do what you like, including calling JHVIEW again.

END

2. CALCULATION SUBROUTINE.

SUBROUTINE MY SUB(NX,NY,XC,YC,ZC)
common/jhvset/side(3),xlimit(2),ylimit(2),zlimit(2), ...
Any other dimension or common, data, etc. ▒ ▒ note, DO NOT attempt to set -any- JHVIEW parameters with data statements. The results will be unpredictable. ▒ ▒

Whatever manipulations you may need to perform.

XC = whatever
YC = whatever
ZC = whatever
RETURN

ENTRY PRESET

This entry, about which little has been said (or maybe nothing) is required. It can be a separate subroutine if you like. It is always called by JHVIEW once, after the parameters have been read but before the plot is begun. It permits you to set up scale factors or any other quantities which will be needed to compute XC, YC, ZC. This would also be a reasonable time to read in data, especially if it changes from one plot to the next. In particular, there is no reason why you can't set XLIMIT or NXDO, etc. at this time if their values are to be dependent on some other calculation or if you prefer not to use the keywords.

When JHVIEW calls entry PRESET, the 5 parameters NX through ZC are actually USER(1) through USER(5), which may or may not prove convenient to you.

RETURN
END

(PS. A clarification: Every appearance of the keywords PLOT, ADD, END, or end-of-file causes a call to PRESET and a plot, provided that there has been some parameter input by means of a keyword. Thus, PLOT END will only generate one plot. But the sequence PLOT LINE 1 END will generate two plots.)

APPENDIX C

COMPLETE EXAMPLE. THIS IS EXACTLY THE ENTIRE SET OF CARDS WHICH PRODUCED THE "POLLEN COUNT" PLOT. DATA WAS ON A PFILE IN THE FORMAT INDICATED.

```
JHH,T2.
ACCOUNT,nnnn,pppp.
ROUTE(PLOT,EJ,EQ=PL,L=WCC)
GET(TAPE1)
GET(JHVIEW/UN=LIBRARY)
FTN.
LOAD(JHVIEW)
LGO.
(789)
      PROGRAM POLLEN(INPUT,OUTPUT,PLOT,TAPE47 = PLOT,TAPE1)
      CALL JHVIEW
      END

      SUBROUTINE MY SUB(NX,NY,XC,YC,ZC)
      DIMENSION ALT(5),DAY(20),SNEEZE(20,5)
      XC = DAY(NX)
      YC = ALT(NY)
      ZC = SNEEZE(NX,NY)
      RETURN

      ENTRY PRESET
      READ(1,100),(ALT(J),J = 1,5)
      READ(1,100),(DAY(J),J = 1,20)
      READ(1,100),((SNEEZE(K,J),J = 1,5),K = 1,20)
  100 FORMAT  (5E10.5)
      RETURN
      END
(789)
XLIMIT $1., 365. $    YLIMIT $0., 10000.$   ZLIMIT $0., 1000.$
NXDO$1,20,1,  1,5,1$             /  NXDO AND NYDO CAN BE READ IN ONE STEP.
ISLICE 1                         /  THIS SETS UP THE SLICE MODE.
NGRAIN 200                       /  SLICE MODE PREFERS HIGH DEFINITION.
IBOX 1,   TITLE *POLLEN COUNT AS DISPLAYED IN SLICE MODE.*
FRAME$4. 4.$                     /  FOR TINY LITTLE PICTURES.
T 20. H 20. P 120.
W -90.  /PROJECTS SLICES HORIZONTALLY, FLIPS DISPLAY LEFT - TO - RIGHT.
END     /   THIS IS AS GOOD A WAY TO TERMINATE AS ANY.
(789)
(6789)
```