RECEIVED BY TIC DEC 2 1976

NOVEMBER 1976

PPPL-1304

PERVUE: A PROGRAM TO PLOT
PERSPECTIVE VIEWS OF SURFACES

RECEIVED BY TIC DEC 2 1976

BY

W. G. PRICE, JR.

PLASMA PHYSICS LABORATORY

PRINCETON UNIVERSITY
PRINCETON, NEW JERSEY

This work was supported by U. S. Energy Research and Development Administration Contract E(11-1)-3073. Reproduction, translation, publication, use and disposal, in whole or in part, by or for the United States Government is permitted.

DISCLAIMER

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

DISCLAIMER

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

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Energy Research and Development Administration, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

Printed in the United States of America.

Available from
National Technical Information Service
U. S. Department of Commerce
5285 Port Royal Road
Springfield, Virginia 22151

Price: Printed Copy \$ * ; Microfiche \$3.00

<u>*Pages</u>	<u>NTIS Selling Price</u>
1-50	\$ 4.00
51-150	5.45
151-325	7.60
326-500	10.60
501-1000	13.60

PERVUE: A Program to Plot Perspective
Views of Surfaces

William G. Price, Jr.

Plasma Physics Laboratory, Princeton University
Princeton, New Jersey 08540

PPPL-1304

November 1976

PERVUE: A Program to Plot Perspective
Views of Surfaces

William G. Price, Jr.
Plasma Physics Laboratory, Princeton University
Princeton, New Jersey 08540

ABSTRACT

PERVUE is a program which basically plots a perspective view of a surface specified, as heights above a plane, by a two-dimensional array. The basal plane may be partitioned by an arbitrary rectangular or polar grid, contour lines may be traced on the surface, and hidden lines may be removed. The Fortran calling sequence is quite simple.

The basic function of PERVUE is to produce a perspective-view plot of a surface described by a two-dimensional array as a mesh of points above a regular grid. The array $Z(I,J)$ provides heights of the surface above a base plane partitioned by either a rectangular grid $(X(I), Y(J))$ or a polar (r, θ) grid $(R(I), T(J))$.

The perspective transformation from 3-D $(X,Y,Z$ or $R,T,Z)$ space to the plotted (X',Y') space is calculated with respect to a particular viewing point, $V(VX, VY, VZ)$. If V recedes to infinity, the transformation becomes a simple orthographic projection. The formulas for the transformation are given in the Appendix.

PERVUE can construct a network of constant-height contour lines on the surface. This is done by linear interpolation between each grid intersection, followed by an attempt to match up and connect individual contour segments into continuous lines. It is possible to plot the contours while suppressing the original grid lines on the surface.

During the plotting of both grid and contour lines it is possible to suppress "hidden" lines, lines which are behind other parts of the surface as seen from the viewing point. Lines are assumed visible if both end points are visible. If only one end is visible, a partial line is drawn to the last visible point. This procedure is clearly not 100% accurate, but it is very effective for most plots.

The visibility of individual points is computed by the following algorithm. A line is passed from the viewpoint V through the point P and projected onto the basal plane. The

intersection of this base line with each grid line (X(I), Y(J), R(I), or T(J)) is then calculated. If the intersection lies between V and P and within the grid region, the Z-value above it on the V-P line is compared to the Z-value on the surface (obtained by interpolation along the grid). The relative heights above each intersection indicate whether the V-P line is wholly above or below the surface, or if it passes through the surface, making P invisible.

An auxiliary entry to PERVUE allows arbitrary lines (specified by endpoint) and special symbols to be plotted, using the same perspective transformation as for a previous surface plot.

PERVUE is written almost entirely in FORTRAN-IV, with calls to basic Calcomp plot routines.

Two FORTRAN entry points are provided for users of PERVUE. To produce a perspective plot of a surface:

```
CALL PERVUE ( LGEOM, X,Y,Z, VX,VY,VZ,  
              NXDIM,NYDIM, NX,NY, LGRID, NCONTR, IDE,  
              ZMIN,ZMAX, KBOX, KOLOR, KROSS,  
              PWIDE, PLONG, NEND, NCHAR, LABEL)
```

To plot individual lines and points:

```
CALL PERPNT( X,Y,Z, KODE,KHAR, XP,YP,ZP, IDE)  
(Then CALL ENDPLT).
```


The parameters are type REAL*4 or INTEGER*4, as indicated by the standard FORTRAN convention. None of the parameters are altered by PERVUE or PERPNT.

Here Z is a FORTRAN matrix which describes the surface to be viewed, while X and Y define the grid structure. If $|LGEOM| = 1 + 10*NZ$, the surface is defined in rectangular coordinates by the points $(X,Y,Z) = (X(I), Y(J), Z(I,J))$; if $|LGEOM| = 2 + 10*NZ$ or $3 + 10*NZ$ then cylindrical coordinates are used, $X = R$, $Y = T = THETA$, so that $(R,T,Z) = (X(I), Y(J), Z(I,J))$. $|LGEOM| = 2 + 10*NZ$ implies that T is in degrees, while $|LGEOM| = 3 + 10*NZ$ implies radians.

IF $NZ > 1$, the matrix Z is assumed to have a third dimension. This allows the specification of NZ different surfaces all to be plotted in the same region. Hidden line removal will be computed with respect to all surfaces jointly.

The dimensions of Z as specified in the FORTRAN calling program are assumed to be $Z(NXDIM, NYDIM, NZ)$, while X and Y are similarly assumed to be dimensioned $X(NXDIM)$ and $Y(NYDIM)$. However, the surface is assumed to be described by only the "top left" subarray $Z(I,J)$, $I = 1$ to NX , $J = 1$ to NY . Implicitly then $XMIN = X(1)$, $XMAX = X(NX)$, $YMIN = Y(1)$, $YMAX = Y(NY)$. The extremes of Z in this region are limited to $ZMIN$ and $ZMAX$. Values of Z outside these limits will not be plotted or used in the hidden line removal evaluation. Each $X(I)$ should be $> X(I-1)$, and similarly $Y(J) > Y(J-1)$. Figure 1 illustrates a surface specification.

The viewing point is defined by VX, VY, and VZ which are, depending on LGEOM, either (X,Y,Z) or (R,T,Z) values. Hidden lines with respect to this viewpoint are plotted only if $IDE < 0$. As noted previously, the hidden line removal is not 100% accurate; in particular, it will not look through internal holes produced when $Z(I,J)$ is outside (ZMIN,ZMAX). For hidden line removal in cylindrical geometry, $T(I,J)$ should be between 0 and 2π and R should not be negative.

The perspective transformation is computed with respect to the line of sight from the "viewing point" through a "viewed point" which defaults to the origin ($X = R = Y = T = Z = 0$).

Scale factors, which default to 1.0, may be applied to any of the axes. Non-default values for the viewed point and the scale factors may be supplied when $LGEOM < 0$, in which case VX, VY, and VZ are assumed to be vectors of length 3, rather than scalars. The first element of each is given the appropriate component (X,Y,Z or R,T,Z) of the viewing point V, as before. The second element gives the component of the viewed point; the third element gives the scale factor.

The plotting of grid lines is controlled by LGRID. If $LGRID = 1$, only the row-grid lines ($X = X(I)$) are plotted; if $LGRID = 2$ only the column-grid lines ($Y = Y(J)$) are plotted. The full grid is plotted if $LGRID = 0$, while nothing is plotted (e.g. to get contours only) if $LGRID < 0$.

The plotting of contour lines is requested when NCONTR is not zero. When $NCONTR > 0$, this integer is taken to be the number of equally-spaced intervals to be bounded by $NCONTR + 1$

contour lines, at values of $Z = ZMIN + N * (ZMAX - ZMIN) / NCONTR$. If $NCONTR < 0$, this parameter is interpreted as the floating-point array ZCON, rather than as an integer scalar. The first element is a negative, whole-integer-valued floating point number $ZCON(1) = -NC$. The following NC elements are assumed to be specific Z-values for which contours are requested. Therefore ZCON must be a vector dimensioned at least NC+1.

Placing the viewpoint at a great distance up the Z-axis will give an orthographic projection of the contours on the X-Y plane, i.e. a contour map. Setting $VX = VY = 0$ will suppress any rotation of the contour map; otherwise a line from the origin to (VX,VY) will be straight down or straight up on the plot, when VZ is + or - respectively.

KBOX controls the drawing of a box frame around the region viewed: if KBOX = 0, nothing is drawn; if KBOX = 1 the full frame is drawn on the edges of the region. If KBOX = 2, the frame is drawn, but with hidden lines removed.

A simple label to be drawn as a plot header may be provided as a vector or a literal constant in the parameter LABEL. NCHAR, which should be < 50 , gives the number of characters in the label; if NCHAR not > 0 , no label is drawn.

Pen color changes are requested during plotting if KOLOR > 0 . The frame is in red; grid lines seen from the top are blue; those seen from the bottom are black; while contour lines and labels are red.

The actual plot size of the perspective image is specified by PWIDE, the width of the plot, and PLONG, the length of the plot (both in inches). PWIDE should not exceed the physical width of the plotter to be used. KROSS controls the plot orientation. If $KROSS < 0$, the plot is rotated 90 degrees counter-clockwise so that the image of the Z-axis is lengthwise rather than crosswise, on the plot paper. This can be used, for example, to obtain tall, narrow views.

The final parameter for PERVUE, NEND, is set < 0 to suppress the normal internal calls to ENDPLT and IGRAPH so that PERPNT may be called. If this is done, it is up to the user to call ENDPLT later to properly close the plot files (and IGRAPH(0,0,0) before doing non-PERVUE plotting).

The first three parameters for PERPNT are the same X, Y and Z previously given to PERVUE. XP, YP and ZP specify an arbitrary point, (X,Y,Z) or (R,T,Z), within the viewing region established by the prior call to PERVUE. The same perspective transformation is applied and the pen is moved from its current location to this new point.

If $KODE > 0$ the pen is lowered so that a line is drawn, otherwise no line is drawn. Furthermore, if $KHAR = 0$ through 255, the Calcomp special symbol # KHAR is plotted centered on the new point. All plotting is subject to hidden line removal if IDE not < 0 .

The PERVUE routines use a named common block /PERCOM/ whose first two variables are XPEN and YPEN. After any call to PERPNT (even one with $KODE = -1$ and $KHAR = -1$), these are set

equal to the real pen X and Y coordinates on the plotter corresponding to (XP,YP,ZP). XPEN and YPEN are specified in inches relative to the absolute plot origin, which is at the lower left corner by the date-and-time label. XPEN and YPEN may be used to superimpose labels, lines, etc. on the plot as long as YPEN < real plotter width and XPEN < 1.0+PLONG.

PERVUE may be used as a function of type REAL*8: the value returned is a date (format A8) identifying the version of PERVUE being used. Setting IGEOM = 0 will cause this date to be returned immediately, with no attempt to do any plotting.

At the Princeton University Computer Center, the load module containing PERVUE may be obtained by including the statements:

```
//LKED.PERVUE DD DISP=SHR,DSN=U,PPL.BINARY
//LKED.SYSIN DD *
      INCLUDE PERVUE(PERVUE)
/*
```

between a FORTRAN procedure's FORT and GO steps. Do not forget to include a //GO.CALCOMP DD SYSOUT=C card and a "plot block" estimate. The core region request should allow about (40000 + NX *NY) bytes for PERVUE. PERVUE, PERPNT, and PERCOM use quite a few internal subroutines with names starting with "\$". These will be obvious in the load map.

Several examples of the applications of PERVUE are given in Figures 2, 3, and 4.

Figure 1. Parameters which describe the grid and surface.
#754148

Figure 2. A line spiraling around a torus; LGEOM = 22, + PERPNT.
#754146

Figure 3. Perspective view of a $Z(I, J)$; LGEOM = 11.
#754147

Figure 4. Contour map obtained with LGRID = -1 and $VZ \rightarrow \infty$.
#754145

ACKNOWLEDGMENTS

PERVUE is the latest improvement upon several codes patched together from decks obtained from Stott Parker of the Princeton University Computer Center Clinic. The perspective view and hidden line removal sections use methods presented by Kubert, Szabo, and Giulieri in JACM, Vol. 15, p. 193 (1968) and originally programmed by W. C. Gray at the University of Texas.

This work was supported by the Energy Research and Development Administration (formerly Atomic Energy Commission) contract E(11-1)-3073; also, use was made of computer facilities supported in part by National Science Foundation Grant GP 579.

APPENDIX

The perspective image plotted by PERVUE is obtained by projecting lines through each "surface point" $\underline{P}$ from the "viewing point" $\underline{V}$ onto a plane perpendicular to the line from $\underline{V}$ to the "viewed point" $\underline{W}$ (which defaults to $\underline{0}$, the origin).

$\underline{V}$, $\underline{W}$, and $\underline{P}$ are given in the absolute 3-D coordinate system with unit vectors $(\hat{i}, \hat{j}, \hat{k})$. They are immediately scaled to $\tilde{V}$, $\tilde{W}$, $\tilde{P}$ by multiplication with a diagonal matrix $\underline{S}$ (which defaults to $\underline{I}$, the identity matrix).

The desired transformation is to a space with unit vectors $(\hat{u}, \hat{v}, \hat{w})$, where $\hat{w}$ lies along $\tilde{H} = \tilde{V} - \tilde{W}$, so that the $\hat{u}$ - $\hat{v}$ plane is perpendicular to this viewing direction. The ambiguity in the rotation of the $\hat{u}$ - $\hat{v}$ plane is resolved by requiring that $\hat{k}$ project only onto $\hat{v}$, i.e. that it be perpendicular to $\hat{u}$.

Defining $D = |\tilde{H}|$, $\alpha = \hat{i} \cdot \tilde{H}/D$, $\beta = \hat{j} \cdot \tilde{H}/D$, $\gamma = \hat{k} \cdot \tilde{H}/D$, an appropriate choice for $(\hat{u}, \hat{v}, \hat{w})$ is

$$\hat{u} = (-\beta\hat{i} + \alpha\hat{j})/\sqrt{1-\gamma^2},$$

$$\hat{v} = (-\alpha\gamma\hat{i} - \beta\gamma\hat{j} + (1-\gamma^2)\hat{k})/\sqrt{1-\gamma^2}$$

$$\hat{w} = (\alpha\hat{i} + \beta\hat{j} + \gamma\hat{k}).$$

Letting (x, y, z) be the $(\hat{u}, \hat{v}, \hat{w})$ coordinates of the rotated space with origin at $\tilde{W}$, obviously $x = \hat{u} \cdot (\tilde{P} - \tilde{W})$, $y = \hat{v} \cdot (\tilde{P} - \tilde{W})$, $z = \hat{w} \cdot (\tilde{P} - \tilde{W})$.

Since the desired perspective image is a projection onto the $\hat{u} - \hat{v}$ plane at $z = 0$, an additional scaling by D/E is required, where $E = \hat{w} \cdot (\tilde{V} - \tilde{P}) = D - \hat{w} \cdot (\tilde{P} - \tilde{W})$:

$\tilde{x} = \hat{u} \cdot (\tilde{P} - \tilde{W}) * D/E$ and $\tilde{y} = \hat{v} \cdot (\tilde{P} - \tilde{W}) * D/E$ are the 2-D coordinates of the scaled perspective image of $\underline{P}$.

PERPNT: Summary of Parameters

X,Y,Z Same as for PERVUE
KODE = -1/+1 for pen up/down
KHAR = -1/+N for no symbol or symbol #N
XP or RP = End point on I-axis
YP or TP = End point on J-axis
ZP = End point on Z-axis
IDE = -1/0 for no/yes hidden line removal

Note: in COMMON /PERCOM/ XPEN,YPEN

XPEN = Real pen X-position, in inches, of last XP, YP, ZP
YPEN = Real pen Y-position, in inches, of last XP, YP, ZP

PERVUE: Summary of Parameters

LGEOM = 1/2/3 + 10*NZ for rectangular/degree/radian plot of NZ surfaces.

X or R(I) Values on I-axis

Y or T(J) Values on J-axis

Z(I,J) Array which defines surface height

VX or VR Viewing point on I-axis, with viewed point and scale if LGEOM<0

VY or VT Viewing point on J-axis, with viewed point and scale if LGEOM<0

VZ Viewing point on Z-axis, with viewed point and scale if LGEOM<0

NXDIM Real I-dimension of Z

NYDIM Real J-dimension of Z

NX Limit on I

NY Limit on J

LGRID = -1/0/1/2 for no/all/I/J grid lines

NCONTR >0 for equally spaced intervals; <0 for -NC and list of Z-values for particular contours

IDE = -1/0 for no/yes hidden line removal

ZMIN Minimum Z value plotted

ZMAX Maximum Z value plotted

KBOX = 0/1/2 for no/all/visible frame

KOLOR = 0/1 for no/yes colored pens

KROSS = -1/0 for lengthwise/crosswise Z-axis

PWIDE Plot region width in inches

PLONG Plot region length in inches

NEND = -1/0 to suppress/allow internal ENDPLT

NCHAR <50 number of characters in LABEL

LABEL Plot label as vector of characters or literal constant