NASA Technical Memorandum 78772

DESCRIPTION OF A FORTRAN

SUBROUTINE FOR PLOTTING

THREE-DIMENSIONAL DATA

(NASA-TM-78772) DESCRIPTION OF A FORTRAN SUBROUTINE FOR PLOTTING THREE-DIMENSIONAL DATA (NASA) 29 p HC A03/MF A01 CSCL 09B

N78-32751

Unclas

G3/61 31581

Michael R. James and Wendell W. Kelley
Langley Research Center
Hampton, Virginia

National Aeronautics and Space Administration

Langley Research Center
Hampton, Virginia 23665

SUMMARY

A FORTRAN subroutine is described which provides the capability to plot three-dimensional data on an interactive cathode-ray-tube computer terminal or conventional plotter. The plotted data, which must be described in terms of a series of two-dimensional curves, appears to form a surface in three-dimensional space. Features of the subroutine include a capability for hidden-line computations. User instructions and sample programs are described.

INTRODUCTION

The subroutine DEPTHVW was developed in connection with a need for plotting wind-shear profiles, in which the magnitude and direction of the wind varied with both altitude and distance from the runway. Conventional two-dimensional plots, which only display functions of one independent variable, were not adequate for presenting the spatial distribution of the wind velocities. The DEPTHVW three-dimensional plots are intended primarily for obtaining a qualitative appraisal of the data rather than quantitative values. Axes and scales are thus omitted from the plots in favor of picture clarity.

The general-purpose design of the subroutine makes it suitable for plotting any data which can be expressed as a function of two independent variables. DEPTHVW is written in FORTRAN and uses the set of plot subroutines contained in the TEKTRONIX Terminal Control System software package (ref. 1). The user must therefore have access to the TEKTRONIX package, although no familiarity with its contents is necessary. The user need only specify in his program a set of parameters which determines various plotting format options and identifies the data to be plotted.

SYMBOLS AND ABBREVIATIONS

CRT	cathode-ray tube
DEPTHVW	name of the three-dimensional plotting subroutine
FORTRAN	a scientific programming language
LRC	Langley Research Center
NOS	Network Operating System

PROGRAM THEORY AND DESCRIPTION

A function of two variables can be represented by a surface in three-dimensional space. The two independent variables define a point on a base plane, and the value of the function is the height of the surface above or below the plane at that point. If one of the independent variables is fixed, and the other is permitted to assume a range of values, the function will represent a "slice" of the three-dimensional surface. We are then restricted to a curve on the surface, and the curve is in a plane, i.e., two-dimensional. If the process is repeated by incrementing the value of the fixed variable, the entire surface may be reduced to a set of curves that can be plotted.

The subroutine DEPTHVW depicts three-dimensional information which is described in terms of these surface slices. The slices are plotted slightly offset from each other to give the appearance of a surface (fig. 1). Using a greater number of slices, which brings the surface lines closer together, generally improves the surface-like appearance. A representation of the base plane and a zero-reference line are plotted to aid the viewer in visualizing the date surface (fig. 2).

The user must provide his own data slices (in the form of a FORTRAN data array) one at a time to the subroutine DEPTHVW, along with several plot-control parameters. The subroutine then plots the data at the user's cathode-ray-tube (CRT) terminal, or alternatively generates a vector file which may be used for off-line plotting. User specification of the control parameters determines the disposition of hidden lines, endpoint connections, and various other options, each of which is explained in the next section.

CRT terminals which are compatible for use with the subroutine include the TEKTRONIX 4010, 4012/13, and 4014/15 Computer Display Terminals. NASA Langley Research Center (LRC) users may also convert the subroutine output to a format suitable for off-line plotting by following the procedures outlined in appendix A.

PROGRAM USE

Prospective users must have access to the DEPTHVW subroutine and the TEKTRONIX Terminal Control System software package. For Langley Research Center users, the following job control statements are required (ref. 2 and 3):

```
GET (DVWLIB/UN = 474795N)
ATTACH (LIBFTEK/UN = LIBRARY, NA)
LDSET (LIB = DVWLIB)
LDSET (LIB = LIBFTEK)
```

DEPTHVW and two supplementary subroutines (a hidden-line routine and an endpoint-connection routine) are contained in the library file DVWLIB, while the TEKTRONIX routines are contained in the library file LIBFTEK. The core storage requirement for DEPTHVW and the necessary TEKTRONIX modules is approximately 65K octal words.

For users who do not have access to DVWLIB, a FORTRAN source listing of the DEPTHVW routines is provided in appendix B. Langley Research Center users may also access the DEPTHVW source statements by using the following command:

```
GET (DVSORCE/UN = 474795N)
```

The user's FORTRAN main program must assign values to several plot control parameters, create or access the data arrays, and perform a subroutine call to DEPTHVW. The syntax of the subroutine call is:

```
CALL DEPTHVW (VLF,MODE,X,Y,M,N,XMIN,XMAX,YMIN,YMAX,IHIDE,IBAUD)
```

Each of the subroutine arguments has a specified type, purpose, values and limitations. For all arguments except X and Y, values must be specified prior to the first call to DEPTHVW and should not be altered on subsequent calls.

The use of each argument is explained below. Also appendix C contains sample programs which were used to plot figures 1-10 on a TEKTRONIX 4014 Computer Display Terminal.

VLF. - Viewpoint Location Factor is a real variable which may take on values between 0.0 and 1.0. The purpose of this parameter is to vary the amount of overlap of slices in the vertical direction, thus altering the apparent viewpoint. Low values for VLF cause large overlap and the viewing direction appears to be almost parallel to the base plane (fig. 2). Values near 1.0 cause little overlap and the viewer appears to be looking down on top of the data surface (fig. 3). The value of VLF required to get the best viewpoint will vary with the number of curves, the smoothness of the data, and user preference. In general, the greater the number of curves, the smaller VLF must be to achieve the same viewpoint.

MODE. - This is an integer variable with possible values 0, 1, 2 and 3. This parameter specifies the method of handling endpoints for each data slice. A value of 0 specifies no endpoint connection (fig. 4), 1 specifies connection to the base plane (fig. 5), 2 causes endpoints to be connected to each other and to the base plane (fig. 6), and 3 causes endpoints to be connected only to each other (fig.7).

X,Y.- These are real arrays containing the data for one slice. Data must be sent to the subroutine one slice at a time in order to reduce subroutine storage requirements. Thus if twenty curves are to be plotted, the subroutine must be called twenty times with the X and Y arrays containing data for a new slice each time. The x-coordinates are in the X array and the corresponding y-coordinates in the Y array. Thus X and Y are of equal length. The values in the X array should be in strictly increasing order if the user intends to use the hidden-line capability. Also, the values in the Y array should be within the minimum and maximum values specified in YMIN, YMAX or clipping will result (especially on the first and last curves). The values in the X array must be within the limits specified in XMIN, XMAX. On the terminal screen, XMAX is in the horizontal direction and YMAX is vertical.

<u>M.</u> - This is an integer variable which specifies the length of X and Y arrays, i.e., number of data points.

<u>N.</u> - This is an integer variable which specifies the number of curves (slices) to be plotted.

<u>XMIN, XMAX.</u> - These are real variables describing the minimum and maximum values of the x-coordinate scale. Data in the X array should be within these limits. Values of XMIN or XMAX may be negative, zero, or positive, but XMAX must be greater than XMIN. The values for XMIN and XMAX should be chosen as close to the actual data limits as possible. If the data exceeds these values the curve will extend off the base plane, and other unpredictable results may occur. On the other hand, if the range of XMIN and XMAX far exceeds the range of data values, the screen area is not utilized efficiently.

<u>YMIN, YMAX.</u> - These are real variables describing the minimum and maximum values of the y-coordinate scale. Values may be any real number within

hardware limitations, but YMAX must be greater than YMIN. Clipping will result, especially in the first and last curves due to screen height limitations, if the data in the Y array exceeds these limits.

IHIDE. - This is an integer variable which specifies the method of handling hidden lines. Possible values of this variable are 0, 1 or 2. A value of 0 specifies no hidden-line computation is desired, i.e., lines which would normally be hidden will be plotted as solid lines (fig. 8). A value of 1 causes hidden lines to be plotted as dashed lines (fig. 9). A value of 2 causes hidden lines to be invisible (fig. 10).

If the hidden-line capability is used (that is, IHIDE nonzero) the user should have at least 100 data points per curve. Fewer points than this will reduce the effectiveness of the hidden-line checking algorithm. For example, a supposedly invisible line will remain visible for some distance even after going behind a surface. Also, the X array should be arranged in strictly increasing order. Finally, the hidden-line routine checks only the previous 10 curves; if a line is hidden only by a slice 11 or more curves back, it will be visible. Despite these limitations, the hidden-line capability is useful in reducing the confusion of several curves close together on the screen.

IBAUD. - This integer variable specifies the computer terminal transmission rate in characters per second.

CONCLUDING REMARKS

DEPTHVW is a subroutine which plots a series of data points so that they appear as a surface in three-dimensional space. It does not draw a continuous surface, but instead plots a series of two-dimensional curves spaced properly so as to give the appearance of a surface.

The subroutine permits a qualitative, easily interpreted visualization of data that depends on two independent variables. It has several limitations, however. It was not designed to depict complex three-dimensional objects such as structures. Also, to make full use of its features, the data must be in strict order and within specified limits.

The subroutine is still in a developmental stage, and may contain deficiencies that will show up with the use of different data. The authors solicit user comments on the subroutine in order that further improvements can be made.

Langley Research Center
National Aeronautics and Space Administration
Hampton, VA 23665
August 1, 1978

APPENDIX A

USING DEPTHVW WITH LRC GRAPHICS POSTPROCESSOR

DEPTHVW may be used to create three-dimensional plots without an interactive CRT terminal, utilizing the LRC graphics postprocessor to create Varian or CalComp plots. The following steps are necessary:

A. Changes to Main Program

Access to the LIBFTEK common block /JTB/ is required. The user must include the following statements in his main program:

```
COMMON/JTB/ NFR, JREQ, DUMMY(8)
.
.
.
JREQ = 1
```

The first is a non-executable statement and must occur before any executable statements in the program.

Setting the value JREQ = 1 in /JTB/ causes the plot vector file to be written to file TTPE16 instead of file OUTPUT (the CRT screen) in order that the data may be plotted off-line. In this case, the plot will not appear on the screen, if the program is being run at a CRT. The user may also obtain a plot on the screen by using the following batch control statements:

```
ASCII.
LNH,F = TTPE16.
```

B. Plot Vector File Conversion

The plot vector file must be converted from TEKTRONIX CRT format to a format acceptable to the LRC graphics postprocessor. This is done with procedure file MCOMP, and can be done interactively or in batch. Control statements necessary are (TTPE16 is a local file):

```
GET, MCOMP/UN = 075768N
REWIND, TTPE16, MCOMP.
CALL (MCOMP)
REWIND, SAVPLT.
```

MCOMP converts TTPE16 to a LRC graphics vector file which is stored in SAVPLT.

C. Postprocessing

NOS does not allow postprocessing to be done interactively. The vector file conversion described above can be done interactively and postprocessing in a separate batch job, or the two steps can be combined in a single batch job.

Note that if conversion and postprocessing are done as a single batch job, the last two control statements in procedure file MCOMP should be deleted before it is called. That is, the statements,

```
EXIT.
REDUCE.
```

should be deleted from the user's copy of MCOMP before it is used in the combined job described above.

Postprocessing requires the following batch control statements (SAVPLT is a local file):

```
ATTACH (PLOT/UN = LIBRARY)
PLOT.  "Device Name". . .
```

Reference 4 gives details on the LRC plotting devices and how to control them with the PLOT control card.

APPENDIX B

SOURCE LISTING OF DEPTHVW SUBROUTINES

```
      SUBROUTINE DEPTHVW(VLF1,NMODE1,XP,YP,M1,N1,
     *XMIN1,XMAX1,YMIN1,YMAX1,IHIDE1,IBAUD)
      DIMENSION XP(M1),YP(M1)
      COMMON/PARAM/ VLF,NMODE,IHIDE,
     *IBLNK,TEST,HFAC,DELX,DELY,M,N,I,J1,
     *IBLNK2,IFLG,XMAX
      DATA IBLNK/0/,IFIRST/1/
C
      IF(IFIRST.NE.1) GO TO 100
C     LOAD THE COMMON BLOCK
      VLF=VLF1
      NMODE=NMODE1
      IHIDE=IHIDE1
      M=M1
      N=N1
      XMAX=XMAX1
      XMIN=XMIN1
      YMIN=YMIN1
      YMAX=YMAX1
C     INITIALIZE AND CALCULATE WINDOW SIZE
      CALL INITT(IBAUD)
      I=1
      IWDTH=1024*2/3
      VHGHT=(YMAX-YMIN)*N*VLF+(YMAX-YMIN)
      VWDTH=1.5*(XMAX-XMIN)
      VBOT=YMIN
      VLFT=XMIN
      CALL VWINDO(VLFT,VWDTH,VBOT,VHGHT)
C     DRAW BASE PLANE
      IHGHT=(VHGHT-(YMAX-YMIN))/VHGHT*780
      CALL MOVABS(0,0)
      CALL DRWABS(IWDTH,0)
      CALL DRWABS(1024,IHGHT)
      IFIRST=0
  100 IF(I.GT.N) GO TO 999
      HFAC=(XP(M)-XP(1))/100.
      TEST=XP(1)
      CALL MOVEA(VLFT,VBOT)
C     DETERMINE OFFSET FOR THIS CURVE
      DELX=VWDTH/3.*(I-1)/(N-1)
      DELY=(VHGHT-(YMAX-YMIN))*(I-1)/(N-1)
      IFLG=0
      DO 200 J=1,M
C     DETERMINE COORDINATES AND GRAPH POINT
      XCRD=XP(J)+DELX
      YCRD=YP(J)+DELY
      IF(J.EQ.1) XCD2=XMIN+DELX
      IF(J.EQ.1) YCD2=YCRD
```

```
      IF(J.EQ.1) CALL MOVEA(XCRD,YCRD)
      J1=J
      IF(IHIDE.NE.0.A.XP(J).GE.TEST) CALL HIDDEN(XCRD,YCRD)
      CALL DASHA(XCRD,YCRD,IBLNK)
  200 CONTINUE
C     ENDPOINT HANDLING
      DOWN1=YP(J1)-YMIN
      CALL MOVEA(XMAX+DELX,YCRD)
      IF(I.EQ.1) DOWN2=YP(1)-YMIN
      CALL ENDPT(XCD1,YCD1,DOWN1,DOWN2,XCD2,YCD2)
      XCD1=XMAX+DELX
      YCD1=YCRD
      I=I+1
  999 RETURN
      END
      SUBROUTINE ENDPT(XCD1,YCD1,DOWN1,DOWN2,
     *XCD2,YCD2)
      COMMON/PARAM/ VLF,NMODE,IHIDE,IBLNK,TEST,HFAC,DELX,
     *DELY,M,N,I,J,IBLNK2,IFLG,XMAX
      DATA IBLNK3/0/
C
C     MODE 0 IS DO NOTHING
      IFLG=1
      IF(NMODE.EQ.0) GO TO 99
C     MODE 3 IS CONNECT ONLY FIRST AND LAST CURVES
C       TO BASE; CONNECT ENDS TO EACH OTHER
      IF(NMODE.EQ.3.A.I.NE.1.A.I.NE.N) GO TO 20
C     MODE 1 IS CONNECT ENDS TO BASE ONLY
      CALL DRAWR(0.,-DOWN1)
      CALL MOVER(0.,DOWN1)
      IF(NMODE.EQ.1) GO TO 99
   20 IF(I.EQ.1) GO TO 30
C     CONNECT ENDS TO EACH OTHER
      CALL DRAWA(XCD1,YCD1)
      CALL MOVEA(DELX+XMAX,DELY)
      CALL DRAWA(XCD4,YCD4)
   30 CALL MOVEA(XCD2,YCD2)
      CALL HIDDEN(XCD2,YCD2)
      IBLNK2=IBLNK
      IF(I.EQ.1) GO TO 50
C     CHECK FOR HIDDEN END-LINES ON LEFT SIDE
   40 IF(IBLNK2.NE.0.O.IBLNK3.NE.0) GO TO 70
      GO TO 80
C     DRAW CONNECT TO BASE ON LEFT SIDE IF FIRST CURVE
   50 CALL DRAWR(0.,-DOWN2)
      CALL MOVER(0.,DOWN2)
C     PUT PRESENT ENDPT COORD. IN OLD COORD. LOCATIONS
   60 XCD3=XCD2
      YCD3=YCD2
```

```
      IBLNK3=IBLNK2
      XCD4=DELX+XMAX
      YCD4=DELY
      GO TO 99
C     DETERMINE HOW MUCH OF LEFT END-LINE IS HIDDEN
C     AND DRAW END-LINE CORRECTLY
   70 IF(IBLNK2.NE.0.A.IBLNK3.NE.0) GO TO 80
      XRT=XCD2
      XLT=XCD3
      YTOP=YCD2
      YBOT=YCD3
      IFLG=1
      DO 79 ITST=1,10
      XMID=(XRT+XLT)/2.
      YMID=(YTOP+YBOT)/2.
      CALL HIDDEN(XMID,YMID)
      IF(IBLNK.EQ.0) GO TO 75
      IF(IBLNK2.EQ.0) GO TO 76
   74 XRT=XMID
      YTOP=YMID
      GO TO 79
   75 IF(IBLNK2.EQ.0) GO TO 74
   76 XLT=XMID
      YBOT=YMID
   79 CONTINUE
      CALL DASHA(XMID,YMID,IBLNK2)
   80 CALL DASHA(XCD3,YCD3,IBLNK3)
      GO TO 60
   99 RETURN
      END
      SUBROUTINE HIDDEN(XCRD,YCRD)
      DIMENSION XCD(10,110),YCD(10,110)
      COMMON/PARAM/ VLF,NMODE,IHIDE,IBLNK,TEST,HFAC,DELX,
     *DELY,M,N,I,J,IBLNK2,IFLG,XMAX
C
C     INITIALIZE IF FIRST CURVE,FIRST POINT
      IF((I.NE.1.OR.J.NE.1).A.IHIDE.NE.0) GO TO 5
      DO 4 JBLA=1,10
      DO 4 KBLA=1,110
      XCD(JBLA,KBLA)=-1000000.
      YCD(JBLA,KBLA)=-1000000.
    4 CONTINUE
    5 IF(J.EQ.1) GOTO30
C      DETERMINE IF POINT IS HIDDEN
    6 IF(I.EQ.1) GO TO 20
      DO 16 L=2,10
      IF(XCRD.GT.XCD(L,LST)) GO TO 16
      IF(XCRD.LT.XCD(L,1)) GO TO 16
      DO 10 K=1,LST
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      IF(XCRD.LT.XCD(L,K)) GO TO 15
   10 CONTINUE
   15 IF(YCRD.LT.YCD(L,K)) GO TO 21
   16 CONTINUE
C     SET HIDDEN FLAG
   20 IBLNK=0
      GO TO 22
   21 IBLNK=-1
      IF(IHIDE.EQ.1) IBLNK=34
   22 IF(IFLG.NE.0) GO TO 99
      XCD(1,ICNT)=XCRD
      YCD(1,ICNT)=YCRD
      IF(J.NE.1) TEST=TEST+HFAC
      ICNT=ICNT+1
      GO TO 99
   30 IF(IFLG.NE.0) GO TO 6
      IF(I.NE.1) LST=ICNT-1
      ICNT=1
      IF(IFLG.NE.0) GO TO 6
C     LOAD PRESENT COORD. INTO OLD COORD. LOCATIONS
      DO 35 KO=1,110
      DO 35 L=1,9
      LO=11-L
      XCD(LO,KO)=XCD(LO-1,KO)
   35 YCD(LO,KO)=YCD(LO-1,KO)
      GO TO 6
   99 RETURN
      END
```

APPENDIX C

SOURCE LISTING OF SAMPLE PROGRAMS

NOTE: Program CHECK1 was used to generate figures 1, 8, 9 and 10. Program HYPLOID was used to generate figures 2-7.

```
      PROGRAM CHECK1(INPUT,OUTPUT)
C     THIS PROGRAM PLOTS A SPIKE
      DIMENSION X(1001),Y(1001)
      PRINT*,"NMODE,VLF,IHIDE"
      READ*,NMODE,VLF,IHIDE
C     CREATE DATA ARRAYS
      DO 20 I=1,21
      Z=EXP(FLOAT((I-1)*(21-I))-100.)
      DO 10 J=1,1001
      X(J)=FLOAT(J-1)/100.
      Y(J)=Z*EXP(X(J)*(10.00-X(J)))*75./EXP(5.**2)
   10 CONTINUE
C     TRANSFER TO DEPTHVW WITH DATA ARRAYS AND CONTROL
C     PARAMETERS - - 21 TIMES
      CALL DEPTHVW(VLF,NMODE,X,Y,1001,21,
     *0.,10.,-10.,50.,IHIDE,120)
   20 CONTINUE
C     WAIT UNTIL USER PRESSES "RETURN"
      CALL TPAUSE
      END
```

```
      PROGRAM HYPLOID(INPUT,OUTPUT)
C     THIS PROGRAM PLOTS A HYPERBOLIC SURFACE
      DIMENSION X(1000),Y(1000)
      DATA XMIN,XMAX,YMIN,YMAX,M,N/-100.,100.,-10000.,
     *10000.,1000,20/
      PRINT*,"MODE,VLF,IHIDE ?"
      READ*,MODE,VLF,IHIDE
C     DATA CREATION AND CALL TO DEPTHVW - - N CURVES,
C       M DATA POINTS PER CURVE,N CALLS TO PLOTTER
      DO 20 I=1,N
      FIX=100.-(I-1)*200./19.
      DO 10 J=1,M
      X(J)=-100.+(J-1)*200./999.
      Y(J)=(X(J)**2-FIX**2)
   10 CONTINUE
      CALL DEPTHVW(VLF,MODE,X,Y,M,N,XMIN,XMAX,
     *YMIN,YMAX,IHIDE,120)
   20 CONTINUE
C     WAIT UNTIL USER PRESSES "RETURN" KEY
      CALL TPAUSE
      END
```

REFERENCES

1. TEKTRONIX Terminal Control Systems Manual, No. 062-1474-00. January, 1976.

2. NOS 1.2 Cyber Loader Reference Manual, No. 60429800. Control Data Corporation. March, 1978.

3. NOS 1.2 Time Share Reference Manual, No. 60435500. Control Data Corporation. March, 1978.

4. LRC Graphics Programming Manual, (Vol. IV of LRC Computer Programming Manuals). Section 1.3. January, 1975.

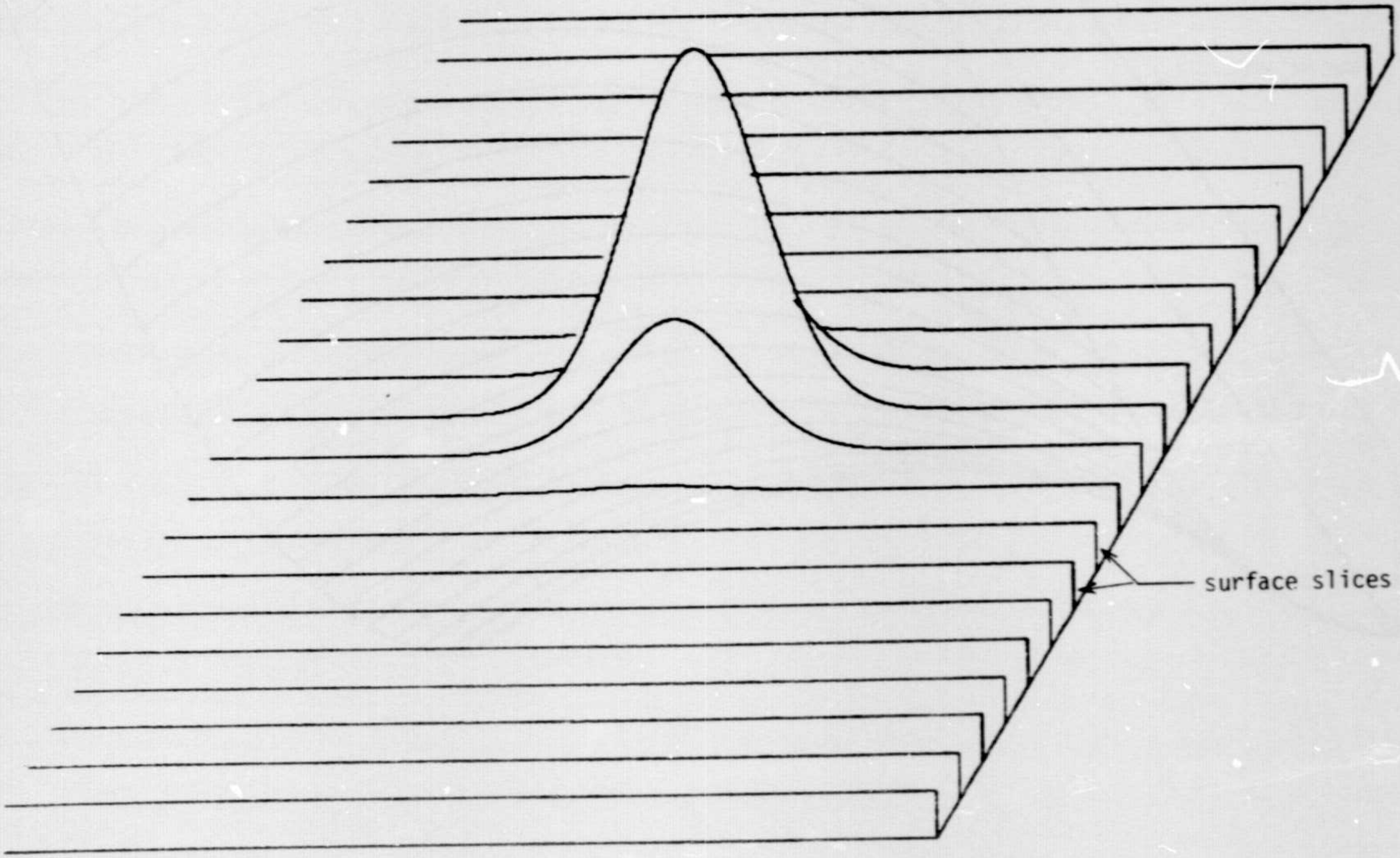

Figure 1.- Three-dimensional surface shown as a series of two-dimensional slices.

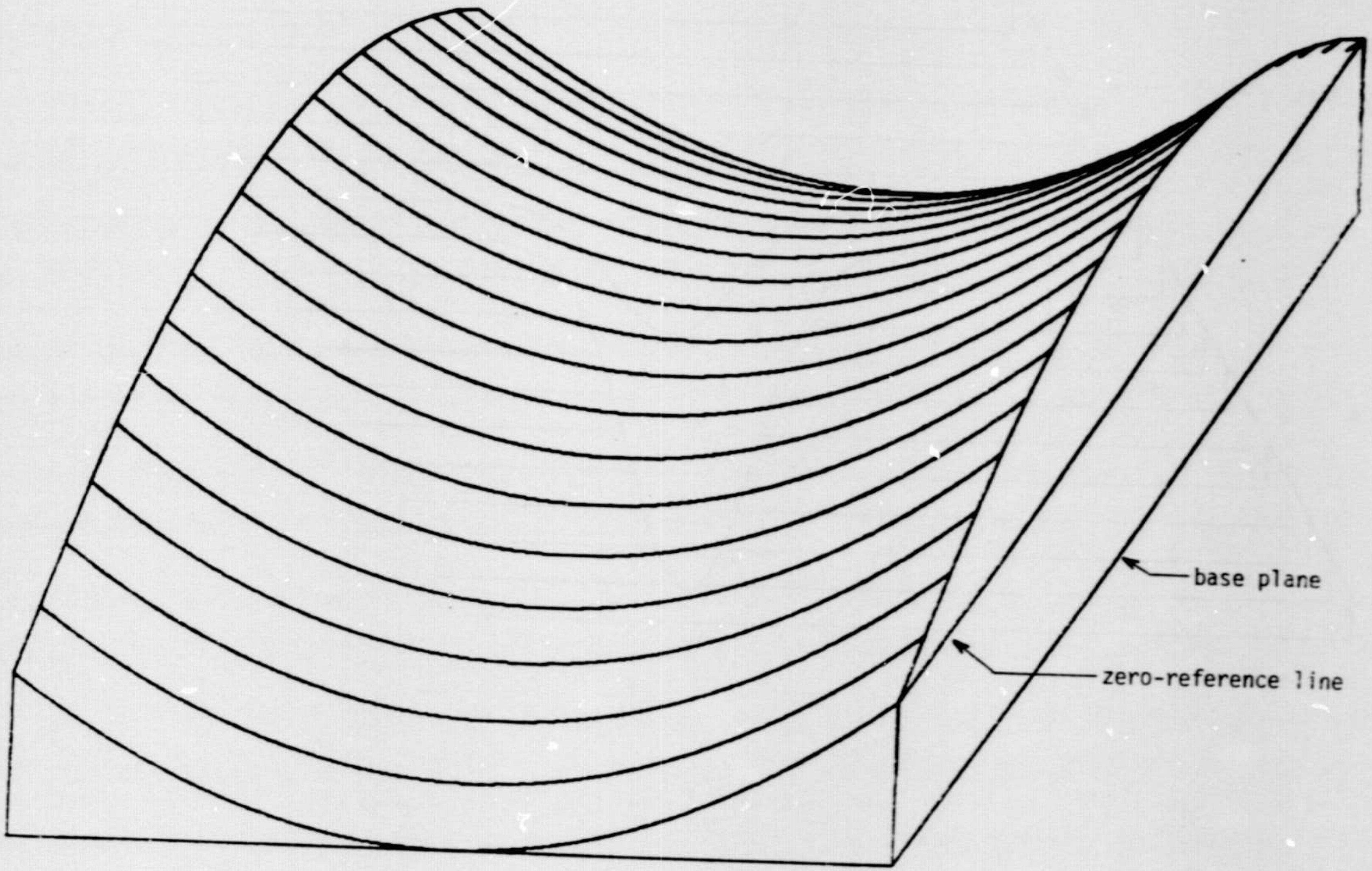

Figure 2.- Hyperbolic surface; VLF= 0.1.

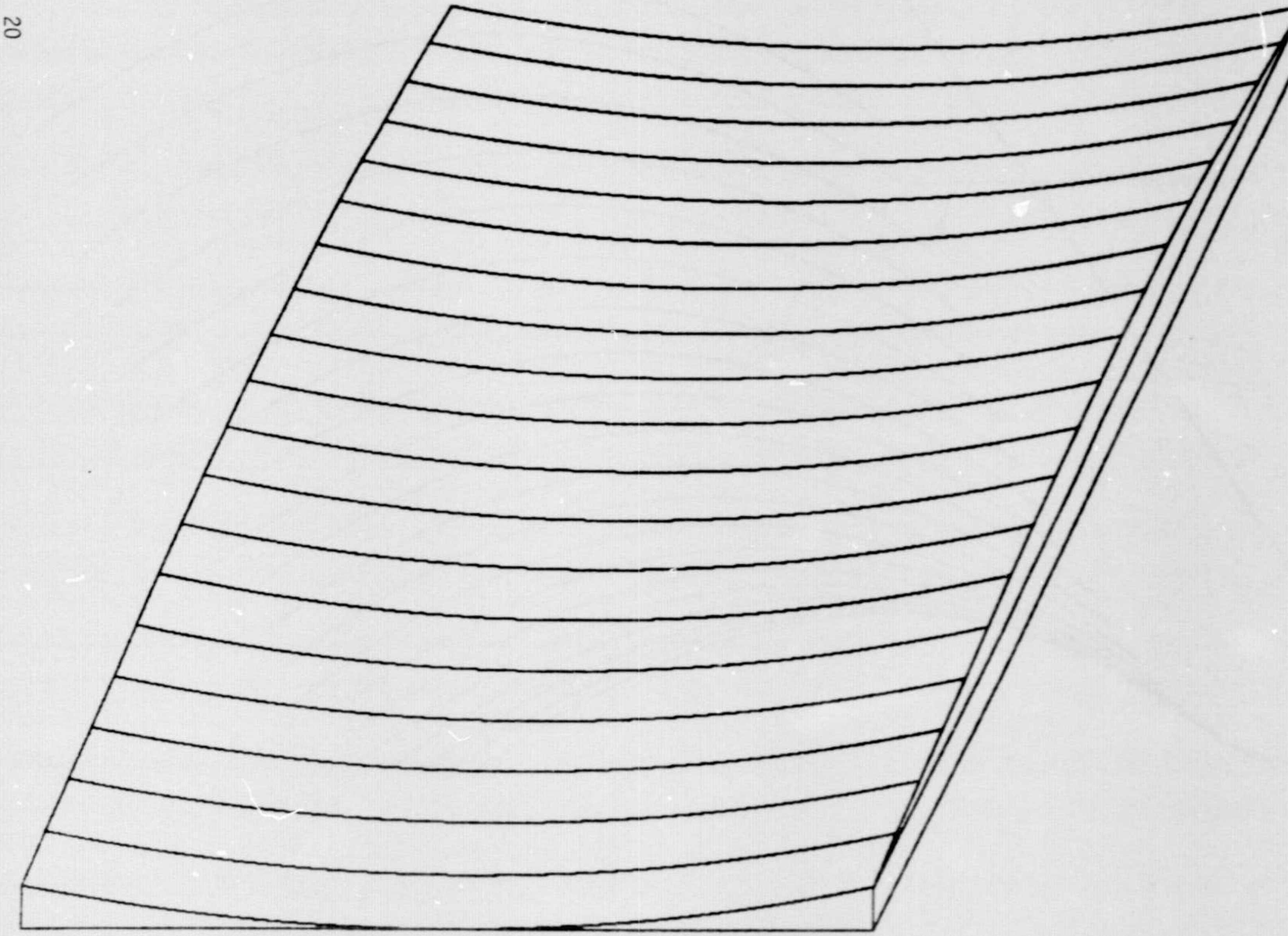

Figure 3.- Hyperbolic surface; VLF= 0.5.

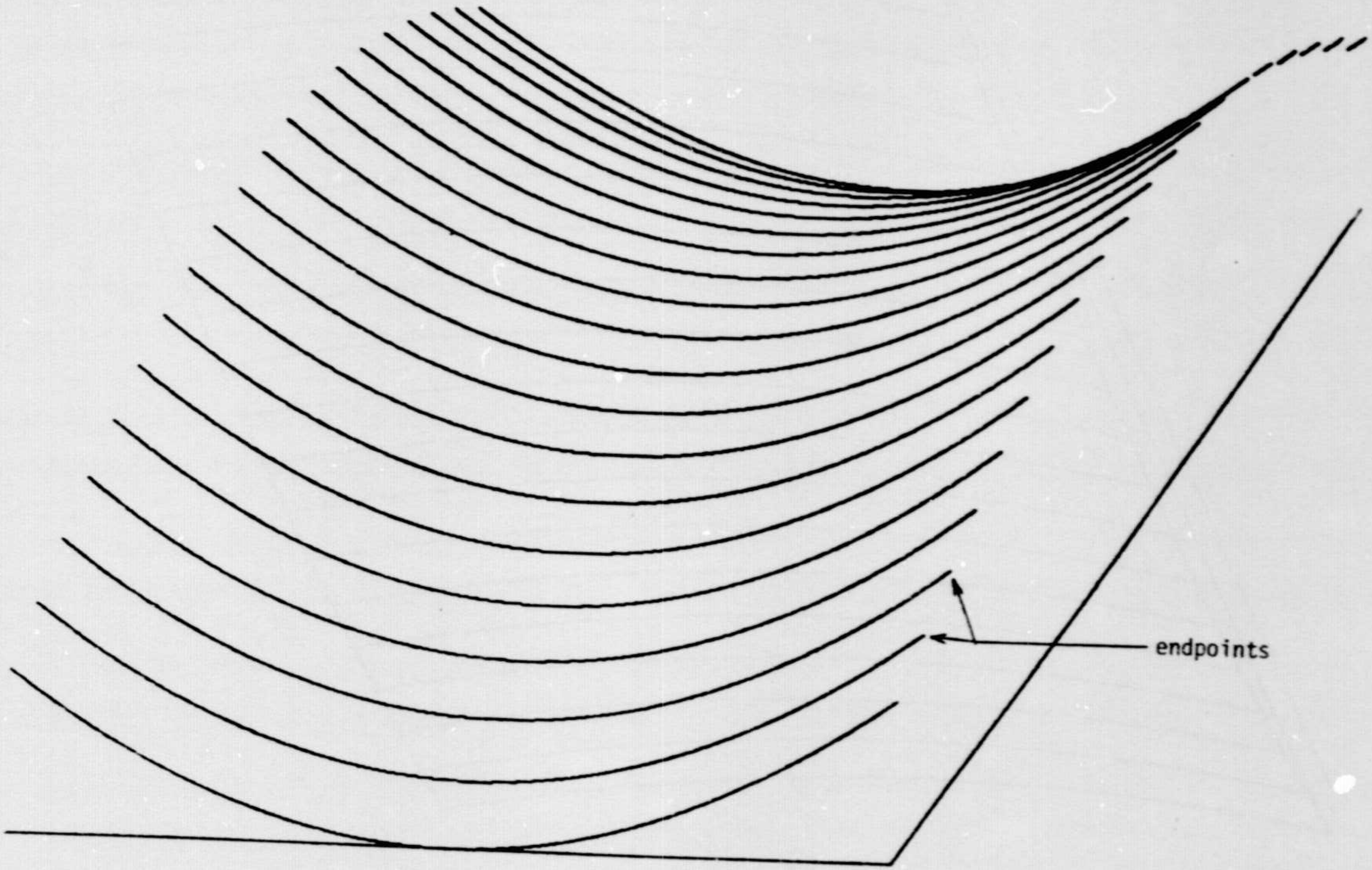

Figure 4.- No endpoint connection; MODE = 0.

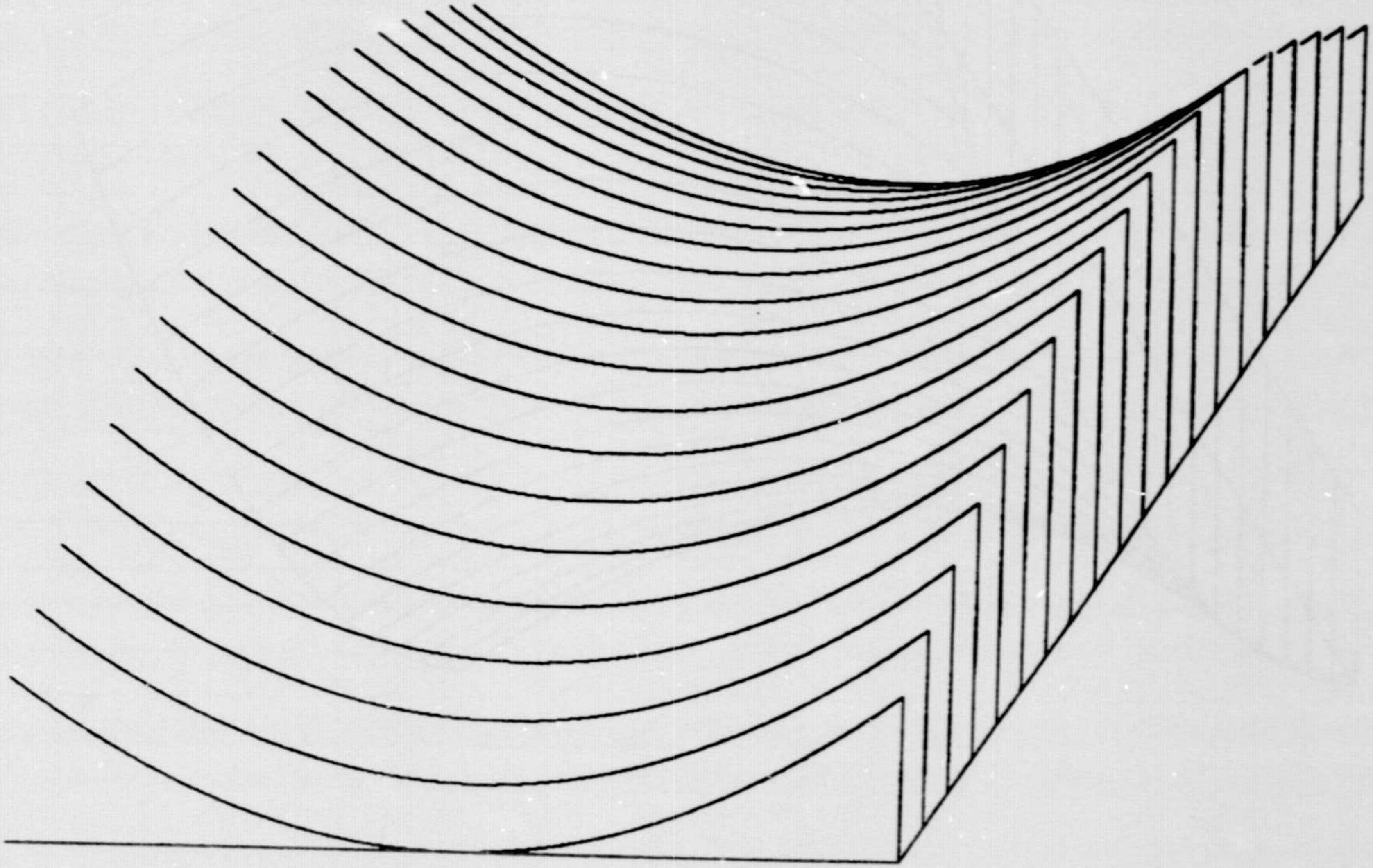

Figure 5.- Endpoints connected to the base plane; MODE = 1.

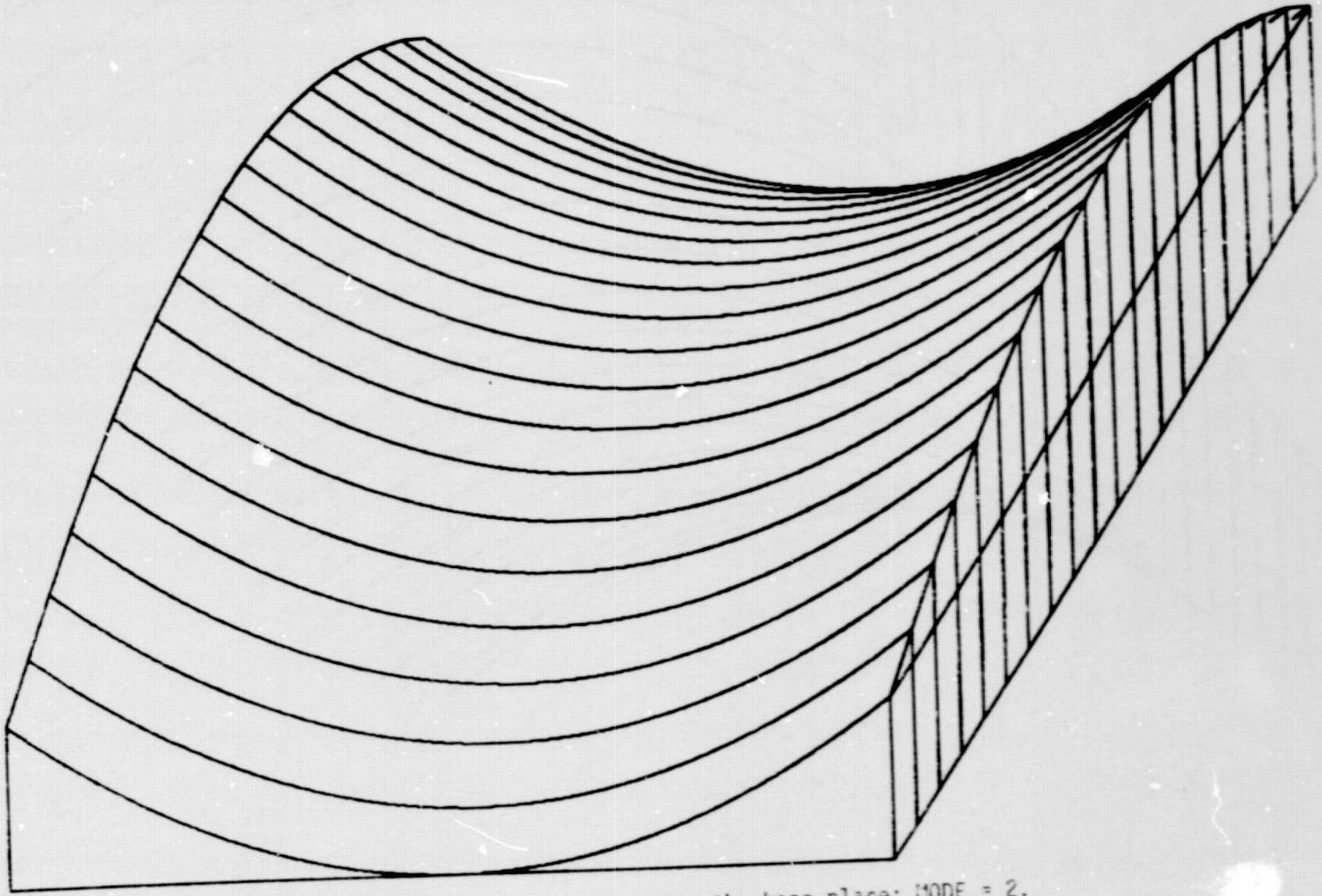

Figure 6.- Endpoints connected together and to the base plane; MODE = 2.

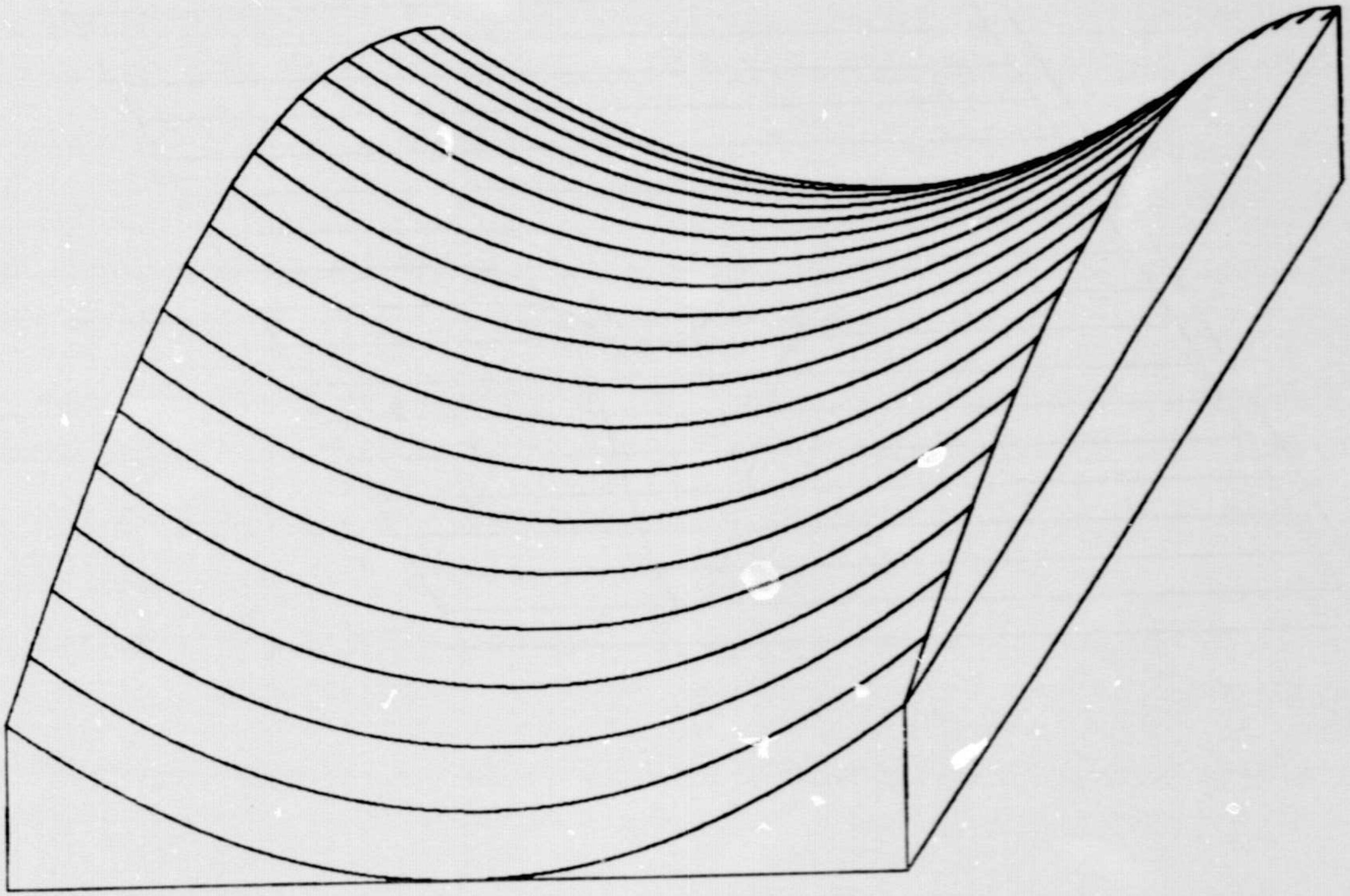

Figure 7.- Endpoints connected together; MODE = 3.

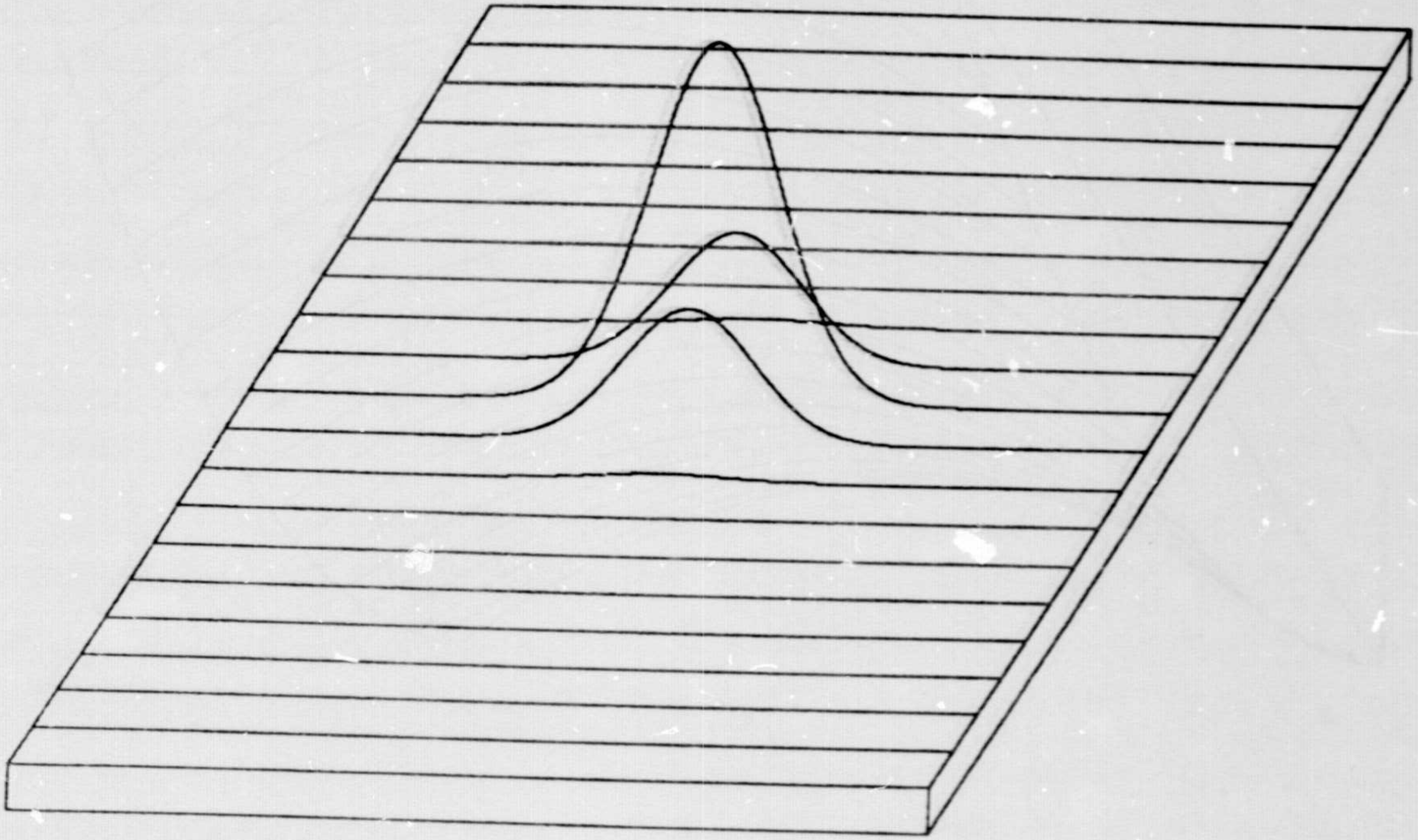

Figure 8.- No hidden-line computations; IHIDE = 0.

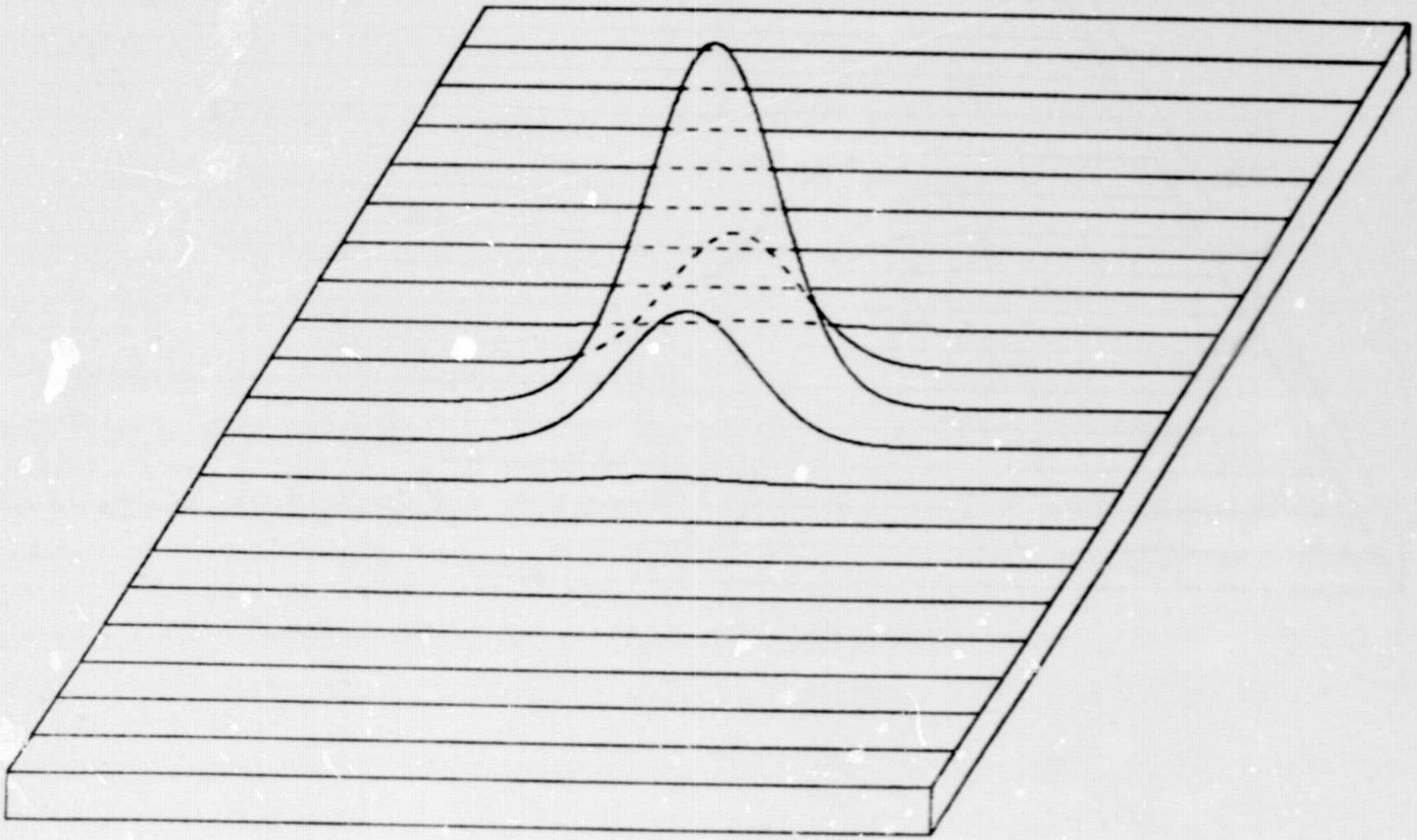

Figure 9.- Hidden lines shown as dashed curves; IHIDE = 1.

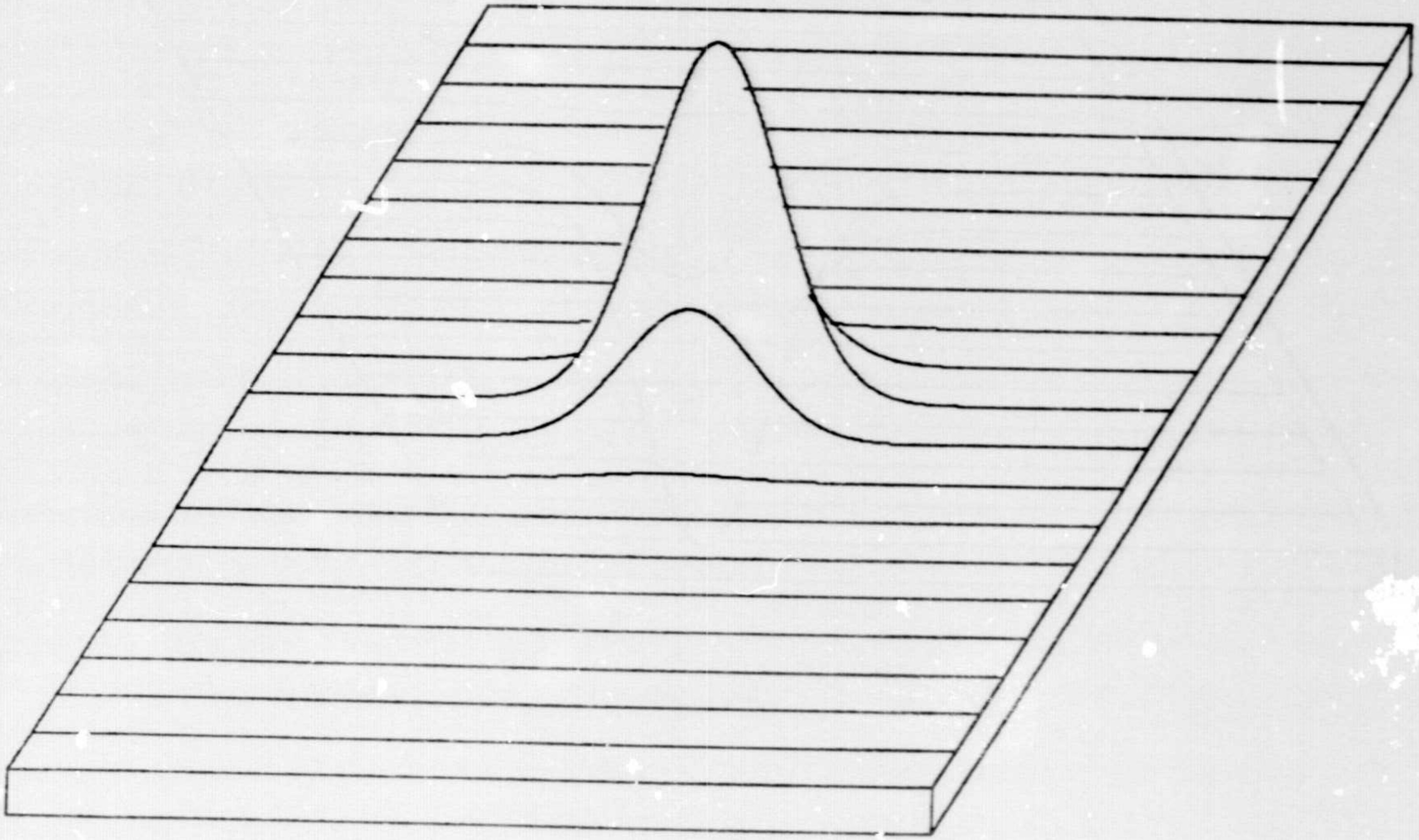

Figure 10.- Hidden lines not visible; IHIDE = 2.