

AD-776 204

A FORTRAN SUBROUTINE TO PRODUCE PLOTS
OF TWO DIMENSIONAL SURFACES USING A
LINE PRINTER

Michael A. Tamny

Naval Research Laboratory
Washington, D. C.

November 1973

DISTRIBUTED BY:

NTIS

National Technical Information Service
U. S. DEPARTMENT OF COMMERCE
5285 Port Royal Road, Springfield Va. 22151

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NRL Memorandum Report 2756 NRL Computer Bulletin 36	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER AD 776204
4. TITLE (and Subtitle) A FORTRAN SUBROUTINE TO PRODUCE PLOTS OF TWO DIMENSIONAL SURFACES USING A LINE PRINTER	5. TYPE OF REPORT & PERIOD COVERED This is a final report on this phase of the problem.	
7. AUTHOR(s) Michael A. Tamny	6. PERFORMING ORG. REPORT NUMBER	
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Research Laboratory Washington, D.C. 20375	8. CONTRACT OR GRANT NUMBER(s)	
11. CONTROLLING OFFICE NAME AND ADDRESS Department of the Navy (Office of Naval Research) Washington, D.C. 20360	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NRL General & Administrative Function R02-46C	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	12. REPORT DATE November 1973	
	13. NUMBER OF PAGES 40	
	15. SECURITY CLASS. (of this report) UNCLASSIFIED	
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE	
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES Reproduced by NATIONAL TECHNICAL INFORMATION SERVICE U S Department of Commerce Springfield VA 22151		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Computer Subroutine Picture Quantization Gray-Scale Plotting		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) A FORTRAN subroutine has been written to produce line printer plots of a function of two variables, $f(x, y)$, using the CDC-3800. Line printer plots are valuable for users who want a fast and economical method of producing plots but do not require a high resolution capability.		

CONTENTS

1.0	IDENTIFICATION	1
2.0	PURPOSE	2
3.0	USAGE	6
4.0	METHOD OR ALGORITHM	15
5.0	SOURCE LANGUAGE LISTING	20
6.0	COMPARISON	35
7.0	TEST METHOD AND RESULTS	35
8.0	REMARKS	37
Figure 1	- Partitioning of Data Range with a Linear Scale	4
Figure 2	- Rough Transfer Characteristics for Data Scales 0 through 3	14
Figure 3	- Linear Scales Relocated	17
Table 1	- Scale Algorithms	18
Table 2	- Definitions of Terms	19

1.0 IDENTIFICATION

1.1 Title

Line Printer Plots of Two-Dimensional Surfaces

1.2 Identification Name

J5-NRL-SURFPLØT

1.3 Classification Code

J5-Output, Plotting

1.4 RCC Identification Number

J5008Ø00

1.5 Entry Points

SURFPLØT

1.6 Programming Language

Language: 3600/3800 FØRTRAN

Routine Type: Subroutine

Operating System: DRUM SCØPE 2.1

1.7 Computer and Configuration

CDC-3800

1.8 Contributor or Programmer

Michael A. Tamny, Code 5365T, Airborne Radar Branch,
Radar Division

1.9 Contributing Organization

NRL - Naval Research Laboratory,
Washington, D.C. 20375

Manuscript submitted February 15, 1974.

1.10 Program Availability

1.10.1 Submittal:

Program write-up, Fortran source deck
with comments, source listing.

1.10.2 On File:

RCC Program Library

1.11 Verification

A sequence of Cassini Ovals, defined by the equation $[(x + a)^2 + y^2][(x - a)^2 + y^2] = k^4$, where k equals the amplitude and a equals the constant 3.0, was used to test SURFPLØT. Several different subroutine parameter values were used. The resultant plots compared favorably with those produced by the contour plotting program J5-ØRNL-CØNTØUR on the CalComp plotter.

1.12 Date

November 1973

2.0 PURPOSE

2.1 Description of the Routine

The routine SURFPLØT generates a plot of two-dimensional surfaces using the line printer. The plot is a plan view of the surface; the amplitude at each (x, y) point is represented by a shade of gray. The gray shading is created by overprinting with the line printer using three line printer strikes for each finished line in the plot. The cost in turn-around time using this routine is modest as it can use either of the NRL CDC-3800 systems and it avoids the queue at the CalComp plotter. There is logic to handle arrays that are too large to or do not reside in memory at one time by doing the plotting piecemeal. The plotting is ordinarily accomplished with minimal user involvement, however the user can control the nature of the plot. The array of $f(x,y)$ values plotted is left undisturbed by SURFPLØT.

The plot returned by SURFPLØT is the same in principle as shading each bounded region in a contour plot with a distinct shade of gray that corresponds to the levels that have defined the region. There are normally ten distinguishable shades of gray (including unmarked paper) that have been created with the line printer character sets. This number is not ironclad and is subject to vagaries of paper and ribbon. However, the gray shades generally are satisfactory using the full ten shades of gray during daytime runs at NRL with either system.

The operation of SURFPLØT is ordinarily as follows:

- (1) The array of $f(x,y)$ values is searched for the maximum and minimum values.

- (2) The range from maximum to minimum is divided by ten partitioning values using a linear scale (see Figure 1).

- (3) The array of $f(x,y)$ values is then searched point-by-point generating a quantized representation of each row, a row at a time, using the partitioning from step (2).

- (4) The points are printed, a row at a time, using the shade of gray unique to the quantization at each (x,y) point.

The physical plot of the array of $f(x,y)$ values is printed so that the columns of the array are parallel to the sprocket holes in the line printer paper. The rows are printed across the page. The line printer has a resolution of ten characters per inch across the page and six lines per inch along the sprocket holes. Hence, the aspect ratio is not one. There is no limitation in the number of rows (equivalent to printer lines) other than the one provided by the line count on the RUN card. However, no more than 128 columns can be printed on any one printer page. For arrays that have more than 128 columns, the plot is printed out in blocks of 100

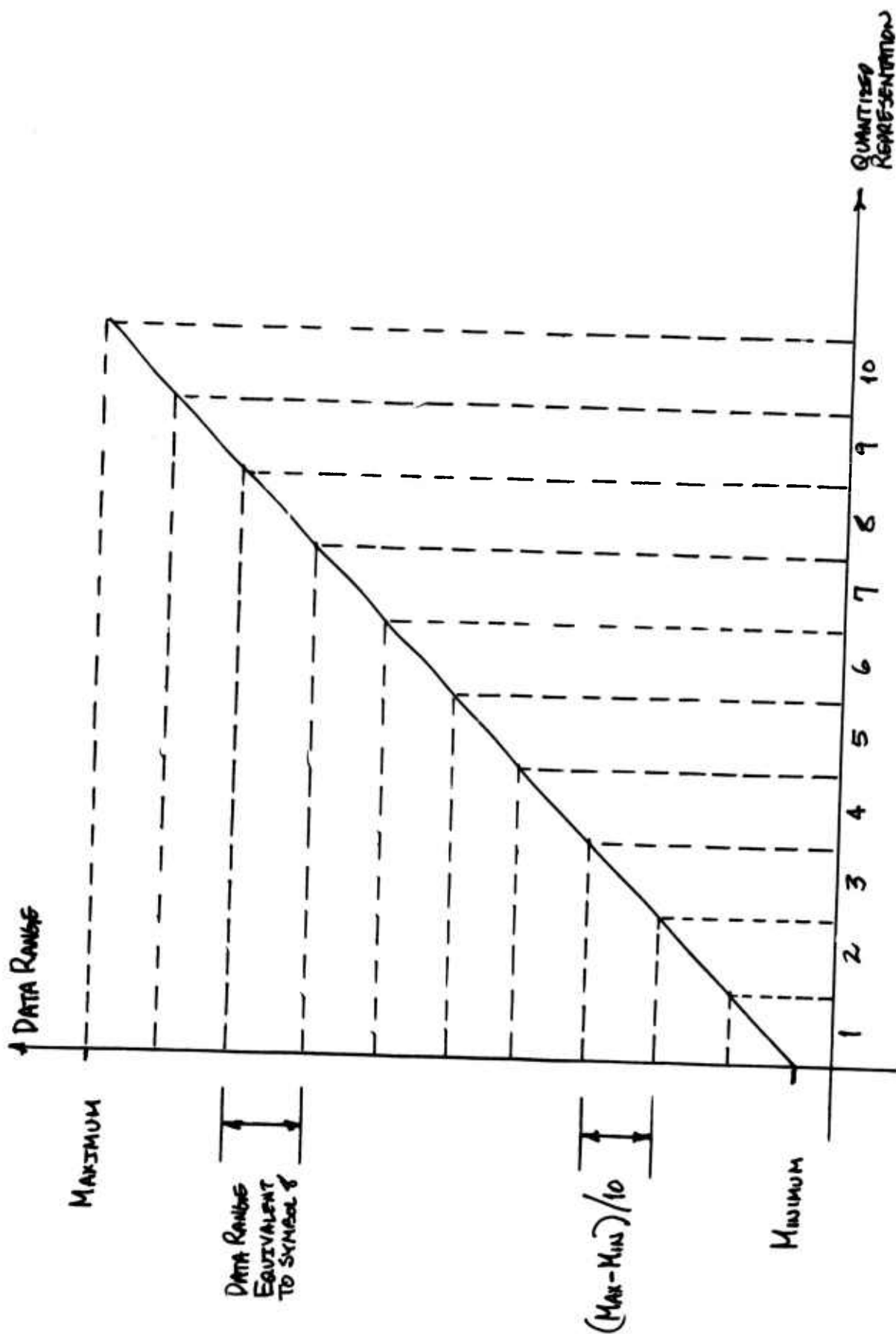


Fig. 1 - Partitioning of Data Range with a Linear Scale

columns per page (the routine automatically accomplishes this step) with each page annotated (see Section 3.8).

There are three functional pieces to SURFPLØT:

(1) generation of the quantization levels used to represent $f(x,y)$ and the equivalent shade of gray,

(2) generation of the annotation, and

(3) generation of the line printer image of the array of $f(x,y)$ values with the notation into line printer statements and the printing.

The user can be involved heavily, partially, or not at all in the first area and thereby exercise control over the nature of the plot. The number of shades of gray and the type of scale are just two of the variables that can be controlled, if desired. Only when the plot is done piecemeal is the user involved in (3) and then only in the most minimal fashion.

2.2 Problem Background

Most of the available routines for plotting in the RCC library are for functions of one variable. Representative routines that can plot an array of $f(x,y)$ values include the CalComp THREE-D Plotting Routine, J5-NRL-PLØTISØM, J5-NRL-SURFACE, and J5-ØRNL-CØNTØUR. These four routines can give an excellent representation of the function surface, however they require the use of a CalComp plotter. SURFPLØT is a simple plotting routine that returns a usable image of the function surface with minimum cost in turn-around time and minimal user involvement. For situations where the point by point generation of an image of $f(x,y)$ is useful, more flexible control of the image generation is available.

3.0 USAGE

3.1 Calling Sequence or Operational Procedure

```
CALL SURFPLØT (ZDATA, IXLENGTH, JYLENGTH,  
ITITLE, ISCALE, ZSCALE, ZLEVEL, ZPARAMTR,  
IEXTEND, ISEARCH)
```

The calling sequence used is the same form whatever the level of control being exercised by the user. The number of parameters that need be specified, however, vary depending on (1) a single or multiple call to SURFPLØT and (2) the degree of control exercised by the user over the nature of the plot. The user always names the array of $f(x,y)$ values, its dimensions and sets up a title. All that need be done thereafter is to dimension properly the remaining arrays and a linearly scaled plot will result. To exercise control over the plot, the parameters used are ISCALE, ZSCALE, and occasionally ZLEVEL and ZPARAMTR. To plot piecemeal or with several calls one appropriately sets ZPARAMTR, IEXTEND, AND ISEARCH.

3.2 Arguments, Parameters, and/or Initial Conditions

The user must dimension all arrays in the calling program.

ZDATA - the name of the array containing the $f(x,y)$ values, which can be either fixed or floating point. The array is left undisturbed by SURFPLØT.
DIMENSION ZDATA (IXLENGTH, JYLENGTH)
IXLENGTH - the number of rows in ZDATA
JYLENGTH - the number of columns in ZDATA
ITITLE - the name of the array containing the plot title, one card of BCD characters.
DIMENSION ITITLE (10). See Section 7.0.

While the above four parameters must be specified in each call, the remaining parameters are optional. The next group: ISCALE, ZSCALE, and occasionally ZLEVEL and/or ZPARAMTR, are used to control the plotting.

ISCALE - the name of an array of four numbers that selects the number of shades of gray, the type of scale, a "positive" or "negative", and indicates whether the elements of ZDATA are fixed or floating point numbers. DIMENSION ISCALE(4).

ISCALE(1) = { 0 - ten shades of gray
(as shown in the sample plot in Section 7.0.)
1-25 ISCALE(1) shades of gray - if more than ten, the user must supply the symbol set using three data cards. These cards are read in under a 25(R1) format. There is an interaction with ISCALE(3); when zero the first symbol on the card will represent the smallest amplitude and when non-zero the largest. See Section 3.7.

ISCALE(2) = { 0 - Linear Scale
1 - Logarithmic Scale
2 - Logarithm of sine to the pth* power
3 - sine to the pth* power
4 - user supplied; the quantizing levels must be in ZLEVEL

ISCALE(3) = { 0 - increasing amplitude printed as increasing darkness - a "negative". Probably the most useable form.
1 - decreasing amplitude printed as increasing darkness - a "positive".

*If ISCALE(2) = 2, p = 2.5
*If ISCALE(2) = 3, p = 0.25

$$\text{ISCALE}(4) = \begin{cases} 0 & \text{the elements of the array} \\ & \text{ZDATA are floating point} \\ & \text{numbers.} \\ 1 & \text{the elements of the array} \\ & \text{ZDATA are fixed point} \\ & \text{numbers.} \end{cases}$$

ZSCALE - the name of an array of three numbers used to generate the quantizing levels in ZLEVEL. A full explanation of the meanings of ZSCALE is deferred to Section 4.0 where the algorithms for generating ZLEVEL are explained. Along with ISCALE, ZSCALE can be used to exercise complete control over the plotting. By leaving ZSCALE(.) = 0.00, the present parameters will be selected. To use ZSCALE(.) with other than nominal values, see Tables 1 & 2, Section 4. DIMENSION ZSCALE(3).

ZLEVEL - the name of an array of quantizing values generated by SURFPLØT. If ISCALE(2) ≠ 4, the user need only dimension ZLEVEL. If ISCALE(2) = 4, the user supplies the quantizing levels in ZLEVEL (see Section 4.) DIMENSION ZLEVEL(10) or ZLEVEL(ISCALE(1)) if more than ten shades of gray are used.

The last group of parameters is used when the plot is made with several successive calls to SURFPLØT.

ZPARAMTR - the name of an array containing:

ZPARAMTR(1) = maximum value of $f(x,y)$
 ZPARAMTR(2) = value of x for $f_{\max}(x,y)$
 ZPARAMTR(3) = value of y for $f_{\max}(x,y)$
 ZPARAMTR(4) = minimum value of $f(x,y)$
 ZPARAMTR(5) = value of x for $f_{\min}(x,y)$
 ZPARAMTR(6) = value of y for $f_{\min}(x,y)$

In the multiple call situation, to have the same scale for all portions of the plot, the scale routines must be given the maximum and minimum values of $f(x,y)$ at the start. This is accomplished by filling ZPARAMTR(.) with the appropriate values. If left zero, the first portion of the array that is plotted will be searched and the scales from this first portion

will be used for the remaining portions. In the single call case, ZPARAMTR returns the maximum and minimum parameters of the plotted array. DIMENSION ZPARAMTR(6).

IEXTEND - the name of an array containing multiple call indicators; IEXTEND(1) denotes that the extensions will be in the x-direction (parallel to the sprocket holes), IEXTEND(2) for y-direction extensions. Either one or both extensions can be called. DIMENSION IEXTEND(2). The following code is used.

$$\text{IEXTEND} = \begin{cases} 0 & \text{- only one call in this direction} \\ 1 & \text{- first of successive calls in this direction} \\ 2 & \text{- neither first nor last call} \\ 3 & \text{- last of successive calls} \end{cases}$$

ISEARCH - a variable used in multiple call situations to force the scales to be recomputed for each separate call in the sequence. This means that the scales change (presumably) for each call. The only reason for including this option is that the maximum and minimum values of $f(x,y)$ can be found for the extended array from the maximum and minimum values returned in ZPARAMTR(.) for each section of $f(x,y)$. A plot of each section also is printed.

$$\text{ISEARCH} = \begin{cases} 0 & \text{- scales are not recomputed} \\ 1 & \text{- scales are recomputed} \end{cases}$$

3.3 Space Required (Decimal and Octal)

3.3.1 Unique Storage

6100 octal (3136 decimal) locations

3.3.2 Common Blocks

None

3.3.3 Temporary Storage

None

3.4 Messages and Instructions to the Operator

None

3.5 Error Returns, Messages, and Codes

None

3.6 Informative Messages to the User

None

3.7 Input

In addition to a card containing the title, input is required when the user supplies the symbol set to define the shades of gray. Up to 25 shades can be defined. Each shade is made up of three characters from the line printer set as read in from three data cards punched in a 25(R1) format. In the example below the first gray level is the character A overprinted with a B and C.

CARD 3
C F I ...
CARD 2
B E H ...
CARD 1
A D G ...

3.8 Output

A typical output is shown in Section 7. In addition to the line printer plot, the following information is printed:

(N,M) SURFACE PLOT.....Page N1 of N2
The (N,M) refers to multiple call situations and is used to annotate each portion of the overall plot as an element in a matrix. N is the row index and M is the column index. For the single call case, $N = M = 1$.

The page numbering accounts for arrays with more than 128 columns; then the plot will be made 100 columns per page with the N1 and N2 annotating this string.

ITITLE - the user supplied title.

DATA MAXIMUM.....the maximum value of $f(x,y)$ and
its (x,y) location in the array
(the first occurrence of $f_{\max}(x,y)$)

DATA MINIMUM.....the minimum value of $f(x,y)$ and
its (x,y) location in the array
(the first occurrence of $f_{\min}(x,y)$)

DATA RANGE.....if linear plot, $(f_{\max} - f_{\min})$; if
logarithmic, $C \cdot \log_{10}(f_{\max}/f_{\min})$
where C is equal to 10.00 unless
otherwise specified by the user.
See Tables 1 & 2, Section 4.

SCALE MAXIMUM.....maximum value of the quantizing
levels

SCALE MINIMUM.....minimum value of the quantizing
levels

SCALE RANGE.....if linear, $(\text{Scale}_{\max} - \text{Scale}_{\min})$
quantizing levels; if log,
 $C \cdot \log_{10}(\text{Scale}_{\max}/\text{Scale}_{\min})$,
 C defined as above.

PLOT REFERENCE.....if linear, $(f_{\max} - f_{\min})$ unless
specified by the user; if log,
either $f_{\max}$, $f_{\max}/2.00$, or user
supplied. See Tables 1 & 2,
Section 4.0.

PLOT BIAS.....for linear scales, nominally
 $f_{\min}(x,y)$ unless otherwise
specified by the user; if log
scale, it will be 0.00.

PLOT EXTENT.....the dimensions of the array of
 $f(x,y)$ values (IXLENGTH, JYLENGTH)

DB REFERENCE.....the "C" used to compute decibels.
If $\text{ZSCALE}(2) = 0$, DB REFERENCE =
10.00; if $\text{ZSCALE}(2) \neq 0$, DB REFERENCE =
 $\text{ZSCALE}(2)$.

HALF TONE DENSITY WEDGE AND SCALE

The equivalence between shades of gray and numerical values and the relation to the data. If linear, percentage of the maximum; if log, decibels relative to the maximum*. Five rows of the gray scale, each gray shade repeated for ten points, are printed. If more than ten shades of gray have been specified the wedge is printed along the page rather than across the page.

The body of the plot follows. A border of **M** is printed around the periphery. Every fifth row and column are numbered, the maximum and minimum of the array of $f(x,y)$ values are labeled (only the first occurrence of each). The program will keep track of multiple call plots so that the numbering is consistent and the borders removed appropriately to ease the cutting and pasting.

After completion of the call, the array ZPARAMTR contains the maximum and minimum parameters of the array (see Section 3.2) and the array ZLEVEL contains the quantizing level values.

3.9 Formats

See Section 3.7.

3.10 External Routines and Symbols

None

3.11 Timing

The routine takes roughly 0.07 seconds per row of the array of $f(x,y)$ values.

*decibels are conventionally defined as $10 \log(\cdot) = \text{dB}$

3.12 Accuracy

Not applicable

3.13 Cautions to Users

The aspect ratio of the plot is not one so a distortion exists due to the stretching of the x-axis relative to the y-axis.

This imaging does not work well with small or thin arrays, for the technique depends on the eye perceiving structures in the data. If the gray shade does not have enough spatial extent, the eye does not perceive a pattern.

3.14 Program Deck Structure

7JØB
9

7FTN
9

user program (contains call to SURFPLØT)
SUBROUTINE SURFPLØT
SCOPE

7LØAD
9

7RUN
9

data card 1 - title card
data card 2 } plotting symbols if more than
data card 3 } ten shades of gray are
data card 4 } requested.
end-of-file card

3.15 References - Literature - Appendices

- [1] D. E. Knuth, The Art of Computer Programming,
Volume 2 - Seminumerical Algorithms,
Addison - Wesley, 1969.

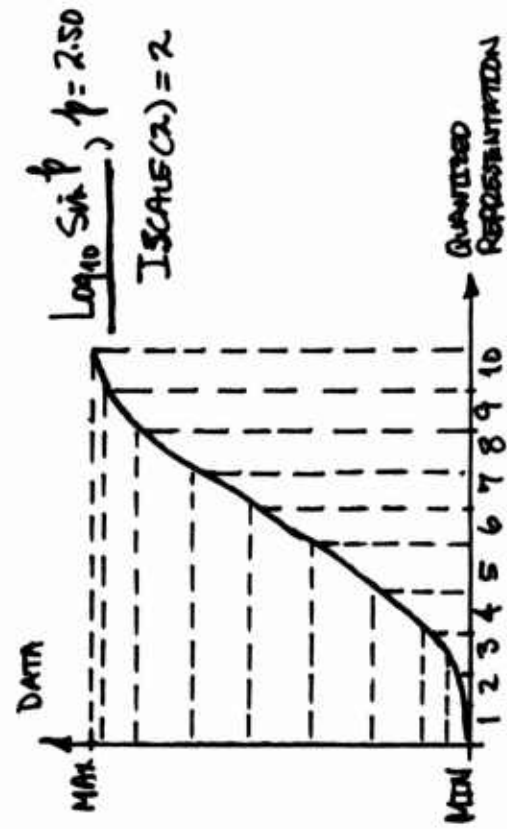
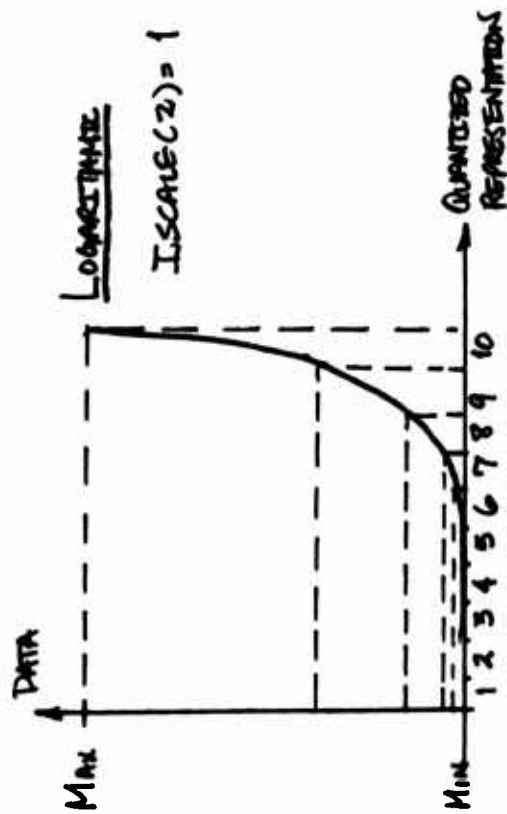
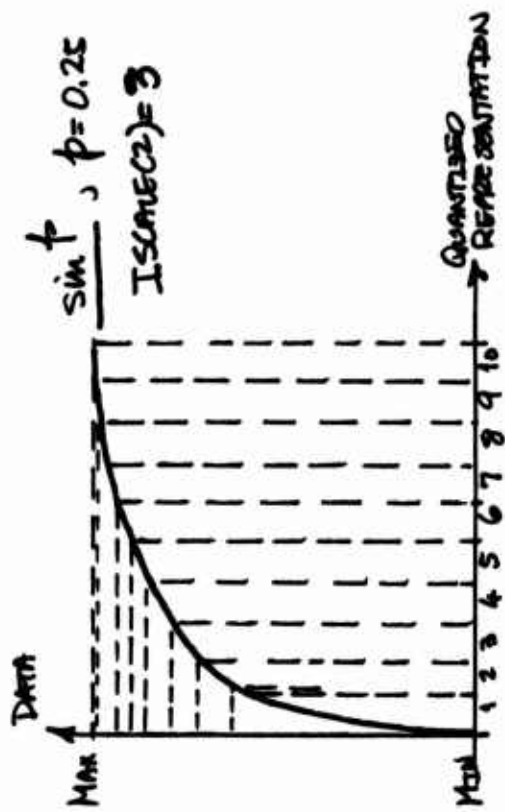
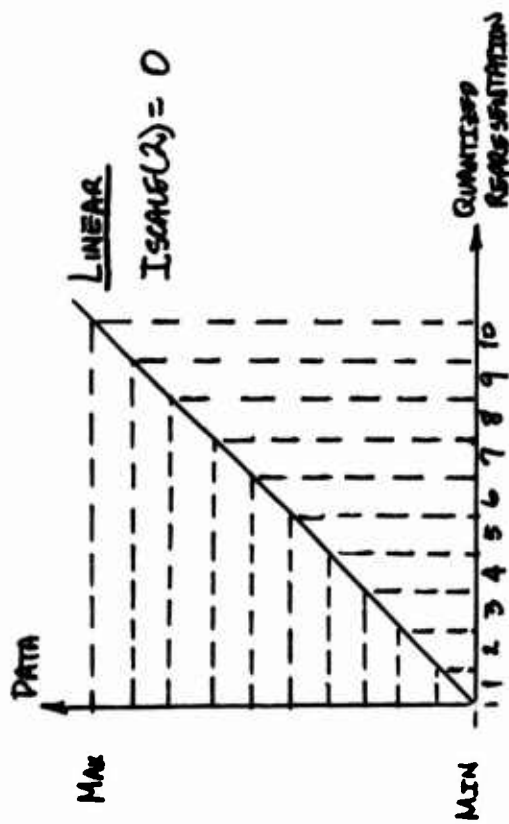


Fig. 2 - Rough Transfer Characteristics for Data Scales 0 through 3

4.0 METHOD OR ALGORITHM

Representing a function of two variables on a flat piece of paper so that its structure is obvious to the eye requires some trickery. The surface can be suggested by encoding the amplitude of each (x,y) point into a pseudo-height interpretable by the eye. Some standard methods are:

- (a) mapping amplitude into brightness as in a photograph,
- (b) mapping constant amplitude contours as in the topographic map,
- (c) using isometric distortions (or projections) by mapping amplitude for each row (or column) into y-axis deflection then offsetting each succeeding row a fixed amount in the x- and y-directions, and
- (d) mapping regions of constant amplitude or amplitude bands into colors or shades of gray as in a color relief map.

SURFPLØT follows method (d). The data $f(x,y)$ is partitioned into bands, each band is assigned a unique shade of gray, and the array of $f(x,y)$ values is printed point by point following this code.

The technique of selecting the amplitude bands is not straightforward. In Figure 2 four different partitionings of the data range are displayed. The attraction of non-linear scales is simply that it allows the resolution of the eye, represented by the ten levels along the abscissa, to be placed over the significant detail of the data. A classical example is a monochromatic image; the array of $f(x,y)$ values representing the image often has considerable dynamic range. However, on closer examination, the distribution of amplitudes often turns out to be confined to a small portion of the dynamic range not far from the minimum value with a few very large specular returns. The information content is in this small portion of the dynamic range near the minimum; hence a logarithmic scale would be an appropriate technique to use for partitioning the dynamic range, (see Figure 2).

In constructing a partition of the data there are three features available to the user for control:

- (1) the number of bands,
- (2) the shape of the partitioning curve, and
- (3) the location of the partitioning curve.

Examples of (2) are shown in Figure 2. An example of (3) for linear scales is shown in Figure 3. In SURFPLØT, the first two features are specified by ISCALE (1) and ISCALE (2) respectively. To specify the third feature, the program uses ISCALE(1), ZSCALE(1), ZSCALE(2) and ZSCALE(3). The partitioning values are located in array ZLEVEL. Table 1 shows how each scale is computed; Table 2 shows the specification of the third feature. The following pattern is used; if ZSCALE(·) is left zero, a preset value is used. If non-zero, that value of ZSCALE(·) is used. In the case of ZSCALE(1), a third option is provided. If ZSCALE(1) = -1.00, then half the preset value is used. This is most useful in logarithmic plots as it slides the scale down 3dB below the maximum.

The plotting consists of printing one of a series of half-tone-like shades of gray at each point corresponding to the $f(x,y)$ value at that point. The shade of gray is determined by finding the partitioning level that just equals or exceeds the $f(x,y)$ value. Assigned to each partitioning level is a unique shade of gray created by overprinting with three line printer characters. The array is examined row by row; printing is also accomplished row by row. The data array is unaltered.

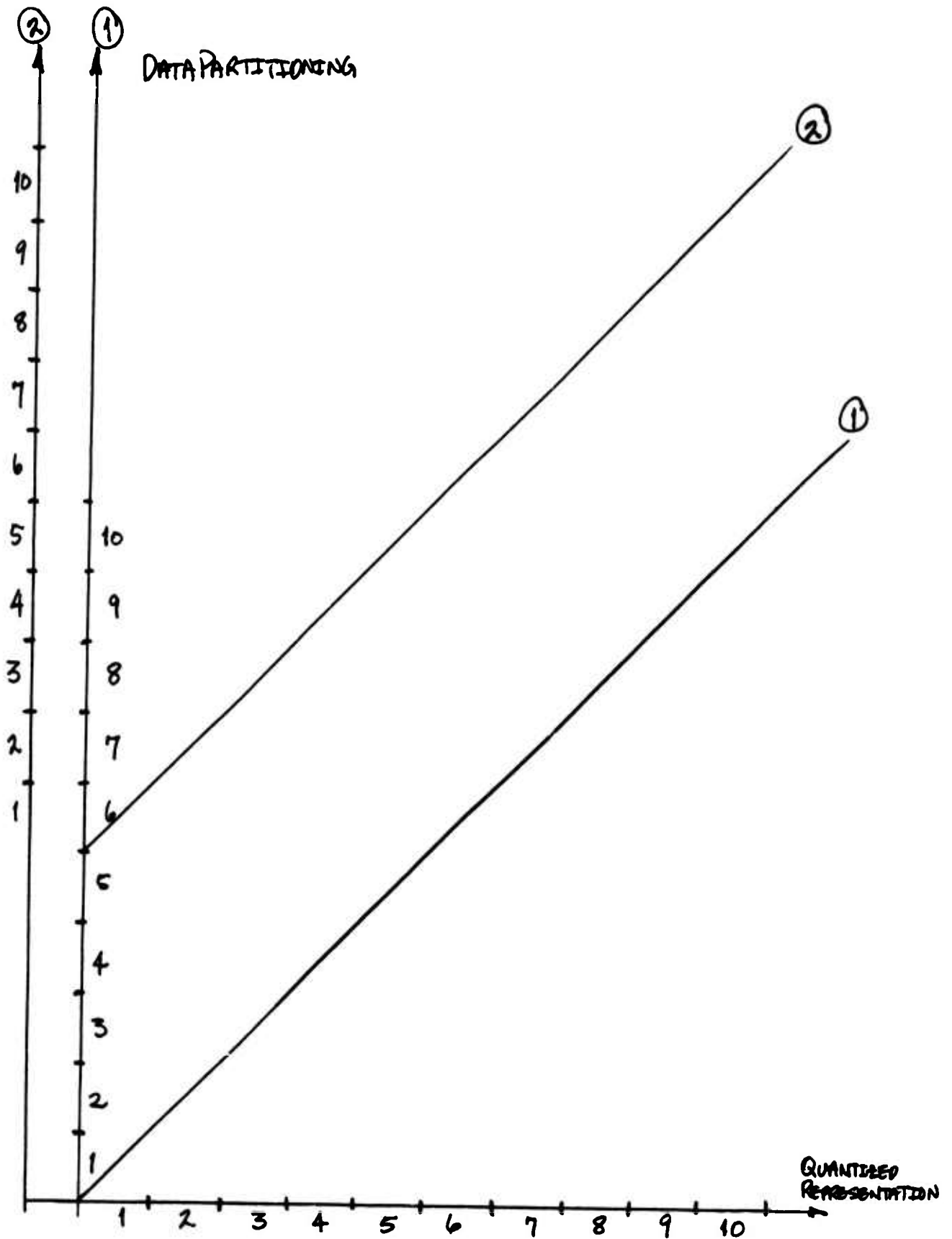


Fig. 3 - Linear Scales Relocated

TABLE 1 - SCALE ALGORITHMS

SCALE TYPE	$ZLEVEL(L) =$	$ZLEVEL(L) / ZMAX$ OR $DBSCALE * \log_{10}(ZLEVEL(L) / ZMAX)$	SCALE RANGE
LINEAR $ISCALE(2)=0$	$ZBIAS + \frac{ZREF}{LEVELS} * L$	$\frac{ZBIAS + \frac{ZREF}{ZMAX} * \frac{L}{LEVELS}}{ZMAX}$	$ZREF * \frac{LEVELS-1}{LEVELS}$
LOGARITHMIC $ISCALE(2)=1$	$ZREF * 10.00 * \left\{ \left(1 - \frac{L}{LEVELS} \right) \right\}$	$DBSCALE * \log_{10} \frac{ZREF}{ZMAX} - DBSCALE * SLOPE * \left(1 - \frac{L}{LEVELS} \right)$ (INDB)	$DBSCALE * SLOPE * \frac{LEVELS-1}{LEVELS}$ (INDB)
LOG-SIN ² $ISCALE(2)=2$	$ZREF * \left\{ \sin^2 \left(\frac{\pi}{2} \frac{L}{LEVELS} \right) \right\} * POWER$	$DBSCALE * \log_{10} \frac{ZREF}{ZMAX} - DBSCALE * POWER * \log_{10} \left(\sin^2 \frac{\pi}{2} \frac{L}{LEVELS} \right)$	$-DBSCALE * POWER * \log_{10} \left(\sin^2 \frac{\pi}{2} \frac{1}{LEVELS} \right)$
SIN ⁴ $ISCALE(2)=3$	$ZBIAS + ZREF * \left\{ \sin^4 \left(\frac{\pi}{2} \frac{L}{LEVELS} \right) \right\} * POWER$	$\frac{ZBIAS + \frac{ZREF}{ZMAX} * \left\{ \sin^4 \left(\frac{\pi}{2} \frac{L}{LEVELS} \right) \right\} * POWER}{ZMAX}$	$ZREF * \left\{ 1 - \left[\sin^4 \frac{\pi}{2} \frac{1}{LEVELS} \right] * POWER \right\}$
USER DEFINED $ISCALE(2)=4$	$ZLEVEL(L)$ defined in calling program	$\frac{ZLEVEL(L)}{ZMAX}$	$ZLEVEL(LEVELS) - ZLEVEL(1)$

TABLE 2 - DEFINITION OF TERMS

SCALE TYPE	REFERENCE VALUE (ZREF) ZSCALE(1) = ZREF =	BINS OR DB CONSTANT ZSCALE(2) =	POWER OR SLOPE ZSCALE(3) =
LINEAR ISCALE(2) = 0	0.00	ZBINS = ZMIN	0.00
	-1.00 NEITHER OF ABOVE	$\frac{(Z_{MAX} - Z_{MIN})}{2}$ ZSCALE(1)	NOT USED ± 0.00
LOGARITHMIC ISCALE(2) = 1	0.00	ZMAX	0.00
	-1.00 NEITHER	$\frac{Z_{MAX} - Z_{MIN}}{2.00}$ ZSCALE(1)	SLOPE = 40/DBSCALE = ZSCALE(3) ± 0.00
LOG _e -SIN P ISCALE(2) = 2	0.00	ZMAX	0.00
	-1.00 NEITHER	$\frac{Z_{MAX} - Z_{MIN}}{2.00}$ ZSCALE(1)	POWER = $2.5 - \frac{LEVELS - 10}{40}$ = ZSCALE(3) ± 0.00
SIN P ISCALE(2) = 3	0.00	ZMAX - ZMIN	0.00
	-1.00 NEITHER	$\frac{(Z_{MAX} - Z_{MIN})}{2.00}$ ZSCALE(1)	POWER = $2.5 - \frac{LEVELS - 10}{40}$ = ZSCALE(3) ± 0.00
USER DEFINED ISCALE(2) = 4	ZREF = ZSCALE(1)	NOT USED	NOT USED

5.0 SOURCE LANGUAGE LISTING

```

J5 NRL SURFLOT LINE PRINTER PLOTS OF TWO-DIMENSIONAL SURFACES
  SUBROUTINE SURFLOT(ZDATA,IXLENGTH,JYLENGTH,ITITLE,ISCALE,ZSCALE, 00000100
  1 ZLEVEL,ZPARAMTR,IEXTEND,ISARCH) 00000200
                                     00000300
C IDENT NUMBER = J5008000 00000400
C TITLE = LINE PRINTER PLOTS OF TWO DIMENSIONAL SURFACES 00000500
C IDENT NAME = J5-NRI-SURFLOT 00000600
C LANGUAGE = FORTRAN 00000700
C COMPUTER = CDC-3800 00000800
C CONTRIBUTOR = MICHAEL A. TAMNY, CODE 5365T, AIRBORNE RADAR BRANCH, 00000900
C RADAR DIVISION 00001000
C ORGANIZATION = NRL - NAVAL RESEARCH LABORATORY - WASHINGTON, D.C. 00001100
C 20375 00001200
C DATE = 31 OCTOBER 1973 00001300
C PURPOSE = TO PLOT AN ARRAY OF F(X,Y) VALUES USING THE LINE PRINTER 00001400
C BY REPRESENTING AMPLITUDE WITH A SHADE OF GRAY 00001500
                                     00001600
                                     00001700
                                     00001800
DIMENSION ZDATA(1) 00001900
DIMENSION ZPARAMTR(1) 00002000
DIMENSION IEXTEND(1) 00002100
DIMENSION ISCALE(1) 00002200
DIMENSION ZSCALE(1) 00002300
DIMENSION ZLEVEL(1) 00002400
DIMENSION ITITLE(1) 00002500
DIMENSION LINEPLOT(135,3) 00002600
DIMENSION LINE1(135), LINE2(135), LINE3(135) 00002700
DIMENSION JLINE(17,3) 00002800
DIMENSION INFSYM(25,3) 00002900
                                     00003000
EQUIVALENCE ( LINEPLOT( 1),LINE1(1) ) 00003100
EQUIVALENCE ( LINEPLOT(136),LINE2(1) ) 00003200
EQUIVALENCE ( LINEPLOT(271),LINE3(1) ) 00003300
EQUIVALENCE ( LINEPLOT( 1),JLINE(1) ) 00003400
EQUIVALENCE ( ZTEST,ITEST ) 00003500
                                     00003600
TYPE INTEGER BLNKFILL, DRFILL 00003700
TYPE INTEGER BLANK, DOT, PLUS 00003800
TYPE INTEGER AFILL, IFILL, MFILL, NFILL, UFILL, WFILL, XFILL 00003900
                                     00004000
DATA (BLNKFILL = 2R ), (DRFILL = 2RDR) 00004100
DATA (BLANK = 1R ), (DOT = 1R.), (PLUS = 1R+) 00004200
DATA (AFILL = 1RA), (IFILL = 1RI), (MFILL = 1RM), (NFILL = 1RN) 00004300
DATA (WFILL = 1RW), (XFILL = 1RX) 00004400
DATA (NX = 1), (NY = 1) 00004500
DATA((I INFSYM(I,1),I=1,10)=1R,1R-,1R+,1R*,1RX,2(1R0),1RU,1R0,1RD) 00004600
DATA((I INFSYM(I,2),I=1,10)=6(1P ),1RX,1RA,1P+,1RB) 00004700
DATA((I INFSYM(I,3),I=1,10)=8(1P ),1RU,1RX) 00004800
DATA (ISHIFT = 1000000000000000R) 00004900

```

DATA (ISHIFT2 = 1000000000000R)	00005000
DATA (ISHIFT3 = 100000000000R)	00005100
DATA (ISHIFT4 = 1000000000R)	00005200
DATA (ISHIFT5 = 10000000R)	00005300
DATA (ISHIFT6 = 10000R)	00005400
DATA (ISHIFT7 = 100R)	00005500
	00005600
	00005700
	00005800
GO TO (1005,1008) NX + NY	00005900
	00006000
	00006100
C THIS TRAP IS USED TO RECOGNIZE THE TYPE OF CYCLE IN EXTENDED	00006110
C CALLS IN BOTH DIMENSIONS. THE FIRST CALL OF THIS TYPE CAUSES	00006120
C NX = NY = 0. OTHERWISE THEY ARE INDICES USED TO KEEP TRACK OF	00006130
C THE ARRAY BLOCKS SENT TO SURFLOT FOR PRINTING AND AS SUCH ARE	00006140
C GREATER THAN OR EQUAL TO 1.	00006150
	00006200
	00006300
1005 IF(IEXTEND(1)-IEXTEND(2) .EQ. 1) GO TO 1006	00006400
NX = 1	00006500
NY = 2	00006600
ILAST = 0	00006700
IOXY = 2	00006800
GO TO 1008	00006900
1006 NY = 1	00007000
NX = 2	00007100
JLAST = 0	00007200
IOXY = 1	00007300
	00007400
C THE ARRAY IS SEARCHED FOR THE MAXIMUM AND MINIMUM VALUES AND	00007410
C THE SCALES COMPUTED IN THE FOLLOWING MANNER..(1) IN SINGLE CALL	00007420
C CASES..(2) IF ISEARCH = 1..(3) IF THE FIRST CALL OF AN EXTENDED	00007430
C CALL AND THE MAXIMUM AND MINIMUM VALUES HAVE NOT BEEN PASSED IN	00007440
C ZPARAMTR. IN THE SEARCH, LOCATIONS ARE NUMBERED IN THE EXTENDED	00007450
C ARRAY BY RIASING WITH ILAST AND JLAST. IF A SEARCH IS MADE THE	00007460
C LEVELS MUST BE COMPUTED.	00007470
	00007500
1008 IF(IEXTEND(1)+IEXTEND(2) .EQ. 0) GO TO 1009	00007600
IF(ISEARCH .EQ. 1) GO TO 1009	00007700
IF(IEXTEND(1) .GT. 1 .OR. IEXTEND(2) .GT. 1) GO TO 1075	00007800
IF(ZPARAMTR(1) .EQ. 0.00 .AND. ZPARAMTR(4) .EQ. 0.00) GO TO 1009	00007900
GO TO 1020	00008000
	00008100
	00008200
1009 ZTEST = ZDATA(1)	00008300
IF(ISCALE(4) .EQ. 0) GO TO 1010	00008400
ZTEST = ITEST	00008500
1010 Z MAX = Z MIN = ZTEST	00008600
MAX X = MIN X = 1 + ILAST	00008700

MAX Y = MIN Y = 1 + JLAST	00008800
DO 1015 J = 1,JYLENGTH	00008900
INDEX = (J-1) * IXLLENGTH	00009000
DO 1015 I = 1,IXLENGTH	00009100
ZTEST = ZDATA(INDEX+1)	00009200
IF(ISCALE(4) .EQ. 0) GO TO 1011	00009300
ZTEST = ITEST	00009400
1011 IF(ZTEST .LE. Z MAX) GO TO 1012	00009500
Z MAX = ZTEST	00009600
MAX X = I + ILAST	00009700
MAX Y = J + JLAST	00009800
GO TO 1015	00009900
1012 IF(ZTEST .GE. Z MIN) GO TO 1015	00010000
Z MIN = ZTEST	00010100
MIN X = I + ILAST	00010200
MIN Y = J + JLAST	00010300
1015 CONTINUE	00010400
ZPARAMTR(1) = Z MAX	00010500
ZPARAMTR(2) = MAX X	00010600
ZPARAMTR(3) = MAX Y	00010700
ZPARAMTR(4) = Z MIN	00010800
ZPARAMTR(5) = MIN X	00010900
ZPARAMTR(6) = MIN Y	00011000
IF(IFXTEND(1) .GT. 1 .OR. IFXTEND(2) .GT. 1) GO TO 1050	00011100
GO TO 1030	00011200
1020 Z MAX = ZPARAMTR(1)	00011300
MAX X = ZPARAMTR(2)	00011400
MAX Y = ZPARAMTR(3)	00011500
Z MIN = ZPARAMTR(4)	00011600
MIN X = ZPARAMTR(5)	00011700
MIN Y = ZPARAMTR(6)	00011800
	00011900
C THIS SECTION SETS THE NUMBER OF GRAY SHADES, AND IF LESS THAN	00011910
C TEN, SELECTS ONES FURTHER APART USING LSTEP AND LBIAS.	00011920
	00012000
1030 IF(ISCALE(1) .GT. 10) GO TO 1040	00012100
IF(ISCALE(1) .EQ. 0) LLEVELS = 10	00012200
IF(ISCALE(1) .NE. 0) LLEVELS = ISCALE(1)	00012300
LSTEP = 10/LLLEVELS	00012400
LBIAS = 10 - LSTEP*LLLEVELS	00012500
GO TO 1050	00012600
1040 LLEVELS = ISCALE(1)	00012700
LSTEP = 1	00012800
LBIAS = 0	00012900
READ 91041, (LINESYM(I,1), L = 1,LLLEVELS)	00013000
READ 91041, (LINESYM(I,2), L = 1,LLLEVELS)	00013100
READ 91041, (LINESYM(I,3), L = 1,LLLEVELS)	00013200
91041 FORMAT(25(R1))	00013300
	00013400
C THIS SECTION COMPUTES THE PARTITIONING LEVELS, ZLEVEL	00013410

1050 GO TO (1051,1055,1060,1065,1070) ISCALE(2) + 1	00013500
	00013600
1051 ITYPE = BLNKFILL	00013700
IF(ZSCALE(1) .EQ. 0.00) Z RFF = (ZMAX-ZMIN)	00013800
IF(ZSCALE(1) .EQ.-1.00) Z RFF = (ZMAX-ZMIN)/2.00	00013900
IF(ZSCALE(1) .NE. 0.00 .AND. ZSCALE(1) .NE.-1.00)	00014000
1 Z RFF = ZSCALE(1)	00014100
IF(ZSCALE(2) .EQ. 0.00) Z RIAS = Z MIN	00014200
IF(ZSCALE(2) .NE. 0.00) Z RIAS = ZSCALE(2)	00014300
DRSCALE = 0.00	00014400
STEP = ZRFF/LLFVFLS	00014500
DO 1052 L = 1,LLFVFLS	00014600
ZLEVEL(L) = Z RIAS + STEP*L	00014700
1052 CONTINUE	00014800
ZRANGE = Z MAX - Z MIN	00014900
SCALE = ZLEVEL(LLFVFLS) - ZLEVEL(1)	00015000
GO TO 1075	00015100
	00015200
	00015300
1055 ITYPE = DRFILL	00015400
IF(ZSCALE(1) .EQ. 0.00) Z RFF = Z MAX	00015500
IF(ZSCALE(1) .EQ.-1.00) Z RFF = ZMAX/2.00	00015600
IF(ZSCALE(1) .NE. 0.00 .AND. ZSCALE(1) .NE.-1.00)	00015700
1 Z RFF = ZSCALE(1)	00015800
IF(ZSCALE(2) .EQ. 0.00) DRSCALE = 10.00	00015900
IF(ZSCALE(2) .NE. 0.00) DRSCALE = ZSCALE(2)	00016000
IF(ZSCALE(3) .EQ. 0.00) SLOPE = 40.00/DRSCALE	00016100
IF(ZSCALE(3) .NE. 0.00) SLOPE = ZSCALE(3)	00016200
Z RIAS = 0.00	00016300
STEP = 1.00/LLFVFLS	00016400
DO 1056 L = 1,LLFVFLS	00016500
POWER = -SLOPE*(1.00-STEP*L)	00016600
ZLEVEL(L) = ZRFF*(10.00**POWER)	00016700
1056 CONTINUE	00016800
ZRANGE = DRSCALE*ALOG10(ZMAX/ZMIN)	00016900
SCALE = DRSCALE*ALOG10(ZLEVEL(LLFVFLS)/ZLEVEL(1))	00017000
GO TO 1075	00017100
	00017200
	00017300
1060 ITYPE = DRFILL	00017400
IF(ZSCALE(1) .EQ. 0.00) Z RFF = Z MAX	00017500
IF(ZSCALE(1) .EQ.-1.00) Z RFF = ZMAX/2.00	00017600
IF(ZSCALE(1) .NE. 0.00 .AND. ZSCALE(1) .NE.-1.00)	00017700
1 Z RFF = ZSCALE(1)	00017800
IF(ZSCALE(2) .EQ. 0.00) DRSCALE = 10.00	00017900
IF(ZSCALE(2) .NE. 0.00) DRSCALE = ZSCALE(2)	00018000
IF(ZSCALE(3) .EQ. 0.00) POWER = 2.50	00018100
IF(ZSCALE(3) .NE. 0.00) POWER = ZSCALE(3)	00018200
Z RIAS = 0.00	00018300
STEP = 1.00/LLFVFLS	00018400
PIOVER2 = 3.141592654/2.00	

RADIAN = P/OVER2*STFP	00018500
DO 1061 L = 1,LLEVELS	00018600
ARGUMENT = RADIAN*L	00018700
ZLEVEL(L) = ZREF*((SINF(ARGUMENT))**POWER)	00018800
1061 CONTINUE	00018900
SCALE = DRSCALE*ALOG10(ZLEVEL(LLEVELS)/ZLEVEL(1))	00019000
ZRANGE = DRSCALE*ALOG10(ZMAX/ZMIN)	00019100
GO TO 1075	00019200
	00019300
1065 ITYPE = BLNKFILL	00019400
IF(ZSCALE(1) .EQ. 0.00) Z REF = (ZMAX-ZMIN)	00019500
IF(ZSCALE(1) .EQ.-1.00) Z REF = (ZMAX-ZMIN)/2.00	00019600
IF(ZSCALE(1) .NE. 0.00 .AND. ZSCALE(1) .NE.-1.00)	00019700
1 Z REF = ZSCALE(1)	00019800
IF(ZSCALE(2) .EQ. 0.00) Z RIAS = Z MIN	00019900
IF(ZSCALE(2) .NE. 0.00) Z RIAS = ZSCALE(2)	00020000
IF(ZSCALE(3) .EQ. 0.00) POWER = 0.25	00020100
IF(ZSCALE(3) .NE. 0.00) POWER = ZSCALE(3)	00020200
DRSCALE = 0.00	00020300
STFP = 1.00/LLEVELS	00020400
P/OVER2 = 3.141592654/2.00	00020500
RADIAN = P/OVER2*STFP	00020600
DO 1066 L = 1,LLEVELS	00020700
ARGUMENT = RADIAN*L	00020800
ZLEVEL(L) = Z RIAS + ZREF*((SINF(ARGUMENT))**POWER)	00020900
1066 CONTINUE	00021000
SCALE = ZLEVEL(LLEVELS) - ZLEVEL(1)	00021100
ZRANGE = Z MAX - Z MIN	00021200
GO TO 1075	00021300
	00021400
1070 ITYPE = BLNKFILL	00021500
ZRANGE = Z MAX - Z MIN	00021600
SCALE = ZLEVEL(LLEVELS) - ZLEVEL(1)	00021700
DRSCALE = 0.00	00021800
Z RIAS = Z MIN	00021900
Z REF = Z MAX	00022000
	00022100
C THIS SECTION IS TESTING THE ARRAY WIDTH. THE LOGIC IS TO PRINT	00022110
C OUT AS MANY OF THE COLUMNS AS WILL FIT ON A COMPUTER PAGE AT ONE	00022120
C TIME. UP TO 128 COLUMNS CAN BE PRINTED ON ONE PAGE. IF MORE THAN	00022130
C 128 COLUMNS IN THE ARRAY, THEY ARE PRINTED IN BLOCKS OF 100.	00022140
C ALL THE ROWS ARE PRINTED IN EACH BLOCK.	00022150
	00022200
1075 JRIAS = JLAST	00022300
IF(JYLENGTH .GT. 128) GO TO 1080	00022400
NPAGES = 1	00022500
JLONG = JYLENGTH	00022600
IF(JYLENGTH .LT. 114) JSHIFT = 10	00022700
IF(JYLENGTH .GT. 113) JSHIFT = 0	00022800
GO TO 2000	00022900

1000	NPAGES = JYLENGTH/100	00023000
	IF(JYLENGTH .GT. NPAGES*100) NPAGES = NPAGES + 1	00023100
	JLONG = 100	00023200
	JSHIFT = 10	00023300
		00023400
		00023500
		00023600
2000	DO 2500 NPAGE = 1, NPAGES	00023700
		00023800
C	THIS IS THE PRINTING SECTION	00023810
C		00023820
C	THIS SECTION SELECTS THE PROPER BLOCK OF COLUMNS FOR PRINTING,	00023830
C	POSITIONS THE PLOTTING ON THE PAGE, AND SELECTS THE TYPE OF	00023840
C	ANNOTATION. A HEADER IS ALWAYS PRINTED FOR THE FIRST CALL AND	00023850
C	EXTENDED CALLS WITH RECOMPUTED SCALES.	00023860
		00023900
	JSTART = JRIAS + 1	00024000
	JEND = MINOF(JRIAS+JLONG, JLAST+JYLENGTH)	00024100
	IF(JSHIFT .EQ. 0) GO TO 2010	00024200
	IF(IEXTEND(2) .GT. 0) GO TO 2005	00024300
2001	IF(NPAGE .GT. 1) JLFFT = 0	00024400
	IF(NPAGE .EQ. 1) JLFFT = 2	00024500
	IF(NPAGE .NE. NPAGES) JRIGHT = 0	00024600
	IF(NPAGE .EQ. NPAGES) JRIGHT = 2	00024700
	GO TO 2020	00024800
2005	IF(ISCALE .EQ. 1) GO TO 2001	00024900
	IF(NPAGE .EQ. 1 .AND. IEXTEND(2) .EQ. 1) JLFFT = 2	00025000
	IF(NPAGE .GT. 1 .OR. IEXTEND(2) .GT. 1) JLFFT = 0	00025100
	IF(NPAGE .EQ. NPAGES .AND. IEXTEND(2) .EQ. 3) JRIGHT = 2	00025200
	IF(NPAGE .LT. NPAGES .OR. IEXTEND(2) .LT. 3) JRIGHT = 0	00025300
	GO TO 2020	00025400
2010	GO TO (2011, 2012, 2013) IEXTEND(2)	00025500
2011	JLFFT = 2	00025600
	JRIGHT = 1	00025700
	GO TO 2020	00025800
2012	IF(ZSCALE(1) .EQ. 0.00 .OR. ZSCALE(1) .EQ. -1.00) GO TO 2011	00025900
	JLFFT = 0	00026000
	JRIGHT = 0	00026100
	GO TO 2020	00026200
2013	JSHIFT = -5	00026300
	JLFFT = 1	00026400
	JRIGHT = 2	00026500
2020	IF(IEXTEND(1) .LT. 2) GO TO 2100	00026600
	IF(ZSCALE(1) .EQ. 0.00 .OR. ZSCALE(1) .EQ. -1.00) GO TO 2100	00026700
	IF(IEXTEND(2) .LT. 1 .AND. NPAGES .EQ. 1) GO TO 2200	00026800
		00026900
C	THIS SECTION GENERATES THE HEADER. NX AND NY ARE MATRIX INDEXING	00026910
C	OF THE ARRAY BLOCKS SENT SURPLOT IN EXTENDED CALLS. NPAGE AND	00026920
C	NPAGES KEEP TRACK OF THE BLOCKS OF COLUMNS USED IN PRINTING THE	00026930
C	LOCAL ARRAY. THE BLOCKS OF COLUMNS USED IN PRINTING THE	00026940

PRINT 92025, NX, NY, NPAGE, NPAGES	00027000
92025 FORMAT(1HR, *(*,I2,*,*,I2,*)*,3X,	00027100
1*SURFACE PLOT PAGE *,I2,* OF *,I2)	00027200
GO TO 2200	00027300
	00027400
	00027500
2100 PRINT 92100, NX, NY, NPAGE, NPAGES	00027600
92100 FORMAT(1HR, //,1H , *(*,I2,*,*,I2,*)*,3X,	00027700
1*SURFACE PLOT PAGE *,I2,* OF *,I2, //)	00027800
PRINT 92101, (1TITLE(I), L = 1,10)	00027900
92101 FORMAT(1H , 20X, 10(AR), //)	00028000
PRINT 92102,	00028100
1 Z MAX, MAX X, MAX Y,	00028200
2 Z MIN, MIN X, MIN Y,	00028300
3 ZRANGE, ITYPE,	00028400
4 ZLEVFL(1:LEVFLS),	00028500
5 ZLEVFL(1),	00028600
6 SCALE, ITYPE,	00028700
7 Z REF,	00028800
8 Z HTAS,	00028900
9 IXLFWGTH, JYLFNGTH,	00029000
4 DSCALE	00029100
92102 FORMAT(1H , 20X,	00029200
1*DATA MAXIMUM *,F11.4,1X,*(*,I4,*,*,I4,*)*, //,1H , 20X,	00029300
2*DATA MINIMUM *,F11.4,1X,*(*,I4,*,*,I4,*)*, //,1H , 20X,	00029400
3*DATA RANGE *,F11.4,R2, //,1H , 20X,	00029500
4*SCALE MAXIMUM *,F11.4, //,1H , 20X,	00029600
5*SCALE MINIMUM *,F11.4, //,1H , 20X,	00029700
6*SCALE RANGE *,F11.4,R2, //,1H , 20X,	00029800
7*PLOT REFERENCE *,F11.4, //,1H , 20X,	00029900
8*PLOT HTAS *,F11.4, //,1H , 20X,	00030000
9*PLOT EXTENT (*,I4,*,*,I4,*)*, //,1H , 20X,	00030100
4*DB REFERENCE *,F5.2, //,1H , 20X,	00030200
8*HALF TONE DENSITY WEDGE AND SCALE*)	00030300
	00030400
C THIS SECTION PRINTS THE DENSITY WEDGE OF GRAY SHADES AND THE	00030410
C EQUIVALENT NUMERICAL VALUES, BOTH IN ABSOLUTE TERMS AND	00030420
C RELATIVE TO THE REFERENCES,	00030430
	00030500
IF(LLEVFLS .GT. 10) GO TO 2125	00030600
IF(ITYPE .EQ. DBFILL) GO TO 2105	00030700
DO 2103 L = 1,LLEVFLS	00030800
ZLEVFL(L) = ZLEVFL(1)/7MAX	00030900
2103 CONTINUE	00031000
PRINT 92103, (ZLEVFL(I), L = 1,LLEVFLS)	00031100
92103 FORMAT(1H , 20X, 10(2X,F5.3,*MAX*))	00031200
DO 2104 L = 1,LLEVFLS	00031300
ZLEVFL(L) = ZLEVFL(L)*7MAX	00031400
2104 CONTINUE	00031500
GO TO 2110	00031600

2105 DO 2106 L = 1, LLEVELS	00031700
ZLEVEL(L) = DRSCALE*ALOG10(ZLEVEL(L)/ZMAX)	00031800
2106 CONTINUE	00031900
PRINT 92106, (ZLEVEL(L), L = 1, LLEVELS)	00032000
92106 FORMAT(1H, 20X, 10(1X, F7.2, *DR*))	00032100
DO 2107 L = 1, LLEVELS	00032200
ZLEVEL(L) = ZMAX*(10.00** (ZLEVEL(L)/DRSCALE))	00032300
2107 CONTINUE	00032400
2110 PRINT 92110, (ZLEVEL(L), L = 1, LLEVELS)	00032500
92110 FORMAT(1H, 20X, 10(1X, F9.2))	00032600
DO 2115 K = 1, 3	00032700
DO 2115 L = 1, LLEVELS	00032800
IF(ISCALE(3) .EQ. 0) LPRINT = LRTAS + LSTEP*I	00032900
IF(ISCALE(3) .NE. 0) LPRINT = LRTAS + LSTEP*(LLEVELS+1-L)	00033000
JEDGE1 = 21 + (L-1)*10	00033100
JEDGE2 = JEDGE1 + 9	00033200
DO 2115 J = JEDGE1, JEDGE2	00033300
LINEPLOT(J, K) = LINESYM(LPRINT, K)	00033400
2115 CONTINUE	00033500
DO 2116 J = 1, 20	00033600
LINE1(J) = LINE2(J) = LINE3(J) = BLANK	00033700
2116 CONTINUE	00033800
JEDGE1 = JEDGE2 + 1	00033900
DO 2117 J = JEDGE1, 135	00034000
LINE1(J) = LINE2(J) = LINE3(J) = BLANK	00034100
2117 CONTINUE	00034200
DO 2120 I = 1, 5	00034300
PRINT 92120, LINE1	00034400
PRINT 92120, LINE2	00034500
92120 FORMAT(1H, 135(R1))	00034600
PRINT 92121, LINE3	00034700
92121 FORMAT(1H, 135(R1))	00034800
2120 CONTINUE	00034900
GO TO 2150	00035000
2125 DO 2126 L = 1, LLEVELS	00035100
DO 2126 J = 1, 135	00035200
LINE1(J) = LINE2(J) = LINE3(J) = BLANK	00035300
2126 CONTINUE	00035400
IF(ITYPE .EQ. DRFTLL) GO TO 2128	00035500
ZLEVEL(L) = ZLEVEL(L)/ZMAX	00035600
PRINT 92127, ZLEVEL(L)	00035700
92127 FORMAT(1H, 21X, F6.3, *MAX*)	00035800
ZLEVEL(L) = ZLEVEL(L)*ZMAX	00035900
GO TO 2130	00036000
2128 ZLEVEL(L) = DRSCALE*ALOG10(ZLEVEL(L)/ZMAX)	00036100
PRINT 92128, ZLEVEL(L)	00036200
92128 FORMAT(1H, 21X, F7.2, *DR*)	00036300
ZLEVEL(L) = ZMAX*(10.00** (ZLEVEL(L)/DRSCALE))	00036400
2130 IF(ISCALE(3) .EQ. 0) LPRINT = LRTAS + LSTEP*I	00036500
IF(ISCALE(3) .NE. 0) LPRINT = LRTAS + LSTEP*(LLEVELS+1-L)	00036600

DO 2131 K = 1,3	00036700
DO 2131 J = 31,40	00036800
LINEPLOT(J,K) = LINESYM(LPRINT,K)	00036900
2131 CONTINUE	00037000
PRINT 92120, LINE1	00037100
PRINT 92120, LINE2	00037200
PRINT 92121, LINE3	00037300
PRINT 92132, ZLEVEL(L)	00037400
92132 FORMAT(1H*, 21X,E9.2)	00037500
PRINT 92120, LINE1	00037600
PRINT 92120, LINE2	00037700
PRINT 92121, LINE3	00037800
2135 CONTINUE	00037900
C THIS SECTION IS NUMBERING THE COLUMNS (IN THE EXTENDED ARRAY	00038000
C NUMEROLOGY).	00038010
2150 DO 2151 K = 1,3	00038020
DO 2151 J = 1,135	00038100
LINEPLOT(J,K) = BLANK	00038200
2151 CONTINUE	00038300
JEDGE1 = JSHIFT + 6	00038400
JEDGE2 = JSHIFT + 8 + (JEND-JSTART)	00038500
DO 2170 J = JSTART,JEND	00038600
JPRINT = JSHIFT + 7 + J - JSTART	00038700
IF(J .NE. MAX Y) GO TO 2155	00038800
LINE1(JPRINT) = MFILL	00038900
LINE2(JPRINT) = AFILL	00039000
LINE3(JPRINT) = XFILL	00039100
GO TO 2170	00039200
2155 IF(J .NE. MIN Y) GO TO 2160	00039300
LINE1(JPRINT) = MFILL	00039400
LINE2(JPRINT) = IFILL	00039500
LINE3(JPRINT) = NFILL	00039600
GO TO 2170	00039700
2160 INDEX = J - (J/5)*5	00039800
IF(INDEX .NE. 0) GO TO 2170	00039900
J1000 = (J/1000)	00040000
J100 = (J-J1000*1000)/100	00040100
J10 = (J-J1000*1000-J100*100)/10	00040200
J1 = (J-J1000*1000-J100*100-J10*10)	00040300
IF(J1000 .EQ. 0) GO TO 2164	00040400
2161 LINE1(JPRINT) = J100	00040500
2162 LINE2(JPRINT) = J10	00040600
2163 LINE3(JPRINT) = J1	00040700
GO TO 2170	00040800
2164 IF(J100 .EQ. 0) GO TO 2165	00040900
GO TO 2161	00041000
2165 LINE1(JPRINT) = DOT	00041100
IF(J10 .EQ. 0) GO TO 2166	00041200
	00041300
	00041400

GO TO 2162	00041500
2166 LINE2(JPRINT) = DOT	00041600
GO TO 2163	00041700
2170 CONTINUE	00041800
PRINT 92121, LINE1	00041900
PRINT 92121, LINE2	00042000
PRINT 92121, LINE3	00042100
2190 DO 2191 J = JFNGF1,JFNGF2	00042200
LINE1(J) = MFILL	00042300
LINE2(J) = WFILL	00042400
LINE3(J) = BLANK	00042500
2191 CONTINUE	00042600
IF(JLEFT .GT. 0) GO TO 2192	00042700
LINE1(JFNGF1) = BLANK	00042800
LINE2(JFNGF1) = BLANK	00042900
2192 IF(JRIGHT .GT. 0) GO TO 2193	00043000
LINE1(JFNGF2) = BLANK	00043100
LINE2(JFNGF2) = BLANK	00043200
2193 PRINT 92120, LINE1	00043300
PRINT 92121, LINE2	00043400
IF(IFXTEND(1) .LT. 2) GO TO 2200	00043500
PRINT 92195	00043600
92195 FORMAT(1HR)	00043700
	00043800
	00043900
C THIS IS THE ACTUAL PRINTING SECTION FOR THE ARRAY OF F(X,Y)	00043910
C VALUES. A SHADE OF GRAY IS ASSIGNED EACH (X,Y) LOCATION BY	00043920
C DETERMINING WHICH ZLEVEL JUST EXCEEDS THE AMPLITUDE AT THAT	00043930
C POINT. THE EQUIVALENT SHADE OF GRAY FOR THAT ZLEVEL IS THE	00043940
C PRINTED REPRESENTATION OF F(X,Y). THIS COMPARISON IS DONE ROW	00043950
C BY ROW. THE REMAINDER OF THIS SECTION IS SPEEDING UP THE I/O TO	00043960
C THE LINE PRINTER.	00043970
	00044000
	00044100
2200 DO 2300 I = 1,IXLENGTH	00044200
IX = I + IIAST	00044300
DO 2201 K = 1,3	00044400
DO 2201 J = 1,135	00044500
LINEPLOT(I,K) = BLANK	00044600
2201 CONTINUE	00044700
DO 2215 J = JSTART,JEND	00044800
JY = J - JIAST	00044900
INDEX = I + (JY-1) * IXLENGTH	00045000
DO 2214 L = 1,LLEVELS	00045100
IF(L .EQ. LLEVELS) GO TO 2205	00045200
ZTEST = ZDATA(INDEX)	00045300
IF(ISCALE(4) .EQ. 0) GO TO 2202	00045400
ZTEST = ITEST	00045500
2202 IF(ZTEST .GT. ZLEVEL(L)) GO TO 2214	00045600
2205 JPRINT = JSHIFT + 7 + J - JSTART	00045700

IF(ISCALE(3) .EQ. 0) LPRINT = LRIAS + LSTEP*L	00045800
IF(ISCALE(3) .NE. 0) LPRINT = LRIAS + LSTEP*(LLFVELS+1)-L)	00045900
DO 2210 K = 1,3	00046000
LINEPLOT(LPRINT,K) = LINESYM(LPRINT,K)	00046100
2210 CONTINUE	00046200
GO TO 2215	00046300
2214 CONTINUE	00046400
2215 CONTINUE	00046500
2220 IF(JLEFT .EQ. 0) GO TO 2221	00046600
LINE1(JEDGF1) = MFILL	00046700
LINE2(JEDGF1) = WFILL	00046800
2221 IF(JRIGHT .EQ. 0) GO TO 2230	00046900
LINE1(JEDGF2) = MFILL	00047000
LINE2(JEDGF2) = WFILL	00047100
2230 IF(IX .NE. MAX X) GO TO 2240	00047200
IF(JLEFT .LT. 2) GO TO 2231	00047300
LINE1(JSHIFT+1) = MFILL	00047400
LINE1(JSHIFT+2) = AFILL	00047500
LINE1(JSHIFT+3) = XFILL	00047600
2231 IF(JRIGHT .LT. 2) GO TO 2232	00047700
LINE1(JEDGF2+3) = MFILL	00047800
LINE1(JEDGF2+4) = AFILL	00047900
LINE1(JEDGF2+5) = XFILL	00048000
GO TO 2260	00048100
2232 IF(LINE1(JSHIFT+1) .EQ. MFILL) GO TO 2260	00048200
2240 IF(IX .NE. MIN X) GO TO 2250	00048300
IF(JLEFT .LT. 2) GO TO 2241	00048400
LINE1(JSHIFT+1) = MFILL	00048500
LINE1(JSHIFT+2) = IFILL	00048600
LINE1(JSHIFT+3) = NFILL	00048700
2241 IF(JRIGHT .LT. 2) GO TO 2242	00048800
LINE1(JEDGF2+3) = MFILL	00048900
LINE1(JEDGF2+4) = IFILL	00049000
LINE1(JEDGF2+5) = NFILL	00049100
GO TO 2260	00049200
2242 IF(LINE1(JSHIFT+1) .EQ. MFILL) GO TO 2260	00049300
2250 IF(JLEFT .LT. 2 .AND. JRIGHT .LT. 2) GO TO 2260	00049400
INDEX = IX - (IX/5)*5	00049500
IF(INDEX .NE. 0) GO TO 2260	00049600
I1000 = (IX/1000)	00049700
I100 = (IX-I1000*1000)/100	00049800
I10 = (IX-I1000*1000-I100*100)/10	00049900
I1 = (IX-I1000*1000-I100*100-I10*10)	00050000
IF(JLEFT .EQ. 2) ASSIGN 2251 TO JNEXT	00050100
IF(JLEFT .LT. 2) ASSIGN 2259 TO JNEXT	00050200
GO TO JNEXT, (2251,2259,2260)	00050300
2251 INDEX = JSHIFT + 1	00050400
ASSIGN 2259 TO JNEXT	00050500
2252 IF(I1000 .EQ. 0) GO TO 2256	00050600
LINE1(INDEX) = I1000	00050700

2253	LINE1(INDEX+1) = 1100	00050800
2254	LINE1(INDEX+2) = 110	00050900
2255	LINE1(INDEX+3) = 11	00051000
	GO TO JNEXT, (2251,2259,2260)	00051100
2256	LINE1(INDEX) = 001	00051200
	IF(1100 .EQ. 0) GO TO 2257	00051300
	GO TO 2253	00051400
2257	LINE1(INDEX+1) = 001	00051500
	IF(110 .EQ. 0) GO TO 2258	00051600
	GO TO 2254	00051700
2258	LINE1(INDEX+2) = 001	00051800
	GO TO 2255	00051900
2259	IF(JRIGHT .LT. 2) GO TO 2260	00052000
	ASSIGN 2260 TO JNEXT	00052100
	INDEX = JEDGE2 + 2	00052200
	GO TO 2252	00052300
2260	DO 2270 K = 1,3	00052400
	DO 2270 J = 1,17	00052500
	IF(J .EQ. 1) GO TO 2265	00052600
	INDEX = (J-2) * 8 + 7	00052700
	LINEPLOT(INDEX+1,K) = ISHIFT1*LINEPLOT(INDEX+1,K)	00052800
	LINEPLOT(INDEX+2,K) = ISHIFT2*LINEPLOT(INDEX+2,K)	00052900
	LINEPLOT(INDEX+3,K) = ISHIFT3*LINEPLOT(INDEX+3,K)	00053000
	LINEPLOT(INDEX+4,K) = ISHIFT4*LINEPLOT(INDEX+4,K)	00053100
	LINEPLOT(INDEX+5,K) = ISHIFT5*LINEPLOT(INDEX+5,K)	00053200
	LINEPLOT(INDEX+6,K) = ISHIFT6*LINEPLOT(INDEX+6,K)	00053300
	LINEPLOT(INDEX+7,K) = ISHIFT7*LINEPLOT(INDEX+7,K)	00053400
	JLINE(1,K) = LINEPLOT(INDEX+1,K) .OR. LINEPLOT(INDEX+2,K)	00053500
1	.OR. LINEPLOT(INDEX+3,K) .OR. LINEPLOT(INDEX+4,K)	00053600
2	.OR. LINEPLOT(INDEX+5,K) .OR. LINEPLOT(INDEX+6,K)	00053700
3	.OR. LINEPLOT(INDEX+7,K) .OR. LINEPLOT(INDEX+8,K)	00053800
	GO TO 2270	00053900
2265	IF(K .EQ. 3) IPAGE = ISHIFT1*BLANK	00054000
	IF(K .NE. 3) IPAGE = ISHIFT1*PLUS	00054100
	LINEPLOT(1,K) = ISHIFT2*LINEPLOT(1,K)	00054200
	LINEPLOT(2,K) = ISHIFT3*LINEPLOT(2,K)	00054300
	LINEPLOT(3,K) = ISHIFT4*LINEPLOT(3,K)	00054400
	LINEPLOT(4,K) = ISHIFT5*LINEPLOT(4,K)	00054500
	LINEPLOT(5,K) = ISHIFT6*LINEPLOT(5,K)	00054600
	LINEPLOT(6,K) = ISHIFT7*LINEPLOT(6,K)	00054700
	JLINE(J,K) = IPAGE .OR. LINEPLOT(1,K)	00054800
1	.OR. LINEPLOT(2,K) .OR. LINEPLOT(3,K)	00054900
2	.OR. LINEPLOT(4,K) .OR. LINEPLOT(5,K)	00055000
3	.OR. LINEPLOT(6,K) .OR. LINEPLOT(7,K)	00055100
2270	CONTINUE	00055200
	PRINT 92270, JLINE	00055300
92270	FORMAT(17(A8))	00055400
2300	CONTINUE	00055500
		00055600
C	THIS SECTION IS NUMBERING THE COLUMNS (IN EXTENDED ARRAY	00055610

C	NUMEROLOGY).	00055620
		00055700
	DO 2301 K = 1,3	00055800
	DO 2301 J = 1,135	00055900
	LINEPLOT(J,K) = BLANK	00056000
2301	CONTINUE	00056100
	IF(IEXTEND(1) .EQ. 0 .OR. IEXTEND(1) .EQ. 3) JTAIL = 1	00056200
	IF(IEXTEND(1) .EQ. 1 .OR. IEXTEND(1) .EQ. 2) JTAIL = 0	00056300
	IF(JTAIL .EQ. 0) GO TO 2400	00056400
	JEDGE1 = JSHIFT + 6	00056500
	JEDGE2 = JSHIFT + 8 + (JEND-JSTART)	00056600
	DO 2302 J = JEDGE1,JEDGE2	00056700
	LINE1(J) = MFILL	00056800
	LINE2(J) = WFILL	00056900
2302	CONTINUE	00057000
	IF(JLEFT .GT. 0) GO TO 2303	00057100
	LINE1(JEDGE1) = BLANK	00057200
	LINE2(JEDGE1) = BLANK	00057300
2303	IF(JRIGHT .GT. 0) GO TO 2304	00057400
	LINE1(JEDGE2) = BLANK	00057500
	LINE2(JEDGE2) = BLANK	00057600
2304	PRINT 92120, LINE1	00057700
	PRINT 92121, LINE2	00057800
	LINE1(JEDGE1) = LINE1(JEDGE2) = BLANK	00057900
	LINE2(JEDGE1) = LINE2(JEDGE2) = BLANK	00058000
	DO 2320 J = JSTART,JEND	00058100
	JPRINT = JSHIFT + 7 + J - JSTART	00058200
	LINE1(JPRINT) = LINE2(JPRINT) = BLANK	00058300
	IF(J .NE. MAX Y) GO TO 2305	00058400
	LINE1(JPRINT) = MFILL	00058500
	LINE2(JPRINT) = AFILL	00058600
	LINE3(JPRINT) = XFILL	00058700
	GO TO 2320	00058800
2305	IF(J .NE. MIN Y) GO TO 2310	00058900
	LINE1(JPRINT) = MFILL	00059000
	LINE2(JPRINT) = IFILL	00059100
	LINE3(JPRINT) = NFILL	00059200
	GO TO 2320	00059300
2310	INDEX = J - (J/5)*5	00059400
	IF(INDEX .NE. 0) GO TO 2320	00059500
	J1000 = J/1000	00059600
	J100 = (J-J1000*1000)/100	00059700
	J10 = (J-J1000*1000-J100*100)/10	00059800
	J1 = (J-J1000*1000-J100*100-J10*10)	00059900
	IF(J1000 .EQ. 0) GO TO 2314	00060000
2311	LINE1(JPRINT) = J100	00060100
2312	LINE2(JPRINT) = J10	00060200
2313	LINE3(JPRINT) = J1	00060300
	GO TO 2320	00060400
2314	IF(J100 .EQ. 0) GO TO 2315	00060500

GO TO 2311	00060600
2315 LINE1(JPRINT) = DOT	00060700
IF(J10 .EQ. 0) GO TO 2316	00060800
GO TO 2312	00060900
2316 LINE2(JPRINT) = DOT	00061000
GO TO 2313	00061100
2320 CONTINUE	00061200
PRINT 92121, LINE1	00061300
PRINT 92121, LINE2	00061400
PRINT 92121, LINE3	00061500
	00061600
2400 JBTAS = JEND	00061700
	00061800
	00061900
2500 CONTINUE	00062000
	00062100
C THIS ENDS THE PRINTING SECTION.	00062110
C THE EXTENDED ARRAY PARAMETERS ARE UPDATED IN THIS SECTION.	00062120
	00062200
	00062300
IF(IEXTEND(1)*IEXTEND(2) .EQ. 0) GO TO 3020	00062400
IF(IEXTEND(1)*IEXTEND(2) .EQ. 9) GO TO 3015	00062500
GO TO (3001,3005,3010) IDXY + 1	00062600
3001 ILAST = IXLNGTH	00062700
JLAST = JYLGTH	00062800
NX = 0	00062900
NY = 0	00063000
GO TO 5000	00063100
3005 IF(IEXTEND(1) .EQ. 3) GO TO 3006	00063200
ILAST = ILAST + IXLNGTH	00063300
NX = NX + 1	00063400
GO TO 5000	00063500
3006 ILAST = 0	00063600
JLAST = JLAST + JYLGTH	00063700
NX = 1	00063800
NY = NY + 1	00063900
GO TO 3016	00064000
3010 IF(IEXTEND(2) .EQ. 3) GO TO 3011	00064100
JLAST = JLAST + JYLGTH	00064200
NY = NY + 1	00064300
GO TO 5000	00064400
3011 JLAST = 0	00064500
ILAST = ILAST + IXLNGTH	00064600
NY = 1	00064700
NX = NX + 1	00064800
GO TO 3016	00064900
3015 ILAST = JLAST = 0	00065000
NX = NY = 1	00065100
IDXY = 0	00065200
3016 PRINT 93016	00065300

93016	FORMAT(JHR. //)	00065400
	GO TO 5000	00065500
3020	IF(IEXTEND(1) .NE. 0) GO TO 3040	00065600
	IF(IEXTEND(2) .NE. 0) GO TO 3030	00065700
	GO TO 3015	00065800
3030	IF(IEXTEND(2) .EQ. 3) GO TO 3015	00065900
	JLAST = JLAST + JYLENGTH	00066000
	NY = NY + 1	00066100
	GO TO 5000	00066200
3040	IF(IEXTEND(1) .EQ. 3) GO TO 3015	00066300
	ILAST = ILAST + IXLENGTH	00066400
	NX = NX + 1	00066500
		00066600
		00066700
		00066800
5000	RETURN	00066900
		00067000
		00067100
		00067200
		00067300
END		00067400

6.0 COMPARISON

No equivalent plotting routine was found in the library.

7.0 TEST METHOD AND RESULTS

The following simple example, PROGRAM TEST, illustrates the use of SURFPLØT.

PROGRAM TEST

C THIS IS AN EXAMPLE PROGRAM USING SURF PLOT. THE FUNCTION BEING PLOTTED IS THE
C SUM OF TWO SINUSOIDS, ONE IN X AND ONE IN Y, ON A RIAS.

```
DIMENSION A(72,100)
DIMENSION ZDATA(72,100), ITITLE(10), ISCALE(4), ZSCALE(3)
DIMENSION IEXTEND(2), ZLEVEL(10), ZPARAMTR(6)

IXLENGTH = 72
JYLENGTH = 100
READ 90000, ( ITITLE(I), I = 1,10 )
90000 FORMAT( 10( A8 ) )

PI = 3.1415926535
RADIAN X = PI/12.0
RADIAN Y = PI/20.0
DO 1000 J = 1,100
  ARG Y = J*RADIAN Y
  PART Y = SIN( ARG Y )
  DO 1000 I = 1,72
    ARG X = I*RADIAN X
    PART X = SIN( ARG X )
    A(I,J) = 2.00 + PART X + PART Y
  1000 CONTINUE

CALL SURF PLOT ( A, IXLENGTH, JYLENGTH, ITITLE, ISCALE, ZSCALE,
1  ZLEVEL, ZPARAMTR, IEXTEND, ISARCH )

END
```

THE FUNCTION IS $F(X,Y) = 2.00 + \sin(\pi \cdot I/12.00) + \sin(\pi \cdot J/20.00)$

THE FUNCTION IS $F(X,Y) = 2.00 + \sin(\pi I/12.00) + \sin(\pi J/20.00)$

DATA MAXIMUM 4.0000+000 (6, 10)
 DATA MINIMUM 0.0000+000 (18, 30)
 DATA RANGE 4.0000+000

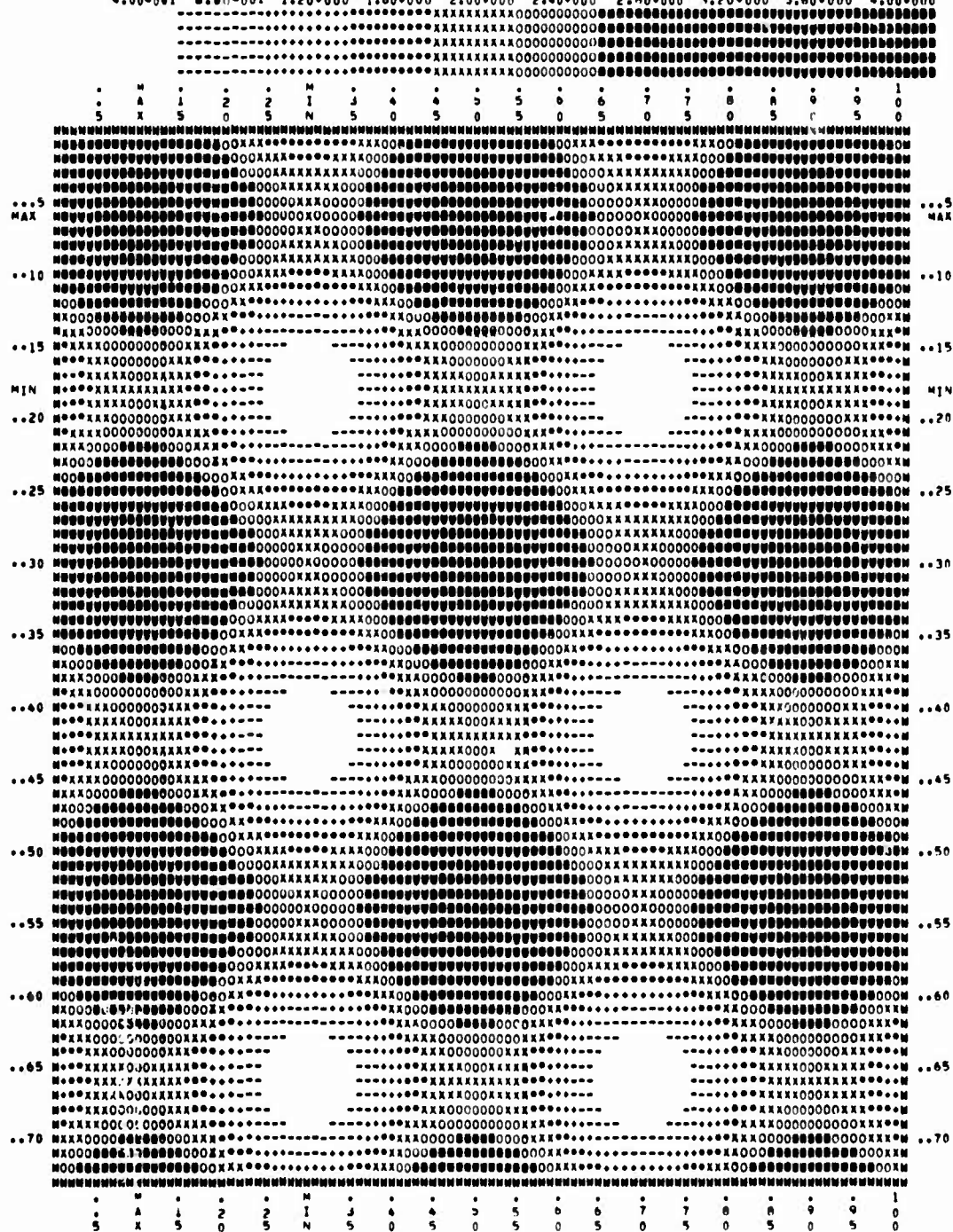
SCALE MAXIMUM 4.0000+000
 SCALE MINIMUM 4.0000+001
 SCALE RANGE 3.6000+000

PLOT REFERENCE 4.0000+000
 PLOT BIAS 0.0000+000
 PLOT EXTENT (72, 100)

DB REFERENCE 0.00

HALF TONE DENSITY WEDGE AND SCALE

0.100MAX 0.4 VMAX 0.300MAX 0.400MAX 0.500MAX 0.600MAX 0.700MAX 0.800MAX 0.900MAX 1.000MAX
 4.00+001 0.00+001 1.20+000 1.60+000 2.00+000 2.40+000 2.80+000 3.20+000 3.60+000 4.00+000



8.0 REMARKS

The "writing speed" of the line printer is not high. The technique used here depends strongly on the natural spatial integration done by the eye. If the structures change quickly, e.g., in the span of one element to its neighbor, then the detail will not be perceived.

The gray scale used in SURFPLØT is one selected after some trial and error. It is by no means unique. There is literature on the subject, although the only reference consulted was Knuth [1]. The user who wants to develop a different gray scale can either (1) use more than ten shades in which case this subroutine requires a new gray scale or (2) change the internal gray scale contained in the data statements for LINESYM. One technique of interest is to "go around the gray scale" several times to obtain more shades. That is, using the same symbols in cycles. The author would be interested to see other gray scales generated using the line printer.