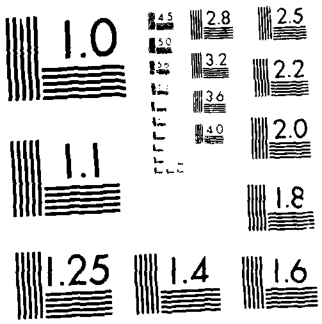
AD-A195 163 IMPLEMENTATION OF THE NCAR (NATIONAL CENTRE FOR ATMOSPHERIC RESEARCH) GRA..(U) AERONAUTICAL RESEARCH LABS MELBOURNE (AUSTRALIA) B D FAIRLIE JAN 88 1/2

UNCLASSIFIED ARL/AERO-TM-394 DODA-AR-004-593 F/G 12/5 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

(4)

ARL-AERO-TM-394 AR-004-593

AUSTRALIA

DEPARTMENT OF DEFENCE

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION

AERONAUTICAL RESEARCH LABORATORY

MELBOURNE, VICTORIA

Aerodynamics Technical Memorandum 394

IMPLEMENTATION OF THE NCAR GRAPHICS PACKAGE AT ARL

by

B.D. FAIRLIE

Approved for Public Release

AR-004-593

DEPARTMENT OF DEFENCE
DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION
AERONAUTICAL RESEARCH LABORATORY

Aerodynamics Technical Memorandum 394

IMPLEMENTATION OF THE NCAR GRAPHICS PACKAGE AT ARL (U)

by

B.D. FAIRLIE

SUMMARY

The NCAR graphics package consists of several graphics utilities and their supporting routines which are designed to provide a wide range of high-level graphics functions. Its capabilities range from simple contouring, labelling and smoothing of lines, through to vector and velocity field plotting and a mapping facility for any part of the Earth. This document describes the philosophy used in implementing the package at ARL, the mechanisms for using the package, and the capabilities of the routines currently implemented. It represents the current status of the package as at August 1987.

DSTO ARL
FISHERMEN'S BEND

POSTAL ADDRESS: Director, Aeronautical Research Laboratory,
P.O. Box 4331, Melbourne, Victoria, 3001, Australia

Contents

List of Figures

1 Introduction

The NCAR graphics package was developed at the National Centre for Atmospheric Research, Boulder, Colorado, USA, to provide their research staff with a wide range of graphics utilities. It consists of a series of high-level graphics utilities and, in its original form, a set of low level routines (known as the System Plot Package) which perform much the same function as does DI3000 on the ARL Elxsi 6400, albeit with considerably lower functionality. The general capabilities of the high-level utilities range from simple contouring, labeling and smoothing of lines through to vector and velocity field plotting and map representation of any section of the Earth [1].

The package was acquired from NCAR in late 1986, and the first implementation on the ARL Elxsi achieved shortly thereafter. This first implementation was aimed specifically at solving problems encountered in presenting data produced by one particular ARL task [5]. Since that time, considerable effort has been invested in improving the user interface and making the system more readily accessible to ARL computer users in general.

This memo describes the philosophy used in implementing the system on the Elxsi 6400, the mechanisms for gaining access to the package, and the capabilities of the routines currently available. It also includes brief descriptions of the many useful non-graphics routines included as part of the tape received from NCAR.

2 Implementation Philosophy

In making the NCAR graphics package available on the ARL Elxsi 6400, two major aims have been:—

- To maintain the highly standard use of the package's language — ANSI 66 FORTRAN — wherever possible. This aim has been largely met, with no need for recourse to "dirty tricks", in all but one small area. This area (described in Section 5.5) still uses standard FORTRAN but in a somewhat "innovative" manner.

- To integrate the package with DI3000 in a way that is as transparent to the user as possible. Although total transparency has not been possible to achieve, the package generally runs "below" DI3000 with the user needing little knowledge of how the two systems interact.

The complete NCAR graphics package contains more than sixty separate user callable routines [4]. In any particular area, the package typically provides a range of routines providing very much the same level of functionality, but varying from the "quick and dirty" at the low end, to the "slow but attractive" at the high end. In general, only the "slow but attractive" versions have been implemented. It was considered that although slower than the low end versions, in no case were execution times sufficiently long to justify providing the quicker versions, at the expense of extra work and possible user confusion. In addition, some routines were not considered worth implementing. The reasons for this decision were either that a similar (usually better) facility was already available via DI3000, or that no immediate use was expected at ARL. In the latter case, if such need should arise, the routines could be added with little difficulty.

Since the package already included a set of low-level routines for outputting primitive plotter instructions (the System Plot Package), it was decided that the simplest and quickest implementation would be obtained by rewriting these, using DI3000 primitive instructions. Only those routines needed for the operation of the selected sub-set of high-level routines were emulated. Although a description of the parameters and operation of these routines is included in Section 5, their direct use is to be discouraged. The same functionality can in general be obtained directly from DI3000 calls, without the overhead of time and data storage imposed by the System Plot Package.

The original NCAR package achieved device independence by generating *meta-code* files, a device independent set of instructions which can be *translated* into the instruction set of an intended target graphics device. Rather than implement this system, device independence has been obtained by taking advantage of the equivalent facility included in DI3000.

3 Description of the Utilities

The following is a brief description of the high-level graphics utilities currently implemented on the Elxsi 6400. Further descriptions of the utilities functions together with detailed user guides are included in individual Appendices.

3.1 Contour Plotting

The two utilities in this family will generate contour lines from either regularly (CONRECSUPER) or irregularly (CONRANSUPER) distributed data in a two-dimensional array. The routines will optionally smooth the data (using splines under tension), label the contour lines and remove contour lines from crowded areas. Both the contouring utilities require the support of the dashed line package DASHSUPER — see Section 3.6.

An example of the output from CONRECSUPER is shown in Figure 1. In this example, contours are plotted for the function

$$z(x,y) = x + y + \frac{1}{(x-0.1)^2 + y^2 + 0.09} - \frac{1}{(x+0.1)^2 + y^2 + 0.09}$$

Note:— CONRANSUPER is currently not implemented due to the presence of an unresolved "bug". It will be implemented as soon as this problem is resolved.

3.2 Three-Dimensional Surface Drawing

The two utilities in this family produce three-dimensional representations of the input data.

ISOSURF draws an approximation to an iso-valued surface (with hidden lines removed — but not perfectly) from a three-dimensional array. The position of the

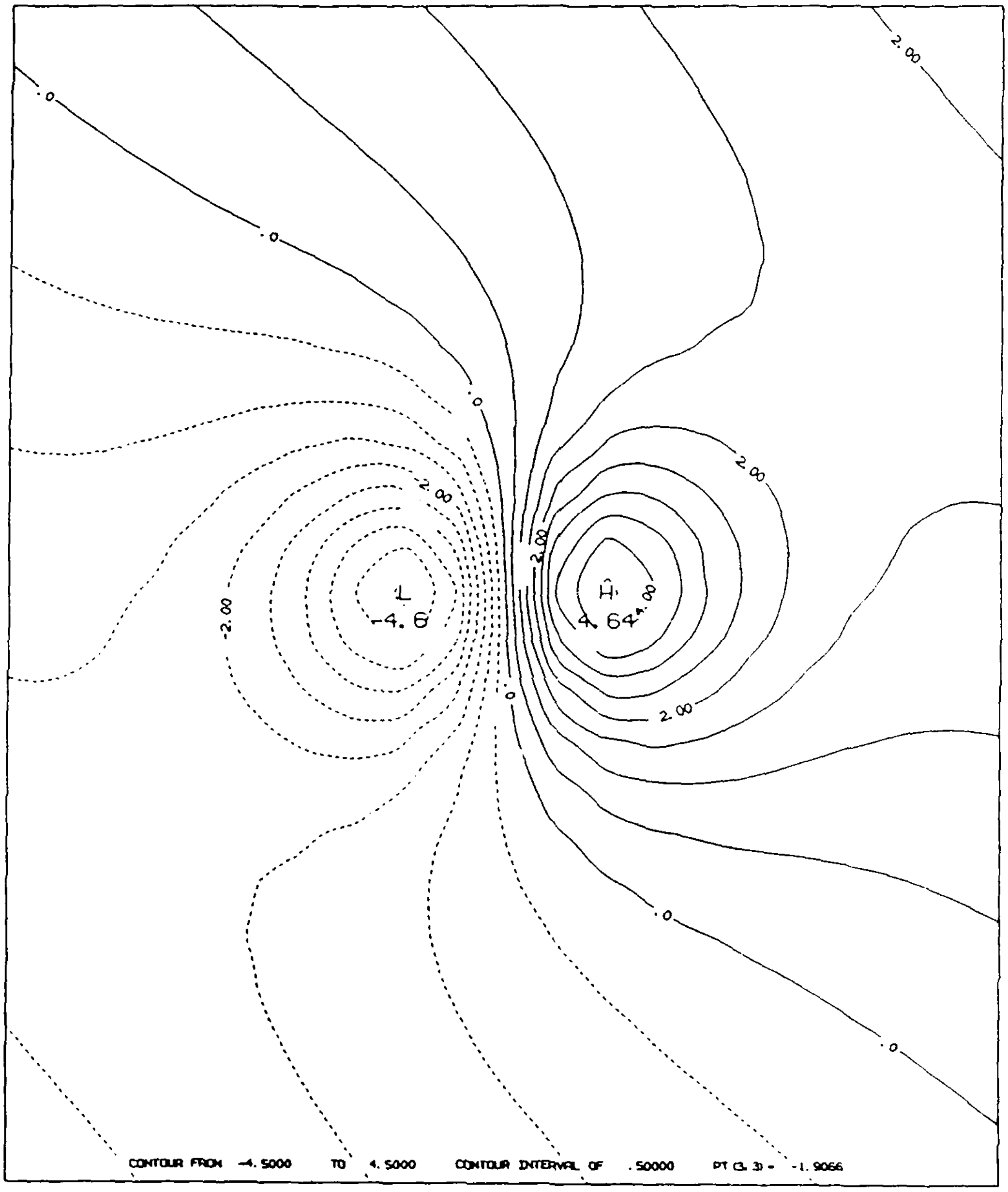


Figure 1: Contour Plot Produced by CONRECSUPER

eye in three-space may be specified, as well as parameters controlling how the surface is to be represented.

A three-dimensional view of two intersecting toroids as produced by ISOSURF is shown in Figure 2. Note the less than perfect hidden line removal.

SURFACE draws a perspective picture of a function of two variables (again with hidden lines removed — in this case, very well). The function is represented by a two-dimensional array of surface heights. Parameters are provide to control the line of sight, and contour lines may be plotted on the surface.

Figure 3 presents an example of the output produced by SURFACE. In this case the plot is of the same function contoured by CONRECSUPER in Figure 1.

3.3 Vector Plotting

The routine VELVECT draws a representation of a two-dimensional velocity field by drawing arrows at each data location, with the length of the arrow proportional to the magnitude of the velocity at that location, and the direction of the arrow indicating the direction of the flow at that location. The size of the vectors may be scaled, and vectors may be omitted from specified parts of the flowfield.

The output from VELVECT presented in Figure 4 is once again a representation of the function contoured in Figure 1. However, in this case, the direction of the field represents the direction of the gradient of the function, and the magnitude of the field represents the logarithm of the absolute value of the gradient.

3.4 Streamline Plotting

The routine STREAMLINE plots a streamline representation of velocity data contained in a two-dimensional array. Arrowheads are added to the streamlines to indicate flow direction. Streamline spacing is under user control. The routine uses various interpolation and extrapolation formulæto deal withmissing data values.

Figure 5 presents the output from STREAMLINE for a field with velocity components described by

$$u(i,j) = \sin(2\pi \frac{i-1}{i_{max}})$$

$$v(i,j) = \sin(2\pi \frac{j-1}{j_{max}})$$

where

$$1 \leq i \leq i_{max} = 21$$

and

$$1 \leq j \leq j_{max} = 25$$

3.5 Map Plotting

The routine EZMAP produces maps of portions of the Earth's surface in any one of nine different projections. Continental, political and/or U.S. state boundaries may

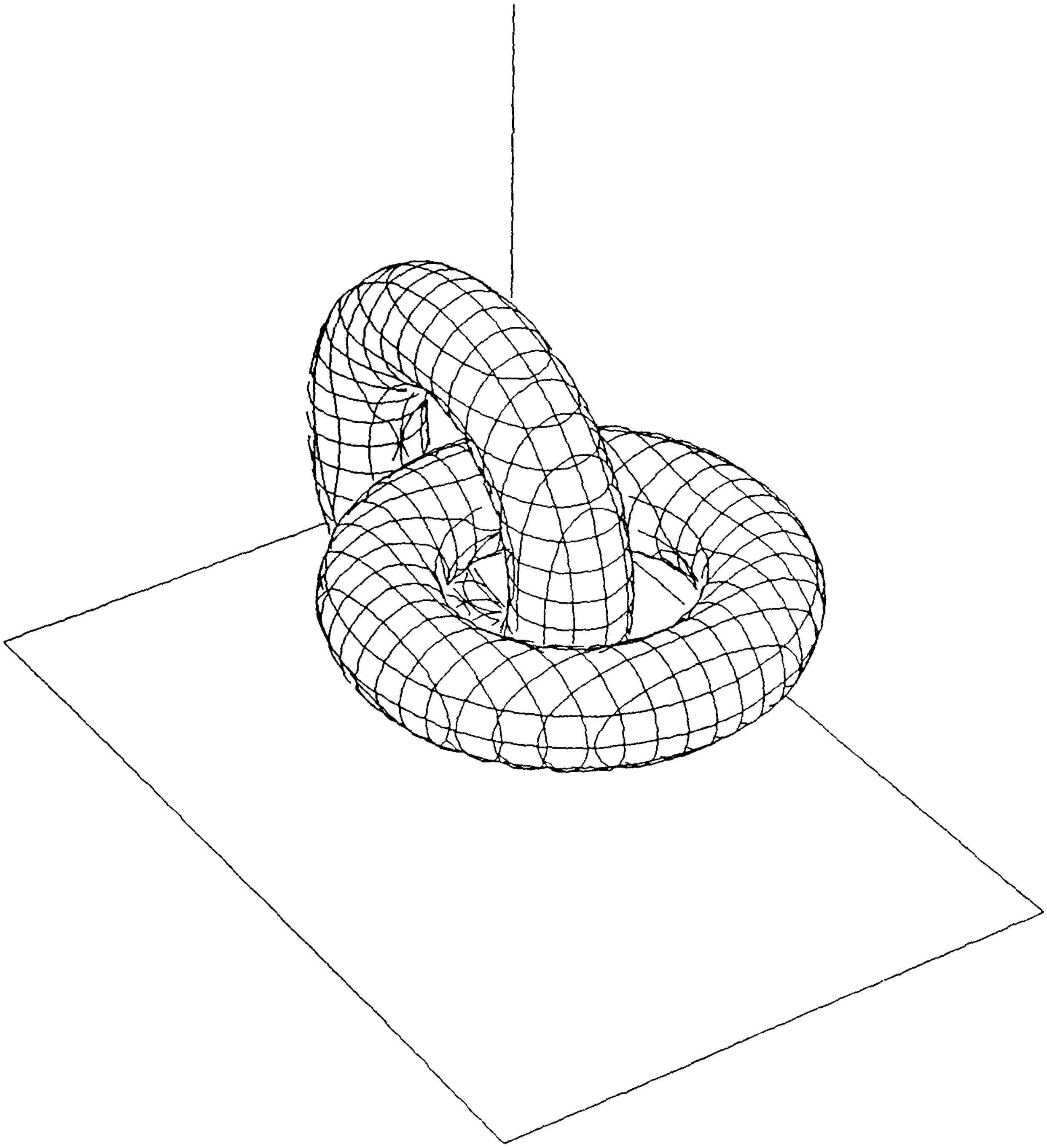


Figure 2: View of Intersecting Toroids Produced by ISOSURF

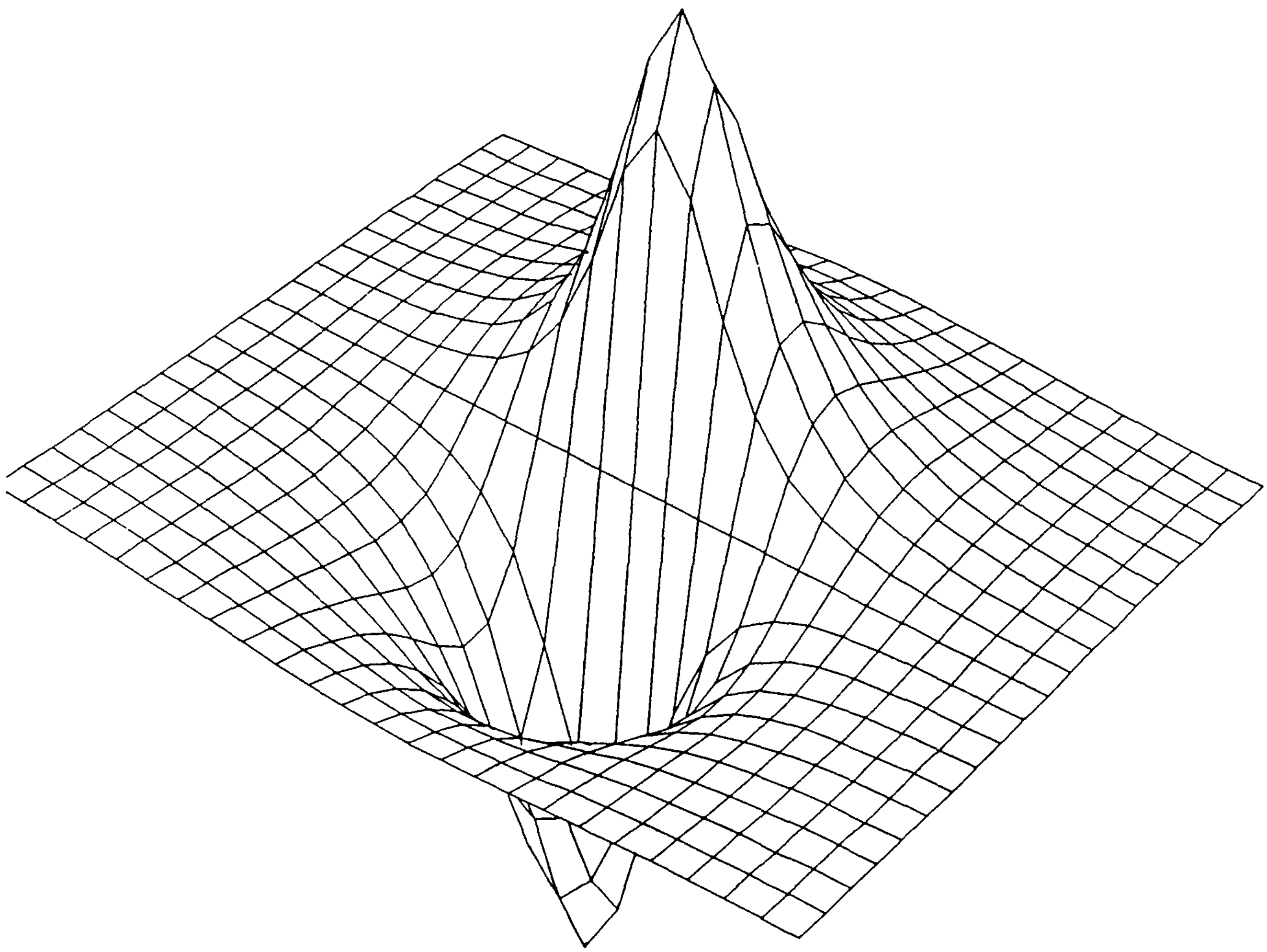


Figure 3: Perspective View Produced by SURFACE

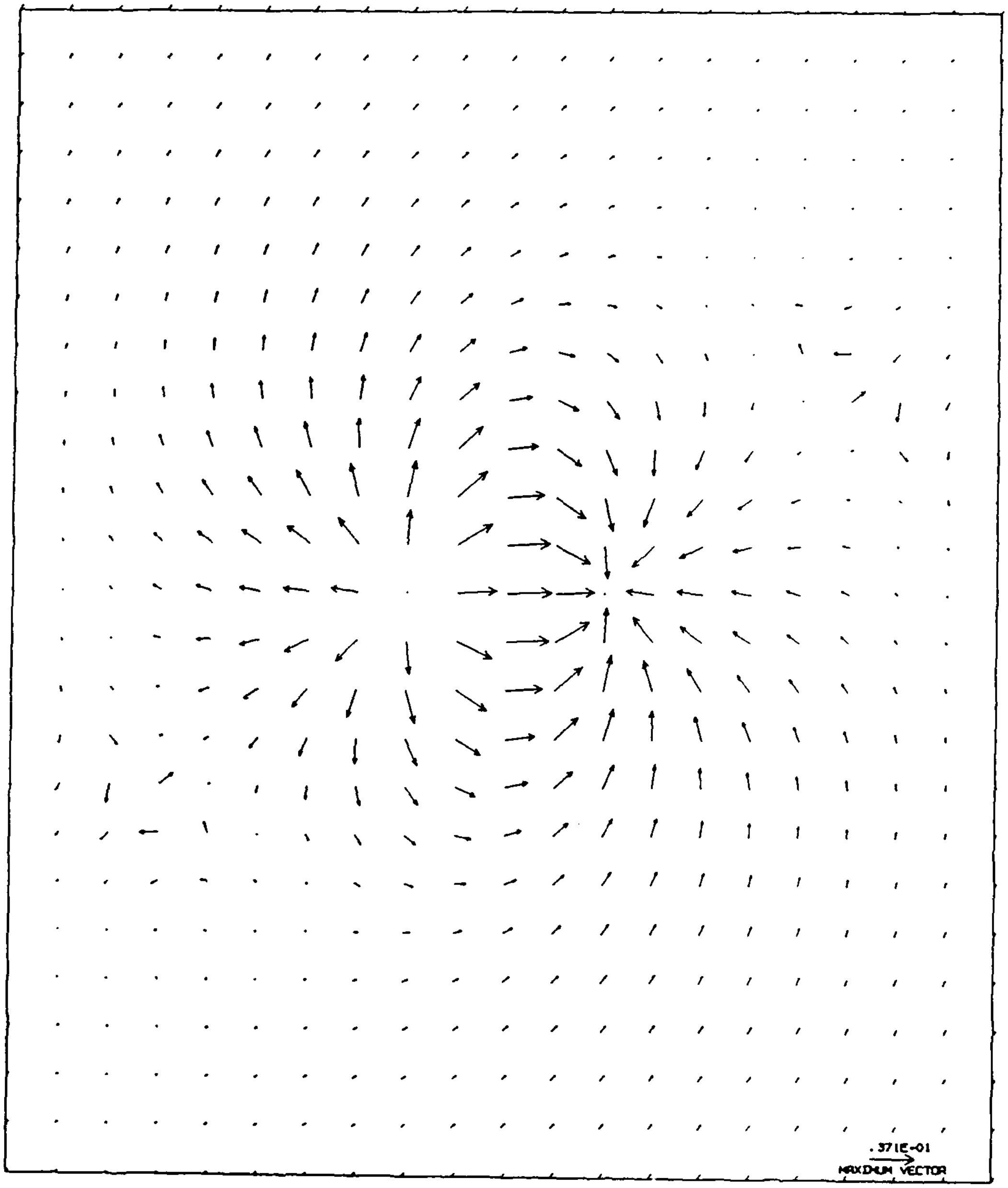


Figure 4: Vector Plot Produced by VELVECT

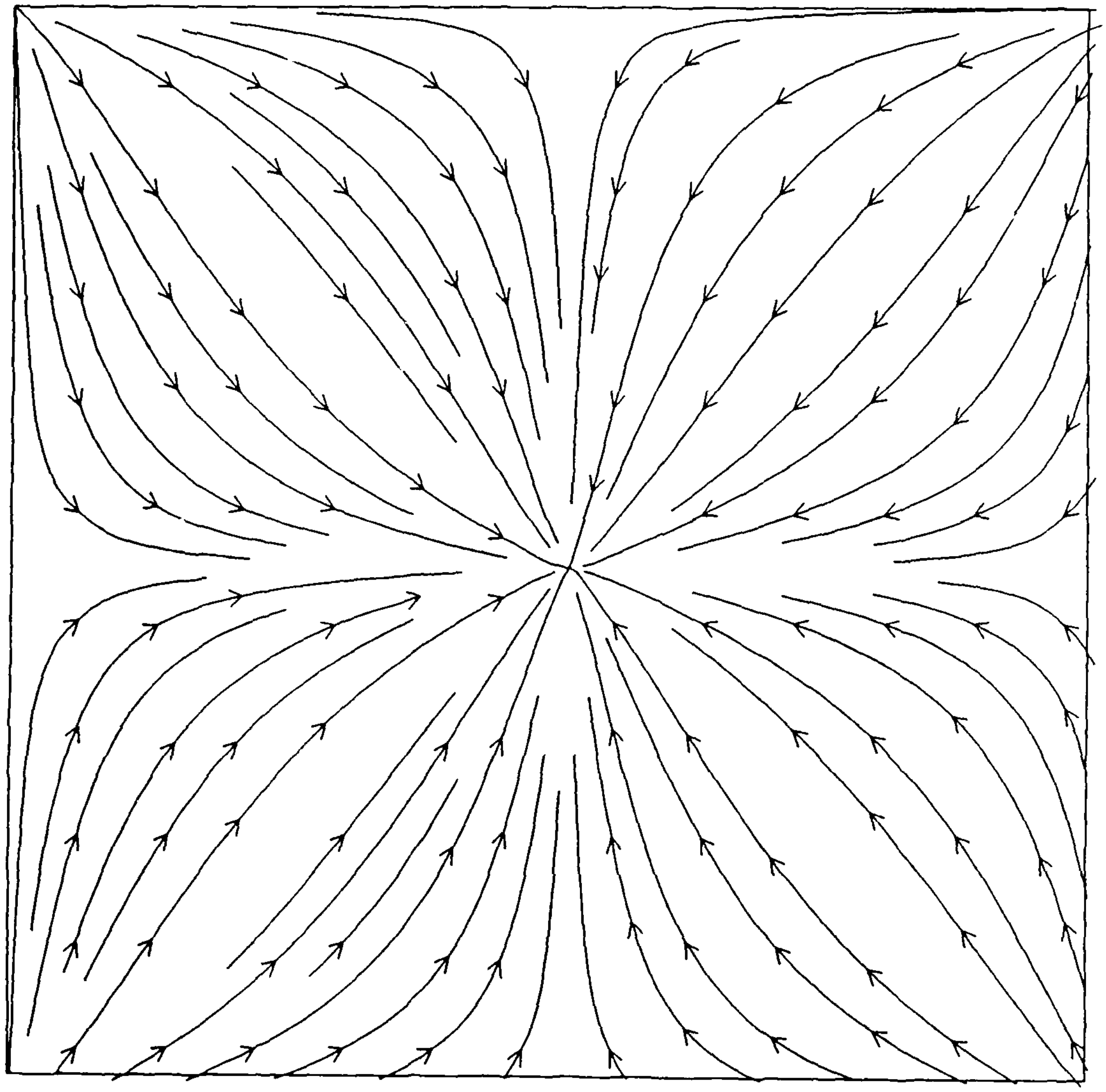


Figure 5: Streamline Plot Produced by STREAMLINE

be included. Many parameters are available to obtain the desired view of any part of the Earth's surface.

The map shown in Figure 6 was produced using the sample program discussed in Section 4.3. The map is an Orthographic projection (JPRL = 'OR') of the region surrounding Australia, with the pole of the projection located at a lattitude of 38° South and a longitude of 150° East (PLAT = -38, PLON = 150, ROTA = 0). The area of the map to be drawn is to be the maximum useful area produced by this projection (JLTS = 'MA'), the default value. In this case, PLM1, PLM2, PLM3, and PLM4 are not used.

3.6 Dashed Line Plotting

The utility DASHSUPER consists of a software generated dashed-line package with smoothing and labeling capabilities. In addition, lines may be removed from areas of high crowding.

3.7 Spline Drawing

Although not strictly part of the NCAR graphics package, the package CURVEPACK has been included in the ARL implementation. Routines from CURVEPACK are used by Both Contouring Utilities, ISOSURF and SURFACE. CURVEPACK consists of a suite of 24 subprograms for fitting splines under tension to one-, two-, and three-dimensional data.

3.8 Routines Not Currently Available

Apart from the "quick and dirty" routines which implement functions similar to those described above, the following utilities are not currently available in the ARL version of the NCAR package:—

ISOSURFHR Similar to ISOSURF except that the iso-valued surfaces are generated from a high resolution array. Intended to replace ISOSURF for arrays much larger than $40 \times 40 \times 40$.

THREED A three-dimensional line drawing package whose capabilities are significantly lower than those provided directly by DI3000.

PWRITX This routine and its companion routines, PWRITY, PWRZI, PWRZS, and PWRZT, generate software characters in a variety of fonts. Generally inferior to the similar capabilities provided by DI3C00.

AUTOGRAPH This routine draws and annotates curves or families of curves. Its capabilities are similar to those of the DI3000 companion system Grafmaker.

HALFTON This routine creates half-tone pictures from a two-dimensional array. Although a potentially useful program, it is not clear how to implement it on any of the graphics hard-copy devices currently available at A.R.L. — at least not without a large amount of effort.

SCROLL This routine produces scrolled movie style titles.

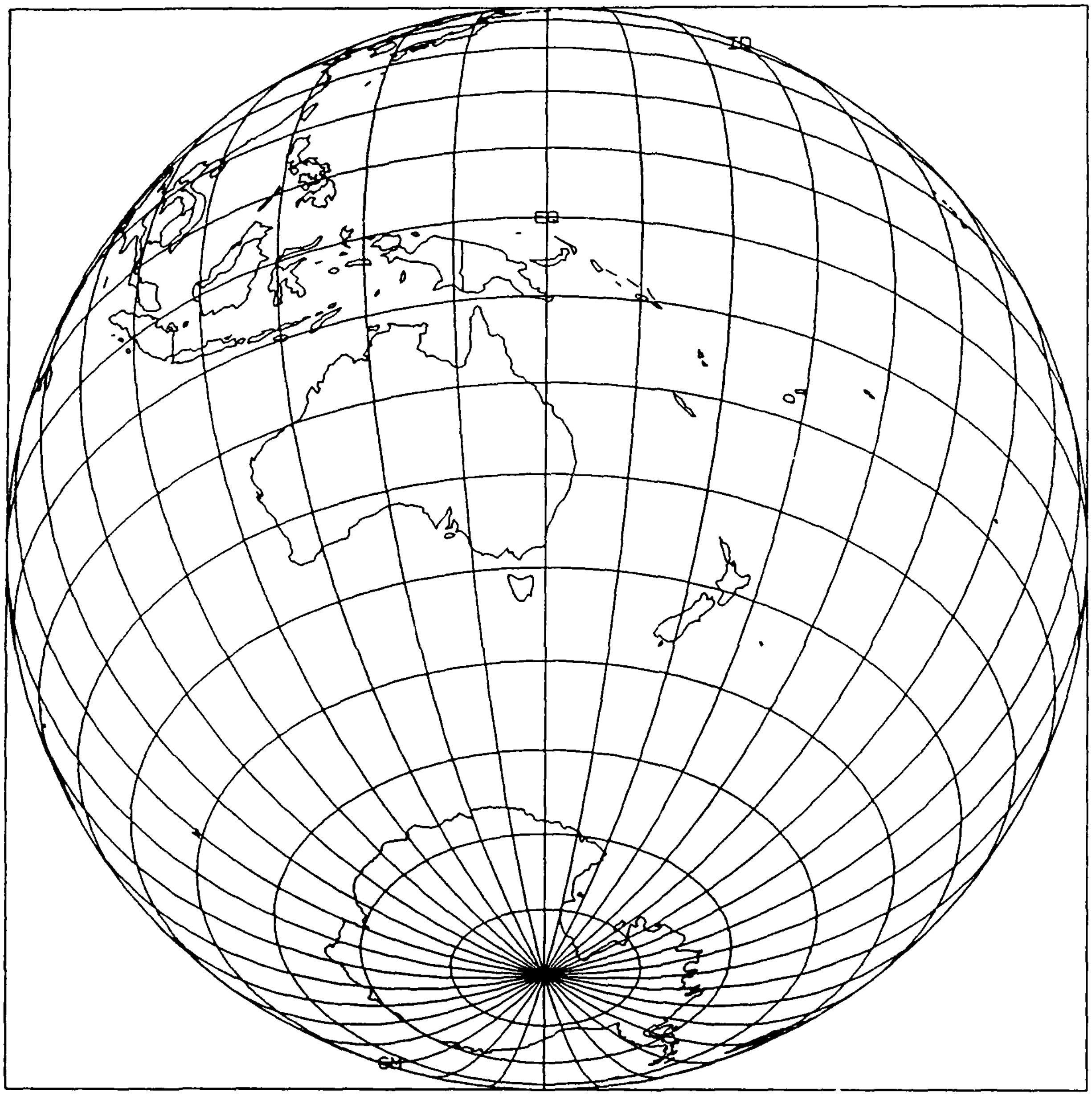


Figure 6: Map Produced by EZMAP

WINDOW This routine provides a window clipping function similar to that provided by DI3000.

4 Using the Utilities on the Elxsi 6400

Access to the available NCAR utilities is provided via object library files — one containing the System Plot Package (see Section 5) and one for each of the utilities described above. To allow simple maintenance and updating, these object libraries are stored in the directory

`subdomains/ARL/user/ae.fairlie/ncar`

with security set to allow read and execute access for all users.

The names and contents of the object library files are as follows:—

Library File	Contents
`ncarsim.lib.o`	The emulation of the System Plot Package
`conrecsuper.lib.o`	Regular data contourer
`conrassuper.lib.o`	Irregular data contourer
`isosurf.lib.o`	Iso-valued surface plotter
`surface.lib.o`	Perspective plot of function of two variables
`velvect.lib.o`	Velocity vector plotter
`streamline.lib.o`	Streamline plotter
`ezmap.lib.o`	Map drawing package
`dashsuper.lib.o`	Software dash-line package
`curvepack.lib.o`	Spline fitting package

4.1 Binding to the Object Libraries

A user program which makes use of one or more of the NCAR Utilities must be bound with a minimum of:—

1. The DI3000 graphics system library.
2. A DI3000 device driver.
3. The NCAR System Plot Package emulator library.
4. The library containing the utility required.

Hence, a user program "example" which uses the velocity vector plotting utility and is to be run on a Tektronix 4010 compatible graphics terminal, would be bound via the command:—

```
bind example libfiles=/applications/di3000/di3000,&
                      /applications/di3000/410,&
                      subdomains/ARL/user/ae.fairlie/ncar/ncarsim,&
                      subdomains/ARL/user/ae.fairlie/ncar/velvect
```

For both of the contouring utilities (CONRECSUPER and CONRANSUPER) and EZMAP, the software dashed-line package DASHSUPER must also be bound using the object library `dashsuper.lib.o`.

4.2 Elxsi Shell File

Since the above binding process is not simple, a shell file called "`ncarbind`" has been written to make the process more convenient. This shell file has been written with a similar form to that of the shell file "`libbind`" which is used to bind libraries stored in the `/applications` directory. However `ncarbind` always includes the DI3000 library, the NCAR Syatem Plot Routine emulation library, and the user's selection of the other NCAR utility libraries. The software dashed-line package library is also included automatically as necessary. A full description of the operation of `ncarbind` is provided in a help file available on the Elxsi system.

4.3 Source File Requirements

Source files which access NCAR Utilities should follow the standard form for programs using DI3000 routines. Hence, DI3000 must be initialized via a call to `JBEGIN`, devices initialized and turned on via calls to `JDINIT` and `JDEVON`, and subsequently devices turned off when terminating DI3000 via a call to `JEND`. Apart from explicit calls to the NCAR Utility routines described in the Appendices, the only other routine which must always be called is `ININCR`. This routine sets up the interface between DI3000 and the emulation of the NCAR System Plot Package, initializing all variables in the common block `\NCAR\` by which the two systems communicate. `ININCR` *must* be called:—

- After DI3000 has been initialized but before any NCAR routines are called. (Strictly, NCAR routines which do not produce any graphic output may preceed the call to `ININCR` (see example below), but the above restriction will *always* work).
- After each call to `JWINDO` or `JVPORT` or any other routine (not part of one of the NCAR Utilities) which could change the mapping between world and viewplane coordinates.
- After each and every change in Metafile status — either output to a Metafile being turned either on or off.

The routine `ININCR` has a single logical argument which should be `.true.` if output is being sent to a Metafile and `.false.` otherwise.

The following short program gives an example of how the NCAR Utility routines are combined with DI3000 calls. The purpose of this program is to allow a user to select a portion of the Earth to be mapped, to select the projection system and parameters, to preview the map on a graphics device and, if satisfied, to output the map to a Metafile for later plotting.

```
PROGRAM MAPDRAW
LOGICAL MONOFF
```

```
      CHARACTER JPRJ*2,JLTS*2,YES*1
C
      MONOFF=.FALSE.
      CALL JBEGIN                    ! Initialize DI3000
10    CONTINUE                       ! Loop back to here
      CALL MAPSTC('OU','PS')         ! Draw internat'l outlines
      CALL MAPSTL('PE',.TRUE.)       ! Draw the perimeter
c
      TYPE *,'Change Projection (y or n)?'
      ACCEPT 20,YES
20    FORMAT(A)
      IF(YES.EQ.'y') THEN
         TYPE *,'Enter Projection Code (JPRJ) :'
         ACCEPT 20,JPRJ
         TYPE *,'Enter PLAT,PLON, and ROTA :'
         ACCEPT *,PLAT,PLON,ROTA
         CALL MAPROJ(JRPJ,PLAT,PLON,ROTA)
      END IF
C
      TYPE *,'Change u/v Plane Specs. (y or n)?'
      ACCEPT 20,YES
      IF(YES.EQ.'y') THEN
         TYPE *,'Enter Limits Code (JLTS) :'
         ACCEPT 20,JLTS
         TYPE *,'Enter PLM1, PLM2, PLM3 and PLM4 :'
         ACCEPT *,PLM1,PLM2,PLM3,PLM4
         CALL MAPSET(JLTS,PLM1,PLM2,PLM3,PLM4)
      END IF
C
      CALL JDINIT(1)
      CALL JDEVON(1)
      CALL ININCR(MONOFF)
      CALL MAPDRW
      CALL JPAUSE
c
c  The following line has been found necessary on some graphics
c  terminals to catch a spurious character(s) echoed following
c  the keystroke which completes the pause period.
c
c     ACCEPT 20,JPRJ
c
       TYPE *,'Create a Metafile of this Map (y or n)?'
       ACCEPT 20,YES
       IF(YES.EQ.'y') THEN
          MONOFF=.TRUE.
          CALL JDINIT(0)
```

```
          CALL JDEVON(0)
          CALL ININCR(MONOFF)
          CALL MAPDRW
          CALL JPAUSE(1)
C         ACCEPT 20,JPRJ
          CALL JDEVOF(1)
          MONOFF=.FALSE.
      END IF
C
      TYPE *,'Continue with Another Map (y or n) ?'
      ACCEPT 20,YES
      IF(YES.EQ.'y') THEN
         GO TO 10
      ELSE
         CALL JDEND
         CALL EXIT
      END IF
      END
```

This program has been used to produce the example map shown in Figure 6.

The following points should be noted:—

1. The calls to the EZMAP Utility routines MAPSTC, MAPSTI, MAPROJ and MAPSET serve only to set parameters for the later call to MAPDRW: they do not produce any graphical output and hence can be called before the first call tc ININCR.

2. MAPDRW *does* produce graphical output (it does the actual drawing of the map), and hence must follow the first call to ININCR.

3. ININCR is called *whenever* the Metafile device id turned either on *or* off.

4. Extra calls to ININCR (i.e. those not strictly required) do no harm, except in the extra (small) amount of cpu time consumed.

4.4 Opening and Closing Segments

In implementing the NCAR Utilities on the Elxsi, one of the most difficult problems has proved to be to ensure that the state of the DI3000 "segment" (either open or closed) is correct for the various calls to DI3000 routines. Many calls to DI3000 routines from within the NCAR Utilities and the emulation of the System Plot Package must be done with the segment closed (e.g a call to JVPORT), or open (e.g a call to JMOVE or other drawing primitives). In general, each NCAR Utility has been modified to allow the state of the current segment to be determined on entry (via a call to J1IGET) and the Utility then proceeds as appropriate. Hence, the state of the segment in the user's program at the time when an NCAR Utility is called *should* be irrelevant. However, there may be (and almost certainly will be) cases where a particular entry point is not treated correctly. In these cases, one simple cure is to enter the NCAR Utility with the segment in the opposite state and try again!

Despite whether this trick works or not, please let me know of all such problems so they can be fixed.

4.5 Processing of Parts of Arrays

Most of the NCAR Utilities have argument lists which make it possible to process not only the entire array, but any part (sub-array) of the array. When an N-dimensional array is being passed to a Utility, the following arguments are required: the name (and origin if not (1,1)) of the array, its first N-1 dimensions in the dimension statement of the calling routine, and the number of array values to be processed in each of the N dimensions.

For example, consider a two-dimensional array Z dimensioned in the calling program as Z(100,100). If only a 50×30 sub-array of Z, starting at Z(10,20), is to contoured using CONRECSUPER, the call to the Utility might appear as

```
CALL CONREC (Z(10,20),100,50,30,........)
```

This call would produce a contour map from the Z values bounded by Z(10,20), Z(60,20), Z(60,50) and Z(10,50).

If on the other hand, the full array was to be processed, the call would appear as

```
CALL CONREC (Z,100,100,100,........)
```

5 The System Plot Package

The System Plot Package is used to perform low-level graphics tasks such as scaling, line drawing and character drawing. More complicated capabilities are available through the use of the NCAR utilities which use the System Plot Package. These utilities are described in Section 3.

The version of the System Plot Package available on the ARL Elxsi 6400 uses DI3000 primitive routines to emulate the original NCAR [2] versions. The following Sections present a description of the capabilities of these routines (more detail is available in Appendix J), but their direct use is not encouraged. All the facilities are available, usually in a more powerful and consistent form, via direct calls to DI3000 routines.

The plotter is assumed to have some finite resolution, initially assumed to be 1024 by 1024 addressable positions. This can be changed by the user, but this document uses the $2^{10} \times 2^{10}$ resolution in its examples. Users of some other resolution should bear this in mind when interpreting the examples.

The System Plot Package, as implemented on the ARL Elxsi 6400 contains routines for scaling, plotting, writing, and determining status.

5.1 Scaling

ININCR Sets up the interface between DI3000 and the System Plot Package, initializing all variables in the system common block /NCAR/. This routine *must* be called from the user's program before calling any of the NCAR routines.

SET Establishes mapping from floating point ("world") coordinates to the (integer) addressable coordinates of the plotter

SETI Changes the number of addressable coordinates (i.e. the resolution) which can be used as input to the plot package.

FL2INT Coordinate conversion based on SET call without plotting

5.2 Plotting

FRSTPT Pen up move.

VECTOR Pen down move.

LINE Connects two coordinate pairs with a line.

POINTS Draws points at a set of coordinates optionally connecting them with line segments.

CURVE Draws (straight) line segments through a sequence of coordinate pairs.

PLOTIT Pen up or down move with high resolution integer coordinates.

FRAME Advances plotter to next picture.

PERIM Draws a perimeter around the current frame.

5.3 Character Drawing

PWRIT Draws character strings on the plotter.

5.4 Status

GETSET Returns most recent SET call parameters.

GETSI Returns the number of addressable coordinates which are allowed as input to the plot package.

MXMY Returns the current pen location in plotter address units.

5.5 Coordinates

Whenever the coordinate arguments of a System Plot Package routine are X and Y (floating point) — see Appendix J — alternate arguments MX and MY (integer) may be used. When floating point arguments are used, the coordinate position on the plotter is determined by the plot package using the parameters of the most recent SET call. When integer arguments are used, they are assumed to be coordinate positions on the plotter in the range 1 to 1024 (unless the upper bound has been changed by a SETI call). Zero is always considered a floating point number. For example, FIRSTPT may be called in any of these ways:—

```
CALL FIRSTPT (X,Y)        CALL FIRSTPT (X,MY)
CALL FIRSTPT (MX,Y)       CALL FIRSTPT (MX,MY).
```

To allow this duality of parameter type, which is not strictly consistent with the FORTRAN-77 Standard but is used throughout the NCAR high-level utilities, the following "dirty trick" has been used. The subroutine FIRSTPT contains the following code:—

```
      SUBROUTINE FRSTPT(X,Y)
C
      LOGICAL INTT
      EQUIVALENCE (MX,RX),(MY,RY)
C
      IF(INTT(X)) THEN
            RX=X
            RY=Y
            CALL TRANS(MX,MY,X1,Y1)
      ELSE
            X1=X
            Y1=Y
      END IF
C
      CALL JMOVE(X1,Y1)
            .
            .
            .
```

Here INTT is a function which returns a value of true if its argument is an INTEGER which would be valid as an integer argument in one of the dual argument type System Plot Package routines, and false otherwise. It contains the following code:—

```
      FUNCTION INTT(IVAL)
      LOGICAL INTT
      DATA MASK/8#17777760000#/
C
      IF(IVAL.LE.0) THEN
            INTT=.FALSE.
      ELSE
            INTT=IAND(MASK,IVAL).EQ.0
      END IF
      RETURN
      END
```

The operation of INTT relies on the fact that an integer argument in the range 1 to 32768 (the range of valid integer arguments to the plot package routines) will have zero's in all bit positions except the 15 least significant. However, all floating point variables must have at least one non-zero bit in the most significant 16 bits since all exponents are always positive, being biased by 127. (Actually, there is one case where this is not true — a denormalized number (a non-zero floating point number whose exponent is the minimum for the format i.e. -127 for R*4 variables, and whose bit pattern is also all zeros) — but the likelihood of this happening in practice is negligible).

The subroutine TRANS converts the integer arguments from plotter units to user ("world") coordinates. The EQUIVALENCE statements is necessary to allow TRANS to be called always with integer arguments.

The alternate arguments MX and MY apply to the X and Y coordinates of the following routines:—

```
FRSTPT       CURVE
VECTOR       PWRIT
        LINE
```

5.6 Support Routines

In addition to the System Plot Package routines listed above, an additional 12 support routines needed to be implemented. These routines provide a set of inherently non-standard functions and include all machine dependent constants [3]. The routines are as follows:—

ENCD Creates labels for plots from floating point arguments.

GETCHR Extracts a specified character from a string.

IAND Calculates the bit by bit logical sum of two integer arguments.

INTT Detremines whether an argument is integer or real (see above).

IOR Calculates the bit by bit logical product of two integer arguments.

ISHIFT Shifts the bits in an integer argument; either a left circular shift or a right end-off shift may be specified.

PERROR Provides a graceful error exit following a fatal error in the System Plot Package.

R1MACH Sets 5 single-precision machine constants.

SETCHR Takes a right-justified, zero-filled character string and inserts it into a particular position in a second character string, replacing its original contents.

SETER Emulates the NCAR system error output routine.

TRANS Converts integer plotter coordinates to real user coordinates.

ULIBER Prints error messages.

References

[1] G. R. McArthur
An Introduction to the SCD Graphics System.
NCAR Technical Note NCAR–TN/161+1A, May 1983.

[2] G. R. McArthur
The System Plot Package. NCAR Technical Note NCAR–TN/162+1A, May 1983.

[3] G. R. McArthur
The Graphics System Implementors Guide. NCAR Technical Note NCAR–TN/165+1A, May 1983.

[4] G. R. McArthur
The SCD Graphics Utilities. NCAR Technical Note NCAR–TN/166+1A, May 1983.

[5] J. M. Lopez
Axisymmetric Vortex Breakdown Part 1. — Confined Swirling Flow, ARL Aerodynamics Report 173, January 1988.

APPENDICES

A Package CONRECSUPER

A.1 Purpose

CONRECSUPER draws a contour map from data stored in a rectangular array, labeling the lines. This is the so-called "super" version, which smooths contour lines and removes crowded lines.

A.2 Usage

The call

```
CALL EZCNTR (Z,M,N)
```

may be used if the following assumptions are met:—

- All of the array is to be contoured.
- Contour levels are chosen internally.
- Contouring routine chooses scale factors.
- Highs and lows are marked.
- Negative contour lines are drawn with a dashed line pattern.
- EZCNTR calls FRAME after drawing the contour map.

If these assumptions are not met, use

```
CALL CONREC (Z,L,M,N,FLO,HI,FINC,NSET,NHI,NDOT)
```

A.3 Algorithm

Each line is followed to completion, and points along each line are found on boundaries of the (rectangular) cells. Then points along each line are connected by straight lines using the software dashed line package DASHSUPER, which also labels the lines. A model of the plotting plane is maintained and if a line to be drawn would overlap previously drawn lines, it is omitted.

A.4 Arguments for Subroutine EZCNTR

On Input:—

Z(M,N) The M by N array to be contoured.

M The first dimension of Z.

N The second dimension of Z.

On Output:—

All arguments are unchanged.

A.5 Arguments for Subroutine CONREC

On Input:—

Z The (origin of the) array to be contoured. Z is L by N.

L The *first dimension* of Z *in the calling program*.

M The number of data values to be contoured in the x-direction (the first subscript direction). When plotting an entire array, L = M. See Section 5 for an explanation of using this argument list to process any part of an array.

N The number of data values to be contoured in the y-direction (the second subscript direction).

FLO The value of the lowest contour level. If FLO = HI = 0., a value rounded up from the minimum Z is generated by CONREC.

HI The value of the highest contour level. If HI = FLO = 0., a value rounded down from the maximum Z is generated by CONREC.

FINC If FINC > 0, the increment between contour levels is FINC. If FINC = 0, a value which produces between *10* and *30 contour levels at nice values is generated* by CONREC. If FINC < 0, the number of levels generated is ABS(FINC).

NSET Flag to control scaling. If NSET = 0, CONREC calls SET to properly scale the plotting instructions to the standard configuration. PERIM is called and puts tick marks corresponding to the data points. If NSET > 0, CONREC assumes that SET has been called by the user in such a way as to properly scale the plotting instructions generated by CONREC. PERIM is not called. If NSET < 0, CONREC calls SET in such a way as to place the (untransformed) contour plot within the limits of the user's last SET call. PERIM is not called.

NHI Flag to control extra information on the contour plot. If NHI = 0, highs and lows are marked with an H or an L as appropriate, and the value of the high or low is plotted under the symbol. If NHI > 0, the values in each Z are plotted at each Z point, with the center indicating the data point location. If NHI < 0, neither of the above are done.

NDOT A 10-bit constant designating the desired dashed line pattern. If NDOT = 0, 1, or 1777_8, solid lines are drawn. If NDOT > 0, NDOT is used as the pattern for all contour levels. If NDOT < 0, IABS(NDOT) is used for negative valued contour levels.

On Output:—

All arguments are unchanged.

A.6 Supplementary Information

The most common modifications are:—

- Non-uniform contour levels. See comments in CLGEN source code.
- Transformations of contour lines. See comments in Section 5.
- Internal parameters such as label sizes, unknown data points, drawing placement, etc. Access is via common blocks as listed in the table below.

COMMON BLOCK CONRE1		
Parameter Name	Default Value	Function
IOFFP	0	Flag to control special value feature. IOFFP = 0 means special value feature not in use. IOFFP non-zero means special value feature in use. SPVAL is set to the special value. Contour lines will then be omitted from any cell with any corner equal to the special value.
SPVAL	0.	Contains the special value when IOFFP is non-zero.
IOFFM	0	Flag to control the message below the plot. IOFFM = 0 means message is plotted. IOFFM non-zero means the message is not plotted.
ISOLID	1777_8	Dash pattern for non-negative lines.
IHILO	3	Flag to control labelling of highs, lows, or both. If NHI = 0, then — IHILO = 0 means neither highs nor lows labelled, IHILO = 1 means highs are labelled, lows are not, IHILO = 2 means lows are labelled, highs are not, IHILO = 3 means that both highs and lows labelled

COMMON BLOCK CONRE2		
Parameter Name	Default Value	Function
SIZEL	1.5	Size of line labels.
SIZEM	2.5	Size of labels for minimums and maximums.
SIZEP	1.0	Size of labels for data point values.
NLA	16	Approximate number of contour levels when internally generated.
NLM	40	Maximum number of contour levels. If this is to be increased, the dimensions of CL and IWORK must be increased by the same amount in CONREC.
XLT	0.05	Left hand edge of the plot (0.0 = left edge of frame, 1.0 = right edge of frame).
YBT	0.05	Bottom edge of the plot (0.0 = bottom of frame, 1.0 = top of frame).
SIDE	0.9	Length of longer edge of plot (see also EXT).
NREP	6	Number of repetitions of the dash pattern between line labels.
NCRT	4	Number of crt units per element (bit) in the dash pattern.
ILAB	1	Flag to control the drawing of line labels. ILAB non-zero means label, ILAB = 0 means do not label the lines.
NULBLL	3	Number of unlabeled lines between labeled lines. For example, when NULBLL = 3, every fourth level is labeled.
IOFFD	0	Flag to control normalization of label numbers. IOFFD = 0 means include decimal point when possible (do not normalize unless required). IOFFD non-zero means normalize all label numbers and output a scale factor in the message below the graph.
EXT	0.25	Lengths of the sides of the plot are proportional to M and N (when CONREC calls SET) except in extreme cases, namely, when the MIN(M,N)/MAX(M,N) is less than EXT. Then CONREC produces a square plot.

B Package CONRANSUPER

B.1 Purpose

CONRANSUPER performs contouring of irregularly distributed data. CONRANSUPER will plot contours and smooth them (if smoothing is requested), and will produce a perimeter or grid, a title, a message about map variables, contour line labels, and additionally will eliminate crowding of contour lines. CONRANSUPER will produce publication-quality plots.

B.2 Usage

```
CALL CONRAS (XD,YD,ZD,NDP,WK,IWK,SCRARR)
```

`FRAME` must be called by the user.

Option-setting routines can be invoked, as discussed in the "Options" section below.

B.3 Algorithm

The sparse data is triangulated and a virtual grid is laid over the triangulated area. Each virtual grid point receives an interpolated value. The grid is scanned once for each contour level and all contours at that level are plotted.

There are two methods of interpolation. The first is a smooth data interpolation scheme based on Lawson's C1 surface interpolation, refined by Hirosha Akima. The second is a linear interpolation scheme.

When data is sparse it is usually better to use the C1 interpolation. If you have dense (over 100 points) then the linear interpolation will give better results.

B.4 Bibiliography

Akima, Hirosha *A method of bivariate interpolation and smooth surface fitting for irregularly distributed data points.*
ACM Transactions on Mathematical Software, Vol 4, no. 2, June 1978, pages 148-159.

Lawson, C.L. *Software for C1 surface interpolation.*
JPL Publication 77-30, August 15, 1977.

B.5 Arguments

On input:—

`XD` Array of dimension `NDP` containing the x-coordinates of the data points.

`YD` Array of dimension `NDP` containing the y-coordinates of the data points.

`ZD` Array of dimension `NDP` containing the data values at the points.

NDP Number of data points. Note that the number of input points must be at least 4. This is equal to the default number of data points to be used for estimation of partial derivatives at each data point. NCP contains the number of these points to be used. The estimated partial derivatives are used for the construction of the interpolating polynomial's coefficients.

WK Real work array of dimension at least 13*NDP.

IWK Integer work array. When using Lawson C1 surfaces the array must be at least IWK((27+NCP)*NDP). NCP = 4 by default. When using linear interpolation the array must be at least IWK((27+4)*NDP).

SCRARR Real work array of dimension at least (RESOLUTION**2) where RESOLUTION is described in the SSZ option below. The default value of RESOLUTION is 40.

On output:—

All arguments remain unchanged except the scratch arrays IWK, WK, and SCRARR. If making multiple runs on the same triangulation, IWK and WK must be saved and returned to the next invocation of CONRAS.

B.6 Options

Plot enhancement options are implemented via calls to the options subroutines CONOP?. Calling a CONOP? routine prior to the CONRAS call can activate or deactivate the options. Each option command must be invoked with separate CONOP calls. There are 4 CONOP? routines: CONOP1, CONOP2, CONOP3, and CONOP4. The numerical value in the name determines the number of parameters in the calling sequence. Arguments to these routines are of the form (Option Name) = (Option Value), with the whole argument a character or Hollerith constant. Only the first two characters on each side of the equal sign are scanned. Therefore only 2 characters for each option are required on input to CONOP? (i.e. 7HSCA=PRI, 'SCA=PRI', 5HSC=PR or 'SC=PR' would be accepted for setting the scaling option to the prior mode).

Note: Option settings remain in effect between calls to any CONRAS entry point.

The following table contains the option names, their purpose and possible values:—

Option	Purpose and Usage
CHL	This flag determines how the high and low contour values are set. These contour values may be set by the program or by the user. If CHL=OFF, the program examines the user's input data and uses both the high and low values. If CHL=ON, the user must specify the desired high (HI) and low (FLO) value should he wish to use different contour values from the input data. The default is CHL=OFF. If program set: CALL CONOP1(7HCHL=OFF) If user set: CALL CONOP3(6HCHL=ON,HI,FLO) Example: CALL CONOP3(6HCHL=ON,5020.,2000.) where the high value desired is 5020. and the low value desired is 2000. Note: The values supplied for contour increment and contour high and low values assumes the unscaled data values. See SDC below.
CIL	This flag determines how the contour increment (CINC) is set. The increment is either calculated by the program (CIL=OFF) using the range of high and low values from the user's input data, or set by the user (CIL=ON). If program set: CALL CONOP1(7HCIL=OFF) If user set:CALL CONOP2(6HCIL=ON,CINC) Example: CALL CONOP2(6HCIL=ON,15.) where 15. represents the contour increment desired by the user. Note: Default option is CIL=OFF. By default the program will examine the user's input data and determine the contour interval (CINC) at some appropriate range between the level of high and low values supplied, usually generating between 15 and 20 contour levels.
CON	This flag determines how the contour levels are set. If CON=ON, the user must specify the number of contour levels (NCL), and an array (ARRAY) containing the values. A maximum of 30 contour levels are permitted. If CON=OFF, default are used. In this case, the program will calculate the values for ARRAY and NCL using input data. If program set: CALL CONOP1(7HCON=OFF) If user set: CALL CONOP3(6HCON=ON,ARRAY,NCL) Example: DATA RLIST(1),...,RLIST(5)/1.,2.,3.,10 CALL CONOP3(6HCON=ON,RLIST,5) where RLIST contains the user specified levels,and 5 is the number of user specified contour levels. Warning on Contour Options: It is illegal to use the CON option if either CIL or CHL are activated. In this error condition the option call that detected the error will not be executed. When assigning the contour array, it must be ordered from smallest to largest.

Option	Purpose and Usage
DAS	This flag determines which contours are represented by dashed lines. The user sets the dashed line pattern. The user may specify that dashed lines be used for contours whose value is less than, equal to, or greater than the dash pattern breakpoint (see BP in the DBP option below), which is zero by default. If DAS=OFF (the default value), all solid lines are used. All solid lines: CALL CONOP1(7HDAS=OFF) If greater: CALL CONOP2(7HDAS=GTR,IPAT) If equal: CALL CONOP2(7HDAS=EQU,IPAT) If less: CALL CONOP2(7HDAS=LSS,IPAT) If all same: CALL CONOP2(7HDAS=ALL,IPAT) Note: IPAT must be a ten character hollerith string with a dollar sign ($) for dash and a single quote (') for blank. This is not the repetition count for line labels. That is controlled by an internal parameter, NREP, found in common block CONR11. Example: CALL CONOP2(7HDAS=GTR,10H$$$$$$$$$$)
DBP	This flag determines how the dash pattern break point (BP) is set. If DBP=ON, BP must be set by the user by specifying BP. If DBP=OFF the program will set BP at the default value which is zero. If program set: CALL CONOP1(7HDBP=OFF) If user set: CALL CONOP2(6HDBP=ON,BP) Example: CALL CONOP2(6HDBP=ON,5.) where 5. is the user specfied break point. Note: BP is a floating point number where the break for GTR and LSS contour dash patterns are defined. BP is assumed to be given relative to the untransformed contours.
DEF	Reset flags to default values. Activating this option sets all flags to the default value. DEF has no "on" or "off" states. To activate: CALL CONOP1(3HDEF)
EXT	Flag to set extrapolation. To turn on: CALL CONOP1(6HEXT=ON) To turn off: CALL CONOP1(7HEXT=OFF) Note: Normally CONRAN will only plot the boundaries of the convex hull defined by the user's data to fill the rectangular area of the frame.

Option	Purpose and Usage
FMT	Flag for the format of the plotted input data values. If FMT=OFF, the default values for FT, L, and IF are used. Default values are:— FT = (G10.3) L = 7 characters including the parentheses IF = 10 characters printed in the output field by the format If FMT=ON, the user must specify values for FT, L, and IF. All user specified values must be given in the correct format. If program set: CALL CONOP1(7HFMT=OFF) If user set: CALL CONOP4(6HFMT=ON,FT,L,IF) Example: CALL CONOP4(6HFMT=ON,(G30.2),7,30) Note: FT is a Hollerith string containing the format. The format must be enclosed by parenthesis. Any format allowed at your installation (up to 10 characters) will be accepted. L is the number of characters in FT. IF is the length of the field created by the format. Warning: CONRAS will not test for a valid format and the format is only allowed to be 10 characters long.
GRI	Flag for the grid. To turn on: CALL CONOP1(6HGRI=ON) To turn off: CALL CONOP1(7HGRI=OFF) Note: If the flag is on, the x and y tick interval will be given. This is the interval in user coordinates that each tick mark represents.
INT	Flag to determine the intensities of the contour lines and other parts of the plot. If INT=OFF, IVAL is the default value. If INT=ON, the user must specify IVAL. If program set: CALL CONOP1(7HINT=OFF) Major lines: CALL CONOP2(7HINT=MAJ,IVAL) Minor lines: CALL CONOP2(7HINT=MIN,IVAL) Title and message: CALL CONOP2(7HINT=LAB,IVAL) Data values: CALL CONOP2(7HINT=DAT,IVAL) Note: This relates to the plotted data values and the plotted maximums and minimums. All the same: CALL CONOP2(7HINT=ALL,IVAL) Example: CALL CONOP2(7HINT=ALL,110) Note: IVAL is the intensity desired. For an explanation of the option value settings see the OPTN routine in the NCAR System Plot Package documentation. Briefly, IVAL values range from 0 to 255 or the character strings 2HLO and 2HHI. The default is 2HHI except for INT=MIN which is set to 2HLO.
ITP	Set the interpolation scheme: C1 or linear. The C1 method takes longer but will give the best results when the data is sparse (less than 100 points). The linear method will produce a better plot when there is a dense data set. The default is C1 surface. For C1 surface: CALL CONOP1(6HITP=C1) For linear: CALL CONOP1(7HITP=LIN)

Option	Purpose and Usage
LAB	This flag can be set to either label the contours (LAB=ON) or not (LAB=OFF). The default value is LAB=ON. To turn on: CALL CONOP1(6HLAB=ON) To turn off: CALL CONOP1(7HLAB=OFF)
LOT	Flag to list options on the printer. The default value is set to off, and no options will be displayed. To turn on: CALL CONOP1(6HLOT=ON) To turn off: CALL CONOP1(7HLOT=OFF) Note: If users want to print the option values, they should turn this option on. The option values will be sent to the standard output unit as defined by the port routine I1MACH.
LSZ	This flag determines the label size. If LSZ=OFF, the default ISIZELSZ value will be used. If LSZ=ON, the user should specify ISIZELSZ. The default value is 9 PWRIT units. If program set: CALL CONOP1(7HLSZ=OFF) If user set: CALL CONOP2(6HLSZ=ON,ISIZELSZ) Example: CALL CONOP2(6HLSZ=ON,4) where 4 is the user desired integer PWRIT units. Note: ISIZELSZ is the requested character size in integer PWRIT units.
MES	Flag to plot a message. To turn on: CALL CONOP1(6HMES=ON) To turn off: CALL CONOP1(7HMES=OFF) Note: If MES=ON a message is printed below the plot giving contour intervals and execution time in seconds. If PER or GRI are on, the message also contains the x and y tick interval.
NCP	Flag for data points used for partial derivative estimation. If NCP=OFF, NUM is set to 4, which is the default value. If NCP=ON, the user must specify NUM greater than or equal to 2. If program set: CALL CONOP1(7HNCP=OFF) If user set: CALL CONOP2(6HNCP=ON,NUM) Example: CALL CONOP2(6HNCP=ON,3) Note: NUM = number of data points used for estimation. Changing this value effects the contours produced and the size of input array IWK.
PDV	Flag to plot the input data values. Default value is PDV=OFF. To turn on: CALL CONOP1(6HPDV=ON) To turn off: CALL CONOP1(7HPDV=OFF) Note: If PDV=ON, the input data values are plotted relative to their location on the contour map. If you only wish to see the locations and not the values, set PDV=ON and change FMT to produce an asterisk (*) such as (I1).

Option	Purpose and Usage
PER	Flag to set the perimeter. The default value is PER=ON, which causes a perimeter to be drawn around the contour plot. To turn on: CALL CONOP1(6HPER=ON) To turn off: CALL CONOP1(7HLPER=OFF) Note: If MES is on, the x and y tick interval will be given. This is the interval in user coordinates that each tick mark represents.
PMM	Flag to plot relative minimums and maximums. To turn off: CALL CONOP1(7HPMM=OFF) To turn on: CALL CONOP1(6HPMM=ON)
PSL	Flag which sets the plot shield option. The outline of the shield will be drawn on the same frame as the contour plot. By default this option is off. Draw the shield: CALL CONOP1(6HPSL=ON) Don't draw shield: CALL CONOP1(7HPSL=OFF)
REP	Flag indicating the use of the same data, but a new execution. Default value is set to off. To turn on: CALL CONOP1(6HREP=ON) To turn off: CALL CONOP1(7HREP=OFF) Note: If REP=ON, the same data and triangulation are to be used but it is assumed the user has changed contour values or resolution for this run. Scratch arrays WK and IWK must remain unchanged.
SCA	Flag for scaling of the plot on a frame. User scaling: CALL CONOP1(7HSCA=OFF) Program scaling: CALL CONOP1(6HSCA=ON) Prior SET call: CALL CONOP1(7HSCA=PRI) Note: With SCA=OFF, plotting instructions will be issued using the user's input coordinates, unless they are transformed via FX and FY transformations. Users will find an extended discussion in the "Interfacing to Other Graphics Routines" Section below. The SCA option assumes the user has previously used a SET call, and that all input data falls into the range of that SET call. With SCA=ON, the entry point will perform the SET call so that the user's plot will fit into the center 90% of the frame. When SCA=PRI, the program maps the user's plot instructions into the portion of the frame defined by the last SET call.

Option	Purpose and Usage
SDC	Flag to determine how to scale the data on the contours. If SDC=OFF, the floating point value is given by SCALE. If SDC=ON, the user may specify scale (see note below.) If program set: CALL CONOP1(7HSDC=OFF) If user set: CALL CONOP2(6HSDC=ON,SCALE) Example: CALL CONOP2(6HSCD=ON,100.) Note: The data plotted on contour lines and the data plotted for relative minimums and maximums will be scaled by the floating point value given by scale. Typical scale values are 10., 100., 1000., etc. The original data values are multiplied by SCALE. It must be a floating point number and is displayed in the message (see MES). The default value for scale is 1.
SLD	Activate or deactivate the shielding option. This option, when activated will display contours only within the region specified by shield given by the user. The shield must be given in the same coordinate range as the data and must define only one polygon. The polygon is described by x,y pairs that sequentially define it. To activate the shield: CALL CONOP4(6HSLD=ON,XSD,YSD,ICSD) where XSD is the x-array of shield coords YSD is the y-array of shield coords ICSD is the number of shield coords To deactivate the shield: CALL CONOP1(7HSLD=OFF)
SML	Flag to determine the size of minimum and maximum labels. If SML=OFF, the value of ISIZESML is 15, which is the default. If SML=ON, the user must specify ISIZESML. If program set: CALL CONOP1(7HSML=OFF) If user set: CALL CONOP1(6HSML=ON,ISIZESML) Example: CALL CONOP1(6HSML=ON,12) Note: ISIZESML is an integer number which is the size of labels in PWRIT units.
SPD	Flag for the size of the plotted input data values. If SPD=OFF, the value of ISIZESPD is 8, which is the default. If SPD=ON, the user must specify ISIZESPD. If program set: CALL CONOP1(7HSPD=OFF) If user set: CALL CONOP2(6HSPD=ON,ISIZESPD) Example: CALL CONOP2(6HSPD=ON,6) Note: ISIZESPD is an integer number giving the size of the data values in PWRIT units.

Option	Purpose and Usage
SSZ	Flag to determine the resolution (number of steps in each direction). If SSZ=ON, the user sets ISTEP, or, if SSZ=OFF, the program will automatically set ISTEP at the default value of 40. If program set: CALL CONOP1(7HSSZ=OFF) If user set: CALL CONOP2(6HSSZ=ON,ISTEP) Example: CALL CONOP2(6HSSZ=ON,25) This ISTEP value will produce a coarse contour. See note below. Note: ISTEP is an integer. It is the density of the virtual grid. Usually, the default value of 40 produces pleasing contours. For coarser but quicker contours, lower the value. For smoother contours, raise the value. Note: For step sizes greater than 200, the array OV in common CONRA1, and ITLOC in common CONRA9 must be expanded to about 10 more than the size of SSZ. See CONRA1 and CONRA9 comments below for more information.
STL	Flag to determine the size of the title. ISIZESTL may be set by the user (STL=ON), or the program will set it at the default size of 16 PWRIT units (STL=OFF). If program set: CALL CONOP1(STL=OFF) If user set: CALL CONOP2(STL=ON,ISIZESTL) Example: CALL CONOP2(STL=ON,13) Note: When 30 to 40 characters are used for the title, the default size of 16 PWRIT units works well. When titles are larger, a smaller title size is required.
TEN	Flag to determine the tension factor applied when smoothing contour lines. The user may set TENS or allow the program to set the value. If user set, TENS must have a value greater than zero and less than or equal to 30. The default value is 2.5. If program set: CALL CONOP1(7HTEN=OFF) If user set: CALL CONOP2(6HTEN=ON,TENS) Example: CALL CONOP2(6HTEN=ON,14.) Smoothing of contour lines is accomplished with splines under tension. To adjust the amount of smoothing applied, adjust the tension factor. The tension must be greater than zero. The default is 2.5. Setting TENS very large (i.e. 30), effectively shuts off smoothing.
TFR	Flag to advance frame before triangulation. The default value is TFR=ON. If program set: CALL CONOP1(6HTFR=ON) To turn off: CALL CONOP1(7HTFR=OFF) Note: Triangles are plotted after the contouring is completed. If the user wished to see the triangles over the contours, turn this switch off.

Option	Purpose and Usage
TLE	Flag to place a title at the top of the plot. If TLE=ON, the user must specify ICHARS and INUM. The default value is off. ICHARS is the Hollerith character string for the title. INUM is the number of characters in ICHARS. To turn on: CALL CONOP3(6HTLE=ON,ICHARS,INUM) Example: CALL CONOP3(6HTLE=ON,13HVECTOR REVIEW,13) To turn off: CALL CONOP1(7HTLE=OFF) Note: The title array is 32 words long, so the maximum title is 32 times the number of characters per word on your machine. Longer titles will require increasing the size of array ISTRNG found in CONRA7.
TOP	Flag to plot triangles only. To turn off: CALL CONOP1(7HTOP=OFF) To turn on: CALL CONOP1(6HTOP=ON) Note: The user may wish to overlay the triangles on some other plot. This option on will allow that. When activated this option will set TRI=ON and TFR=OFF. If the user wants TFR=ON, it should be set after TOP is set. If the user sets TOP=OFF it will set TRI=OFF and TFR=ON. If the user wants TRI and TFR different, then set them after the TOP call.
TRI	Flag to plot the triangulation. To turn on: CALL CONOP1(6HTRI=ON) To turn off: CALL CONOP1(7HTRI=OFF) Note: Plotting the triangles will indicate to the user where good and bad points of interpolation are occuring in the contour map. Equilateral triangles are optimal for interpolation. Quality degrades as triangles approach a long and narrow shape. The convex hull of the triangulation is also a poor point of interpolation.

B.7 Option Default Values

Listed below are the program set default values for the various options given above. Unless the user specifies otherwise, these values will be used in execution of the various options.

CHL=OFF	LOT=OFF	SLD=OFF
CIL=OFF	LSZ=OFF	SML=OFF
CON=OFF	MES=ON	SPD=OFF
DAS=OFF	NCP=OFF	SPT=OFF
DBP=OFF	PDV=OFF	SSZ=OFF
EXT=OFF	PER=ON	STL=OFF
FMT=OFF	PMM=OFF	TEN=OFF
GRI=OFF	REP=OFF	TFR=ON
ITP=C1	SCA=ON	TOP=OFF
LAB=ON	SDC=OFF	TRI=OFF

The option default values given above, if user specified, will set default values for the parameters in the following table:—

Parameter	Default Values
ARRAY	Up to 30 contour levels allowed. Values are computed by the program, on input.
BP	0.0
CINC	Computed by program based on the range of high and low values of the input data.
FLO	Computed by program based on the lowest unscaled input data.
FT	(G10.3) Parentheses must be included.
HI	Computed by program based on the highest unscaled input data.
ICHARS	No title
IF	10 characters.
INUM	No title.
IPAT	10H$$$$$$$$$$
ISIZELSZ	9 PWRIT units.
ISIZESML	15 PWRIT units.
ISIZESPD	8 PWRIT units.
ISIZESTL	16 PWRIT units.
ISTEP	40
IVAL	2HHI for all except minor contour lines which are 2HLO.
L	7 characters including both parentheses.
NCL	Computed by program based on input data. Up to 30 contour levels are permitted.
NUM	4 data points.
SCALE	1 (No scaling performed).
TENS	2.5
XSD,YSD	Used by the program via loc so there is no limit at present.
ICSD	0 — No shield.

B.8 Options Which Affect Contours

To create different styled contours use options NCP and SSZ. NCP will modify the interpolating functions and therefore cause changes in the plots produced. Increasing NCP causes more of the surrounding data to influence the point of interploation. This is useful when triangles are long and narrow). By modifying NCP and thereby changing the interpolation functions, a user can fine-tune a plot. Increasing SSZ will smooth out the contour lines and pick up more detail (new contours will appear as SSZ increases and old ones will sometimes break into more distinct units).

The C1 interpolation method is recommended when the data is sparse. It will smooth the data and add trends (false hills and valleys). The linear method is superior when data is dense (> 50 to 100). It will not smooth the data or add trends.

B.9 Interfacing with Other NCAR Utilities

Statement functions, FX and FY can be modified to perform other mappings. Users are referred to Appendix B of the NCAR *SCD Graphics Utilities* Technical Note for a complete description of FX and FY transformation. The routines having FX and FY transformations are:—

CONDRW, CONPDV, CONTLK, CONPMM, CONGEN

In most cases mapping can be performed before calling a CONRAS entry point, saving the user from modifying the source file. If reasonable results cannot be obtained, then the statement functions will have to be replaced.

Note:— If NCP > 25, arrays DSQO and IPCO in CONDET must be adjusted accordingly. Also NCPSZ in CONBDN (25 by default), will have to equal NCP. The default value of NCP, which is 4, produces pleasing pictures in most cases. However, fine tuning of the interpolation can be obtained by increasing the size of NCP, with a corresponding linear increase in work space. CONRAS calls SETI to guarantee 1-1024 integer range for PWRIT.

B.10 Error Messages

The following table includes all possible error messages which can occur in CONRAS. All errors are considered to be recoverable.

Error	Routine	Message
1	CONRAS	Input parameter NDP < NCP.
2	CONRAS	NCP greater than its maximum size or < 2.
3	CONTNG	All colinear data points.
4	CONTNG	Identical input data points.
5	CONSET	Undefined option.
6	CONCLS	Constant input field.
7	CONSET	Incorrect CONOP call used.
8	CONSET	Illegal use of CON option with CIL or CHL options.
9	CONSET	Number of contour levels greater than 30.
10	CONDRW	Contour storage exhausted. Error is trapped and nullified by CONRAS, but contour levels may be incomplete.
11	CONSTP	Aspect ratio of x and y greater than 5 to 1. This error may cause a poor plot. Usually this can be fixed by multiplying x or y by a constant factor. If this solution is not successful, then increasing SSZ to a very large value may help (Note: This can be expensive).

B.11 Common Blocks

The common blocks listed in the following tables include all the common used by the entire CONRANSUPER Utility.

COMMON BLOCK CONRA1	
CL	Array of contour levels.
NCL	Number of contour levels.
OLDZ	z value of left neighbor to current location.
PV	Array of previous row values.
HI	Largest contour plotted.
FLO	Lowest contour plotted.
FINC	Increment level between equally spaced contours.

COMMON BLOCK CONRA2	
REPEAT	Flag to triangulate and draw or just draw.
EXTRAP	Plot data outside of convex data hull.
PER	Put perimeter arround plot.
MESS	Flag to indicate message output.
ISCALE	Scaling switch.
LOOK	Plot triangles flag.
PLDVLS	Plot the data values flag.
GRD	Plot grid flag.
CON	User set or program set contours flag.
CINC	User or program set increment flag.
CHILO	User or program set hi low contours.
LABON	Flag to control labeling of contours.
PMIMX	Flag to control the plotting of min's max's.
SCALE	The scale factor for contour line values. Min, max plotted values.
FRADV	Advance frame before plotting triangualtion.
EXTRI	Only plot triangulation.
BPSIZ	Breakpoint size for dash patterns.
LISTOP	List options on UNIT6 flag.

COMMON BLOCK CONRA3	
IREC	Recoverable error flag.

COMMON BLOCK CONRA4	
NCP	Number of data points used at each point for polynomial construction.
NCPSZ	Max size allowed for NCP.

COMMON BLOCK CONRA5	
NIT	Flag to indicate status of search data base.
ITIPV	Last triangle interpolation occurred in.

COMMON BLOCK CONRA6	
XST	x coordinate start point for contouring.
YST	y coordinate start point for contouring.
XED	x coordinate end point for contouring.
YED	y coordinate end point for contouring.
STPSZ	Step size for x,y change when contouring.
IGRAD	Number of graduations for contouring (step size).
IG	Reset value for IGRAD.
XRG	x range of coordinates.
YRG	y range of coordinates.
BORD	Percent of frame used for contour plot.
PXST	x plotter start address for contours.
PYST	y plotter start address for contours.
PXED	x plotter end address for contours.
PYED	y plotter end address for contours.
ITICK	Number of tick marks for grids and perimeters.

COMMON BLOCK CONRA7	
TITLE	Switch to indicate if title option on or off.
ISTRNG	Character string of title.
ICNT	Character count of ISTRNG.
ITLSIZ	Size of title in PWRIT units.

COMMON BLOCK CONRA8	
IHIGH	Default intensity setting.
INMAJ	Contour level intensity for major lines.
INMIN	Contour level intensity for minor lines.
INLAB	Title and message intensity.
INDAT	Data value intensity.
FORM	The format for plotting the data values.
LEN	The number of characters in the format.
IFMT	Size of the format field.
LEND	Default format length.
IFMTD	Default format field size.
ISIZEP	Size of the plotted data values.

COMMON BLOCK CONRA9	
X	Array of x coordinates of contours drawn at current contour level.
Y	Array of y coordinates of contours drawn at current contour level.
NP	Count in X and Y.
MXXY	Size of X and Y.
TR	Top right corner value of current cell.
BR	Bottom right corner value of current cell.
TL	Top left corner value of current cell.
BL	Bottom left corner value of current cell.
CONV	Current contour value.
XN	x position where contour is being drawn.
YN	y position where contour is being drawn.
ITLL	Triangle where top left corner of current cell lies.
IBLL	Triangle of bottom left corner.
ITRL	Triangle of top right corner.
IBRL	Triangle of bottom left corner.
XC	x coordinate of current cell.
YC	y coordinate of current cell.
ITLOC	In conjunction with PV stores the triangle where PV value came from.

COMMON BLOCK CONR10	
NT	Number of triangles generated.
NL	Number of line segments.
NTNL	NT+NL.
JWIPT	Pointer into IWK where where triangle point numbers are stored.
JWIWL	In IWK the location of a scratch space.
JWIWP	In IWK the location of a scratch space.
JWIPL	In IWK the location of end points for border line segments.
IPR	In WK the location of the partial derivatives at each data point.
ITPV	The triangle where the previous value came from.

COMMON BLOCK CONR11	
NREP	Number of repetitions of dash pattern before a label.
NCRT	Number of crt units for a dash mark or blank.
ISIZEL	Size of contour line labels.
NDASH	Array containing the negative contour dash pattern value.
MINGAP	Number of unlabeled lines between each labeled one.
IDASH	Positive valued contour dash pattern.
ISIZEM	Size of plotted minimums and maximums.
EDASH	Equal valued contour dash pattern.
TENS	Default tension setting for smoothing.

COMMON BLOCK CONR12	
IXMAX,IYMAX	Maxinum x and y coordinates relative to the scratch array, SCRARR.
XMAX,YMAX	Maximum x and y coordinates relative to users coordinate space.

COMMON BLOCK CONR13	
XVS	Array of the x coordinates for shielding.
YVS	Array of the y coordinates for shielding.
IXVST	Pointer (via LOC) to the users x array for shielding.
IYVST	Pointer (via LOC) to the users y array for shielding.
ICOUNT	Count of the shield elements.
SPVAL	Special value used to halt contouring at the shield boundary.
SHIELD	Logical flag to signal status of shielding.
SLDPLT	Logical flag.

C Package ISOSURF

C.1 Purpose

To draw an approximation of an iso-valued surface from a three-dimensional array with hidden lines removed. See package ISOSURFHR for a high-resolution version of package ISOSURF.

C.2 Usage

This package contains two user entries — ISOSRF and EZISOS. The call

```
CALL EZISOS (T,MU,MV,MW,EYE,SLAB,TISO)
```

may be used if the following assumptions are met:—

- The entire three-dimensional input array is to be used.
- Iso-lines, visible lines, and boundary treatment of the surface to be drawn are internally chosen rather than user-specified.
- Routine FRAME is called internally.

If these assumptions are not met use

```
CALL ISOSRF (T,LU,MU,LV,MV,MW,EYE,MUVWP2,SLAB,TISO,IFLAG)
```

C.3 Algorithm

Cuts through the three-dimensional array are contoured with a smoothing contourer which also marks a model of the plotting plane. Interiors of boundaries are filled in and the result is or'ed into another model of the plotting plane which is used to test subsequent contour lines for visibility.

The hidden-line algorithm is not exact, so visibility errors can occur.

C.4 Arguments for Subroutine ISOSRF

On input:—

T(LU,LV,MW) Three-dimensional array of data that is used to determine the iso-valued surface.

LU First dimension of T in the calling program.

MU The number of data values of T to be processed in the u-direction (the first subscript direction). When processing the entire array, LU = MU (and LV = MV). See Section 5 for an explanation of using this argument list to process any part of an array.

LV Second dimension of T in the calling program.

MV The number of data values of T to be processed in the v-direction (the second subscript direction).

MW The number of data values of T to be processed in the w-direction (the third subscript direction).

EYE(3) The position of the eye in three-space. T is considered to be in a box with opposite corners (1,1,1) and (MU,MV,MW). The eye is at (EYE(1),EYE(2),EYE(3)), which must be outside the box that T is in. While gaining experience with the routine, a good choice for eye might be (5.0*MU,3.5*MV,2.0*MW).

MUVWP2 The maximum of (MU,MV,MW)+2 i.e. MUVWP2 = MAX0(MU,MV,MW)+2.

SLAB(MUVWP2,MUVWP2) A work space used for internal storage. SLAB must be at least MUVWP2*MUVWP2 words long.

TISO The iso-value used to define the surface; the surface drawn will separate volumes of T that have value greater than TISO from volumes of T that have value less than TISO.

IFLAG A flag which serves two purposes. First, the absolute value of IFLAG determines which types of lines are drawn to approximate the surface. Three types of lines are considered: lines of constant u, lines of constant v and lines of constant w. The following table lists the types of lines drawn.

IABS(IFLAG)	Lines of Constant		
	u	v	w
1	no	no	yes
2	no	yes	no
3	no	yes	yes
4	yes	no	no
5	yes	no	yes
6	yes	yes	no
0, 7 or more	yes	yes	yes

Second, the sign of IFLAG determines what is inside and what is outside, hence, which lines are visible and what is done at the boundary of T.

IFLAG	Action
positive	T values greater than TISO are assumed to be inside the solid formed by the drawn surface.
negative	T values less than TISO are assumed to be inside the solid formed by the drawn surface.

If the algorithm draws a cube, reverse the sign of IFLAG.

On Output:—

Only array SLAB has been overwritten. Its contents are not needed.

C.5 Arguments of the Subroutine EZISOS

On Input:—

T, MU, MV, MW, EYE, SLAB, and TISO have the same descriptions as for the routine ISOSRF above.

On Output:—

Only array SLAB has been overwritten. Its contents are not needed.

C.6 Supplementary Information

IREF is set internally via a DATA statement. It is a flag to control the drawing of axes. For IREF non-zero, axes are drawn. For IREF zero, axes are not drawn. The current value of IREF is 1.0.

D Package SURFACE

D.1 Purpose

SURFACE draws a perspective picture of a function of two variables with hidden lines removed. The function is approximated by a two-dimensional array of heights.

D.2 Usage

This package contains two user entries, SRFACE and EZSRFC. The call

```
CALL EZSRFC (Z,M,N,ANGH,ANGV,WORK)
```

may be used if the following assumptions are met:—

- The entire array is to be drawn.
- The data is equally spaced (in the x-y plane).
- No stereo pairs.
- Scaling is chosen internally.

If these assumptions are not met use

```
CALL SRFACE (X,Y,Z,M,MX,NX,NY,S,STEREO)
```

D.3 Algorithm

The data are processed from the near side of the surface towards the far side. Visibility information is stored. Highest so far is visible from above.

The algorithm is based on the work of Wright, T.J., "A two space solution to the hidden line problem for plotting a function of two variables". IEEE Trans. Comp., pp 28-33, January 1973.

D.4 Arguments of Subroutine SRFACE

On Input:—

X(NX) A linear array NX long containing the x coordinates of the points in the surface approximation. See special conditions below.

Y(NY) The linear array NY long containing the y coordinates of the points in the surface approximation. See special conditions below.

Z(MX,NY) An array MX by NY containing the surface to be drawn in NX by NY cells. Z(I,J) = F(X(I),Y(J)). See special conditions below.

M(2,NX,NY) Scratch array at least 2*NX*NY words long.

MX First dimension of Z.

NX Number of points in the x-direction in Z. When plotting an entire array, MX=NX. See Section 5 for an explanation of using this argument list to process any part of an array.

NY Number of points in the y-direction in Z.

S(6) S defines the line of sight. The viewers eye is at (S(1),S(2),S(3)) and the point looked at is at (S(4),S(5),S(6)). The eye should be outside the block with opposite corners (X(1),Y(1),ZMIN) and (X(NX),Y(NY),ZMAX) and the point looked at should be inside it For a nice perspective effect, the distance between the eye and the point looked at should be 5 to 10 times the size of the block. See special conditions below.

STEREO Flag to indicate if stereo pairs are to be drawn. Zero means no stereo pair (one picture). Non-zero means put out two pictures. The value of STEREO is the relative angle between the eyes. A value of 1.0 produces standard separation. Negative STEREO reverses the left and right figures.

On Output:—

X, Y, Z, MX, NX, NY, S, STEREO are unchanged. M has been written in.

D.4.1 Special Conditions

- The range of Z compared with the range of X and Y determines the shape of the picture. They are assumed to be in the same units and not wildly different in magnitude. S is assumed to be in the same units as X, Y, and Z.
- Picture size can be made relative to distance. See comments in SETR.
- TRN32S can be used to translate from three- to two-space. See comments there.
- Data with extreme discontinuities may cause visibility errors. If this problem occurs, use a distant eye position away from the +z axis.

D.5 Arguments for Subroutine EZSRFC

On Input:—

Z(M,N) The M by N array to be drawn.

M The first dimension of Z.

N The second dimension of Z.

ANGH Angle in degrees in the x-y plane to the line of sight (counter-clock wise from the plus-x axis).

ANGV Angle in degrees from the x-y plane to the line of sight (positive angles are above the middle Z, negative below).

WORK(2*M*N+M+N) A scratch storage dimensioned at least 2*M*N+M+N.

On Output:—

Z, M, N, ANGH, ANGV are unchanged. WORK been written in.

D.6 Supplementary Information

The values of many internal parameters which control the detailed format of the output are accessible via the Common Block SFRIP1. The parameter names, default values and descriptions of the varibles in this Common Block are listed in the following table.

COMMON BLOCK SFRIP1		
Parameter Name	Default Value	Function
IFR	1	−1 Call FRAME first. 0 Do not call FRAME. +1 Call FRAME when done.
ISTP	0	Stereo type if STEREO non-zero. −1 Alternating frames, slightly offset (for movies, IROTS = 0). 0 Blank frame between (for stereo slide IROTS = 1). +1 Both on same frame, (left picture to left side. IROTS = 0).
IROTS	0	0 +z in vertical plotting direction (cine mode). +1 +z in horizontal plotting direction (comic mode).
IDRX	1	+1 Draw lines of constant x. 0 Do not.
IDRY	1	+1 Draw lines of constant y. 0 Do not.
IDRZ	0	+1 Draw lines of constant z (contour lines). 0 Do not.

COMMON BLOCK SFRIP1 (CONTINUED)		
Parameter Name	Default Value	Function
IUPPER	0	+1 Draw upper side of surface. 0 Draw both sides. −1 Draw lower side.
ISKIRT	0	+1 Draw a skirt around the surface. Bottom = HSKIRT. 0 Do not.
NCLA	6	Approximate number of levels of constant z that are drawn if levels are not specified. 40 levels maximum.
THETA	0.02	Angle in radians between eyes for stereo pairs.
HSKIRT	0.0	Height of skirt (If ISKIRT = 1).
CHI	0.0	Highest level of constant z.
CLO	0.0	Lowest level of constant z.
CINC	0.0	Increment between levels. (If CHI, CLO, or CINC is zero, a nice value is generated automatically).

E Package VELVECT

E.1 Purpose

To graphically represent a two-dimensional velocity field by drawing arrows from each data location, with the length of the arrow proportional to the strength of the field at that location and the direction of the arrow indicating the direction of the flow at that location.

E.2 Usage

This package contains two user entry points — EZVEC and VELVCT. The call

```
CALL EZVEC (U,V,M,N)
```

may be used if the following assumptions are met:—

- The whole array is processed.
- The scale factor is chosen internally.
- The perimeter is drawn.
- FRAME is called after plotting.
- There are no special values.

If these assumptions are not met, use

```
CALL VELVCT (U,LU,V,LV,M,N,FLO,HI,NSET,LENGTH,ISPV,SPV)
```

E.3 Algorithm

Each vector is examined, possibly transformed, and then plotted as follows.

The endpoints of each arrow drawn are

```
FX(X,Y),FY(X,Y))
and
(MXF(X,Y,U,V,SFX,SFY,MX,MY),MYF(X,Y,U,V,SFX,SFY,MX,MY))
```

where X = I, Y = J, U = U(I,J), V = V(I,J), and SFX and SFY are scale factors. Here I is the X index and J is the Y index. (MX,MY) is the location of the tail. Thus the actual length of the arrow is SQRT(DX**2+DY**2) and the direction is ATAN2(DY,DX), where DX=MX-MXF(...) and DY=MY-MYF(...).

E.4 Arguments for Subroutine VELVCT

On Input:—

U(LU,N) Two-dimensional array containing the first component of the velocity field to be plotted.

LU The first dimension of U in the calling program.

V(LV,N) Two-dimensional array containing the second component of the velocity field to be plotted. The velocity field vector at (I,J) has magnitude SQRT(U(I,J)**2+V(I,J)**2) and direction ATAN2(V(I,J),U(I,J)). Other representations (such as (r,θ)) can be plotted by changing statement functions in this routine.

LV The first dimension of V in the calling program.

M The number of data values to be plotted in the x-direction (the first subscript direction). When plotting the entire array, LU = LV = M.

N The number of data values to be plotted in the y-direction (the second subscript direction).

FLO The minimum vector magnitude to be shown.

HI The maximum vector magnitude. (A value less than or equal to zero causes the maximum value of SQRT(U**2+V**2) to be used.)

NSET Flag to control scaling —

If NSET is zero, VELVCT calls SET to properly scale plotting instructions to the standard configuration. PERIM is called to draw a border.

If NSET is greater than zero, VELVCT assumes that SET has been called by the user in such a way as to properly scale the plotting instructions generated by VELVCT. PERIM not called.

If NSET is less than zero, VELVCT calls SET in such a way as to place the contour plot within the limits of the user's last SET call. PERIM is not called.

LENGTH The length, in crt units, of a vector having magnitude HI (or, if HI = 0, the length in crt units of the longest vector). If LENGTH = 0, a value is chosen such that the longest vector could just reach to the tail of the next vector. If the horizontal and vertical resolutions of the plotter are different, LENGTH should be non-zero and specified as a horizontal distance.

ISPV Flag to control the special value feature.

ISPV zero means that the feature is not in use.

ISPV = 1 means that if the value of U(I,J) = SPV(1) the vector will not be plotted.

ISPV = 2 means that if the value of V(I,J) = SPV(2) the vector will not be plotted.

IPSV = 3 means that if either U(I,J) = SPV(1) or V(I,J) = SPV(2) then the vector will not be plotted.

ISPV = 4 is identical to ISPV = 3 with SPV(2) = SPV(1).

SPV(2) Array of real values, of dimension two, whose usage is described in the ISVP description immediately above.

On Output:—

All arguments remain unchanged.

E.5 Arguments for Subroutine EZVEC

On Input:—

`U(M,N)` Two-dimensional array containing the first component of the velocity field to be plotted.

`V(M,N)` Two-dimensional array containing the second component of the velocity field to be plotted. The velocity field vector at `(I,J)` has magnitude `SQRT(U(I,J)**2+V(I,J)**2)` and direction `ATAN2(V(I,J),U(I,J))`. Other representations (such as (r,θ)) can be plotted by changing statement functions in this routine.

`M` The number of data values to be plotted in the x-direction (the first subscript direction).

`N` The number of data values to be plotted in the y-direction (the second subscript direction).

On Output:—

All arguments remain unchanged.

E.6 Using VELVCT with the Mapping Packege EZMAP

Using this routine to put vectors on an arbitrary background drawn by EZMAP is a bit tricky. The arithmetic statement functions `FX` and `FY` are easy to replace — the problem arises in replacing `MXF` and `MYF` — the following example may be helpful.

Suppose that we have two arrays — `CLON(36,9)` and `CLAT(36,9)` — which contain the E–W and N–S components of a wind flow field on the surface of the earth. `CLON(I,J)` is the magnitude of the easterly flow and `CLAT(I,J)` the magnitude of the northerly flow at a longitude `(I-1)*10` degrees east of Greenwich and a latitude `(J-1)*10` degrees north of the equator. Subroutine `SUPMAP` from EZMAP is to be used to draw a polar projection of the earth and `VELVCT` is to be used to superimpose vectors representing the flow field on it. the following steps would be necessary.

1. `CALL SUPMAP (1,90.,0.,-90.,90.,90.,90.,90.,-4,10,0,1,IER)` to draw the map.

2. `CALL VELVCT (CLON,36,CLAT,36,36,9,0.,0.,1,50,0,0.)` to put vectors on it. Notice that `NSET` has the value 1 to tell `VELVCT` that `SUPMAP` has done the required `SET` call.

3. In order to ensure that step 2 will work properly, delete the arithmetic statement functions `FX`, `FY`, `MXF`, and `MYF` from `VELVCT` and include the following functions:—

```
      FUNCTION FX(XX,YY)
      CALL SUPCON (10.*(YY-1.),10.*(XX-1.),X,Y)
      FX=X
      RETURN
      END

      FUNCTION FY(XX,YY)
      CALL SUPCON (10.*(YY-1.),10.*(XX-1.),X,Y)
      FY=Y
      RETURN
      END

      FUNCTION MXF(XX,YY,UU,VV,SFX,SFY,MX,MY)
      CFCT=COS(.17453292519943*(YY-1.))
      CALL SUPCON(10.*(YY-1.),10.*(XX-1.),X1,Y1)
      CALL SUPCON(10.*(YY-1.)+1.E-6*VV,10.*(XX-1.)+1.E-6*UU/CFCT,X2,Y2)
      U=((X2-X1)/SQRT((X2-X1)**2+(Y2-Y1)**2))*SQRT(UU**2+VV**2)
      MXF=MX+IFIX(SFX*U)
      RETURN
      END

      FUNCTION MYF(XX,YY,UU,VV,SFX,SFY,MX,MY)
      CFCT=COS(.17453292519943*(YY-1.))
      CALL SUPCON(10.*(YY-1.),10.*(XX-1.),X1,Y1)
      CALL SUPCON(10.*(YY-1.)+1.E-6*VV,10.*(XX-1.)+1.E-6*UU/CFCT,X2,Y2)
      V=((Y2-Y1)/SQRT((X2-X1)**2+(Y2-Y1)**2))*SQRT(UU**2+VV**2)
      MYF=MY+IFIX(SFY*V)
      RETURN
      END
```

The basic notion behind the coding of the MXF and MYF functions is that, since UU and VV are the longitudinal and latitudinal components, respectively, of a velocity vector having units of distance over time, 1.E-6*UU/COS(LATITUDE) and 1.E-6*VV represent the change in longitude and latitude, respectively, of a particle moving with the flow field for a very short period of time. The routine SUPCON is used to find the position of the particle's projection at the beginning and end of that tiny time slice and, therefore, the direction in which to draw the arrow representing the velocity vector so that it will be tangent to a projected flow line of the field at that point. The values U and V are computed so as to give the arrow the length implied by UU and VV — that is, the code ensures that SQRT(U**2+V**2) is equal to SQRT(UU**2+VV**2). The length of the arrow represents the magnitude of the velocity vector, unaffected by perspective — the scaling set up by VELVCT will therefore be appropriate for the arrows drawn.

This method is rather heuristic and has three inherent problems. First, the constant 1.E-6 may need to be made larger or smaller, depending on the magnitude of your U/V data. Second, the north and south poles must be avoided — at either pole, CFCT goes to zero, giving a division by zero — in a small region near the pole, the method may try to use SUPCON with a latitude outside the range (−90,+90). Third, the projection must be set up so as to avoid having vector basepoints at the exact edge of the map — vectors there will be of the correct length, but they may be drawn in the wrong direction (when the projected particle track determining the direction crosses the edge and reappears elsewhere on the map). With a little care, the desired results may be obtained.

F Package STREAMLINE

F.1 Purpose

STREAMLN draws a streamline representation of the flow field. The representation is independent of the flow speed.

F.2 Usage

This package contains two user entries — STRMLN and EZSTRM. The call

```
CALL EZSTRM (U,V,WORK,IFLG,IMAX,JMAX)
```

may be used if the following assumptions are met:—

- The whole array is to be processed.
- The arrays are dimensioned U(IMAX,JMAX), V(IMAX,JMAX) WORK(2*IMAX*JMAX), and IFLG(IMAX,JMAX).
- Routines SET and PERIM are to be called automatically

If these assumptions are not met, use

```
CALL STRMLN (U,V,WORK,IFLG,IMAX,IPTSX,JPTSY,NSET,IER)
```

Note that the user must call FRAME in the calling routine.

F.3 Algorithm

Wind components are normalized to the value of DISPL. Judicious use of flags prevents overcrowding of streamlines and directional arrows. Experience indicates that a final pleasing picture is produced when streamlines are initiated in the center of a grid box. The streamlines are drawn in one direction then in the opposite direction.

The techniques utilized here are described in an article by Thomas Whittaker (University of Wisconsin) which appeared in the notes and correspondence section of Monthly Weather Review, June 1977.

F.4 Arguments for Subroutine STRMLN

On Input:—

U(IMAX,JPTSY),V(IMAX,JPTSY) The two dimensional arrays containing the velocity fields to be plotted. Note: If the u- and v-components are, for example, defined in Cartesian coordinates and the user wishes to plot them on, for example, a polar stereographic projection, then an appropriate transformation must be made to the u- and v-components via the functions FU and FV located in DRWSTR — see the NCAR Graphics Manual for more information on coordinate transformations.

WORK User provided WORK array. The dimension of this array must be greater than or equal to 2*IMAX*JPTSY. Caution: This routine does not check the size of the work array.

IFLG User provided INTEGER*1 scratch array. The dimensions of this array must be greater than or equal to (IMAX,JPTSY).

IMAX The first dimension of U and V in the calling program. (x-direction)

IPTSX The number of points to be plotted in the first subscript direction. (x-direction)

JPTSY The number of points to be plotted in the second subscript direction. (y-direction)

Note: See the Section 5 on using these arguments to plot a portion of an array.

NSET Flag to control scaling. If NSET > 0, STRMLN assumes that SET has been called by the user in such a way as to properly scale the plotting instructions generated by STRMLN. PERIM is not called. If NSET = 0, STRMLN calls SET to properly scale the plotting instructions to the standard configuration. PERIM is called to draw the border. If NSET < 0, STRMLN calls SET in such a way as to place the streamlines within the limits of the users last set call. PERIM is not called.

On Output:—

Only the IER argument and WORK array are changed.

IER If IER = 0, no errors are detected. When the routine is called with ICYC $\neq$ 0, and the data are not cyclic, then IER = -1 (ICYC is an internal parameter and is described below). In this case the routine will draw the streamlines with the non-cyclic interpolation formulas.

F.5 Arguments for Subroutine EZSTRM

On Input:—

U(IMAX,JMAX),V(IMAX,JMAX) The two dimensional arrays containing the velocity fields to be plotted. Note: If the u- and v-components are, for example, defined in Cartesian coordinates and the user wishes to plot them on, for example, a polar stereographic projection, then an appropriate transformation must be made to the u- and v-components via the functions FU and FV located in DRWSTR— see the NCAR Graphics Manual for more information on coordinate transformations.

WORK User provided WORK array. The dimension of this array must be greater than or equal to 2*IMAX*JMAX. Caution: This routine does not check the size of the work array.

IFLG User provided INTEGER*1 scratch array. The dimensions of this array must be greater than or equal to (IMAX,JMAX).

IMAX The first dimension of U and V in the calling program. (x-direction)

JMAX The second dimension of U and V in the calling program. (y-direction)

On Output:—

Only the WORK array is changed.

F.6 Supplementary Information

The values of many internal parameters which control label and arrowhead sizes, drawing placement, etc. are accessable via two Common Blocks. The names, default values and descriptions of the variables contained in these Common Blocks are listed in the following tables.

COMMON BLOCK STR02		
Parameter Name	Default Value	Function
CRTX,CRTY	1024	Maximum number of plotter points in the x- and y-directions.
EXT	0.25	Lengths of the sides of the plot are proportional to IPTSX and JPTSY except in the case when MIN(IPTSX,JPTSY)/MAX(IPTSX,JPTSY) < EXT in which case a square graph is plotted.
SIDE	0.90	Length of the longer edge of the plot. (See also EXT).
XLT	0.05	Left hand edge of the plot. (0.0 = left edge of frame, 1.0 = right edge of frame).
YBT	0.05	Bottom edge of the plot. (0.0 = bottom , 1.0 = top). (YBT+SIDE and XLT+SIDE must be less than or equal to 1.).

COMMON BLOCK STR03		
Parameter Name	Default Value	Function
INITA	2	Used to pre-condition grid boxes to be eligible to start a streamline. (e.g., a value of four means that every fourth grid box is eligible ; a value of 2 means yhat every other grid box is eligible.
INITB	2	Used to pre-condition grid boxes to be eligible for direction arrows. If the user changes the default values of INITA and/or INIT it should be done such that MOD(INITA,INITB). If the user has a reasonable dense grid try INITA = 4 , INITB = 2. It will cut cpu time down.
AROWL	0.33	Length of directional arrow — 0.33, for example, means each directional arrow will up a third of a grid box.
ITERP	35	Every ITERP iterations the streamline progress is checked.
ITERC	-99	The default value of this parameter is such that it it has no effect on the code. When set to some positive value the program will check for streamline crossover every ITERC iterations. (The routine currently does this every time it enters a new grid box). Caution: When this parameter is activated, cpu time will increase.
IGFLG	0	A value of zero means that the sixteen point Bessel interpolation formula will utilized where possible. When near the grid edges quadratic blinear interpolation will used. This mixing of interpolation scheme can cause slight raggedness near the edges. This, however, does not occur very often. If IGFLG is non-zero, then the bilinear interpolation formula (only) is to be use this will result in slightly faster plot times. However, in general the plots will be less pleasing.

COMMON BLOCK STR03 (CONTINUED)		
Parameter Name	Default Value	Function
IMSG	0	If zero, then no missing u- and v-components are present. If non-zero, then STRMLN will utilize the bi-linear interpolation scheme and if any of the points are missing the the streamline will be terminated.
UVMSG	1.E+36	Value assigned to a missing point.
ICYC	0	Zero means the data are non-cyclic in the x-direction. If non-zero, then the cyclic interpolation formulas will be used. Note: Even if the data are cyclic in x, leaving ICYC = 0 will do no harm.
DISPL	0.33	The wind speed is normalized to this value. (See below for a discussion of the DISPL, DISPC and CSTOP parameters.)
DISPC	0.67	The critical displacement. If after ITERP iterations the streamline has not moved this distance, then the streamline will be terminated.
CSTOP	0.50	This parameter controls the spacing between streamlines. The checking is done when a new grid box is entered.

F.6.1 Discussion of DISPL, DISPC and CSTOP

Assume a value of 0.33 for DISPL. This means that it will take three steps to move across one grid box if the flow was all in x-direction. If the flow is zonal, then a larger value of DISPL is in order. If, however, the flow is highly turbulent then a smaller value is in order. Note: The smaller DISPL, the more the computer time. A value of 2 to 4 times DISPL is a reasonable value for DISPC. DISPC should always be greater than DISPL. A value of 0.33 for CSTOP would mean that a maximum of three streamlines will be drawn per grid box. This maximum will normally only occur in areas of singular points.

G Package EZMAP

G.1 Purpose

To plot maps of the earth according to any one of ten different projections, showing continental, international, and/or U.S. state outlines, parallels, and meridians. The origin and orientation of the projection are selected by the user. Points on the earth defined by latitude and longitude are mapped to points in the plane of projection — the u/v plane. The u and v axes are parallel to the x and y axes of the plotter, respectively. A rectangular frame whose sides are parallel to the u and v axes is chosen and material within that frame (or an inscribed elliptical frame) is plotted.

G.2 Usage

The routine MAPDRW draws a complete map, as directed by the current values of parameters in the EZMAP package. To change the values of those parameters, and thus the appearance of the map, one may first call one of the routines MAPROJ (to change the projection to be used), MAPSET (to change what portion of the u/v plane is to be viewed), MAPPOS (to change what portion of the plotter frame is to be used), or one of the parameter-setting routines MAPSTC, MAPSTI, MAPSTL, and MAPSTR (to change various other parameters, of types character, integer, logical, and real). The parameter-retrieval routines MAPGTC, MAPGTI, MAPGTL, and MAPGTR allow the user to retrieve the values of EZMAP parameters.

The routine MAPSAV allows one to save the current state of EZMAP, the routine MAPRST to restore a saved state.

Users with special needs may wish to call the lower-level routines MAPINT (to initialize the package — it must be called initially and again whenever certain parameters are changed), MAPGRD (to draw parallels and meridians), MAPLBL (to label the international date line, the equator, the Greenwich meridian, and the poles, and to draw the perimeter), and MAPLOT (to draw the selected geographic outlines). These routines are normally called by MAPDRW.

Intensities of various map portions may be set by calls to the routine MAPSTI. The routine MAPUSR is called by EZMAP just before/after drawing various portions of the map; the default version, which does nothing, may be replaced by a user version which sets/restores color, spot size, intensity, dash pattern, etc.

The routine MAPEOS is called by EZMAP once for each outline segment. The user may supply a version which examines the segment to see if it ought to be plotted and, if not, to delete it. This may be used, for example, to reduce the clutter in northern Canada.

To overlay objects of one's own on the map drawn by MAPDRW, one may use one or more of the routines MAPTRN (to compute the u/v coordinates of a point, given its latitude and longitude), MAPIT (to do "pen-up/down" moves), MAPFST (to do "pen-up" moves), and MAPVEC (to do "pen-down" moves).

The routine SUPMAP, from which EZMAP g. ·v, is implemented within it and allows one to draw a complete map with a single, rather lengthy, call.

G.3 Error Conditions

When an error occurs during a call to an EZMAP routine, an error message is logged, using the NCAR error routine SETERR; the program is then aborted.

Possible error flags are as follows:—

Number	Routine	Meaning
1	MAPGTC	Unknown parameter name xx
2	MAPGTI	Unknown parameter name xx
3	MAPGTL	Unknown parameter name xx
4	MAPGTR	Unknown parameter name xx
5	MAPINT	Attempt to use non-existent projection
6	MAPINT	Angular limits too great
7	MAPINT	Map has zero area
8	MAPINT	Map limits inappropiate
9	MAPROJ	Unknown projection name xx
10	MAPSET	Unknown map area specifier xx
11	MAPSTC	Unknown outline name xx
12	MAPSTC	Unknown parameter name xx
13	MAPSTI	Unknown parameter name xx
14	MAPSTL	Unknown parameter name xx
15	MAPSTR	Unknown parameter name xx
16	MAPTRN	Attempt to use non-existent projection
17	MAPIO	Outline dataset is unreadable
18	MAPIO	EOF encountered in outline dataset
19	MAPPOS	Arguments are incorrect
20	MAPRST	Error on read
21	MAPRST	EOF on read
22	MAPSAV	Error on write

G.4 Bibliography

1. Hershey, A.V., *The Plotting of Maps on a CRT Printer.* NWL Report no. 1844, 1963.

2. Lee, Tso-Hwa, *Students' Summary Reports, Work-Study Program in Scientific Computing.* NCAR, 1968.

3. Parker, R.L., *2UCSD SUPERMAP: World Plotting Package.*

4. Steers, J.A., *An Introduction to the Study of Map Projections.* University of London Press, 1962.

G.5 Accuracy

The definition of the map produced is limited by two factors: the resolution of the outline data and the resolution of the graphics device.

Data points in the continental outlines are about one degree apart and the coordinates are accurate to .01 degree. Data points in U.S. state outlines are about .05 degrees apart and the coordinates are accurate to .001 degree. Both the spacing and the accuracy ot the international boundaries falls somewhere between these two extremes.

G.6 Subroutine MAPDRW

G.6.1 Purpose

To draw the complete map described by the current values of the EZMAP parameters.

`MAPDRW` calls `MAPINT` (if required), `MAPGRD`, `MAPLBL`, and `MAPLCT`, in that order. The user may wish to call these routines directly.

G.6.2 Usage

```
CALL MAPDRW
```

G.6.3 Arguments

None.

G.7 Subroutine MAPEOS

G.7.1 Purpose

`MAPEOS` is called by EZMAP to examine each segment in the outline datasets. The default version does nothing. A user-supplied version may cause selected segments to be deleted (to reduce the clutter in northern Canada, for example).

G.7.2 Usage (by EZMAP)

```
CALL MAPEOS (NOUT,NSEG,IGID,NPTS,PNTS)
```

G.7.3 Arguments

`NOUT` is the number of the outline dataset from which the segment comes, as follows:—

NOUT	Dataset to which segment belongs
1	'CO' - Continental outlines only.
2	'US' - U.S state outlines only.
3	'PS' - Continental, U.S state, and outlines.
4	'PO' - Continental and international outlines.

NSEG is the number of the segment within the outline dataset.

IGID identifies the group to which the segment belongs, as follows:—

IGID	Group to which segment belongs
1	Continental outlines.
2	U.S. state boundaries.
3	International boundaries.

NPTS is the number of points defining the outline segment. NPTS may be zeroed to suppress plotting of the segment.

PNTS is an array of coordinates. PNTS(1) and PNTS(2) are the latitude and longitude of the first point, PNTS(3) and PNTS(4) the latitude and longitude of the second point, ...PNTS(2*NPTS-1) and PNTS(2*NPTS) the latitude and longitude of the last point.

G.8 Subroutine MAPFST

G.8.1 Purpose

To draw lines on the map produced by a call to MAPDRW — used in conjunction with MAPVEC.

G.8.2 Usage

```
CALL MAPFST (RLAT,RLON)
```

This call is exactly equivalent to the call

```
CALL MAPIT (RLAT,RLON,0)
```

G.8.3 Arguments

RLAT and RLON are defined as for MAPIT. See the description of MAPIT.

G.9 Subroutine MAPGRD

G.9.1 Purpose

To draw a grid made up of lines of latitude and longitude. If EZMAP needs initialization or if the error flag 'ER' is non-zero, MAPGRD does nothing.

G.9.2 Usage

```
CALL MAPGRD
```

G.9.3 Arguments

None.

G.10 Subroutine MAPGTx

G.10.1 Purpose

To get the values of EZMAP parameters.

G.10.2 Usage

```
CALL MAPGTC (WHCH,CVAL)
CALL MAPGTI (WHCH,IVAL)
CALL MAPGTL (WHCH,LVAL)
CALL MAPGTR (WHCH,RVAL)
```

G.10.3 Arguments

WHCH is a character string specifying the parameter to get.

CVAL,IVAL,LVAL, or RVAL is a variable to receive the value of the parameter specified by WHCH — of type character, integer, logical, or real, respectively.

All of the parameters listed in the discussion of MAPSTx may be retrieved. The following may also be retrieved:—

WHCH	Type	Meaning
ARea	C	The value of the map limits specifier JLTS from the last call to MAPSET. The default value is 'MA'.
ERror	I	The current value of the error flag. Default is zero.
INitialize	I,L	Initialization flag. If true (non-zero), EZMAP is in need of initialization (by means of a CALL MAPINT). The default value is true (non-zero).

WHCH	Type	Meaning
PRojection	C	The value of the projection specifier JPRJ from the last call to MAPROJ. The default value is 'CE'.
PN	I,R	The value of PLON from the last call to MAPROJ. The default value is zero.
PT	I,R	The value of PLAT from the last call to MAPROJ. The default value is zero.
Pn	I,R	"n" is an integer between 1 and 8. Retrieves values from the last call to MAPSET. P1 through P4 get you PLM1(1), PLM2(1), PLM3(1), and PLM4(1), while P5 through P8 get you PLM1(2), PLM2(2), PLM3(2), and PLM4(2). The default values are all zero.
ROtation	I,R	The value of ROTA from the last call to MAPROJ. The default value is zero.
XLeft	R	The parameters XLOW, XROW,
XRight	R	YBOW, and YTOW from the last
YBottom	R	call to MAPPOS. Defaults
YTop	R	are .05, .95, .05, and .95.

G.11 Subroutine MAPINT

G.11.1 Purpose

To initialize the package after the values of some parameters have been changed. The flag 'IN', which may be retrieved by a call to MAPGTI or MAPGTL, indicates whether or not initialization is required at a given time. (Some parameters may be reset at any time and do not require MAPINT to be called again.)

G.11.2 Usage

```
CALL MAPINT
```

G.11.3 Arguments

None.

G.12 Subroutine MAPIT

G.12.1 Purpose

To draw lines on the map produced by a call to MAPDRW. MAPIT attempts to omit non-visible portions and to handle "cross-over" — a jump from one end of the map to the other caused by the projection's having slit the globe along some half of a great circle and laid it open with the two sides of the slit at opposite ends of the map. Cross-over can occur on cylindrical and conical projections; MAPIT handles it very well on the former and not so well on the latter.

The EZMAP parameter 'DL' determines whether MAPIT draws solid lines or dotted lines. The parameters 'DD' and 'MV' also affect MAPIT's behavior. See the description of the routine MAPSTx, below.

A sequence of calls to MAPIT should be followed by a call to MAPIQ (which see, above) to flush its buffers.

Points in two contiguous pen-down calls to MAPIT should not be far apart on the globe.

G.12.2 Usage

```
CALL MAPIT (RLAT,RLON,IFST)
```

G.12.3 Arguments

RLAT and RLON are the latitude and longitude of a point to which the "pen" is to be moved. Both are given in degrees. RLAT must be between -90. and +90., inclusive; RLON must be between -540. and +540., inclusive.

IFST is 0 to do a "pen-up" move, 1 to do a "pen-down" move if the distance from the last point to the new point is greater than 'MV' plotter units, 2 or greater to do the move regardless of the distance from the last point to the new one.

G.13 Subroutine MAPIQ

G.13.1 Purpose

To flush MAPIT's buffers. This is particularly important before a STOP or a CALL FRAME and before changing intensity, dash pattern, color, etc.

G.13.2 Usage

```
CALL MAPIQ
```

G.13.3 Arguments

None.

G.14 Subroutine MAPLBL

G.14.1 Purpose

To label the International Date Line (ID), the equator (EQ), the Greenwich Meridian (GM), and the poles (NP and SP), and to draw the border around the map. If EZMAP needs initialization or if the error flag 'ER' is set, MAPLBL does nothing.

G.14.2 Usage

```
CALL MAPLBL
```

G.14.3 Arguments

None.

G.15 Subroutine MAPLOT

G.15.1 Purpose

To draw the continental and/or international and/or U.S. state outlines selected by the parameter 'OU'. If EZMAP currently needs initialization or if the error flag 'ER' is set, MAPLOT does nothing.

G.15.2 Usage

```
CALL MAPLOT
```

G.15.3 Arguments

None.

G.16 Subroutine MAPPOS

G.16.1 Purpose

To specify the position of the map on the plotter frame.

G.16.2 Usage

```
CALL MAPPOS (XLOW,XROW,YBOW,YTOW)
```

G.16.3 Arguments

The arguments are fractions between 0.0 and 1.0 determining the position of a window in the plotter frame within which the map is to be drawn. XLOW and XROW position the left and right edges and are stated as fractions of the distance from left to right in the plotter frame. YBOW and YTOW position the bottom and top edges and are stated as fractions of the distance from bottom to top in the plotter frame. The map is centered in the specified window and made as large as possible while maintaining its proper shape.

The default values of the internal parameters changed by this routine are .05, .95, .05, and .95, respectively.

G.17 Subroutine MAPROJ

G.17.1 Purpose

To specify the projection to be used.

G.17.2 Usage

CALL MAPROJ (JPRJ,PLAT,PLON,ROTA)

G.17.3 Arguments

JPRJ is a character variable defining the desired projection type, as follows:—

- The conic projection:—
  - 'LC' — Lambert conformal conic with two standard parallels.
- The azimuthal projections:—
  - 'ST' — Stereographic.
  - 'OR' — Orthographic. Causes the parameter 'SA' (which see, in the description of the routine MAPSTx) to be zeroed.
  - 'LE' — Lambert equal area.
  - 'GN' — Gnomonic.
  - 'AE' — Azimuthal equidistant.
  - 'SV' — Satellite-view. If the parameter 'SA' (which see, in the description of the routine MAPSTx) is greater than 1 or less than -1, it is left alone; otherwise, it is given the value 6.631.
- The cylindrical projections:—
  - 'CE' — Cylindrical equidistant.
  - 'ME' — Mercator.
  - 'MO' — Mollweide. The projection used is not actually a true Mollweide.

PLAT, PLON, ROTA are reals specifying the values of angular quantities, in degrees. How they are used depends on the value of JPRJ, as follows:— If JPRJ is not equal to 'LC': PLAT and PLON define the latitude and longitude of the pole of the projection — the point on the globe which is to be projected to the origin of the u/v plane. PLAT must be between -90. and +90., inclusive, positive in the northern hemisphere, negative in the southern. PLON must be between -180. and +180., inclusive, positive to the east, and negative to the west, of Greenwich. ROTA is the angle between the v axis and north at the origin. It is taken to be positive if the angular movement from north to the v axis is counter-clockwise, negative otherwise. If the origin is at the north pole, "north" is considered to be in the direction of PLON+180. If the origin is at the south pole, "north" is considered to be in the direction of PLON. For the cylindrical projections, the axis of the projection is parallel to the v axis.

If JPRJ is equal to 'LC' (Lambert conformal conic with two standard parallels): PLON defines the central meridian of the projection, while PLAT and ROTA define the two standard parallels. If PLAT and ROTA are equal, a conic projection with one standard parallel is used.

More detailed descriptions of the projections may be found in the NCAR Graphics Manual, together with helpful diagrams, but a few words may be helpful here:—

The conical projection maps the surface of the earth onto the surface of a cone intersecting the earth along the two standard parallels. The cone is then slit along a line opposite the central meridian and opened up (with some stretching) onto a flat surface.

The azimuthal projections map the surface of the earth (or of one hemisphere of the earth) onto a plane whose origin is tangent to it at the point (PLAT,PLON). The several azimuthal projections differ only in the function used to map the great-circle distance of a point from the pole (PLAT,PLON) to a linear distance of the projected point from the origin (0,0). The projected image may be rotated using the parameter ROTA.

The cylindrical projections map the surface of the earth onto a cylinder which is tangent to it along a great circle passing through the point (PLAT,PLON) at an angle determined by ROTA. The cylinder is then slit along its length through the point opposite (PLAT,PLON) and opened up onto the plane. The several cylindrical projections differ principally in the function used to map the distance from the great circle of tangency to a distance along the cylinder. If PLAT is zero and ROTA is either 0.0 or 180.0, the cylindrical projections are particularly simple to do and a faster path through the code is used.

G.18 Subroutine MAPRS

G.18.1 Purpose

Recalls SET. Intended to be used when data is to be plotted over a map generated in a different overlay, and when the system plot package does not reside in an outer overlay. Not necessary in the ARL Elxsi implementation.

G.18.2 Usage

```
CALL MAPRS
```

G.18.3 Arguments

None.

G.19 Subroutine MAPRST

G.19.1 Purpose

Restores a saved state of EZMAP. This is done by reading saved parameter values from a user unit and then calling MAPINT. See MAPSAV.

G.19.2 Usage

```
CALL MAPRST (IFNO)
```

G.19.3 Arguments

IFNO is the number of a unit from which a single unformatted record is to be read. It is the user's responsibility to open a file on this unit and to position the file. MAPRST does not rewind it, either before or after reading the record.

G.20 Subroutine MAPSAV

G.20.1 Purpose

Saves the current state of EZMAP by writing parameter values onto a user unit. See MAPRST.

G.20.2 Usage

```
CALL MAPSAV (IFNO)
```

G.20.3 Arguments

IFNO is the number of a unit to which a single unformatted record is to be written. It is the user's responsibility to open this unit. MAPSAV does not rewind it, either before or after writing the record.

G.21 Subroutine MAPSET

G.21.1 Purpose

To specify the rectangular portion of the u/v plane to be drawn.

G.21.2 Usage

```
CALL MAPSET (JLTS,PLM1,PLM2,PLM3,PLM4)
```

G.21.3 Arguments

JLTS can have the following character values. It specifies one of five ways in which the limits of the map are defined by the parameters PLM1, PLM2, PLM3, and PLM4.

JLTS='MA' (MAXIMUM) The maximum useful area produced by the projection is plotted. PLM1, PLM2, PLM3, and PLM4 are not used.

JLTS='CO' (CORNERS) The points (PLM1,PLM2) and (PLM3,PLM4) are to be at opposite corners of the map. PLM1 and PLM3 are latitudes, in degrees. PLM2 and PLM4 are longitudes, in degrees. If a cylindrical projection is being used, the first point should be on the left edge of the map and the second point on the right edge; otherwise, the order makes no difference.

JLTS='PO' (POINTS) PLM1, PLM2, PLM3, and PLM4 are two-element arrays giving the latitudes and longitudes, in degrees, of four points which are to be on the edges of the rectangular map. If a cylindrical projection is being used, the first point should be on the left edge and the second point on the right edge; otherwise, the order makes no difference. Note that the calling program should include the following statement:—

```
DIMENSION PLM1(2),PLM2(2),PLM3(2),PLM4(2)
```

(In fact, strict adherence to the FORTRAN-77 standard requires this, no matter what the value of JLTS.)

JLTS='AN' (ANGLES) PLM1, PLM2, PLM3, and PLM4 are positive angles, in degrees, representing angular distances from a point on the map to the left, right, bottom, and top edges of the map. For most projections, these angles are measured with the center of the earth at the vertex and represent angular distances from the point which projects to the origin of the u/v plane; on a satellite-view projection, they are measured with the satellite at the vertex and represent angular deviations from the line of sight. Angular limits are particularly useful for polar projections and the satellite-view projection; they are not appropriate for the Lambert conformal conic and an error will result if one attempts to use JLTS='AN' with JPRJ='LC'.

JLTS='LI' (LIMITS) PLM1, PLM2, PLM3, and PLM4 specify the minimum value of u, the maximum value of u, the minimum value of v, and the maximum value of v, respectively. Knowledge of the projection equations is necessary in order to use this option correctly.

G.22 Subroutine MAPSTx

G.22.1 Purpose

To set the values of EZMAP parameters.

G.22.2 Usage

```
CALL MAPSTC (WHCH,CVAL)
CALL MAPSTI (WHCH,IVAL)
CALL MAPSTL (WHCH,LVAL)
CALL MAPSTR (WHCH,RVAL)
```

G.22.3 Arguments

WHCH is a character string specifying the parameter to be set.

CVAL,IVAL,LVAL,RVAL is the value to be given to the parameter specified by WHCH — of type character, integer, logical, or real, respectively.

Some parameters may be set in more than one way. For example, the parameter 'GR' (Grid), which specifies the grid spacing, may be given the value 10.0 in either of two ways:—

```
CALL MAPSTI ('GR',10)
CALL MAPSTR ('GR',10.)
```

The flag which controls dotting of outlines may be turned on using either of these calls:—

```
CALL MAPSTI ('DO',1)
CALL MAPSTL ('DO',.TRUE.)
```

The important point to remember is that the last character of the routine name implies the type of the argument.

Only the first two characters of WHCH are examined. For the sake of code readability, a longer character string may be used.

Below is a list of all the parameters which may be set using these routines:—

WHCH	Type	Meaning
DAshpattern	I	Dashed-line pattern for the grids. A 16-bit quantity. Default is 21845 (octal 52525 or binary 0101010101010101).
DD	I,R	Distance between dots along a dotted line drawn by MAPIT. The default value is 12 *(out of 4096; see* 'RE', *below).*
DL	I,L	If true (non-zero), user calls to MAPIT draw dotted lines. Default is false (zero); lines drawn by MAPIT are solid or dashed, depending on the current state of the DASHSUPER package.
DOt	I,L	If true (non-zero), outlines are dotted. Default is false (zero); outlines are solid.
ELliptical	I,L	If true (non-zero), only that part of the map which falls inside an ellipse inscribed within the normal rectangular perimeter is drawn. This is particularly appropriate for use with azimuthal projections and angular limits specifying a square, in which case the ellipse becomes a circle, but it will work for any map. The default value is zero.
GD	R	The distance between points used to draw the grid, in degrees. The default value is 1.0; user values must fall between .001 and 10.
GRid	I,R	The desired grid spacing. A zero suppresses the grid. The default is 10 degrees.

<table>
<tr><th>WHCH</th><th>Type</th><th>Meaning</th></tr>
<tr><td>In</td><td>I</td><td>“n” is an integer between 1 and 7. Each “In” specifies the intensity of some portion of the map. Values are in the range 0-255. Defaults are:—
<table>
<tr><th>n</th><th>Use</th><th>Default</th></tr>
<tr><td>1</td><td>perimeter</td><td>240</td></tr>
<tr><td>2</td><td>grid</td><td>150</td></tr>
<tr><td>3</td><td>labels</td><td>210</td></tr>
<tr><td>4</td><td>limbs</td><td>240</td></tr>
<tr><td>5</td><td>continents</td><td>240</td></tr>
<tr><td>6</td><td>U.S. states</td><td>180</td></tr>
<tr><td>7</td><td>countries</td><td>210</td></tr>
</table>
</td></tr>
<tr><td>LAbel</td><td>I,L</td><td>If true (non-zero), label the meridians and poles. Default is true (non-zero).</td></tr>
<tr><td>LS</td><td>I</td><td>Controls label size. A character width, to be used in calling PWRIT. The default value is 1, which gives a character width of 12 plotter units.</td></tr>
<tr><td>MV</td><td>I,R</td><td>Minimum vector length for outlines. A point closer to the previous point than this is omitted. Default value is 4 (out of 4096; see 'RE', below).</td></tr>
<tr><td>OUtline</td><td>C</td><td>Says which set of outline data to use. Possible values are 'NO', for no outlines, 'CO', for the continental outlines (the default), 'US', for U.S. state outlines, 'PS', for continental outlines plus international outlines plus U.S. state outlines, and 'PO', for continental outlines plus international outlines. Default is 'CO'.</td></tr>
<tr><td>PErim</td><td>I,L</td><td>If true (non-zero), draw the perimeter. Default is true (non-zero).</td></tr>
<tr><td>REsolution</td><td>I,R</td><td>The width of the target plotter, in plotter units. Default value is 4096.</td></tr>
</table>

WHCH	Type	Meaning
SAtellite	I,R	If less than -1 or greater than 1, changes orthographic projection to satellite-view. Absolute value is the distance of satellite from the center of the earth, in multiples of the earth's radius. The sign indicates whether a normal projection (positive) or an extended projection (negative) is to be used. The extended projection is useful when one is overlaying CONREC output on a map. The default value of 'SA' is zero. See also 'S1' and 'S2', below.
S1 and S2	I,R	Used only when 'SA' is outside [-1,1]. Both are angles, in degrees. 'S1' measures the angle between the center of the earth and the aim point of the satellite's camera, as seen from the satellite. If 'S1' is zero, the projection shows the earth as seen by a satellite looking straight down; call this the "basic view". If 'S1' is nonzero, 'S2' measures the angle from the positive u axis of the basic view to the line OP, where O is the origin of the basic view and P is the projection of the desired line of sight on the basic view, positive if measured counter-clockwise.
SR	R	A search radius, in degrees. Used by MAPINT in finding the latitude/longitude range of the map. The default value is 1.0; user values must fall between .001 and 10. This parameter should probably not be changed except by advice of a knowledgeable consultant.

G.23 Subroutine MAPTRN

G.23.1 Purpose

To find the projection in the u/v plane of a point whose latitude and longitude are known. May be called at any time after EZMAP has been initialized (by calling MAPINT or otherwise).

G.23.2 Usage

```
CALL MAPTRN (RLAT,RLON,UVAL,VVAL)
```

G.23.3 Arguments

RLAT,RLON are the latitude and longitude, respectively, of a point on the globe. RLAT must be between -90. and +90., inclusive; RLON must be between -540. and +540., inclusive.

UVAL,VVAL is the projection in the u/v plane of (RLAT,RLON). The units of UVAL and VVAL depend on the projection.

If the point is not projectable, UVAL is returned equal to 1.E12. Note that, if the point is projectable, but outside the boundary of the map, as defined by the last call to MAPSET, its u and v coordinates are still returned by MAPTRN. The user must do the test required to determine if the point is within limits, if that is necessary.

G.24 Subroutine MAPUSR

G.24.1 Purpose

The routine MAPUSR is called by EZMAP just before and just after portions of the map are drawn. The default version does nothing. (Actually, that's not quite true; for the sake of efficiency, the default version resets the dash pattern for grid lines to "solid"). A user-supplied version may set/reset the dotting parameter 'DL', the DASHSUPER dash pattern, the intensity, the color, etc., so as to achieve a desired effect.

G.24.2 Usage (by EZMAP)

```
CALL MAPUSR (IPRT)
```

G.24.3 Arguments

IPRT, if positive, says that a particular part of the map is about to be drawn, as follows:—

IPRT	Part
1	Perimeter
2	Grid
3	Labels
4	Limb lines
5	Continental outlines
6	U.S. state outlines
7	International outlines

If IPRT is negative, it says that drawing of the last part is complete. The absolute value of IPRT will be one of the above values. Changed quantities should be restored.

G.25 Subroutine MAPVEC

G.25.1 Purpose

To draw lines on the map produced by a call to MAPDRW — used in conjunction with MAPFST.

G.25.2 Usage

```
CALL MAPVEC (RLAT,RLON)
```

This call is exactly equivalent to the call

```
CALL MAPIT (RLAT,RLON,1)
```

G.25.3 Arguments

RLAT and RLON are defined as for MAPIT. See the description of MAPIT.

G.26 Subroutine SUPMAP

G.26.1 Purpose

An implementation of the routine from which EZMAP grew. A single call to SUPMAP creates a map of a desired portion of the globe, according to a desired projection, with desired outlines drawn in, and with lines of latitude and longitude at desired intervals. An appropriate call to the routine SET is performed.

G.26.2 Usage

```
CALL SUPMAP (JPRJ,PLAT,PLON,ROTA,PLM1,PLM2,
             PLM3,PLM4,JLTS,JGRD,IOUT,IDOT,IERR)
```

G.26.3 Arguments

IABS(JPRJ) defines the projection type, as follows (values less than 1 or greater than 10 are treated as 1 or 10, respectively):—

IABS(JPRJ)	Projection
1	Stereographic.
2	Orthographic.
3	Lambert conformal conic.
4	Lambert equal area.
5	Gnomonic.
6	Azimuthal equidistant.
7	Satellite view.
8	Cylindrical equidistant.
9	Mercator.
10	Mollweide.

Using the value 2 causes the parameter 'SA' to be zeroed. Using the value 7 causes 'SA' to be examined. If it has a non-zero value, the value is left alone. If it has a zero value, its value is reset to 6.631, which is about right for a satellite in a geosynchronous equatorial orbit (for whatever that's worth).

The sign of JPRJ, when IOUT is -1, 0, or 1, indicates whether the continental outlines are to be plotted or not. See IOUT, below.

PLAT,PLON,ROTA define the origin of the projection and its rotation angle and are used in the same way as they would be in a call to the routine MAPROJ (which see).

JLTS,PLM1,PLM2,PLM3,PLM4 specify the rectangular limits of the map. These arguments are used in the same way as they would be in a call to MAPSET (which see), except that JLTS is an integer instead of a character string. IABS(JLTS) may take on the values 1 through 5, as follows:—

- 1 Like JLTS='MA' in a call to MAPSET.
- 2 Like JLTS='CO' in a call to MAPSET.
- 3 Like JLTS='LI' in a call to MAPSET.
- 4 Like JLTS='AN' in a call to MAPSET.
- 5 Like JLTS='PO' in a call to MAPSET.

MOD(IABS(JGRD),1000) is the value, in degrees, of the interval at which lines of latitude and longitude are to be plotted. If the given interval is zero, grid lines and labels are not plotted. If JGRD is less than zero, the perimeter is not plotted. Set JGRD to -1000 to suppress both grid lines and perimeter and to +1000 to suppress the grid lines, but leave the perimeter. The value -0 may have a meaning on ones' complement machines, but should be avoided; use -1000 instead.

If IOUT has the value 0, U.S. state outlines are omitted. If it has the absolute value 1, they are plotted. In both of these cases, the sign of JPRJ indicates whether continental outlines are to be plotted (JPRJ positive) or not (JPRJ negative). Originally, SUPMAP recognized only these values of IOUT; now, if IOUT is less than −1 or greater than 1, the sign of JPRJ is ignored, and IOUT selects an outline group, as follows:—

IOUT	Outline Group Selected
-2 or less	'NO' (no outlines).
2	'CO' (continental outlines).
3	'US' (U.S. state outlines).
4	'PS' (continental outlines plus international outlines plus U.S. state outlines).
5 or greater	'PO' (continental outlines plus international outlines, but no U.S. state outlines).

IDOT=0 to get continuous outlines, 1 to get dotted outlines.

IERR is an output parameter. A non-zero value indicates that an error has occurred.

H Package DASHSUPER

H.1 Purpose

DASHSUPER is a software dashed line package with smoothing capabilities and the capability of removing crowded lines.

H.2 Usage

DASHSUPER contains a set of routines corresponding to the line drawing routine included in the NCAR System Plot Package. However, rather than producing straight, solid lines, the routines from DASHSUPER produce smoothed lines drawn with a user specified dash pattern. The first call should be

```
CALL RESET
```

RESET initializes the model-picture array and should be called before each new frame to be produced. Then use,

```
CALL DASHD (IPAT,NC,JCRT,JSIZE)
```

to specify the dash pattern to be used in subsequent calls. Lines may then be drawn with calls to either

```
CALL CURVED (X,Y,N)
```

or to

```
CALL FRSTD (X,Y)
CALL VECTD (X,Y)
CALL LASTD
```

FRSTD processes the first point of a line. VECTD is called for the remaining points of a line. LASTD is called only after the last point of a line has been processed in VECTD.

The following routine may also be called, but no smoothing will result:—

```
CALL LINED (XA,YA,XB,YB)
```

PWRM can be called to draw characters and mark the regions where the characters have been drawn in the model picture.

```
CALL PWRM (X,Y,IDPC,NC,ISIZ,IOR,ICNT)
```

NOTE

1. When using FRSTD and VECTD, LASTD must be called (no arguments needed). LASTD sets up the calls to the smoothing routines KURV1S and KURV2S.
2. When switching from the regular plotting routines to a dashed line package the first call should not be to VECTD, and when switching from a dashed line package to the regular plotting routines the first call should not be to VECTOR.

H.3 Algorithm

Points for each line segment are processed and passed to the routines, KURV1S and KURV2S, which compute splines under tension passing through these points. New points are generated between the given points, resulting in smooth lines. As each line is drawn, a test is done to see if that part of the plotting plane has been drawn on. If it has, then that line, or part of that line, is not drawn. As the lines are drawn, they are also marked into the model picture.

Note: Since this algorithm eliminates the lines in the order that they are drawn, the user should draw all lines of major importance first and then lines of minor importance.

H.4 Arguments for the Subroutines

Except for DASHD, all arguments match those in the corresponding routine in the System Plot Package. See Appendix J for an explanation of the arguments.

Wherever the system plot package allows either fixed or floating point arguments, the software dashed line package does also.

H.4.1 Arguments for Subroutine DASHD

On Input:—

IPAT A dash line pattern that may be defined in one of two ways:

- If IPAT is an integer in the range 0–177777_8 , these 16 bits are treated as a code for solid or gap with a one bit as 3 plotter address units solid and a zero bit as 3 plotter address units of gap. Thus, for example, the pattern 147000_8 = 1100111000000000 means 6 on, 6 off, 9 on, 27 off etc.
- If IPAT is not of the form above, it is assumed to be a Hollerith string , NC characters long. NC may be any number, although about 60 seems to be a practical limit. A solid is indicated by a $. A gap is indicated by a single quote ("'"). Blanks are ignored. Any other Hollerith characters become part of the pattern with a hole of appropriate length being left in the line for the character string. No single Hollerith string to be printed as such may be longer than 15 characters.

When IPAT is to represent a bit pattern the other parameters to DASHD have to be set to zero.

NC The number of characters in IPAT.

JCRT The length in user plotter address units per $ or quote.

JSIZE Is the size of the PWRIT character.

- If between 0 and 3 , it is 1., 1.5, 2. and 3. times the 8 plotter address unit width.
- If greater than 3, it is the character width in user plotter address units.

On Output:—

All arguments are unchanged for all routines.

H.5 Supplementary Information

The values of many internal parameters which control the details of the dashed lines output are accessibile via two Common Blocks — INTPR and SMFLAG. The parameter names, default values and function of the variables in these Common Blocks are described in the tables below.

COMMON BLOCK INTPR		
Parameter Name	Default Value	Function
IPAU	3	Number of plotter addresses per element (bit) in the dash pattern when $\leq 177777_8$. Thus, the pattern is repeated every IPAU*16 plotter address units.
FPART	1.0	Multiplying factor for first solid line segment. This can be used to off-set labels. For example, if FPART = 0.5, the first solid line segment is only one-half as long as it is the rest of the time. This moves all labels on this line towards the beginning, which reduces the probability of the label being written on top of a label of a nearby line drawn with FPART = 1.
TENSN	2.5	Tension factor. Must be greater than 0. A large tension factor (30.) would essentially turn off smoothing.
NP	150	Twice the maximum number of interpolated points on a horizontal line with length equal to that of the grid. More points per unit length are interpolated for short lines than for long lines.
SMALL	128.	The minimum distance in plotter address units between points which are saved. When the points on a line are being processed, and two or more consecutive points are less than 128 plotter address units apart, only the first one of these consecutive points is saved. This procedure is to prevent cusps.

COMMON BLOCK INTPR (CONTINUED)		
Parameter Name	Default Value	Function
L1	70	The maximum number of points saved at one time. If there are more than 70 points on a given line, 70 points are processed, then the next 70, until the entire line is processed. Smoothness between segments is maintained automatically. If L1 is increased, the dimensions of XSAVE, YSAVE, XP, YP, and TEMP in FDVDLD must be increased to the new value of L1.
ADDLR	2.0	Number of plotter addresses added to each character string to the left and to the right as free space.
ADDTB	2.0	The number of plotter addresses added to each character string on the top and on the bottom as free space.
MLLINE	384	The maximum length in each coordinate of a single line to be processed. Lines longer than MLLINE plotter address units in a coordinate are cut up into smaller segments and each of these segments is processed separately. This is to prevent anomalies in the removal of long lines, because only starting point and end point of lines are checked in that process.
ICLOSE	6	An internal or external call to set the pen (pen-up) to a specific position is only executed if this position is more than ICLOSE plotter address units away from the current pen position (distance= difference in x-coordinates + difference in y-coordinates).

COMMON BLOCK SMFLAG		
Parameter Name	Default Value	Function
IOFFS	0	Flag to turn on smoothing code. = 0 smoothing and marking (DASHSUPR). = −1 smoothing but no marking (DASHSMTH). = 1 no smoothing or marking (DASHCHAR).

I Package CURVEPACK

I.1 Contents of the Package

This package contains 24 subprograms:--

CURV1 For setting-up one dimensional interpolation problems.

CURV2 For performing one dimensional interpolation (given the output of CURV1).

CURVD For differentiating the one dimensional interpolating funtion (given the output of CURV1).

CURVI For integrating the one dimensional interpolating function (given the output of CURV1).

KURV1 For setting-up curve fitting problems in the plane.

KURV2 For performing planar curve fitting (given the output of KURV1).

KURVP1 For setting-up closed-curve fitting problems in the plane.

KURVP2 For performing planar closed-curve fitting (given the output of KURVP1).

QURV1 For setting-up three dimensional curve fitting problems.

QURV2 For performing three dimensional curve fitting (given the output of QURV1).

SURF1 For setting-up surface fitting problems over rectangular grids.

SURF2 For performing surface fitting (given the output of SURF1).

BSURF1 For setting-up surface fitting problems over rectangular grids equally spaced in both coordinates (and without user specified boundary conditions).

BSURF2 For performing surface fitting to an entire second grid equally spaced in both coordinates (given the output of BSURF1).

XSURF1 For setting-up surface fitting problems over rectangular grids equally spaced in the x coordinate (and without user specified boundary conditions).

XSURF2 For performing surface fitting to an entire second grid equally spaced in both coordinates (given the output of XSURF1).

YSURF1 For setting-up surface fitting problems over rectangular grids equally spaced in the y coordinate (and without user specified boundary conditions).

YSURF2 For performing surface fitting to an entire second grid equally spaced in both coordinates (given the output of YSURF1).

NSURF1 For setting-up surface fitting problems over rectangular grids (without user specified boundary conditions).

NSURF2 For performing surface fitting to an entire second grid equally spaced in both coordinates (given the output of NSURF1).

CEEZ An auxilliary subroutine used by CURV1, KURV1, QURV1, SURF1, BSURF1, XSURF1, YSURF1, and NSURF1 for determining certain coefficients used to approximate the slopes of curves at their ends when this information is not user specified.

TERMS An auxilliary subroutine used by CURV1, KURV1, KURVP1, QURV1, SURF1, BSURF1, XSURF1, YSURF1, and NSURF1 for determining certain coefficients in the linear system specifying curve parameters.

SNHCSH An auxilliary subroutine used by all of the above package modules which computes 14 digit approximations to the functions SINH(X)-X, COSH(X)-1, and COSH(X)-1-X*X/2.

INTRVL An auxilliary fuction used by CURV2, CURVD, CURVI, KURV2, KURVP2, QURV2, and SURF2 for searching an increasing sequence of values for a pair which bracket another given value.

I.2 Capabilities

The basic capabilities of the package are those described under "Contents", i.e. the fitting of curves and surfaces (and in one case, the differentiation and integration of such curves) in various spaces. Further discussion is found in the Bibliography below. With small alterations, the package can yeild many additional capabilities. Examples are:-

1. Finding second and higher order derivatives of one dimensional, interpolatory functions
2. Finding tangents and normals to curves in the plane and in three dimensions
3. Locating the intersection of two planar curves or the intersection of a plane with a curve in three dimensions
4. Determining tangent planes and normals to surfaces
5. Determining arc-lengths along curves
6. Determining planar areas enclosed by curves
7. Determining extrema of curves and surfaces

I.3 Bibliography

1. Cline, A. K., *Scalar- and planar-valued curve fitting using splines under tension*, Comm. A. C. M. 17, 4 (April 1974), 218-220.
2. Cline, A. K., *Six subprograms for curve fitting using splines under tension*, Comm. A. C. M. 17, 4 (April 1974), 220-223.
3. Cline, A. K., *Curve fitting using splines under tension*, Atmos. Technology 3 (Sept 1973), 60-65.

I.4 Subroutine CURV1

I.4.1 Purpose

This subroutine determines the parameters necessary to compute an interpolatory spline under tension through a sequence of functional values. The slopes at the two ends of the curve may be specified or omitted. For actual computation of points on the curve it is necessary to call the function CURV2.

I.4.2 Usage

```
SUBROUTINE CURV1 (N,X,Y,SLP1,SLPN,ISLPSW,YP,TEMP,SIGMA,IERR)

INTEGER N,ISLPSW,IERR
REAL X(N),Y(N),SLP1,SLPN,YP(N),TEMP(N),SIGMA
```

I.4.3 Arguments

On Input:—

N is the number of values to be interpolated (N≥2).

X is an array of the N increasing abscissae of the functional values.

Y is an array of the N ordinates of the values, (i.e. Y(K) is the functional value corresponding to X(K)).

SLP1,SLPN contain the desired values for the first derivative of the curve at X(1) and X(N), respectively. The user may omit values for either or both of these parameters and signal this with ISLPSW.

ISLPSW contains a switch indicating which slope data should be used and which should be estimated by this subroutine,

= 0 if SLP1 and SLPN are to be used,

= 1 if SLP1 is to be used but not SLPN,

= 2 if SLPN is to be used but not SLP1,

= 3 if both SLP1 and SLPN are to be estimated internally.

YP is an array of length at least N.

TEMP is an array of length at least N which is used for scratch storage.

SIGMA contains the tension factor. This value indicates the curviness desired. If ABS(SIGMA) is nearly zero (e.g. .001) the resulting curve is approximately a cubic spline. If ABS(SIGMA) is large (e.g. 50.) the resulting curve is nearly a polygonal line. If SIGMA equals zero a cubic spline results. A standard value for SIGMA is approximately 1. in absolute value.

On Output:—

YP contains the values of the second derivative of the curve at the given nodes.

IERR contains an error flag,

= 0 for normal return,

= 1 if N is less than 2,

= 2 if X-values are not strictly increasing.

N, X, Y, SLP1, SLPN, ISLPSW and SIGMA are unaltered.

This subroutine references package modules CEEZ, TERMS, and SNHCSH.

I.5 Function CURV2

I.5.1 Purpose

This function interpolates a curve at a given point using a spline under tension. The subroutine CURV1 should be called earlier to determine certain necessary parameters.

I.5.2 Usage

```
FUNCTION CURV2 (T,N,X,Y,YP,SIGMA)

INTEGER N
REAL T,X(N),Y(N),YP(N),SIGMA
```

I.5.3 Arguments

On Input:—

T contains a real value to be mapped onto the interpolating curve.

N contains the number of points which were specified to determine the curve.

X,Y are arrays containing the abscissae and ordinates, respectively, of the specified points.

YP is an array of second derivative values of the curve at the nodes.

SIGMA contains the tension factor (its sign is ignored).

The parameters N, X, Y, YP, and SIGMA should be input unaltered from the output of CURV1.

On Output:—

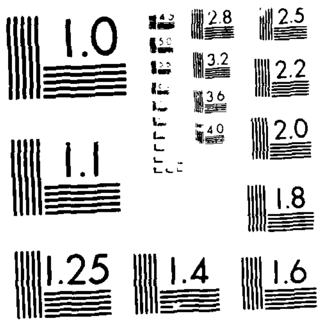
CURV2 contains the interpolated value, and none of the input parameters are altered.

This function references package modules INTRVL and SNHCSH.

AD-A195 163 IMPLEMENTATION OF THE NCAR (NATIONAL CENTRE FOR ATMOSPHERIC RESEARCH) GRA..(U) AERONAUTICAL RESEARCH LABS MELBOURNE (AUSTRALIA) B D FAIRLIE JAN 88 2/2

UNCLASSIFIED ARL/AERO-TM-394 DODA-AR-004-593 F/G 12/5 NL

END
DATE
FILMED
8 88



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

I.6 Function CURVD

I.6.1 Purpose

This function differentiates a curve at a given point using a spline under tension. The subroutine CURV1 should be called earlier to determine certain necessary parameters.

I.6.2 Usage

```
FUNCTION CURVD (T,N,X,Y,YP,SIGMA)

INTEGER N
REAL T,X(N),Y(N),YP(N),SIGMA
```

I.6.3 Arguments

On Input:—

T contains a real value at which the derivative is to be determined.

N contains the number of points which were specified to determine the curve.

X,Y are arrays containing the abscissae and ordinates, respectively, of the specified points.

YP is an array of second derivative values of the curve at the nodes.

SIGMA contains the tension factor (its sign is ignored).

The parameters N, X, Y, YP, and SIGMA should be input unaltered from the output of CURV1.

On Output:—

CURVD contains the derivative value, and none of the input parameters are altered.

This function references package modules INTRVL and SNHCSH.

I.7 Function CURVI

I.7.1 Purpose

This function integrates a curve specified by a spline under tension between two given limits. The subroutine CURV1 should be called earlier to determine necessary parameters.

I.7.2 Usage

```
FUNCTION CURVI (XL,XU,N,X,Y,YP,SIGMA)

INTEGER N
REAL XL,XU,X(N),Y(N),YP(N),SIGMA
```

I.7.3 Arguments

On Input:—

XL,XU contain the upper and lower limits of integration, respectively (XL need not be less than or equal to XU, CURVI(XL,XU,\ldots) = -CURVI(XU,XL,\ldots)).

N contains the number of points which were specified to determine the curve.

X,Y are arrays containing the abscissae and ordinates, respectively, of the specified points.

YP is an array from subroutine CURV1 containing at the nodes.

SIGMA contains the tension factor (its sign is ignored).

The parameters N, X, Y, YP, and SIGMA should be input unaltered from the output of CURV1.

On Output:—

CURVI contains the integral value, and none of the input parameters are altered.

This function references package modules INTRVL and SNHCSH.

I.8 Function KURVI

I.8.1 Purpose

This subroutine determines the parameters necessary to compute a spline under tension forming a curve in the plane and passing through a sequence of pairs (X(1),Y(1)), ..., (X(N),Y(N)). For actual computation of points on the curve it is necessary to call the subroutine KURV2.

I.8.2 Usage

```
SUBROUTINE KURV1 (N,X,Y,SLP1,SLPN,ISLPSW,XP,YP,TEMP,S,SIGMA,IERR)

INTEGER N,ISLPSW,IERR
REAL X(N),Y(N),SLP1,SLPN,XP(N),YP(N),TEMP(N),S(N),SIGMA
```

I.8.3 Arguments

On Input:—

N is the number of points to be interpolated (N$\geq$2).

X is an array containing the N x-coordinates of the points.

Y is an array containing the N y-coordinates of the points. (Adjacent X–Y pairs must be distinct, i.e. either X(I) $\neq$ X(I+1) or Y(I) $\neq$ Y(I+1), for I=1,...,N-1.)

SLP1,SLPN contain the desired values for the angles (in radians) of the slope at (X(1),Y(1)) and (X(N),Y(N)) respectively. The angles are measured counter-clock-wise from the x-axis and the positive sense of the curve is assumed to be that moving from point 1 to point N. The user may omit values for either or both of these parameters and signal this with ISLPSW.

ISLPSW contains a switch indicating which slope data should be used and which should be estimated by this subroutine,

= 0 if SLP1 and SLPN are to be used,

= 1 if SLP1 is to be used but not SLPN,

= 2 if SLPN is to be used but not SLP1,

= 3 if both SLP1 and SLPN are to be estimated internally.

XP,YP are arrays of length at least N.

TEMP is an array of length at least N which is used for scratch storage.

S is an array of length at least N.

SIGMA contains the tension factor. This value indicates the curviness desired. If ABS(SIGMA) is nearly zero (e.g. .001) the resulting curve is approximately a cubic spline. If ABS(SIGMA) is large (e.g. 50.) the resulting curve is nearly a polygonal line. If SIGMA is zero, a cubic spline results. A standard value for SIGMA is approximately 1.0 in absolute value.

On Output:—

XP,YP contain information about the curvature of the curve at the given nodes.

S contains the polygonal arclengths of the curve.

IERR contains an error flag,

= 0 for normal return,

= 1 if N is less than 2,

= 2 if adjacent coordinate pairs coincide.

N, X, Y, SLP1, SLPN, ISLPSW, and SIGMA are unaltered.

This subroutine references package modules CEEZ, TERMS, and SNHCSH.

I.9 Subroutine KURV2

I.9.1 Purpose

This subroutine performs the mapping of points in the interval (0.,1.) onto a curve in the plane. The subroutine KURV1 should be called earlier to determine certain necessary parameters. The resulting curve has a parametric representation both of whose components are splines under tension and functions of the polygonal arclength parameter.

I.9.2 Usage

```
SUBROUTINE KURV2 (T,XS,YS,N,X,Y,XP,YP,S,SIGMA)

INTEGER N
REAL T,XS,YS,X(N),Y(N),XP(N),YP(N),S(N),SIGMA
```

On Input:—

T contains a real value to be mapped to a point on the curve. The interval (0.,1.) is mapped onto the entire curve, with 0.0 mapping to (X(1),Y(1)) and 1.0 mapping to (X(N),Y(N)). Values outside this interval result in extrapolation.

N contains the number of points which were specified to determine the curve.

X,Y are arrays containing the x- and y-coordinates of the specified points.

XP,YP are the arrays output from KURV1 containing curvature information.

S is an array containing the polygonal arclengths of the curve.

SIGMA contains the tension factor (its sign is ignored).

The parameters N, X, Y, XP, YP, S, and SIGMA should be input unaltered from the output of KURV1.

On Output:—

XS,YS contain the x- and y-coordinates of the image point on the curve.

None of the input parameters are altered.

This subroutine references package modules INTRVL and SNHCSH.

I.10 Subroutine KURVP1

I.10.1 Purpose

This subroutine determines the parameters necessary to compute a spline under tension forming a closed curve in the plane and passing through a sequence of pairs (X(1),Y(1)),...,(X(N),Y(N)). For actual computation of points on the curve it is necessary to call the subroutine KURVP2.

I.10.2 Usage

```
SUBROUTINE KURVP1 (N,X,Y,XP,YP,TEMP,S,SIGMA,IERR)

INTEGER N,IERR
REAL X(N),Y(N),XP(N),YP(N),TEMP(2*N),S(N),SIGMA
```

I.10.3 Arguments

On Input:—

N is the number of points to be interpolated ($N \geq 2$).

X is an array containing the N x-coordinates of the points.

Y is an array containing the N y-coordinates of the points. (Adjacent x-y pairs must be distinct, i.e. either X(I)$\neq$X(I+1) or Y(I)$\neq$Y(I+1), for I=1,...,N-1.)

XP,YP are arrays of length at least N.

TEMP is an array of length at least 2*N which is used for scratch storage.

S is an array of length at least N.

SIGMA contains the tension factor. This value indicates the curviness desired. If ABS(SIGMA) is nearly zero (e.g. .001) the resulting curve is approximately a cubic spline. If ABS(SIGMA) is large (e.g. 50.) the resulting curve is nearly a polygonal line. If SIGMA equals zero a cubic spline results. A standard value for SIGMA is approximately 1.0 in absolute value.

On Output:—

XP,YP contain information about the curvature of the curve at the given nodes.

S contains the polygonal arclengths of the curve.

IERR contains an error flag,

= 0 for normal return,

= 1 if N is less than 2,

= 2 if adjacent coordinate pairs coincide.

N, X, Y, and SIGMA are unaltered,

This subroutine references package modules TERMS and SNHCSH.

I.11 Subroutine KURVP2

I.11.1 Purpose

This subroutine performs the mapping of points in the interval (0.,1.) onto a closed curve in the plane. The subroutine KURVP1 should be called earlier to determine certain necessary parameters. The resulting curve has a parametric representation both of whose components are periodic splines under tension and functions of the polygonal arclength parameter.

I.11.2 Usage

```
SUBROUTINE KURVP2 (T,XS,YS,N,X,Y,XP,YP,S,SIGMA)

INTEGER N
REAL T,XS,YS,X(N),Y(N),XP(N),YP(N),S(N),SIGMA
```

I.11.3 Arguments

On Input:—

`T` contains a value to be mapped onto the curve. The interval (0.,1.) is mapped onto the entire closed curve with both 0.0 and 1.0 mapping to (`X(1),Y(1)`). The mapping is periodic with period one thus any interval of the form (`TT,TT+1.`) maps onto the entire curve.

`N` contains the number of points which were specified to determine the curve.

`X,Y` are arrays containing the x- and y-coordinates of the specified points.

`XP,YP` are the arrays output from `KURVP1` containing curvature information.

`S` is an array containing the polygonal arclengths of the curve.

`SIGMA` contains the tension factor (its sign is ignored).

The parameters `N`, `X`, `Y`, `XP`, `YP`, `S` and `SIGMA` should be input unaltered from the output of `KURVP1`.

On Output:—

`XS,YS` contain the x- and y-coordinates of the image point on the curve.

None of the input parameters are altered.

This subroutine references package modules INTRVL and SNHCSH.

I.12 Subroutine QURV1

I.12.1 Purpose

This subroutine determines the parameters necessary to compute a spline under tension passing through a sequence of triples (`X(1),Y(1),Z(1)`),...,(`X(N),Y(N),Z(N)`). The slopes at the two ends of the curve may be specified or omitted. For actual computation of points on the curve it is necessary to call the subroutine `QURV2`.

I.12.2 Usage

```
SUBROUTINE QURV1 (N,X,Y,Z,SLP1X,SLP1Y,SLP1Z,SLPNX,SLPNY,
                  SLPNZ,ISLPSW,XP,YP,ZP,TEMP,S,SIGMA,IERR)

INTEGER N,ISLPSW,IERR
REAL X(N),Y(N),Z(N),SLP1X,SLP1Y,SLP1Z,SLPNX,SLPNY,SLPNZ,
                      XP(N),YP(N),ZP(N),TEMP(N),S(N),SIGMA
```

I.12.3 Arguments

On Input:—

N is the number of points to be interpolated (N$\geq$2).

X is an array containing the N x-coordinates of the points.

Y is an array containing the N y-coordinates of the points.

Z is an array containing the N z-coordinates of the points. (Adjacent x-y-z triples must be distinct, i.e. either X(I)$\neq$X(I+1) or Y(I)$\neq$ Y(I+1) or Z(I)$\neq$Z(I+1), for I=1, ...,N-1).

SLP1X,SLP1Y,SLP1Z,SLPNX,SLPNY,SLPNZ contain the desired values of the components of tangent vectors to the curve at (X(1),Y(1),Z(1)) and (X(N),Y(N),Z(N)), respectively. The positive sense of the curve is assumed to be that moving from point 1 to point N. The user may omit values for either or both of these triples and signal this with ISLPSW.

ISLPSW contains a switch indicating which slope data should be used and which should be estimated by this subroutine,

= 0 if SLP1X, SLP1Y, SLP1Z and SLPNX, SLPNY, SLPNZ are to be used,

= 1 if SLP1X, SLP1Y, SLP1Z are to be used but not SLPNX, SLPNY, SLPNZ,

= 2 if SLPNX, SLPNY, SLPNZ are to be used but not SLP1X, SLP1Y, SLP1Z,

= 3 if both end-tangents are to be estimated internally.

XP,YP,ZP are arrays of length at least N.

TEMP is an array of length at least N which is used for scratch storage.

S is an array of length at least N.

SIGMA contains the tension factor. This value indicates the curviness desired. If ABS(SIGMA) is nearly zero (e.g. .001) the resulting curve is approximately a cubic spline. If ABS(SIGMA) is large (e.g. 50.) the resulting curve is nearly a polygonal line. If SIGMA equals zero a cubic spline results. A standard value for SIGMA is approximately 1.0 in absolute value.

On Output:—

XP,YP,ZP contain information about the curvature of the curve at the given nodes.

S contains the polygonal arclengths of the curve.

IERR contains an error flag,

= 0 for normal return,

= 1 if N is less than 2,

= 2 if adjacent triples coincide.

N, X, Y, Z, SLP1X, SLP1Y, SLP1Z, SLPNX, SLPNY, SLPNZ, ISLPSW, and SIGMA are unaltered,

This subroutine references package modules CEEZ, TERMS, and SNHCSH.

I.13 Subroutine QURV2

I.13.1 Purpose

This subroutine performs the mapping of points in the interval (0.,1.) onto a curve in space. The subroutine QURV1 should be called earlier to determine certain necessary parameters. The resulting curve has a parametric representation all of whose components are splines under tension and functions of the polygonal arclength parameter.

I.13.2 Usage

```
SUBROUTINE QURV2 (T,XS,YS,ZS,N,X,Y,Z,XP,YP,ZP,S,SIGMA)

INTEGER N
REAL T,XS,YS,ZS,X(N),Y(N),Z(N),XP(N),YP(N),ZP(N),SIGMA
```

I.13.3 Arguments

On Input:—

T contains a real value to be mapped to a point on the curve. The interval (0.,1.) is mapped onto the entire curve, with 0.0 mapping to (X(1),Y(1),Z(1)) and 1.0 mapping to (X(N),Y(N),Z(N)). Values outside this interval result in extrapolation.

N contains the number of points which were specified to determine the curve.

X,Y,Z are arrays containing the x-, y- and z-coordinates of the specified points.

XP,YP,ZP are the arrays output from QURV1 containing curvature information.

S is an array containing the polygonal arclengths of the curve.

SIGMA contains the tension factor (its sign is ignored).

The parameters N, X, Y, Z, XP, YP, ZP, S, and SIGMA should be input unaltered from the output of QURV1.

On Output:—

XS,YS,ZS contain the x-, y- and z-coordinates of the image point on the curve.

None of the input parameters are altered.

This subroutine references package modules INTRVL and SNHCSH.

I.14 Subroutine SURF1

I.14.1 Purpose

This subroutine determines the parameters necessary to compute an interpolatory surface passing through a rectangular grid of functional values. The surface determined can be represented as the tensor product of splines under tension. The x- and y-partial derivatives around the boundary and the x-y-partial derivatives at the four corners may be specified or omitted. For actual mapping of points onto the surface it is necessary to call the function SURF2.

I.14.2 Usage

```
SUBROUTINE SURF1 (M,N,X,Y,Z,IZ,ZX1,ZXM,ZY1,ZYN,ZXY11,
             ZXYM1,ZXY1N,ZXYMN,ISLPSW,ZP,TEMP,SIGMA,IERR)

INTEGER M,N,IZ,ISLPSW,IERR
REAL X(M),Y(N),Z(IZ,N),ZX1(N),ZXM(N),ZY1(M),ZYN(M),ZXY11,
            ZXYM1,ZXY1N,ZXYMN,ZP(M,N,3),TEMP(N+M+N),SIGMA
```

I.14.3 Arguments

On Input:—

M is the number of grid lines in the x-direction, i.e. lines parallel to the y-axis (M≥2).

N is the number of grid lines in the y-direction, i.e. lines parallel to the x-axis (N≥2).

X is an array of the M x-coordinates of the grid lines in the x-direction. These should be strictly increasing.

Y is an array of the N y-coordinates of the grid lines in the y-direction. These should be strictly increasing.

Z is an array of the M*N functional values at the grid points, i.e. Z(I,J) contains the functional value at (X(I),Y(J)) for I=1,...,M and J=1, ...,N.

IZ is the row dimension of the matrix Z used in the calling program (IZ≥M).

ZX1,ZXM are arrays of the M x-partial derivatives of the function along the X(1) and X(M) grid lines, respectively. Thus ZX1(J) and ZXM(J) contain the x-partial derivatives at the points (X(1),Y(J)) and (X(M),Y(J)), respectively, for J=1,...,N. Either of these parameters will be ignored (and approximations supplied internally) if ISLPSW so indicates.

ZY1,ZYN are arrays of the N y-partial derivatives of the function along the Y(1) and Y(N) grid lines, respectively. Thus ZY1(I) and ZYN(I) contain the y-partial derivatives at the points (X(I),Y(1)) and (X(I),Y(N)), respectively, for I=1,...,M. Either of these parameters will be ignored (and estimations supplied internally) if ISLPSW so indicates.

ZXY11,ZXYM1,ZXY1N,ZXYMN are the x-y-partial derivatives of the function at the four corners, (X(1),Y(1)), (X(M),Y(1)), (X(1),Y(N)), and (X(M),Y(N)),respectively. Any of the parameters will be ignored (and estimations supplied internally) if ISLPSW so indicates.

ISLPSW contains a switch indicating which boundary derivative information is user-supplied and which should be estimated by this subroutine. To determine ISLPSW, let

I1=0 if ZX1 is user-supplied (and = 1 otherwise),

I2=0 if ZXM is user-supplied (and = 1 otherwise),

I3=0 if ZY1 is user-supplied (and = 1 otherwise),

I4=0 if ZYN is user-supplied (and = 1 otherwise),

I5=0 if ZXY11 is user-supplied (and = 1 otherwise),

I6=0 if ZXYM1 is user-supplied (and = 1 otherwise),

I7=0 if ZXY1N is user-supplied (and = 1 otherwise),

I8=0 if ZXYMN is user-supplied (and = 1 otherwise),

then ISLPSW = I1 + 2*I2 + 4*I3 + 8*I4 + 16*I5 + 32*I6 + 64*I7 + 128*I8

Thus ISLPSW=0 indicates all derivative information is user-supplied and ISLPSW=255 indicates no derivative information is user-supplied. Any value between these limits is valid.

ZP is an array of at least 3*M*N locations.

TEMP is an array of at least N+N+M locations which is used for scratch storage.

SIGMA contains the tension factor. This value indicates the curviness desired. If ABS(SIGMA) is nearly zero (e.g. .001) the resulting surface is approximately the tensor product of cubic splines. If ABS(SIGMA) is large (e.g. 50.) the resulting surface is approximately bi-linear. If SIGMA equals zero tensor products of cubic splines result. A standard value for SIGMA is approximately 1.0 in absolute value.

On Output:—

ZP contains the values of the xx-, yy-, and xxyy-partial derivatives of the surface at the given nodes.

IERR contains an error flag,

= 0 for normal return,

= 1 if N is less than 2 or M is less than 2,

= 2 if the x-values or y-values are not strictly increasing.

M, N, X, Y, Z, IZ, ZX1, ZXM, ZY1, ZYN, ZXY11, ZXYM1, ZXY1N, ZXYMN, ISLPSW, and SIGMA are unaltered.

This subroutine references package modules CEEZ, TERMS, and SNHCSH.

I.15 Subroutine SURF2

I.15.1 Purpose

This function interpolates a surface at a given coordinate pair using a bi-spline under tension. The subroutine SURF1 should be called earlier to determine certain necessary parameters.

I.15.2 Usage

```
FUNCTION SURF2 (XX,YY,M,N,X,Y,Z,IZ,ZP,SIGMA)

INTEGER M,N,IZ
REAL XX,YY,X(M),Y(N),Z(IZ,N),ZP(M,N,3),SIGMA
```

I.15.3 Arguments

On Input:—

XX,YY contain the x- and y-coordinates of the point to be mapped onto the interpolating surface.

M,N contain the number of grid lines in the x- and y-directions, respectively, of the rectangular grid which specified the surface.

X,Y are arrays containing the x- and y-grid values, respectively, each in increasing order.

Z is a matrix containing the M*N functional values corresponding to the grid values (i.e. Z(I,J) is the surface value at the point (X(I),Y(J)) for I=1, ...,M and J=1,...,N).

IZ contains the row dimension of the array Z as declared in the calling program.

ZP is an array of 3*M*N locations stored with the various surface derivative information determined by SURF1.

SIGMA contains the tension factor (its sign is ignored).

The parameters M, N, X, Y, Z, IZ, ZP, and SIGMA should be input unaltered from the output of SURF1.

On Output:—

SURF2 contains the interpolated surface value.

None of the input parameters are altered.

This function references package modules INTRVL and SNHCSH.

I.16 Subroutine BSURF1

I.16.1 Purpose

This subroutine determines the parameters necessary to compute an interpolatory surface passing through a rectangular grid of functional values. The x and y values are assumed equally spaced in the grid. The surface determined can be represented as the tensor product of splines under tension. For actual interpolation at a grid of points equally spaced in both x and y coordinates it is necessary to call subroutine BSURF2.

I.16.2 Usage

```
SUBROUTINE BSURF1 (M,N,XMIN,XMAX,YMIN,YMAX,Z,IZ,ZP,TEMP,
                              SIGMA,IERR)

INTEGER M,N,IZ,IERR
REAL XMIN,XMAX,YMIN,YMAX,Z(IZ,N),ZP(M,N,3),TEMP(N+M+N),SIGMA
```

I.16.3 Arguments

On Input:—

M is the number of grid lines in the x-direction, i.e. lines parallel to the y-axis (M≥2).

N is the number of grid lines in the y-direction, i.e. lines parallel to the x-axis (N≥2).

XMIN,XMAX are the lower and upper limits, respectively, of the grid in the x-direction. XMAX should be greater than XMIN.

YMIN,YMAX are the lower and upper limits, respectively, of the grid in the y-direction. YMAX should be greater than YMIN.

Z is an array of the M*N functional values at the grid points, i.e. Z(I,J) contains the functional value at (X(I),Y(J)) for I=1,...,M and J=1,...,N, where X(I) represents the I^{th} equispaced x value, and Y(J) represents the J^{th} equispaced y value.

IZ is the row dimension of the matrix Z used in the calling program (IZ≥M).

ZP is an array of at least 3*M*N locations.

TEMP is an array of at least N+N+M locations which is used for scratch storage.

SIGMA contains the tension factor. This value indicates the curviness desired. If ABS(SIGMA) is nearly zero (e.g. .001) the resulting surface is approximately the tensor product of cubic splines. If ABS(SIGMA) is large (e.g. 50.) the resulting surface is approximately bi-linear. If SIGMA equals zero tensor products of cubic splines result. A standard value for SIGMA is approximately 1.0 in absolute value.

On Output:—

ZP contains the values of the xx-, yy-, and xxyy-partial derivatives of the surface at the given nodes.

IERR contains an error flag,

= 0 for normal return,

= 1 if N is less than 2 or M is less than 2,

= 2 if XMAX is not greater than XMIN or YMAX is not greater than YMIN.

M, N, XMIN, XMAX, YMIN, YMAX, Z, IZ, and SIGMA are unaltered.

This subroutine references package modules CEEZ, TERMS, and SNHCSH.

I.17 Subroutine BSURF2

I.17.1 Purpose

This subroutine maps values onto a surface at every point of a grid equally spaced in both x and y coordinates. The surface interpolation is performed using a bi-spline under tension. The subroutine BSURF1 should be called earlier to determine certain necessary parameters. In both BSURF1 and BSURF2, the original grid is assumed to be equally spaced in the x and y coordinates.

I.17.2 Usage

```
SUBROUTINE BSURF2 (DXMIN,DXMAX,MD,DYMIN,DYMAX,ND,DZ,IDZ,M,
                  N,XMIN,XMAX,YMIN,YMAX,Z,IZ,ZP,WORK,SIGMA)

INTEGER MD,ND,IDZ,M,N,IZ
REAL DXMIN,DXMAX,DYMIN,DYMAX,DZ(IDZ,ND),XMIN,XMAX,YMIN,
                 YMAX,Z(IZ,N),ZP(M,N,3),WORK(4,MD),SIGMA
```

I.17.3 Arguments

On Input:—

DXMIN,DXMAX contain the lower and upper limits, respectively, of the x-coordinates of the second grid.

MD contains the number of grid lines in the x direction of the second grid (MD$\geq$1).

DYMIN,DYMAX contain the lower and upper limits, respectively, of the y-coordinates of the second grid.

ND contains the number of grid lines in the y direction of the second grid (ND$\geq$1).

IDZ contains the row dimension of the array DZ as declared in the calling program.

M,N contain the number of grid lines in the x- and y-directions, respectively, of the rectangular grid which specified the surface.

XMIN,XMAX are the lower and upper limits, respectively, of the grid in the x direction.

YMIN,YMAX are the lower and upper limits, respectively, of the grid in the y direction.

Z is a matrix containing the M*N functional values corresponding to the grid values (i.e. Z(I,J) is the surface value at the point (X(I),Y(J)) for I=1, ...,M and J=1,...,N, where X(I) represents the I^{th} equispaced x value and Y(J) represents the J^{th} equispaced y value).

IZ contains the row dimension of the array Z as declared in the calling program.

ZP is an array of 3*M*N locations stored with the various surface derivative information determined by SURF1.

WORK is an array of 4*MD locations to be used internally for workspace.

SIGMA contains the tension factor (its sign is ignored).

The parameters M, N, XMIN, XMAX, YMIN, YMAX, Z, IZ, ZP, and SIGMA should be input unaltered from the output of BSURF1.

On Output:—

DZ contains the MD by ND array of surface values interpolated at the points of the second grid.

None of the input parameters are altered.

This function references package module SNHCSH.

I.18 Subroutine XSURF1

I.18.1 Purpose

This subroutine determines the parameters necessary to compute an interpolatory surface passing through a rectangular grid of functional values. The x values are assumed equally spaced in the grid. The surface determined can be represented as the tensor product of splines under tension. For actual interpolation at a grid of points equally spaced in both x and y coordinates it is necessary to call subroutine XSURF2.

I.18.2 Usage

```
SUBROUTINE XSURF1 (M,N,XMIN,XMAX,Y,Z,IZ,ZP,TEMP,SIGMA,IERR)

INTEGER M,N,IZ,IERR
REAL XMIN,XMAX,Y(N),Z(IZ,N),ZP(M,N,3),TEMP(N+N+M),SIGMA
```

I.18.3 Arguments

On Input:—

M is the number of grid lines in the x-direction, i.e. lines parallel to the y-axis ($M \geq 2$).

N is the number of grid lines in the y-direction, i.e. lines parallel to the x-axis ($N \geq 2$).

XMIN,XMAX are the lower and upper limits, respectively, of the grid in the x-direction. XMAX should be greater than XMIN.

Y is an array of the N y-coordinates of the grid lines in the y-direction. These should be strictly increasing.

Z is an array of the M*N functional values at the grid points, i.e. Z(I,J) contains the functional value at (X(I),Y(J)) for I=1,...,M and J=1,...,N, where X(I) represents the I^{th} equispaced x value.

IZ is the row dimension of the matrix Z used in the calling program ($IZ \geq M$).

ZP is an array of at least 3*M*N locations.

TEMP is an array of at least N+N+M locations which is used for scratch storage.

SIGMA contains the tension factor. This value indicates the curviness desired. If ABS(SIGMA) is nearly zero (e.g. .001) the resulting surface is approximately the tensor product of cubic splines. If ABS(SIGMA) is large (e.g. 50.) the resulting surface is approximately bi-linear. If SIGMA equals zero tensor products of cubic splines result. A standard value for SIGMA is approximately 1.0 in absolute value.

On Output:—

ZP contains the values of the xx-, yy-, and xxyy-partial derivatives of the surface at the given nodes.

IERR contains an error flag,

= 0 for normal return,

= 1 if N is less than 2 or M is less than 2,

= 2 if the y-values are not strictly increasing or XMAX is not greater than XMIN.

M, N, XMIN, XMAX, Y, Z, IZ, and SIGMA are unaltered.

This subroutine references package modules CEEZ, TERMS, and SNHCSH.

I.19 Subroutine XSURF2

I.19.1 Purpose

This subroutine maps values onto a surface at every point of a grid equally spaced in both x and y coordinates. The surface interpolation is performed using a bi-spline under tension. The subroutine XSURF1 should be called earlier to determine certain necessary parameters. In both XSURF1 and XSURF2, the original grid is assumed to be equally spaced in the x coordinate.

I.19.2 Usage

```
SUBROUTINE XSURF2 (DXMIN,DXMAX,MD,DYMIN,DYMAX,ND,DZ,IDZ,
                   M,N,XMIN,XMAX,Y,Z,IZ,ZP,WORK,SIGMA)

INTEGER MD,ND,IDZ,M,N,IZ
REAL DXMIN,DXMAX,DYMIN,DYMAX,DZ(IDZ,ND),XMIN,XMAX,Y(N),
                   Z(IZ,N),ZP(M,N,3),WORK(4,MD),SIGMA
```

I.19.3 Arguments

On Input:—

DXMIN,DXMAX contain the lower and upper limits, respectively, of the x-coordinates of the second grid.

MD contains the number of grid lines in the x direction of the second grid (MD$\geq$1).

DYMIN,DYMAX contain the lower and upper limits, respectively, of the y-coordinates of the second grid.

ND contains the number of grid lines in the y direction of the second grid (ND$\geq$1).

IDZ contains the row dimension of the array DZ as declared in the calling program.

M,N contain the number of grid lines in the x- and y-directions, respectively, of the rectangular grid which specified the surface.

XMIN,XMAX are the lower and upper limits, respectively, of the grid in the x direction.

Y is an array containing the y grid values in increasing order.

Z is a matrix containing the M*N functional values corresponding to the grid values (i.e. Z(I,J) is the surface value at the point (X(I),Y(J)) for I=1,...,M and J=1,...,N, where X(I) represents the I^{th} equispaced x value).

IZ contains the row dimension of the array Z as declared in the calling program.

ZP is an array of 3*M*N locations stored with the various surface derivative information determined by SURF1.

WORK is an array of 4*MD locations to be used internally for workspace.

SIGMA contains the tension factor (its sign is ignored).

The parameters M, N, XMIN, XMAX, Y, Z, IZ, ZP, and SIGMA should be input unaltered from the output of XSURF1.

On Output:—

DZ contains the MD by ND array of surface values interpolated at the points of the second grid.

None of the input parameters are altered.

This function references package module SNHCSH.

I.20 Subroutine YSURF1

I.20.1 Purpose

This subroutine determines the parameters necessary to compute an interpolatory surface passing through a rectangular grid of functional values. The y values are assumed equally spaced in the grid. The surface determined can be represented as the tensor product of splines under tension. For actual interpolation at a grid of points equally spaced in both x and y coordinates it is necessary to call subroutine YSURF2.

I.20.2 Usage

```
SUBROUTINE YSURF1 (M,N,X,YMIN,YMAX,Z,IZ,ZP,TEMP,SIGMA,IERR)

INTEGER M,N,IZ,IERR
REAL X(M),YMIN,YMAX,Z(IZ,N),ZP(M,N,3),TEMP(N+N+M),SIGMA
```

I.20.3 Arguments

On Input:—

M is the number of grid lines in the x-direction, i.e. lines parallel to the y-axis (M≥2).

N is the number of grid lines in the y-direction, i.e. lines parallel to the x-axis (N≥2).

X is an array of the M x-coordinates *of the grid lines in the* x-direction. these should be strictly increasing.

YMIN,YMAX are the lower and upper limits, respectively, of the grid in the y direction. YMAX should be greater than YMIN.

Z is an array of the M*N functional values at the grid points, i.e. Z(I,J) contains the functional value at (X(I),Y(J)) for I=1,...,M and J=1,...,N, where Y(J) represents the J^{th} equispaced y value.

IZ is the row dimension of the matrix Z used in the calling program (IZ≥M).

ZP is an array of at least 3*M*N locations.

TEMP is an array of at least N+N+M locations which is used for scratch storage.

SIGMA contains the tension factor. This value indicates the curviness desired. If ABS(SIGMA) is nearly zero (e.g. .001) the resulting surface is approximately the tensor product of cubic splines. If ABS(SIGMA) is large (e.g. 50.) the resulting surface is approximately bi-linear. If SIGMA equals zero tensor products of cubic splines result. A standard value for SIGMA is approximately 1.0 in absolute value.

On Output:—

ZP contains the values of the xx-, yy-, and xxyy-partial derivatives of the surface at the given nodes.

IERR contains an error flag,

= 0 for normal return,

= 1 if N is less than 2 or M is less than 2,

= 2 if the x-values are not strictly increasing or YMAX is not greater than YMIN.

M, N, X, YMIN, YMAX, Z, IZ, and SIGMA are unaltered.

This subroutine references package modules CEEZ, TERMS, and SNHCSH.

I.21 Subroutine YSURF2

I.21.1 Purpose

This subroutine maps values onto a surface at every point of a grid equally spaced in both x and y coordinates. The surface interpolation is performed using a bi-spline under tension. The subroutine YSURF1 should be called earlier to determine certain necessary parameters. In both YSURF1 and YSURF2, the original grid is assumed to be equally spaced in the x coordinate.

I.21.2 Usage

```
SUBROUTINE YSURF2 (DXMIN,DXMAX,MD,DYMIN,DYMAX,ND,DZ,IDZ,
                   M,N,X,YMIN,YMAX,Z,IZ,ZP,WORK,SIGMA)

INTEGER MD,ND,IDZ,M,N,IZ
REAL DXMIN,DXMAX,DYMIN,DYMAX,DZ(IDZ,ND),X(M),YMIN,YMAX,
                   Z(IZ,N),ZP(M,N,3),WORK(4,MD),SIGMA
```

I.21.3 Arguments

On Input:—

DXMIN,DXMAX contain the lower and upper limits, respectively, of the x-coordinates of the second grid.

MD contains the number of grid lines in the x direction of the second grid (MD≥2).

DYMIN,DYMAX contain the lower and upper limits, respectively, of the y-coordinates of the second grid.

ND contains the number of grid lines in the y direction of the second grid (ND≥2).

IDZ contains the row dimension of the array DZ as declared in the calling program.

M,N contain the number of grid lines in the x- and y-directions, respectively, of the rectangular grid which specified the surface.

X is an array containing the x grid values in increasing order.

YMIN,YMAX are the lower and upper limits, respectively, of the grid in the y direction.

Z is a matrix containing the M*N functional values corresponding to the grid values (i.e. Z(I,J) is the surface value at the point (X(I),Y(J)) for I=1 ,...,M and J=1,...,N, where Y(J) represents the Jth equispaced y value).

IZ contains the row dimension of the array Z as declared in the calling program.

ZP is an array of 3*M*N locations stored with the various surface derivative information determined by YSURF1.

WORK is an array of 4*MD locations to be used internally for workspace.

SIGMA contains the tension factor (its sign is ignored).

The parameters M, N, X, YMIN, YMAX, Z, IZ, ZP, and SIGMA should be input unaltered from the output of YSURF1.

On Output:—

DZ contains the MD by ND array of surface values interpolated at the points of the second grid.

None of the input parameters are altered.

This function references package module SNHCSH.

I.22 Subroutine NSURF1

I.22.1 Purpose

This subroutine determines the parameters necessary to compute an interpolatory surface passing through a rectangular grid of functional values. The surface determined can be represented as the tensor product of splines under tension. For actual mapping of points onto the surface it necessary to call the function SURF2. For actual interpolation at a grid of points equally spaced in both x and y coordinates it is necessary to call subroutine NSURF2.

I.22.2 Usage

```
SUBROUTINE NSURF1 (M,N,X,Y,Z,IZ,ZP,TEMP,SIGMA,IERR)

INTEGER M,N,IZ,IERR
REAL X(M),Y(N),Z(IZ,N),ZP(M,N,3),TEMP(N+N+M),SIGMA
```

I.22.3 Arguments

On Input:—

M is the number of grid lines in the x-direction, i.e. lines parallel to the y-axis (M≥2).

N is the number of grid lines in the y-direction, i.e. lines parallel to the x-axis (N≥2).

X is an array of the M x-coordinates of the grid lines in the x-direction. These should be strictly increasing.

Y is an array of the N y-coordinates of the grid lines in the y-direction. These should be strictly increasing.

Z is an array of the M*N functional values at the grid points, i.e. Z(I,J) contains the functional value at (X(I),Y(J)) for I=1,...,M and J=1,...,N.

IZ is the row dimension of the matrix Z used in the calling program (IZ$\geq$M).

ZP is an array of at least 3*M*N locations.

TEMP is an array of at least N+N+M locations which is used for scratch storage.

SIGMA contains the tension factor. This value indicates the curviness desired. If ABS(SIGMA) is nearly zero (e.g. .001) the resulting surface is approximately the tensor product of cubic splines. If ABS(SIGMA) is large (e.g. 50.) the resulting surface is approximately bi-linear. If SIGMA equals zero tensor products of cubic splines result. A standard value for SIGMA is approximately 1.0 in absolute value.

On Output:—

ZP contains the values of the xx-, yy-, and xxyy-partial derivatives of the surface at the given nodes.

IERR contains an error flag,

= 0 for normal return,

= 1 if N is less than 2 or M is less than 2,

= 2 if the x-values or y-values are not strictly increasing.

M, N, X, Y, Z, IZ, and SIGMA are unaltered.

This subroutine references package modules CEEZ, TERMS, and SNHCSH.

I.23 Subroutine NSURF2

I.23.1 Purpose

This subroutine maps values onto a surface at every point of a grid equally spaced in both x and y coordinates. The surface interpolation is performed using a bi-spline under tension. The subroutine SURF1 or NSURF1 should be called earlier to determine certain necessary parameters.

I.23.2 Usage

```
SUBROUTINE NSURF2 (DXMIN,DXMAX,MD,DYMIN,DYMAX,ND,DZ,IDZ,
                   M,N,X,Y,Z,IZ,ZP,WORK,SIGMA)

INTEGER MD,ND,IDZ,M,N,IZ
REAL DXMIN,DXMAX,DYMIN,DYMAX,DZ(IDZ,ND),X(M),Y(N),Z(IZ,N),
               ZP(M,N,3),WORK(4,MD),SIGMA
```

I.23.3 Arguments

On Input:—

DXMIN,DXMAX contain the lower and upper limits, respectively, of the x-coordinates of the second grid.

MD contains the number of grid lines in the x direction of the second grid (MD$\geq$1).

DYMIN,DYMAX contain the lower and upper limits, respectively, of the y-coordinates of the second grid.

ND contains the number of grid lines in the y direction of the second grid (ND$\geq$1).

IDZ contains the row dimension of the array DZ as declared in the calling program.

M,N contain the number of grid lines in the x- and y-directions, respectively, of the rectangular grid which specified the surface.

X,Y are arrays containing the x- and y-grid values, respectively, each in increasing order.

Z is a matrix containing the M*N functional values corresponding to the grid values (i.e. Z(I,J) is the surface value at the point (X(I),Y(J)) for I=1,...,M and J=1,...,N).

IZ contains the row dimension of the array Z as declared in the calling program.

ZP is an array of 3*M*N locations stored with the various surface derivative information determined by SURF1.

WORK is an array of 4*MD locations to be used internally for workspace.

SIGMA contains the tension factor (its sign is ignored).

The parameters M, N, X, Y, Z, IZ, ZP, and SIGMA should be input unaltered from the output of SURF1 or NSURF1.

On Output:—

DZ contains the MD by ND array of surface values interpolated at the points of the second grid.

None of the input parameters are altered.

This function references package module SNHCSH.

I.24 Subroutine CEEZ

I.24.1 Purpose

This subroutine determines the coefficients C1, C2, and C3 used to determine endpoint slopes. Specifically, if function values Y1, Y2, and Y3 are given at points X1, X2, and X3, respectively, the quantity C1*Y1 + C2*Y2 + C3*Y3 is the value of the derivative at X1 of a spline under tension (with tension factor SIGMA) passing through the three points and having third derivative equal to zero at X1. Optionally, only two values, C1 and C2 are determined.

I.24.2 Usage

```
SUBROUTINE CEEZ (DEL1,DEL2,SIGMA,C1,C2,C3,N)

REAL DEL1,DEL2,SIGMA,C1,C2,C3
```

I.24.3 Arguments

On Input:—

DEL1 is X2-X1 (0.0).

DEL2 is X3-X1 (0.0). If N=2, this parameter is ignored.

SIGMA is the tension factor.

N is a switch indicating the number of coefficients to be returned. If N=2 only two coefficients are returned. Otherwise all three are returned.

On Output:—

C1,C2,C3 contain the coefficients.

None of the input parameters are altered.

This subroutine references package module SNHCSH.

I.25 Subroutine TERMS

I.25.1 Purpose

This subroutine computes the diagonal and superdiagonal terms of the tridiagonal linear system associated with spline under tension interpolation.

I.25.2 Usage

```
SUBROUTINE TERMS (DIAG,SDIAG,SIGMA,DEL)

REAL DIAG,SDIAG,SIGMA,DEL
```

I.25.3 Arguments

On Input:—

SIGMA contains the tension factor.

DEL contains the step size.

On Output:—

```
                (SIGMA*DEL*COSH(SIGMA*DEL) - SINH(SIGMA*DEL)
DIAG = DEL*-------------------------------------------.
                      (SIGMA*DEL)**2 * SINH(SIGMA*DEL)

                     SINH(SIGMA*DEL) - SIGMA*DEL
SDIAG = DEL*----------------------------------.
                 (SIGMA*DEL)**2 * SINH(SIGMA*DEL)
```

SIGMA and DEL are unaltered.
This subroutine references package module SNHCSH.

I.26 Subroutine SNHCSH

I.26.1 Purpose

This subroutine returns approximations to

```
SINHM(X) = SINH(X)-X
COSHM(X) = COSH(X)-1
```

and

```
COSHMM(X) = COSH(X)-1-X*X/2
```

with relative error less than 3.42E–14

I.26.2 Usage

```
SUBROUTINE SNHCSH (SINHM,COSHM,X,ISW)

INTEGER ISW
REAL SINHM,COSHM,X
```

I.26.3 Arguments

On Input:—

X contains the value of the independent variable.

ISW indicates the function desired

= –1 if only SINHM is desired,

= 0 if both SINHM and COSHM are desired,

= 1 if only COSHM is desired,

= 2 if only COSHMM is desired,

= 3 if both SINHM and COSHMM are desired.

On Output:—

SINHM contains the value of SINHM(X) if ISW$\leq$0 or ISW= 3 (SINHM is unaltered if ISW=1 or ISW=2).

COSHM contains the value of COSHM(X) if ISW=0 or ISW=1 and contains the value of COSHMM(X) if ISW$\geq$2 (COSHM is unaltered if ISW=−1).

X and ISW are unaltered.

I.27 Function INTRVL

I.27.1 Purpose

This function determines the index of the interval (determined by a given increasing sequence) in which a given value lies.

I.27.2 Usage

```
FUNCTION INTRVL (T,X,N)

INTEGER N
REAL T,X(N)
```

I.27.3 Arguments

On Input:—

T is the given value.

X is a vector of strictly increasing values.

N is the length of X (N$\geq$2).

On Output:—

INTRVL returns an integer I such that

I=1 if T$\leq$X(2),

I=N-1 if X(N-1)$\leq$T,

Otherwise X(I)$\leq$T$\leq$X(I+1)

None of the input parameters are altered.

J The System Plot Package

J.1 Purpose

The System Plot Package provides low-level primitive routines to support the operation of the NCAR Utilities. Its capabilities include scaling, line and character drawing and the provision of status information.

J.2 Coordinates

Many of the plot package routines can accept either integer or floating point input coordinates. The action of these routines depend on the type of arguments supplied. See Section 5.5 for a complete explanation of this feature.

J.3 Subroutine ININCR

J.3.1 Purpose

Sets up the interface between DI3000 and the System Plot Package, initializing all variables in the system common block `/NCAR/`. This routine *must* be called from the users program:—

- After DI3000 has been initialized.
- After each call to `JWINDO` or `JVPORT`.
- After each and every change in Metafile status (on or off).

It must *not* be called while any segments are open.

J.3.2 Usage

```
CALL ININCR(MONOFF)
```

J.3.3 Arguments

The logical variable `MONOFF` is true if output is to beb sent to a Metafile, false otherwise.

J.4 Subroutine SET

J.4.1 Purpose

`SET` establishes a mapping between an area of the plotter and the user's ("world" coordinate) plane. `SET` does no plotting. The mapping equation used as follows:—

```
I = AX + B             J = CY + D
```

where `(I,J)` is a point on the plotter corresponding to the point `(X,Y)` on the user's ("world" coordinate) plane. `B` and `D` are initially set to 0.0. Both `I` and `J` are in the range 1 to 1024, unless the plotter resolution has been altered by a call to `SETI`.

NOTE:— this scaling will be in effect for all subsequent plotting until changed by another call to `SET`.

J.4.2 Usage

```
CALL SET (XA,XB,YA,YB,XC,XD,YC,YD,LTYPE)
```

J.4.3 Arguments

XA,XB,YA,YB Numbers between 0.0 and 1.0 defining the portion of the plotter to be used. (0.,1.,0.,1.) specifies the entire area. the following restrictions apply:—

$$0.0 \leq \mathtt{XA} \leq \mathtt{XB} \leq 1.0$$

$$0.0 \leq \mathtt{YA} \leq \mathtt{YB} \leq 1.0$$

Alternate integer arguments MXA, MXB, MYA, MYB may be used.

XC,XD,YC,YD Numbers indicating minimum and maximum values of the array to be plotted in the plotter area specified by arguments 1-4.

XC maps to XA. XD maps to XB. YC maps to YA. YD maps to YB.

The only restrictions are that

$$\mathtt{XC} \neq \mathtt{XD}$$

$$\mathtt{YC} \neq \mathtt{YD}$$

That is, XD may be larger or smaller than XC, similarly, YD may be larger or smaller than YC.

LTYPE In the original NCAR package, a flag indicating log or linear mapping. In the ARL implementation, only linear mapping is available and hence LTYPE is redundant.

NOTE:— If set is not called the default provides the same scaling as

```
CALL SET (0.,1.,0.,1.,0.,1.,0.,1.,1).
```

J.5 Subroutine SETI

J.5.1 Purpose

SETI changes the resolution of the plotter.

J.5.2 Purpose

```
CALL SETI (LX,LY)
```

J.5.3 Arguments

After calling SETI, all integer coordinates are assumed to be in the range

$$1 \leq \mathtt{MX} \leq 2^{\mathtt{LX}}$$

$$1 \leq \mathtt{MY} \leq 2^{\mathtt{LY}}$$

Thus, SETI can be used to make the plot package compatible with software written for any resolution graphics device.

```
CALL SETI (10,10)
```

returns the plot package to its original state.

NOTE:— The range of LX and LY is

$$1 \leq \mathtt{LX}, \mathtt{LY} \leq 15$$

J.6 Subroutine FL2INT

J.6.1 Purpose

Maps floating point numbers into integer plotting space without plotting.

J.6.2 Usage

```
CALL FL2INT (X,Y,MX,MY)
```

J.6.3 Arguments

For input floating coordinates (X,Y), an integer coordinate (MX,MY) is returned giving the plotter space location of (X,Y) based on the most recent SET call. Nothing is plotted. MX and MY are in the range 0 to 32767 regardless of any SETI call.

J.7 Subroutine FRSTPT

J.7.1 Purpose

To establish the base point of a line. No plotting is done by FRSTPT.

J.7.2 Usage

```
CALL FRSTPT (X,Y)
```

J.7.3 Arguments

X and Y are floating point or integer coordinates of the beginning point of a line.

J.8 Subroutine VECTOR

J.8.1 Purpose

VECTOR plots a straight line segment from the (x,y) coordinates of the immediately previous plotter instruction to the coordinates given by (X,Y). A curve is often drawn with an initial call to FRSTPT followed by as many successive calls to VECTOR as there are points along the line.

J.8.2 Usage

```
CALL VECTOR (X,Y)
```

J.8.3 Arguments

X and Y are floating point or integer coordinates of an end point of a line.

J.9 Subroutine LINE

J.9.1 Purpose

To draw a straight line between two points.

J.9.2 Usage

```
CALL LINE (XA,YA,XB,YB)
```

J.9.3 Arguments

XA,YA Coordinates of beginning point.

XB,YB Coordinates of end point.

J.10 Subroutine POINTS

J.10.1 Purpose

POINTS draws a series of markers at the set of coordinates specified in the arguments. The coordinates can be connected with lines. If integers are used in the arrays, they will be plotted without scaling

J.10.2 Usage

```
CALL POINTS (X,Y,N,ICHAR,IPEN)
```

J.10.3 Arguments

X An array of x coordinates.

Y An array of y coordinates.

N The number of elements in X and Y.

ICHAR If zero, markers (of the marker type currently selected in DI3000) are drawn at the coordinates. Otherwise, ICHAR is the character to be drawn at each point. Note that this latter capability has not been implemented (yet?) in the ARL version.

IPEN If zero, only the points or characters are drawn. If 1, besides the points straight line segments are drawn connecting the coordinates ((X(1),Y(1)) to (X(2),Y(2)), (X(2),Y(2)) to (X(3),Y(3)), etc.) using the current dash pattern.

J.11 Subroutine CURVE

J.11.1 Purpose

Curve draws a series of straight line segments connecting the points specified in the arguments. The routine assumes the input arrays are ordered. If integer arrays are input, they will be plotted without scaling

J.11.2 Usage

```
CALL CURVE (X,Y,N)
```

J.11.3 Arguments

X An array of x coordinates.

Y An array of y coordinates.

N The number of points on the curve.

J.12 Subroutine PLOTIT

J.12.1 Purpose

To plot lines with only integer input.

J.12.2 Usage

```
CALL PLOTIT (NX,NY,NPEN)
```

J.12.3 Arguments

NX,NY Coordinates in the range 0 to 32767

NPEN Zero for a pen up move (like FRSTPT). One for a pen down move.

J.13 Subroutine FRSTPT

J.13.1 Purpose

PWRIT writes character information on the plotter.

J.13.2 Usage

```
CALL PWRIT (X,Y,ICHARS,N,ISIZ,IOR,ICENT)
```

J.13.3 Arguments

X,Y Coordinates of the characters to be written, see ICENT, below.

ICHARS An integer variable containing the characters to be written.

N The number of characters to be written. ICHARS should be dimensioned appropriately if more than NCPW characters are to be written. (NCPW = number of characters per word on the computer, 4 on Elxsi 6400.)

ISIZ Character width in plotter address units. However, if ISIZ≤3, then the following:—

ISIZ	Width
0	8
1	12
2	16
3	24

IOR Character string orientation in degrees counter-clockwise from horizontal.

ICENT Centering option.
If ICENT=-1 then (X,Y) is the center of the left edge of the first character.
If ICENT=0 then (X,Y) is the center of the entire string.
If ICENT=1 then (X,Y) is the center of the right edge of the last character.

J.14 Subroutine FRAME

J.14.1 Purpose

FRAME advances the plotter one frame. On non-plotter display devices, it triggers whatever action is appropriate for termination of the current pixt·ure and initiation of the next picture.

J.14.2 Usage

```
CALL FRAME
```

J.14.3 Arguments

None

J.15 Subroutine GETSET

J.15.1 Purpose

Returns the most recent SET parameters.

J.15.2 Usage

```
CALL GETSET (MXA,MXB,MYA,MYB,XC,XD,YC,YD,LTYPE)
```

J.15.3 Arguments

See the description of SET for an explanation of the meanings of the arguments. MXA, MXB, MYA, MYB are always returned as integers, even if SET was called with floating point values.

J.16 Subroutine GETSI

J.16.1 Purpose

Returns most recent SETI parameters

J.16.2 Usage

```
CALL GETSI (LX,LY)
```

J.16.3 Arguments

LX and LY are the most recent SETI parameters (10 and 10 if SETI has not been called).

J.17 Subroutine MXMY

J.17.1 Purpose

MXMY returns the current pen location.

J.17.2 Usage

```
CALL MXMY (MX,MY)
```

J.17.3 Arguments

(MX,MY) is returned by MXMY and is the current pen position in plotter address units in the range $1 \leq MX \leq 1024$, $1 \leq MY \leq 1024$. If SETI has been called, the range is $1 \leq MX \leq 2^{LX}$, $1 \leq MY \leq 2^{LY}$.

J.18 Subroutine PERIM

J.18.1 Purpose

Draws a perimeter line (border) around the plotter area.

J.18.2 Usage

```
CALL PERIM (MAJRX,MINRX,MAJRY,MINRY)
```

J.18.3 Arguments

In the original NCAR version, this routine is set up to draw tick marks along the perimeter according to the values of its four arguments. This capability is not implemented in the ARL version, and hence the arguments are redundant.

K Non-Graphics NCAR Routines

K.1 General Routines

The following table lists the many non-graphics routines included on the NCAR distribution tape. Each entry consists of the routine name and a short description of the purpose of the routine. Each routine has an extension of either `.loc` or `.port` indicating whether the routine was originally part of either the NCAR "local" or "portable" libraries. In general, the `.port` versions are considered more portable, although little trouble has been encountered in getting any of these routines to run on the Elxsi. An entry in the "test" column indicates that a separate routine with a `.test` extension is available for testing the functionality of the parent routine.

All the NCAR routines are available on the ANSI labeled tape M/194 (volume label "0194") on the Elxsi system.

Routine Name	.loc	.port	.test	Description
AQUAD	†		†	Evaluates $\int_b^a f(x)dx$ with specified error limits — uses ACM Algorithm number 468.
ALFPACK	†			Calculates Legendre functions of the first kind.
AMI	†		†	Solves systems of n first order ordinary differential equations using the Adam-Moulton fourth order method.
AUTOGRAPH	†		†	The Autograph package — see the description in Section 3.8.
BESC	†		†	Calculates Bessel functions I and J for complex arguments — *integer orders only*.
BESR	†		†	As for BESC but for real arguments.
BESSKA	†			Computes modified Bessel Functions K_A of Z for Z complex and A real.
BIVAR	†		†	Carries out bivariate interpolation and smooth surface fitting to irregularly distributed data.
CAIRY	†			Computes the Airy Function for an arbitrary complex argument.
CUBSPL	†		†	One- and two-dimensional cubic spline routine — various boudary conditions.
DATEJ	†		†	Converts the number of hours since December 31^{st}, 1920, to the form year, month, day, time (see also HOURS).

Routine Name	.loc	.port	.test	Description
EZMAPDAT		†		Digitized world map data base — used by EZMAP (Note: This data base must be converted to binary form using the program `CONVRT.F` contained in `EZMAP.F`).
FDPACN	†			Calculates "tap-weights" of the impulse response of discrete-time filters.
FDPACR	†			Generates continuous- and discrete-time filters and plots their characteristics.
FFAFFS	†			A fast fourier analysis routine.
FFT	†		†	Fast Fourier transform routines for data of arbitrary length.
FFTPACK	†			Fast Fourier transform routine package for complex and real periodic sequences.
FINDIF	†			Generates finite difference formulæ for the m^{th} derivative of a function.
FINPDF	†			Generates finite difference formulæ for mixed partial derivatives of a function.
FISHPACK??		†		A series of routines for solving elliptic partial differential equations — see separate table for a description.
GAUSL	†		†	Calculates Gauss-Legendre weights and abscissa.
GAUSS	†		†	Calculates Gaussian quadrature abscissa and weights.
HIORDQ	†			Function to approximate the integral of a function specified as an equispaced array — uses trapezoidal rule and Richardson extrapolation.
HOURS	†		†	Converts a time in the form year, month, day, hour, to the number of hours since December 31^{st}, 1920 (see also DATEJ).
HSHSLV	†			Calculates the solution vector for an overdetermined set of linear equations using least squares.
IFTRAN		†		Preprocessor for FORTRAN programs allowing conditional compilation etc. — machine independent (FORTRAN) version.
IFTRANMI		†		Tailorable version of IFTRAN — written in IFTRAN.

Routine Name	.loc	.port	.test	Description
LSPOLY	†		†	Calculates weighted least-squares variable order polynomial approximation to data.
LTPBVP	†			Calculates the solution to linear two point boundary value problems to either second or fourth order.
NLLSSQ	†			Minimizes the Euclidian norm of a vector.
READLX	†		†	A general purpose data-directed FORTRAN I/O program.
RKI	†		†	Generates fourth order Runge Kutta solutions.
RKFPDE	†			Integrates a system of partial differential equations.
SIMPSN	†		†	Simpsons rule integration for equally or unequally spaced data.
SPLPAK	†		†	Fits multidimensional cubic splines to arbitrary data using a least squares approach.
SURPLS	†		†	Generates least squares solutions to overdetermined linear systems.

K.2 The "Fishpack" Package

This packege was included on the NCAR distribution tape. It consists of a series of routines for solving separable elliptic partial differential equations. The package also includes general purpose matrix solution routines and sample programs which are designed for use by the pde solver routines. For more information, see file FSHPKINDX.PORT on the NCAR distribution tape. The following table provides a *brief* description of the contents of the package. The files are named FISHPACKn for n less than 10 and FISHPAKn for n greater than 10.

FILE NUMBER	CONTENTS OF FILE
1	File directory for FISHPACK — Version 3.1.
2	Subroutine HWSCRT solves the standard five-point finite difference approximation to the Helmholtz equation in cartesian coordinates using a centered finite difference grid.
3	Subroutine HWSPLR solves a five-point finite difference approximation to the Helmholtz equation in polar coordinates using a centered finite difference grid.
4	Subroutine HWSCYL solves a five-point finite difference approximation to the modified Helmholtz equation in cylindrical coordinates using a centered finite difference grid.
5	Subroutine HWSSSP solves a five-point finite difference approximation to the Helmholtz equation in spherical coordinates and on the surface of the unit sphere using a centered finite difference grid.
6	Subroutine HWSCSP solves a five-point finite difference approximation to the modified Helmholtz equation in spherical coordinate assuming axisymmetry (no dependence on longitude) using a centered finite difference grid.
7	Subroutine HSTCRT solves the standard five-point finite difference approximation to the Helmholtz equation in cartesian coordinates using a staggered finite difference grid.
8	Subroutine HSTPLR solves a five-point finite difference approximation to the Helmholtz equation in polar coordinates using a staggered finite difference grid.
9	Subroutine HSTCYL solves a five-point finite difference approximation to the modified Helmholtz equation in cylindrical coordinates using a staggered finite difference grid.

FILE NUMBER	CONTENTS OF FILE
10	Subroutine HSTSSP solves a five-point finite difference approximation to the Helmholtz equation in spherical coordinates and on the surface of the unit sphere using a staggered finite difference grid.
11	Subroutine HSTCSP solves a five-point finite difference approximation to the modified Helmholtz equation in spherical coordinate assuming axisymmetry (no dependence on longitude) using a staggered finite difference grid.
12	Subroutine HW3CRT solves the standard seven-point finite difference approximation to the Helmholtz equation in cartesian coordinates using a centered finite difference grid.
13	Subroutine SEPX4 automatically discretizes and solves second and (optionally) fourth order finite difference approximations on a uniform grid to certain separable elliptic partial differential equations with constant coefficients in one direction.
14	Subroutine SEPELI automatically discretizes and solves second and (optionally) fourth order finite difference approximations on a uniform grid to a separable elliptic partial differential equation on a rectangle.
15	Subroutine GENBUN solves the linear system of equations that results from a finite difference approximation on a centered grid to certain two-dimensional elliptic partial differential equations with constant coefficients in one direction.
16	Subroutine BLKTRI solves block tridiagonal linear systems that arise from finite difference approximations to separable two-dimensional elliptic partial differential equations.
17	Subroutine POISTG solves a block tridiagonal linear system of equations that arises from finite difference approximations on a staggered grid to two-dimensional elliptic partial differential equations with constant coefficients in one direction.
18	Subroutine POIS3D solves a block tridiagonal linear system of equations that arises from finite difference approximations to three-dimensional elliptic partial differential equations in a box.
19	Subroutine CMGNBN solves a complex block tridiagonal linear system arising from finite difference approximations to separable complex two-dimensional elliptic partial differential equations.
20	Subroutine CBLKTRI solves a complex block tridiagonal linear system of equations arising from finite difference approximatio to separable complex two-dimensional elliptic partial differential equations.

FILE NUMBER	CONTENTS OF FILE
21	A package of fast Fourier transforms.
22	Documentation for the package of fast Fourier transforms in the preceeding file.
23	GNBNAUX is a file of internal subroutines used by subroutines GENBUN and POISTG.
24	COMF is a file containing machine dependent constants. If this package is to be used on machines other than the Control Data 6000 or 7000 series, these constants should be changed.
25	Sample program for subroutine HWSCRT.
26	Sample program for subroutine HWSPLR.
27	Sample program for subroutine HWSCYL.
28	Sample program for subroutine HWSSSP.
29	Sample program for subroutine HWSCSP.
30	Sample program for subroutine HSTCRT.
31	Sample program for subroutine HSTPLR.
32	Sample program for subroutine HSTCYL.
33	Sample program for subroutine HSTSSP.
34	Sample program for subroutine HSTCSP.
35	Sample program for subroutine HW3CRT.
36	Sample program for subroutine SEPX4.
37	Sample program for subroutine SEPELI.
38	Sample program for subroutine GENBUN.
39	Sample program for subroutine BLKTRI.
40	Sample program for subroutine POISTG.
41	Sample program for subroutine POIS3D.
42	Sample program for subroutine CMGNBN.
43	Sample program for subroutine CBLKTRI.

DISTRIBUTION

AUSTRALIA

Department of Defence

Defence Central
- Chief Defence Scientist
- Assist Chief Defence Scientist, Operations (shared copy)
- Assist Chief Defence Scientist, Policy (shared copy)
- Director, Departmental Publications
- Counsellor, Defence Science (London) (Doc Data sheet only)
- Counsellor, Defence Science (Washington) (Doc Data sheet only)
- S.A. to Thailand Military R and D Centre (Doc Data sheet only)
- S.A. to the DRC (Kuala Lumpur) (Doc Data sheet only)
- OIC TRS, Defence Central Library
- Document Exchange Centre, DISB (18 copies)
- Joint Intelligence Organisation
- Librarian H Block, Victoria Barracks, Melbourne
- Director General - Army Development (NSO) (4 copies)

Aeronautical Research Laboratory
- Director
- Library
- Superintendent - Aerodynamics and Aero Propulsion
- Head - Aerodynamics Branch (5 copies)
- Head - Aeropropulsion Branch (5 copies)
- Divisional File - Aerodynamics and Aero Propulsion
- Aircraft Materials Division (5 copies)
- Aircraft Structures Division (5 copies)
- Aircraft Systems Division (5 copies)
- Engineering Facilities Division (3 copies)
- Mathematical Research and Computer Services (3 copies)
- Author: B.D. Fairlie

Materials Research Laboratory
- Director/Library

Defence Science & Technology Organisation - Salisbury
- Library

Navy Office
- Navy Scientific Adviser (3 copies Doc Data sheet)

Army Office
- Scientific Adviser - Army (Doc Data sheet only)

Air Force Office
- Air Force Scientific Adviser (Doc Data sheet only)
- Aircraft Research and Development Unit
  - Scientific Flight Group
  - Library

Universities and Colleges

NSW

Library, Australian Defence Force Academy

SPARES (10 copies)

TOTAL (78 copies)

AL 149
REVISED DECEMBER 87

DEPARTMENT OF DEFENCE

DOCUMENT CONTROL DATA

PAGE CLASSIFICATION
UNCLASSIFIED

PRIVACY MARKING

1a. AR NUMBER	1b. ESTABLISHMENT NUMBER	2. DOCUMENT DATE	3. TASK NUMBER
AR-004-593	ARL-AERO-TM-394	JANUARY 1988	DST 85/025

4. TITLE
IMPLEMENTATION OF THE NCAR GRAPHICS PACKAGE AT ARL

5. SECURITY CLASSIFICATION
(PLACE APPROPRIATE CLASSIFICATION IN BOX (S) IE. SECRET (S), CONFIDENTIAL (C), RESTRICTED (R), UNCLASSIFIED (U).)

DOCUMENT	TITLE	ABSTRACT
U	U	U

6. No. PAGES
132

7. No. REFS.
5

8. AUTHOR (S)
B.D. FAIRLIE

9. DOWNGRADING/DELIMITING INSTRUCTIONS

10. CORPORATE AUTHOR AND ADDRESS

AERONAUTICAL RESEARCH LABORATORY
P.O. BOX 4331, MELBOURNE VIC. 3001

11. OFFICE/POSITION RESPONSIBLE FOR
SPONSOR DSTO
SECURITY
DOWNGRADING
APPROVAL

12. SECONDARY DISTRIBUTION (OF THIS DOCUMENT)
Approved for public release.

OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH ASDIS, DEFENCE INFORMATION SERVICES BRANCH, DEPARTMENT OF DEFENCE, CAMPBELL PARK, CANBERRA, ACT 2601

13a. THIS DOCUMENT MAY BE ANNOUNCED IN CATALOGUES AND AWARENESS SERVICES AVAILABLE TO.....
No limitations

13b. CITATION FOR OTHER PURPOSES (IE. CASUAL ANNOUNCEMENT) MAY BE [X] UNRESTRICTED OR [ ] AS FOR 13a.

14. DESCRIPTORS
Computer Graphics

15. DRDA SUBJECT CATEGORIES
0045E

16. ABSTRACT
The NCAR graphics package consists of several graphics utilities and their supporting routines which are designed to provide a wide range of high-level graphics functions. Its capabilities range from simple contouring, labelling and smoothing of lines, through to vector and velocity field plotting and a mapping facility for any part of the Earth. This document describes the philosophy used in implementing the package at ARL, the mechanisms for using the package, and the capabilities of the routines currently implemented. It represents the current status of the package as at August 1987.

PAGE CLASSIFICATION
UNCLASSIFIED
PRIVACY MARKING

THIS PAGE IS TO BE USED TO RECORD INFORMATION WHICH IS REQUIRED BY THE ESTABLISHMENT FOR ITS OWN USE BUT WHICH WILL NOT BE ADDED TO THE DISTIS DATA UNLESS SPECIFICALLY REQUESTED.

16. ABSTRACT (CONT.)		
17. IMPRINT AERONAUTICAL RESEARCH LABORATORY, MELBOURNE		
18. DOCUMENT SERIES AND NUMBER AERODYNAMICS TECHNICAL MEMORANDUM 394	19. COST CODE 545006	20. TYPE OF REPORT AND PERIOD COVERED
21. COMPUTER PROGRAMS USED		
22. ESTABLISHMENT FILE REF. (S)		
23. ADDITIONAL INFORMATION (AS REQUIRED)		