

# **DISLIN 11.5**

**A Data Plotting Extension**

**for the**

**Programming Language**

**Go**

**by**

**Helmut Michels**



# Contents

|          |                                                         |          |
|----------|---------------------------------------------------------|----------|
| <b>1</b> | <b>Overview</b>                                         | <b>1</b> |
| 1.1      | Introduction . . . . .                                  | 1        |
| 1.2      | Dislin Features . . . . .                               | 1        |
| 1.3      | Passing Parameters from Go to Dislin Routines . . . . . | 2        |
| 1.4      | FTP Site, Dislin Home Page . . . . .                    | 2        |
| 1.5      | Reporting Bugs . . . . .                                | 2        |
| <b>A</b> | <b>Short Description of Dislin Routines</b>             | <b>3</b> |
| A.1      | Initialization and Introductory Routines . . . . .      | 3        |
| A.2      | Termination and Parameter Resetting . . . . .           | 4        |
| A.3      | Plotting Text and Numbers . . . . .                     | 5        |
| A.4      | Colours . . . . .                                       | 5        |
| A.5      | Fonts . . . . .                                         | 6        |
| A.6      | Symbols . . . . .                                       | 6        |
| A.7      | Axis Systems . . . . .                                  | 7        |
| A.8      | Secondary Axes . . . . .                                | 8        |
| A.9      | Modification of Axes . . . . .                          | 8        |
| A.10     | Axis System Titles . . . . .                            | 9        |
| A.11     | Plotting Data Points . . . . .                          | 9        |
| A.12     | Legends . . . . .                                       | 10       |
| A.13     | Line Styles and Shading Patterns . . . . .              | 11       |
| A.14     | Cycles . . . . .                                        | 12       |
| A.15     | Base Transformations . . . . .                          | 12       |
| A.16     | Shielding . . . . .                                     | 12       |
| A.17     | Parameter Requesting Routines . . . . .                 | 12       |
| A.18     | Elementary Plot Routines . . . . .                      | 14       |
| A.19     | Conversion of Coordinates . . . . .                     | 15       |
| A.20     | Utility Routines . . . . .                              | 15       |
| A.21     | Binary File I/O . . . . .                               | 16       |
| A.22     | Date Routines . . . . .                                 | 16       |
| A.23     | Cursor Routines . . . . .                               | 17       |
| A.24     | Transparency . . . . .                                  | 17       |
| A.25     | Bar Graphs . . . . .                                    | 17       |
| A.26     | Pie Charts . . . . .                                    | 18       |
| A.27     | Coloured 3-D Graphics . . . . .                         | 19       |
| A.28     | 3-D Graphics . . . . .                                  | 20       |
| A.29     | Geographical Projections . . . . .                      | 22       |
| A.30     | Contouring . . . . .                                    | 23       |
| A.31     | Image Routines . . . . .                                | 24       |
| A.32     | Window Routines . . . . .                               | 25       |
| A.33     | Widget Routines . . . . .                               | 25       |
| A.34     | Dislin Quickplots . . . . .                             | 28       |

|                                           |           |
|-------------------------------------------|-----------|
| A.35 Using Threads . . . . .              | 28        |
| A.36 Reading FITS Files . . . . .         | 28        |
| A.37 MPS Logo . . . . .                   | 28        |
| <b>B Examples</b>                         | <b>29</b> |
| B.1 Demonstration of CURVE . . . . .      | 30        |
| B.2 Symbols . . . . .                     | 32        |
| B.3 Logarithmic Scaling . . . . .         | 34        |
| B.4 Interpolation Methods . . . . .       | 36        |
| B.5 Line Styles . . . . .                 | 38        |
| B.6 Legends . . . . .                     | 40        |
| B.7 Shading Patterns (AREAF) . . . . .    | 42        |
| B.8 Vectors . . . . .                     | 44        |
| B.9 3-D Colour Plot . . . . .             | 46        |
| B.10 Surface Plot . . . . .               | 48        |
| B.11 Surface Plot . . . . .               | 50        |
| B.12 Polar Plots . . . . .                | 52        |
| B.13 Contour Plot . . . . .               | 55        |
| B.14 Shaded Contour Plot . . . . .        | 58        |
| B.15 Pie Charts . . . . .                 | 60        |
| B.16 World Coastlines and Lakes . . . . . | 62        |

# Chapter 1

## Overview

### 1.1 Introduction

This manual describes a data plotting extension for the programming language Go. The plotting extension is based on the data plotting library Dislin that is available for several C, Fortran 77 and Fortran 90/95 compilers.

Dislin is a high-level plotting library that contains subroutines and functions for displaying data graphically as curves, bar graphs, pie charts, 3-D colour plots, surfaces, contours and maps. The library contains about 800 plotting and parameter setting routines which are now available from Go.

### 1.2 Dislin Features

The following features are supported by Dislin:

- Several output formats can be selected such as X11, PostScript, PDF, CGM, WMF, PNG, BMP, PPM, GIF, TIFF and HPGL.
- 9 software fonts are available where each font provides 6 alphabets and special european characters. Hardware fonts for PostScript printers and X11 and Windows displays can also be used.
- Plotting of two- and three-dimensional axis systems. Axes can be linearly or logarithmically scaled and labeled with linear, logarithmic, date, time, map and user-defined formats.
- Plotting of curves. Several curves can appear in one axis system and can be differentiated by colour, line style and pattern. Multiple axis systems can be displayed on one page.
- Plotting of legends.
- Elementary plot routines for lines, vectors and outlined or filled regions such as rectangles, circles, arcs, ellipses and polygons.
- Shielded regions can be defined.
- Business graphics.
- 3-D colour graphics.
- 3-D graphics.
- Elementary image routines.
- Geographical projections and plotting of maps.
- Contouring.

## 1.3 Passing Parameters from Go to Dislin Routines

The passing of parameters from Go to Dislin routines is not so strict as in other programming languages. The following rules are applied:

- Parameters can be passed from Go to Dislin routines as variables, constants and expressions.
- String constants must be enclosed in a pair of quotation marks.
- Floating point parameters can be passed from Go to Dislin as integer and floating point numbers.
- Integer parameters can be passed from Go to Dislin as integer and floating point values. If a floating point value is passed for an integer parameter, the fractional part of the floating point value will be truncated.
- The address of the first array element must be passed to Dislin for an array parameter. One-dimensional or multi-dimensional arrays can be used. The type of the array elements must be float64 for floating point arrays, int32 for integer arrays and int8 or uint8 for byte arrays.
- Memory must be allocated for arrays in a parameter list which are used as output parameters from Dislin.

Note: Normally, the number and meaning of parameters passed to Dislin routines are identical with the syntax description of the routines in the Dislin manual.

## 1.4 FTP Site, Dislin Home Page

The Dislin software is available via ftp anonymous from the site:

`ftp://ftp.gwdg.de/pub/grafik/dislin`

The Dislin home page is:

`https://www.dislin.de`

The Go home page is:

`https://www.golang.org`

## 1.5 Reporting Bugs

Dislin is well tested by many users and should be very bug free. However, no software is perfect. If you have any problems with Dislin, contact the author:

Helmut Michels  
Dislin Software  
Am Hachweg 10  
37083 Göttingen, Germany  
Email: [michels@dislin.de](mailto:michels@dislin.de)

# Appendix A

## Short Description of Dislin Routines

This appendix presents a short description of all Dislin routines that can be called from Go. A complete description of the routines can be found in the Dislin manual or via the online help of Dislin. For parameters, the following conventions are used:

- int variables begin with the character n or i;
- strings and character arrays begin with the character c. c<sub>ray</sub> means a byte array of unsigned int8 values and c<sub>buf</sub> an array of signed int8 values. Other variables beginning with the character c are strings;
- all other variables are floating point numbers;
- one-dimensional arrays end with the keyword 'ray', two-dimensional arrays with the keyword 'mat'. The address of the first array element must be passed to Dislin. 32-bit int arrays (int32) and 64-bit float arrays (float64) are required.

### A.1 Initialization and Introductory Routines

| Routine                        | Meaning                                                           |
|--------------------------------|-------------------------------------------------------------------|
| Bmpmod (n, cval, copt)         | defines the physical resolution of BMP files.                     |
| Bufmod (cmod, ckey)            | modifies the behaviour of the output buffer.                      |
| Cgmbgd (xr, xg, xb)            | defines the background colour for CGM files.                      |
| Cgmpic (cstr)                  | sets the picture ID for CGM files.                                |
| Disenv (cenv)                  | defines the Dislin environment.                                   |
| Disini ()                      | initializes Dislin.                                               |
| Erase ()                       | clears the screen.                                                |
| Errdev (cdev)                  | defines the error device.                                         |
| Errfil (cfil)                  | sets the name of the error file.                                  |
| Errmod (ckey, copt)            | modifies the printing of error messages.                          |
| Filbox (nx, ny, nw, nh)        | defines the position and size of included metafiles.              |
| Filmod (cmode)                 | defines the file creation mode.                                   |
| Filopt (copt, ckey)            | modifies rules for creating file versions.                        |
| iret = Filsiz (cfil, &nw, &nh) | returns the size on an image file.                                |
| iret = Filtyp (ctype)          | returns the type of a file.                                       |
| Filwin (nx, ny, nw, nh)        | defines a rectangle of the image that will be included by Incfil. |
| Gifmod (cmod, ckey)            | enables transparency for GIF files.                               |
| Hpgmod (cmod, ckey)            | defines options for HPGL files.                                   |
| Hworig (nx, ny)                | defines the origin of the PostScript hardware page.               |

| <b>Routine</b>              | <b>Meaning</b>                                      |
|-----------------------------|-----------------------------------------------------|
| Hpgmod (cmod, ckey)         | defines options for HPGL files.                     |
| Hworig (nx, ny)             | defines the origin of the PostScript hardware page. |
| Hwpage (nw, nw)             | defines the size of the PostScript hardware page.   |
| Hwscal (xscl)               | modifies the scale operator in PostScript files.    |
| Imgfmt (copt)               | defines the format of image files.                  |
| Incfil (cfil)               | includes metafiles into a graphics.                 |
| Metafl (cfmt)               | defines the plotfile format.                        |
| Newpag ()                   | creates a new page.                                 |
| Origin (nx, ny)             | defines the origin.                                 |
| Page (nw, nh)               | sets the page size.                                 |
| Pagera ()                   | plots a page border.                                |
| Pagfl (iclr)                | fills the page with a colour.                       |
| Paghdr (c1, c2, iopt, idir) | plots a page header.                                |
| Pagmod (copt)               | selects a page rotation.                            |
| Pagorg (copt)               | defines the origin of the page.                     |
| n = Pdfbuf (cbuf, nmax)     | copies a PDF file to a string.                      |
| Pdfmod (cmod, ckey)         | defines PDF options.                                |
| Pdfmrk (cstr, copt)         | defines kookmarks for PDF files.                    |
| Pngmod (cmod, ckey)         | enables transparency for PNG files.                 |
| Sclfac (x)                  | defines a scaling factor for the entire plot.       |
| Sclmod (copt)               | defines a scaling mode.                             |
| Scrmod (copt)               | swaps back- and foreground colours.                 |
| Sendbf ()                   | flushes the output buffer.                          |
| Setfil (cfil)               | sets the plotfile name.                             |
| Setpag (copt)               | selects a predefined page format.                   |
| Setxid (id, copt)           | defines an external X Window or pixmap.             |
| Symfil (cdev, cstat)        | sends a plotfile to a device.                       |
| Tifmod (n, cval, copt)      | defines the physical resolution of TIFF files.      |

Figure A.1: Initialization and Introductory Routines

## A.2 Termination and Parameter Resetting

| <b>Routine</b>      | <b>Meaning</b>                                     |
|---------------------|----------------------------------------------------|
| Units (copt)        | defines the plot units.                            |
| Wmfmod (cmod, ckey) | modifies the format of WMF files.                  |
| Delglb ()           | frees space allocated for global parameters.       |
| Disfin ()           | terminates Dislin.                                 |
| Endgrf ()           | terminates an axis system and sets the level to 1. |
| Reset (copt)        | resets parameters to default values.               |

Figure A.2: Termination and Parameter Resetting



### A.3 Plotting Text and Numbers

| Routine                  | Meaning                                                                            |
|--------------------------|------------------------------------------------------------------------------------|
| Angle (n)                | defines the character angle.                                                       |
| Chaang (x)               | defines an inclination angle for characters.                                       |
| Chaspc (x)               | affects character spacing.                                                         |
| Chawth (x)               | affects the width of characters.                                                   |
| Fixspc (x)               | sets a constant character width.                                                   |
| Frmess (nfrm)            | defines the thickness of text frames.                                              |
| Height (n)               | defines the character height.                                                      |
| Messag (cstr, nx, ny)    | plots text.                                                                        |
| Mixalf ()                | enables control signs in character strings for plotting indices and exponents.     |
| Newmix ()                | defines an alternate set of control characters for plotting indices and exponents. |
| n = Nmless (cstr)        | returns the length of character strings in plot coordinates.                       |
| Number (x, ndig, nx, ny) | plots floating point numbers.                                                      |
| Numfmt (copt)            | determines the format of numbers.                                                  |
| Numode (c1, c2, c3, c4)  | modifies the appearance of numbers.                                                |
| Rlmess (cstr, x, y)      | plots text.                                                                        |
| Rlnumb (x, ndig, xp, yp) | plots numbers.                                                                     |
| Setbas (xfac)            | determines the position of indices and exponents.                                  |
| Setexp (xfac)            | sets the height of indices and exponents.                                          |
| Setmix (char, cmix)      | defines global control signs for plotting indices and exponents.                   |
| Texmod (cmode)           | enables TeX mode for plotting mathematical formulas.                               |
| Texopt (copt, ctype)     | defines TeX options.                                                               |
| Texval (x, copt)         | defines TeX values.                                                                |
| Txtbgd (nclr)            | defines a background colour for text and numbers.                                  |
| Txtjus (copt)            | defines the alignment of text and numbers.                                         |

Figure A.3: Plotting Text and Numbers

### A.4 Colours

| Routine                            | Meaning                              |
|------------------------------------|--------------------------------------|
| Color (color)                      | defines colours.                     |
| Hsvrgb (xh, xs, xv, &xr, &xg, &xb) | converts HSV to RGB coordinates.     |
| n = Indrgb (xr, xg, xb)            | calculates a colour index.           |
| n = Intrgb (xr, xg, xb)            | calculates an explicit colour value. |
| Myvlt (rray, gray, bray, n)        | changes the current colour table.    |
| Rgbhsv (xr, xg, xb, &xh, &xs, &xv) | converts RGB to HSV coordinates.     |
| Setclr (nclr)                      | defines colours.                     |

| <b>Routine</b>         | <b>Meaning</b>                    |
|------------------------|-----------------------------------|
| Setind (i, xr, xg, xb) | changes the current colour table. |
| Setrgb (xr, xg, xb)    | defines colours.                  |
| Setvlt (cvlt)          | selects a colour table.           |
| Vltfil (cfil, cmod)    | store or loads a colour table.    |

Figure A.4: Colours

## A.5 Fonts

| <b>Routine</b>            | <b>Meaning</b>                                           |
|---------------------------|----------------------------------------------------------|
| Basalf (calph)            | defines the base alphabet.                               |
| Bmpfnt (cfont)            | defines a bitmap font.                                   |
| Chacod (copt)             | defines the character coding.                            |
| Complx ()                 | sets a complex font.                                     |
| Duplx ()                  | sets a double-stroke font.                               |
| Disalf ()                 | sets the default font.                                   |
| Eushft (cnat, char)       | defines a shift character for European characters.       |
| Gothic ()                 | sets a gothic font.                                      |
| Helve ()                  | sets a shaded font.                                      |
| Helves ()                 | sets a shaded font with small characters.                |
| Helvetica ()              | sets a shaded font with thick characters.                |
| Hwfont ()                 | sets a standard hardware font.                           |
| Psfnt (cfont)             | sets a PostScript font.                                  |
| Psmode (cmode)            | enables Greek and Italic characters in PostScript fonts. |
| Serif ()                  | sets a complex shaded font.                              |
| Simplx ()                 | sets a single-stroke font.                               |
| Smxalf (calph, c1, c2, n) | defines shift characters for alternate alphabets.        |
| Triplx ()                 | sets a triple-stroke font.                               |
| Ttfnt (cfont)             | loads a TrueType font.                                   |
| Winfnt (cfont)            | sets a TrueType font.                                    |
| X11fnt (cfont, copt)      | sets an X11 font.                                        |

Figure A.5: Fonts

## A.6 Symbols

| <b>Routine</b>                      | <b>Meaning</b>                        |
|-------------------------------------|---------------------------------------|
| Hsymbl (n)                          | defines the height of symbols.        |
| Mysymb (xray, yray, n, isym, iflag) | defines an user-defined symbol.       |
| Rlsymb (nsym, x, y)                 | plots symbols for user coordinates.   |
| Symbol (nsym, nx, ny)               | plots symbols.                        |
| Symrot (xang)                       | defines a rotation angle for symbols. |

Figure A.6: Symbols

## A.7 Axis Systems

| Routine                                                            | Meaning                                                         |
|--------------------------------------------------------------------|-----------------------------------------------------------------|
| Addlab (cstr, v, itic, cax)                                        | plots additional single labels.                                 |
| Ax2grf ()                                                          | suppresses the plotting of the upper X- and left Y-axis.        |
| Ax3len (nxl, nyl, nzl)                                             | defines axis lengths for a coloured 3-D axis system.            |
| Axsbgd (iclr)                                                      | defines the background colour.                                  |
| Axsers ()                                                          | erases the contents of an axis system.                          |
| Axslen (nxl, nyl)                                                  | defines axis lengths for a 2-D axis system.                     |
| Axsorg (nx, ny)                                                    | determines the position of a crossed axis system.               |
| Axspos (nxp, nyp)                                                  | determines the position of axis systems.                        |
| Axstyp (ctype)                                                     | select rectangular or crossed axis systems.                     |
| Axgit ()                                                           | plots the lines $X = 0$ and $Y = 0$ .                           |
| Box2d ()                                                           | plots a border around an axis system.                           |
| Center ()                                                          | centres axis systems.                                           |
| Cross ()                                                           | plots the lines $X = 0$ and $Y = 0$ marked with ticks.          |
| Endgrf ()                                                          | terminates an axis system.                                      |
| Frame (nfrm)                                                       | defines the frame thickness of axis systems.                    |
| Frmclr (nclr)                                                      | defines the colour of frames.                                   |
| Gaxpar (v1, v2, copt, cax,<br>&a, &b, &or, &stp, &ndig)            | calculates axis parameters.                                     |
| Grace (ngrace)                                                     | affects the clipping margin of axis systems.                    |
| Graf (xa, xe, xor, xstp,<br>ya, ye, yor, ystp)                     | plots a two-dimensional axis system.                            |
| Graf3 (xa, xe, xor, xstp, ya, ye,<br>yor, ystp, za, ze, zor, zstp) | plots an axis system for colour graphics.                       |
| Grafp (xe, xor, xstp, yor, ystp)                                   | plots a polar axis system.                                      |
| Grafr (xray, n, yray, m)                                           | plots an axis system for a Smith chart.                         |
| Grdpol (nx, ny)                                                    | plots a polar grid.                                             |
| Grid (nx, ny)                                                      | overlays a grid on an axis system.                              |
| Gridim (zimg, zre1, zre2, n)                                       | plots a grid line with constant imaginary part in Smith charts. |
| Gridre (zre, zimg1, zimg2, n)                                      | plots a grid line with a constant real part in Smith charts.    |
| Noclip ()                                                          | suppresses clipping of user coordinates.                        |
| Nograf ()                                                          | suppresses the plotting of an axis system.                      |
| Polmod (cpos, cdir)                                                | modifies the appearance of polar labels.                        |
| Setgrf (c1, c2, c3, c4)                                            | suppresses parts of an axis system.                             |
| Setscl (xray, n, cax)                                              | sets automatic scaling.                                         |
| Title ()                                                           | plots a title over an axis system.                              |
| Xaxgit ()                                                          | plots the line $Y = 0$ .                                        |
| Xcross ()                                                          | plots the line $Y = 0$ and marks it with ticks.                 |
| Yaxgit ()                                                          | plots the line $X = 0$ .                                        |
| Ycross ()                                                          | plots the line $X = 0$ and marks it with ticks.                 |

Figure A.7: Axis Systems

## A.8 Secondary Axes

| Routine                                             | Meaning                                    |
|-----------------------------------------------------|--------------------------------------------|
| Xaxis (xa, xe, xor, xstp, nl, cstr, it, nx, ny)     | plots a linear X-axis.                     |
| Xaxlg (xa, xe, xor, xstp, nl, cstr, it, nx, ny)     | plots a logarithmic X-axis.                |
| Yaxis (ya, ye, yor, ystp, nl, cstr, it, nx, ny)     | plots a linear Y-axis.                     |
| Yaxlg (ya, ye, yor, ystp, nl, cstr, it, nx, ny)     | plots a logarithmic Y-axis.                |
| Ypolar (ya, yb, yor, ystp, cstr, ndist)             | plots a polar Y-axis.                      |
| Zaxis (za, ze, zor, zstp, nl, cstr, it, id, nx, ny) | plots a linearly scaled colour bar.        |
| Zaxlg (za, ze, zor, zstp, nl, cstr, it, id, nx, ny) | plots a logarithmically scaled colour bar. |

Figure A.8: Secondary Axes

## A.9 Modification of Axes

| Routine                  | Meaning                                          |
|--------------------------|--------------------------------------------------|
| Axclrs (nclr, copt, cax) | defines colours for axis elements.               |
| Axends (copt, cax)       | suppresses certain labels.                       |
| Axsscl (copt, cax)       | defines the axis scaling.                        |
| Hname (nh)               | defines the character height of axis names.      |
| Intax ()                 | defines integer numbering for all axes.          |
| Labdig (ndig, cax)       | sets the number of decimal places for labels.    |
| Labdis (ndis, cax)       | sets the distance between labels and ticks.      |
| Labels (copt, cax)       | selects labels.                                  |
| Labjus (copt, cax)       | defines the alignment of axis labels.            |
| Labmod (ckey, cval, cax) | modifies date labels.                            |
| Labpos (copt, cax)       | determines the position of labels.               |
| Labtyp (copt, cax)       | defines vertical or horizontal labels.           |
| Logtic (copt)            | modifies the appearance of logarithmic ticks.    |
| Mylab (cstr, itic, cax)  | sets user-defined labels.                        |
| Namdis (ndis, cax)       | sets the distance between axis names and labels. |
| Name (cstr, cax)         | defines axis titles.                             |
| Namjus (copt, cax)       | defines the alignment of axis titles.            |
| Noline (cax)             | suppresses the plotting of axis lines.           |
| Rgtlab ()                | right-justifies labels.                          |
| Rvynam ()                | defines an angle for Y-axis names.               |
| Ticks (ntics, cax)       | sets the number of ticks.                        |
| Ticlen (nmaj, nmin)      | sets the length of ticks.                        |

| <b>Routine</b>     | <b>Meaning</b>                                   |
|--------------------|--------------------------------------------------|
| Ticmod (copt, cax) | modifies the plotting of ticks at calendar axes. |
| Ticpos (copt, cax) | determines the position of ticks.                |
| Timopt ()          | modifies time labels.                            |

Figure A.9: Modification of Axes

## A.10 Axis System Titles

| <b>Routine</b>      | <b>Meaning</b>                           |
|---------------------|------------------------------------------|
| Htitle (nh)         | defines the character height of titles.  |
| Lfttit ()           | left-justifies title lines.              |
| Linesp (xfac)       | defines line spacing.                    |
| Titjus (copt)       | defines the alignment of titles.         |
| Title ()            | plots axis system titles.                |
| Titlin (cstr, ilin) | defines text lines for titles.           |
| Titpos (copt)       | defines the position of titles.          |
| Vkyltit (nshift)    | shifts titles in the vertical direction. |

Figure A.10: System Titles

## A.11 Plotting Data Points

| <b>Routine</b>                                               | <b>Meaning</b>                                                  |
|--------------------------------------------------------------|-----------------------------------------------------------------|
| Bars (xray, y1ray, y2ray, n)                                 | plots a bar graph.                                              |
| Bars3d (xray, yray, z1ray, z2ray,<br>xwray, ywray, icray, n) | plots 3-D bars.                                                 |
| Chnatt ()                                                    | changes curve attributes.                                       |
| chnrcv (copt)                                                | defines attributes changed automatically by CURVE.              |
| Color (color)                                                | defines the colour used for text and lines.                     |
| Crvmat (zmat, n, m, ixpts, iypts)                            | plots a coloured surface.                                       |
| Curve (xray, yray, n)                                        | plots curves.                                                   |
| Curve3 (xray, yray, zray, n)                                 | plots coloured rectangles.                                      |
| Curvx3 (xray, y, zray, n)                                    | plots rows of coloured rectangles.                              |
| Curvy3 (x, yray, zray, n)                                    | plots columns of coloured rectangles.                           |
| Errbar (xray, yray,<br>e1ray, e2ray, n)                      | plots error bars.                                               |
| Fbars (x1ray, x2ray, x3ray,<br>x4ray, x5ray, n)              | plots financial bars.                                           |
| Field (x1ray, y1ray,<br>x2ray, y2ray, n, ivec)               | plots a vector field.                                           |
| Gapcrv (xgap)                                                | defines gaps plotted by CURVE.                                  |
| Inccrv (ncrv)                                                | defines the number of curves plotted with equal attributes.     |
| Licpts (xvmat, yvmat, nx, ny,<br>itmat, wmat)                | calculates a Line Integral Convolution image of a vector field. |

| <b>Routine</b>                                                                              | <b>Meaning</b>                                  |
|---------------------------------------------------------------------------------------------|-------------------------------------------------|
| Incmrk (nmrk)                                                                               | selects symbols or lines for CURVE.             |
| Licmod (cmod, ckey)                                                                         | sets modes for the LIC algorithm.               |
| Linfit (xray, yray, n, &a, &b, &r, copt)                                                    | plots a fitted line.                            |
| Marker (nsym)                                                                               | sets the symbols plotted by CURVE.              |
| Mrkclr (nclr)                                                                               | defines the colour of symbols plotted by CURVE. |
| Nancrv (copt)                                                                               | enables handling of NaN values in curves.       |
| Nochek ()                                                                                   | suppresses listing of out of range data points. |
| Piegrf (cbuf, nlin, xray, n)                                                                | plots a pie chart.                              |
| Polcrv (copt)                                                                               | defines the interpolation method used by CURVE. |
| Resatt ()                                                                                   | resets curve attributes.                        |
| Setres (nx, ny)                                                                             | sets the size of coloured rectangles.           |
| Shdcrv (x1ray, y1ray, n1,<br>x2ray, y2ray, n2)                                              | plots shaded areas between curves.              |
| Splmod (ngrad, npts)                                                                        | modifies spline interpolation.                  |
| Stmmod (cmod, ckey)                                                                         | sets streamline modes.                          |
| Stmopt (n, ckey)                                                                            | defines integer options for streamlines.        |
| n = Stmpts (xvmat, yvmat, nx,<br>ny, xpray, ypray, x0, y0,<br>xray, yray, nmax)             | generates a streamline.                         |
| Stmtri (xvray, yvray, xpray, ypray,<br>n, i1ray, i2ray, i3ray,<br>ntri, xsray, ysray, nray) | plots streamlines from triangulated data.       |
| Stmval (x, ckey)                                                                            | defines floating point options for streamlines. |
| Stream (xvmat, yvmat, nx, ny,<br>xpray, ypray, xsray, ysray, n)                             | plots streamlines.                              |
| Thkcrv (nthk)                                                                               | defines the thickness of curves.                |
| itmat = Txture (nx, ny)                                                                     | generates a texture array for LICPTS.           |
| Vecfld (xvray, yvray,<br>xpray, ypray, n, ivec)                                             | plots a vector field.                           |
| Vecmat (xvmat, yvamt, nx, ny,<br>xpray, ypray, ivec)                                        | plots a vector field on a regular grid.         |

Figure A.11: Plotting Data Points

## A.12 Legends

| <b>Routine</b>              | <b>Meaning</b>                                                                                              |
|-----------------------------|-------------------------------------------------------------------------------------------------------------|
| Frame (nfrm)                | sets the frame thickness of legends.                                                                        |
| Legbgd (nclr)               | defines the background colour of legends.                                                                   |
| Legend (cbuf, ncor)         | plots legends. See Legini for cbuf.                                                                         |
| Legini (cbuf, nlin, nmaxln) | initializes legends. cbuf must be an int8 array with allocated space for at least nlin * nmaxln characters. |

| Routine                                     | Meaning                                                                                 |
|---------------------------------------------|-----------------------------------------------------------------------------------------|
| Leglin (cbuf, cstr, ilin)                   | stores text for a legend line in the buffer cbuf.<br>See Legini for the parameter cbuf. |
| Legopt (xf1, xf2, xf3)                      | modifies the appearance of legends.                                                     |
| Legpat (ityp, ithk, isym, iclr, ipat, ilin) | stores curve attributes.                                                                |
| Legpos (nxp, nyp)                           | determines the position of legends.                                                     |
| Legsel (nray, n)                            | selects legend lines.                                                                   |
| Legtbl (n, copt)                            | sets the number of columns in table legends.                                            |
| Legtit (ctitle)                             | defines the legend title.                                                               |
| Legtyp (ctype)                              | defines horizontal or vertical legend lines.                                            |
| Legval (x, copt)                            | modifies the appearance of legends.                                                     |
| Linesp (xfac)                               | affects line spacing.                                                                   |
| Mixleg ()                                   | enables multiple text lines in legends.                                                 |
| nxl = Nxlegn (cbuf)                         | returns the width of legends in plot coordinates.                                       |
| nyl = Nylegn (cbuf)                         | returns the height of legends in plot coordinates.                                      |

Figure A.12: Legends

### A.13 Line Styles and Shading Patterns

| Routine                           | Meaning                                                                     |
|-----------------------------------|-----------------------------------------------------------------------------|
| Chndot ()                         | sets a dotted-dashed line style.                                            |
| Chndsh ()                         | sets a dashed-dotted line style.                                            |
| Color (color)                     | sets a colour.                                                              |
| Dash ()                           | sets a dashed line style.                                                   |
| Dashl ()                          | sets a long-dashed line style.                                              |
| Dashm ()                          | sets a medium-dashed line style.                                            |
| Dot ()                            | sets a dotted line style.                                                   |
| Dotl ()                           | sets a long-dotted line style.                                              |
| Hwmode (cmod, ckey)               | enables or disables hardware features for line styles and shading patterns. |
| Linclr (nray, n)                  | defines colours for line styles.                                            |
| Lintyp (itype)                    | defines a line style.                                                       |
| Linwid (nwidth)                   | sets the line width.                                                        |
| Lncap (copt)                      | sets the line cap parameter.                                                |
| Lnjoin (copt)                     | sets the line join parameter.                                               |
| Lnmlt (xfac)                      | sets the miter limit parameter.                                             |
| Myline (nray, n)                  | sets a user-defined line style.                                             |
| Mypat (iang, ityp, idens, icross) | defines a global shading pattern.                                           |
| Penwid (nwidth)                   | sets the pen width.                                                         |
| Shdfac (xfac)                     | modifies the distance of scan lines for software shading.                   |
| Shdpat (ipat)                     | selects a shading pattern.                                                  |
| Solid ()                          | sets a solid line style.                                                    |

Figure A.13: Line Styles and Shading Patterns

## A.14 Cycles

| Routine               | Meaning                        |
|-----------------------|--------------------------------|
| Clrcyc (index, iclr)  | modifies the colour cycle.     |
| Lincyc (index, itype) | modifies the line style cycle. |
| Patcyc (index, ipat)  | modifies the pattern cycle.    |

Figure A.14: Cycles

## A.15 Base Transformations

| Routine                   | Meaning                                     |
|---------------------------|---------------------------------------------|
| Tr3axs (x, y, z, a)       | defines a rotation about an arbitrary axis. |
| Tr3res ()                 | resets 3-D base transformations.            |
| Tr3rot (xrot, yrot, zrot) | affects the 3-D rotation of plot vectors.   |
| Tr3scl (xscl, yscl, zscl) | affects the 3-D scaling of plot vectors.    |
| Tr3shf (xshf, yshf, zshf) | affects the 3-D shifting of plot vectors.   |
| Trfres ()                 | resets base transformations.                |
| Trfrot (xang, nx, ny)     | affects the rotation of plot vectors.       |
| Trfscl (xscl, yscl)       | affects the scaling of plot vectors.        |
| Trfshf (nx, ny)           | affects the shifting of plot vectors.       |

Figure A.15: Base Transformations

## A.16 Shielding

| Routine                    | Meaning                                       |
|----------------------------|-----------------------------------------------|
| Shield (careas, cmode)     | defines automatic shielding.                  |
| Shlcir (nx, ny, nr)        | defines circles as shielded areas.            |
| Shldel (id)                | deletes shielded areas.                       |
| Shlell (nx, ny, na, nb, t) | defines ellipses as shielded areas.           |
| id = Shlind ()             | returns the index of a shielded area.         |
| Shlpie (nx, ny, nr, a, b)  | defines pie segments as shielded areas.       |
| Shlpol (nxray, nyray, n)   | defines polygons as shielded areas.           |
| Shlrct (nx, ny, nw, nh, t) | defines rotated rectangles as shielded areas. |
| Shlrec (nx, ny, nw, nh)    | defines rectangles as shielded areas.         |
| Shlres (n)                 | deletes shielded areas.                       |
| Shlvis (id, cmode)         | enables or disables shielded areas.           |

Figure A.16: Shielding

## A.17 Parameter Requesting Routines

| Routine          | Meaning                                              |
|------------------|------------------------------------------------------|
| calf = Getalf () | returns the base alphabet.                           |
| n = Getang ()    | returns the current angle used for text and numbers. |



| Routine                         | Meaning                                                 |
|---------------------------------|---------------------------------------------------------|
| Getclp (&nx, &ny, &nz)          | returns the current clipping window.                    |
| n = Getclr ()                   | returns the current colour number.                      |
| Getdig (&nx, &ny, &nz)          | returns the number of decimal places used in labels.    |
| cdsp = Getdsp ()                | returns the terminal type.                              |
| cfil = Getfil ()                | returns the current plotfile name.                      |
| Getgrf (&a, &b, &or, &stp, cax) | returns the scaling of the current axis system.         |
| n = Gethgt ()                   | returns the current character height.                   |
| n = Gethnm ()                   | returns the character height of axis titles.            |
| Getind (i, &xr, &xg, &xb)       | returns the RGB coordinates for a colour index.         |
| ctype = Getlab (iax)            | returns the current label types, where iax              |
|                                 | denotes the axis and can have the values 1, 2 and 3.    |
| Getlen (&nx, &ny, &nz)          | returns the current axis lengths.                       |
| n = Getlev ()                   | returns the current level.                              |
| n = Getlin ()                   | returns the current line width.                         |
| cmfl = Getmfl ()                | returns the current file format.                        |
| c = Getmix (copt)               | returns shift characters for indices and exponents.     |
| Getor (&nx, &ny)                | returns the current origin.                             |
| Getpag (&nw, &nh)               | returns the current page size.                          |
| n = Getpat ()                   | returns the current shading pattern.                    |
| n = Getplv ()                   | returns the patch level of Dislin.                      |
| Getpos (&nx, &ny)               | returns the position of the axis system.                |
| Getran (&nx, &ny)               | returns the range of colour bars.                       |
| Getres (&nx, &ny)               | returns the size of points used in 3-D colour graphics. |
| Getrgb (&xr, &xg, &xb)          | returns the RGB coordinates of the current colour.      |
| Getscl (&nx, &ny, &nz)          | returns the current axis scaling.                       |
| Getscm (&nx, &ny, &nz)          | informs if automatic scaling is enabled.                |
| Getscr (&nw, &nh)               | returns the screen size in pixels.                      |
| c = Getshf (copt)               | returns shift characters for European characters.       |
| Getsp1 (&nx, &ny, &nz)          | returns the distance between axis ticks and labels.     |
| Getsp2 (&nx, &ny, &nz)          | returns the distance between axis labels and names.     |
| Getsym (&nsym, &nh)             | returns the current symbol number and height.           |
| Gettcl (&nmaj, &nmin)           | returns the current tick lengths.                       |
| Gettic (&nx, &ny, &nz)          | returns the number of ticks plotted between labels.     |
| n = Gettyp ()                   | returns the current line style.                         |
| n = Getuni ()                   | returns the current unit used for messages.             |
| x = Getver ()                   | returns the Dislin version number.                      |
| Getvk (&nyt, &nxbar, &nybar)    | returns the current lengths used for shifting.          |
| cvlt = Getvlt ()                | returns the current colour table.                       |
| n = Getwid ()                   | returns the width of colour bars.                       |
| Getwin (&nx, &ny, &nw, &nh)     | returns the position and size of the graphics window.   |
| id = Getxid ("WINDOW")          | returns the X window ID.                                |

Figure A.17: Parameter Requesting Routines

## A.18 Elementary Plot Routines

| Routine                                     | Meaning                                                                      |
|---------------------------------------------|------------------------------------------------------------------------------|
| Arcell (nx, ny, na, nb, alpha, beta, theta) | plots elliptical arcs.                                                       |
| Areaf (nxray, nyray, n)                     | plots polygons.                                                              |
| Circle (nx, ny, nr)                         | plots circles.                                                               |
| Connpt (x, y)                               | plots a line to a point.                                                     |
| Ellips (nx, ny, nr1, nr2)                   | plots ellipses.                                                              |
| Line (nx, ny, nu, nv)                       | plots lines.                                                                 |
| Noarln ()                                   | suppresses the outline of geometric figures.                                 |
| Pie (nx, ny, nr, a, b)                      | plots pie segments.                                                          |
| Point (nx, ny, nb, nh, nc)                  | plots coloured rectangles where the position is defined by the centre point. |
| Recfl (nx, ny, nw, nh, nc)                  | plots coloured rectangles.                                                   |
| Rectan (nx, ny, nw, nh)                     | plots rectangles.                                                            |
| Rndrec (nx, ny, nw, nh, iopt)               | plots a rectangle with rounded corners.                                      |
| Rlarc (x, y, r1, r2, a, b, t)               | plots elliptical arcs for user coordinates.                                  |
| Rlarea (xray, yray, n)                      | plots polygons for user coordinates.                                         |
| Rlcirc (x, y, r)                            | plots circles for user coordinates.                                          |
| Rlell (x, y, r1, r2)                        | plots ellipses for user coordinates.                                         |
| Rline (x, y, u, v)                          | plots lines for user coordinates.                                            |
| Rlpie (x, y, r, a, b)                       | plots pie segments for user coordinates.                                     |
| Rlpoin (x, y, nw, nh, nc)                   | plots coloured rectangles for user coordinates.                              |
| Rlrec (x, y, xw, xh)                        | plots rectangles for user coordinates.                                       |
| Rlrnd (x, y, xw, xh, iopt)                  | plots for user coordinates a rectangle with rounded corners.                 |
| Rlsec (x, y, r1, r2, a, b, ncol)            | plots coloured pie sectors for user coordinates.                             |
| Rlvec (x1, y1, x2, y2, ivec)                | plots vectors for user coordinates.                                          |
| Rlwind (x, xp, yp, nw, a)                   | plots wind speed symbols for user coordinates.                               |
| Sector (nx, ny, nr1, nr2, a, b, ncol)       | plots coloured pie sectors.                                                  |
| Strtpt (x, y)                               | moves the pen to a point.                                                    |
| Triflc (xray, yray, icray, n)               | plots solid filled rectangles.                                               |
| Trifll (xray, yray)                         | plots solid filled rectangles.                                               |
| Vecclr (nclr)                               | defines colour for arrow heads.                                              |
| Vecopt (xo, key)                            | defines vector options.                                                      |
| Vector (nx, ny, nu, nv, ivec)               | plots vectors.                                                               |
| Windbr (x, nx, ny, nw, a)                   | plots wind speed symbols.                                                    |
| Xmove (x, y)                                | moves the pen to a point.                                                    |
| Xdraw (x, y)                                | plots a line to a point.                                                     |

Figure A.18: Elementary Plot Routines

## A.19 Conversion of Coordinates

| Routine                                | Meaning                                               |
|----------------------------------------|-------------------------------------------------------|
| Colray (zray, icray, n)                | converts Z-coordinates to colour numbers.             |
| Getico (x, y, &xp, &yp)                | converts a complex reflection factor to an impedance. |
| Getrco (x, y, &xp, &yp)                | converts a complex impedance to a reflection factor.  |
| n = Nxpixl (ix, iy)                    | converts X plot coordinates to pixel.                 |
| n = Nxposn (x)                         | converts X-coordinates to plot coordinates.           |
| n = Nypixl (ix, iy)                    | converts Y plot coordinates to pixel.                 |
| n = Nyposn (y)                         | converts Y-coordinates to plot coordinates.           |
| n = Nzposn (z)                         | converts Z-coordinates to colour numbers.             |
| Trfco1 (xray, n, cfrom, cto)           | converts one-dimensional coordinates.                 |
| Trfco2 (xray, yray, n, cfr, cto)       | converts two-dimensional coordinates.                 |
| Trfco3 (xray, yray, zray, n, cfr, cto) | converts three-dimensional coordinates.               |
| Trfrel (xray, yray, n)                 | converts X- and Y-coordinates to plot coordinates.    |
| x = Xinvrs (nx)                        | converts X plot coordinates to user coordinates.      |
| x = Xposn (x)                          | converts X-coordinates to real plot coordinates.      |
| y = Yinvs (ny)                         | converts Y plot coordinates to user coordinates.      |
| y = Yposn (y)                          | converts Y-coordinates to real plot coordinates.      |

Figure A.19: Conversion of Coordinates

## A.20 Utility Routines

| Routine                                                      | Meaning                                                      |
|--------------------------------------------------------------|--------------------------------------------------------------|
| Bezier (xray, yray, n,<br>xpray, ypray, np)                  | calculates a Bezier interpolation.                           |
| n = Bitsi4 (nbits, ninp, iinp,<br>nout, iout)                | allows bit manipulation on 32 bit variables.                 |
| Circ3p (x1, y1, x2, y2, x3, y3,<br>&xm, &ym, &r)             | calculates a circle specified by 3 points.                   |
| cstr = Fcha (x, ndig)                                        | converts floating point numbers to character strings.        |
| n = Flen (x, ndig)                                           | calculates the number of digits for floating point numbers.  |
| nh = Histog (xray, n, xhray, yhray)                          | calculates a histogram.                                      |
| cstr = Intcha (nx)                                           | converts integers to character strings.                      |
| n = Intlen (nx)                                              | calculates the number of digits for integers.                |
| cstr = Intutf (iray, n)                                      | converts Unicode numbers to an UTF8 string.                  |
| n = Nlmess (cstr)                                            | returns the length of character strings in plot coordinates. |
| n = Nlnumb (x, ndig)                                         | returns the length of numbers in plot coordinates.           |
| n = Polc1p (xray, yray, n, x2ray,<br>y2ray, nmax, xv, cedge) | clips a polygon.                                             |
| Sortr1 (xray, n, copt)                                       | sorts floating point numbers.                                |
| Sortr2 (xray, yray, n, copt)                                 | sorts points in the X-direction.                             |

| <b>Routine</b>                                              | <b>Meaning</b>                                            |
|-------------------------------------------------------------|-----------------------------------------------------------|
| npt = Spline (xray, yray, n,<br>xrray, yrray)               | returns splined points as calculated in CURVE.            |
| Swapi2 (iray, n)                                            | swaps the bytes of 16 bit integer variables.              |
| Swapi4 (iray, n)                                            | swaps the bytes of 32 bit integer variables.              |
| Trfmat (zmat, nx, ny,<br>zmat2, nx2, ny2)                   | converts matrices.                                        |
| ntri = Triang (xray, yray, n,<br>i1ray, i2ray, i3ray, nmax) | calculates the Delaunay triangulation.                    |
| n = Trmlen (cstr)                                           | calculates the number of characters in character strings. |
| cstr = Upstr (cstr)                                         | converts a character string to uppercase letters.         |
| n = Utfint (cstr, iray, nray)                               | converts an UTF8 string to Unicode numbers.               |

Figure A.20: Utility Routines

## A.21 Binary File I/O

| <b>Routine</b>                 | <b>Meaning</b>                                     |
|--------------------------------|----------------------------------------------------|
| istat = Closfl (nu)            | closes a file.                                     |
| istat = Openfl (cfil, nu, irw) | opens a file for binary I/O.                       |
| istat = Posifl (nu, nbyte)     | skips to a certain position relative to the start. |
| n = Readfl (nu, cray, nbyte)   | reads a given number of bytes.                     |
| istat = Skipfl (n, nbyte)      | skips a number of bytes from the current position. |
| n = Tellfl (nu)                | returns the file position.                         |
| n = Writfl (nu, cray, nbyte)   | writes a given number of bytes.                    |

Figure A.21: Binary File I/O

## A.22 Date Routines

| <b>Routine</b>            | <b>Meaning</b>                       |
|---------------------------|--------------------------------------|
| Basdat (id, im, iy)       | defines the base date.               |
| n = Incdat (id, im, iy)   | returns incremented days.            |
| n = Nwkday (id, im, iy)   | returns the weekday of a date.       |
| Trfdat (n, &id, &im, &iy) | converts incremented days to a date. |

Figure A.22: Date Routines

## A.23 Cursor Routines

| Routine                         | Meaning                                         |
|---------------------------------|-------------------------------------------------|
| n = Csrkey ()                   | returns a character key.                        |
| Csrlin (&nx1, &ny1, &nx2, &ny2) | returns the end points of a line.               |
| n = Csrmov (nxray, nyray, nmax) | collects cursor movements.                      |
| n = Csrpol (nxray, nyray, nmax) | returns collected cursor positions.             |
| Csrpos (&nx, &ny, &nkey)        | sets and returns the cursor position.           |
| Csrpt1 (&nx, &ny)               | returns a pressed cursor position.              |
| n = Csrpts (nxray, nyray, nmax) | collects cursor positions.                      |
| Csrec (&nx1, &ny1, &nx2, &ny2)  | returns opposite corners of a rectangle.        |
| Csrtp (copt)                    | selects the cursor type.                        |
| Csrni (copt)                    | selects the unit of returned cursor positions.  |
| Setcsr (copt)                   | defines the cursor type of the graphics window. |

Figure A.23: Cursor Routines

## A.24 Transparency

| Routine              | Meaning                     |
|----------------------|-----------------------------|
| Tprfin ()            | terminates alpha blending.  |
| Tprini ()            | initializes alpha blending. |
| Tprmod (cmode, ckey) | modifies alpha blending.    |
| Tprval (x)           | sets the alpha value.       |

Figure A.24: Transparency

## A.25 Bar Graphs

| Routine                      | Meaning                                             |
|------------------------------|-----------------------------------------------------|
| Barbor (iclr)                | defines the colour of bar borders.                  |
| Barclr (ic1, ic2, ic3)       | defines bar colours.                                |
| Bargrp (ngrp, gap)           | affects clustered bars.                             |
| Barmod (copt, ckey)          | enables variable bars.                              |
| Baropt (xf, ang)             | modifies the appearance of 3-D bars.                |
| Barpos (copt)                | selects predefined positions for bars.              |
| Bars (xray, y1ray, y2ray, n) | plots bar graphs.                                   |
| Bartyp (copt)                | selects vertical or horizontal bars.                |
| Chnbar (copt)                | modifies the appearance of bars.                    |
| Labclr (nclr, "BARS")        | defines the colour of bar labels.                   |
| Labdig (ndig, "BARS")        | defines the number of decimal places in bar labels. |
| Labels (copt, "BARS")        | defines bar labels.                                 |
| Labpos (copt, "BARS")        | defines the position of bar labels.                 |

Figure A.25: Bar Graphs

## A.26 Pie Charts

| Routine                      | Meaning                                                      |
|------------------------------|--------------------------------------------------------------|
| Chnpie (copt)                | defines colour and pattern attributes for pie segments.      |
| Labelr (nclr, "PIE")         | defines the colour of segment labels.                        |
| Labdig (ndig, "PIE")         | defines the number of decimal places in segment labels.      |
| Labels (copt, "PIE")         | defines pie labels.                                          |
| Labpos (copt, "PIE")         | defines the position of segment labels.                      |
| Labtyp (copt, "PIE")         | modifies the appearance of segment labels.                   |
| Piebor (iclr)                | defines the colour of pie borders.                           |
| Piecbk (Routine)             | defines a callback routine for PIEGRF.                       |
| Pieclr (ic1ray, ic2ray, n)   | defines pie colours.                                         |
| Pieexp ()                    | defines exploded pie segments.                               |
| Piegrf (cbuf, nlin, xray, n) | plots pie charts.                                            |
| Pielab (clab, cpos)          | sets additional character strings plotted in segment labels. |
| Pieopt (xf, ang)             | modifies the appearance of 3-D pies.                         |
| Pierot (angle)               | sets a rotation angle for 2-D pie charts.                    |
| Pietyp (copt)                | selects 2-D or 3-D pie charts.                               |
| Pieval (x, ckey)             | modifies parameters for pie charts.                          |
| Pievec (ivec, copt)          | modifies the arrow plotted between labels and segments.      |

Figure A.26: Pie Charts

## A.27 Coloured 3-D Graphics

| Routine                                                               | Meaning                                                                         |
|-----------------------------------------------------------------------|---------------------------------------------------------------------------------|
| Ax3len (nx, ny, nz)                                                   | defines axis lengths.                                                           |
| Colran (nx, ny)                                                       | defines the range of colour bars.                                               |
| Crvmat (zmat, n, m, ixp, iyp)                                         | plots a coloured surface.                                                       |
| Crvqdr (xray, yray, zray, n)                                          | plots coloured quadrangles.                                                     |
| Crvtri (xray, yray, zray, n,<br>i1ray, i2ray, i3ray, ntri)            | plots the coloured surface of an Delaunay triangulation.                        |
| Curve3 (xray, yray, zray, n)                                          | plots coloured rectangles.                                                      |
| Curvx3 (xray, y, zray, n)                                             | plots rows of coloured rectangles.                                              |
| Curvy3 (x, yray, zray, n)                                             | plots columns of coloured rectangles.                                           |
| Erase ()                                                              | erases the screen.                                                              |
| Frmbar (nfrm)                                                         | defines the thickness of frames around colour bars.                             |
| Graf3 (xa, xe, xor, xstp,<br>ya, ye, yor, ystp,<br>za, ze, zor, zstp) | plots a coloured axis system.                                                   |
| Jusbar (copt)                                                         | defines the alignment of colour bars.                                           |
| Nobar ()                                                              | suppresses the plotting of colour bars.                                         |
| Nobgd ()                                                              | suppresses the plotting of points which have the same colour as the background. |
| n = Nzposn (z)                                                        | converts a Z-coordinate to a colour number.                                     |
| Point (nx, ny, nb, nh, nc)                                            | plots coloured rectangles.                                                      |
| Posbar (copt)                                                         | sets the position of colour bars.                                               |
| Recfl (nx, ny, nw, nh, nc)                                            | plots coloured rectangles.                                                      |
| Rlpoin (x, y, nw, nh, nc)                                             | plots coloured rectangles for user coordinates.                                 |
| Rlsec (x, y, r1, r2, a, b, ncol)                                      | plots coloured pie sectors for user coordinates.                                |
| Sector (nx, ny, nr1, nr2, a, b, ncol)                                 | plots coloured pie sectors.                                                     |
| Setres (nx, ny)                                                       | defines the size of coloured rectangles.                                        |
| Spcbar (nspc)                                                         | sets the space between colour bars and axis systems.                            |
| Vxkbar (nshift)                                                       | shifts colour bars in the X-direction.                                          |
| Vkybar (nshift)                                                       | shifts colour bars in the Y-direction.                                          |
| Widbar (nw)                                                           | defines the width of colour bars.                                               |
| Zaxis (za, ze, zor, zstp,<br>nl, cstr, it, id, nx, ny)                | plots a linearly scaled colour bar.                                             |
| Zaxlg (za, ze, zor, zstp,<br>nl, cstr, it, id, nx, ny)                | plots a logarithmically scaled colour bar.                                      |

Figure A.27: Coloured 3-D Graphics

## A.28 3-D Graphics

| Routine                                                                          | Meaning                                                             |
|----------------------------------------------------------------------------------|---------------------------------------------------------------------|
| Abs3pt (x, y, z, &xp, &yp)                                                       | converts absolute 3-D coordinates to plot coordinates.              |
| Axis3d (x, y, z)                                                                 | defines the lengths of the 3-D box.                                 |
| Bars3d (xray, yray, z1ray, z2ray,<br>xwray, ywray, icray, n)                     | plots 3-D bars.                                                     |
| Box3d ()                                                                         | plots a border around the 3-D box.                                  |
| Cone3d (xm, ym, zm, r, h1, h2, n, m)                                             | plots a cone.                                                       |
| Conn3d (x, y, z)                                                                 | plots a line to a point in 3-D space.                               |
| Conshd3d (xray, n, yray, m,<br>zmat, zlvray, nlev)                               | plots 3-D contours.                                                 |
| Curv3d (xray, yray, zray, n)                                                     | plots curves or symbols.                                            |
| Curv4d (xray, yray, zray, wray, n)                                               | plots coloured 3-d symbols.                                         |
| Cyli3d (xm, ym, zm, r, h, n, m)                                                  | plots a cylinder.                                                   |
| Dbffin ()                                                                        | terminates a depth sort.                                            |
| iret = Dbfini ()                                                                 | initializes a depth depth sort.                                     |
| Dbfmod (cmod)                                                                    | can disable the depth sort.                                         |
| Disk3d (xm, ym, zm, r1, r2, n, m)                                                | plots a disk.                                                       |
| Field3d (x1ray, y1ray, z1ray,<br>x2ray, y2ray, z2ray, n, ivec)                   | plots a vector field.                                               |
| Flab3d ()                                                                        | disables the suppression of axis labels.                            |
| nclr = Getlit (xp, yp, zp, xn, yn, zn)                                           | calculates colour values.                                           |
| Getmat (xray, yray, zray, n,<br>zmat, nx, ny, zv)                                | calculates a function matrix from randomly distributed data points. |
| Graf3d (xa, xe, xor, xstp, ya, ye,<br>yor, ystp, za, ze, zor, zstp)              | plots an axis system.                                               |
| Grffin ()                                                                        | terminates a projection into 3-D space.                             |
| Grfimg (cfil)                                                                    | includes an image into 3-D space.                                   |
| Grfini (x1, y1, z1, x2, y2, z2,<br>x3, y3, z3)                                   | initializes projections in 3-D space.                               |
| Grid3d (nx, ny, copt)                                                            | plots a grid.                                                       |
| Hsym3d (xh)                                                                      | sets the height of 3-D symbols.                                     |
| n = Isopts (xray, nx, yray, ny, zray, nz,<br>wmat, wlev, xtri, ytri, ztri, nmax) | calculates isosurfaces.                                             |
| Labl3d (copt)                                                                    | modifies the appearance of labels on the 3-D box.                   |
| Light (cmode)                                                                    | turns lighting on or off.                                           |
| Litmod (id, cmode)                                                               | turns single light sources on or off.                               |
| Litop3 (id, xr, xg, xb, ctype)                                                   | modifies light parameters.                                          |
| Litopt (id, xval, ctype)                                                         | modifies light parameters.                                          |
| Litpos (id, xp, yp, zp, copt)                                                    | sets the position of light sources.                                 |
| Matop3 (xr, xg, xb, ctype)                                                       | modifies material parameters.                                       |
| Matopt (xval, ctype)                                                             | modifies material parameters.                                       |
| Mdfmat (ix, iy, w)                                                               | modifies the algorithm used in GETMAT.                              |



| Routine                                                                                                                              | Meaning                                                |
|--------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------|
| Mshclr (iclr)                                                                                                                        | defines the colour of surface meshes.                  |
| Msherv (n)                                                                                                                           | sets the resolution of meshes for 3-D curves.          |
| Nohide ()                                                                                                                            | disables the hidden-line algorithm.                    |
| Pike3d (x1, y1, z1, x2, y2, z2,<br>r, nsk1, nsk2)                                                                                    | plots a cone.                                          |
| Plat3d (xm, ym, zm, xl, copt)                                                                                                        | plots a Platonic solid.                                |
| Plyfin (cfil, cobj)                                                                                                                  | terminates output of polygons to a PLY format.         |
| Plyini (copt)                                                                                                                        | initializes output of polygons to a PLY format.        |
| Pos3pt (x, y, z, &xp, &yp, &zp)                                                                                                      | converts user coordinates to absolute 3-D coordinates. |
| Pyra3d (xm, ym, zm, xl, h1, h2, n)                                                                                                   | plots a pyramid.                                       |
| Quad3d (xm, ym, zm, xl, yl, zl)                                                                                                      | plots a quad.                                          |
| Rel3pt (x, y, z, &xp, &yp)                                                                                                           | converts user coordinates to plot coordinates.         |
| Rot3d (a, b, c)                                                                                                                      | defines rotation angles for symbols and solids.        |
| Setfce (copt)                                                                                                                        | sets a face side for defining material parameters.     |
| Shlsur ()                                                                                                                            | protects surfaces from overwriting.                    |
| Sphe3d (xm, ym, zm, r, n, m)<br>n = Stmpts3d (xv, yv, zv, nx, ny, nz,<br>xpray, ypray, zpray, x0, y0, z0,<br>xray, yray, zray, nmax) | plots a sphere.<br>generates a streamline.             |
| Stream3d (xv, yv, zv, nx, ny, nz,<br>xpray, ypray, zpray, xsray, ysray, n)                                                           | plots streamlines.                                     |
| Strt3d (x, y, z)                                                                                                                     | moves the pen to a point.                              |
| Surclr (itop, ibot)                                                                                                                  | selects surface colours.                               |
| Surfce (xray, nx, yray, ny, zmat)                                                                                                    | plots the surface of a function matrix.                |
| Surfcp (cfunc, a1, a2, astp,<br>b1, b2, bstp)                                                                                        | plots the surface of a parametric function.            |
| Surfun (cfunc, ixp, xdel, iyp, ydel)                                                                                                 | plots the surface grid of a function.                  |
| Suriso (xray, nx, yray, ny,<br>zray, nz, wmat, wlev)                                                                                 | plots isosurfaces.                                     |
| Surmat (zmat, nx, ny, ixpts, iypts)                                                                                                  | plots the surface of a function matrix.                |
| Surmsh (copt)                                                                                                                        | enables grid lines for surfcp and surshd.              |
| Suropt (copt)                                                                                                                        | suppresses surface lines for surfce.                   |
| Surshc (xray, nx, yray, ny, zmat, wmat)                                                                                              | plots a coloured surface.                              |
| Surshd (xray, nx, yray, ny, zmat)                                                                                                    | plots a coloured surface.                              |
| Surtri (xray, yray, zray, n,<br>i1ray, i2ray, i3ray, ntri)                                                                           | plots the surface of an Delaunay triangulation.        |
| Survis (copt)                                                                                                                        | determines the visible part of surfaces.               |
| Symb3d (n, xm, ym, zm)                                                                                                               | plots a 3-D symbol.                                    |
| Thkc3d (nthk)                                                                                                                        | defines the thickness of 3-D curves.                   |
| Torus3d (xm, ym, zm, r1, r2, h,<br>a1, a2, n, m)                                                                                     | plots a torus.                                         |
| Tube3d (x1, y1, z1, x2, y2, z2, r, n, m)                                                                                             | plots a tube.                                          |

| <b>Routine</b>                                                  | <b>Meaning</b>                               |
|-----------------------------------------------------------------|----------------------------------------------|
| Vang3d (ang)                                                    | defines the field of view.                   |
| Vecf3d (xvray, yvray, zvray,<br>xpray, ypray, zpray, n, ivec)   | plots a vector field.                        |
| Vectr3 (x1, y1, z1, x2, y2, z2, ivec)                           | plots vectors in 3-D space.                  |
| Vecmat3d (xv, yv, zv, nx, ny, nz,<br>xpray, ypray, zpray, ivec) | plots a vector field on a regular grid.      |
| Vfoc3d (x, y, z, copt)                                          | defines the focus point.                     |
| View3d (x, y, z, copt)                                          | defines the viewpoint.                       |
| Vscl3d (xfac)                                                   | sets a scaling factor for orthographic view. |
| Vtx3d (xray, yray, zray, n, copt)                               | plots faces from vertices.                   |
| Vtxc3d (xray, yray, zray, icray, n, copt)                       | plots faces from vertices.                   |
| Vtxn3d (xray, yray, zray, xnray,<br>ynray, znray, n, copt)      | plots faces from vertices.                   |
| Vup3d (ang)                                                     | defines the camera orientation.              |
| Zbfers ()                                                       | erases the frame buffer of a Z-buffer.       |
| Zbffin ()                                                       | terminates the Z-buffer.                     |
| iret = Zbfini ()                                                | allocates space for a Z-buffer.              |
| Zbflin (x1, y1, z1, x2, y2, z2)                                 | plots lines.                                 |
| Zbfmod (cmod)                                                   | can disable the Z-buffer.                    |
| Zbfres ()                                                       | resets the Z-buffer.                         |
| Zbfsc1 (x)                                                      | scales the internal image for PDF output.    |
| Zbftri (xray, yray, zray, iray)                                 | plots triangles.                             |
| Zscale (zmin, zmax)                                             | defines a Z-scaling for coloured surfaces.   |

Figure A.28: 3-D Graphics

## A.29 Geographical Projections

| <b>Routine</b>                                   | <b>Meaning</b>                                       |
|--------------------------------------------------|------------------------------------------------------|
| Curvmp (xray, yray, n)                           | plots curves or symbols.                             |
| Grafmp (xa, xe, xor, xstp,<br>ya, ye, yor, ystp) | plots a geographical axis system.                    |
| Gridmp (nx, ny)                                  | plots a grid.                                        |
| Mapbas (copt)                                    | defines a base map.                                  |
| Mapfil (cfil, copt)                              | defines an external map file.                        |
| Mapimg (cfil, x1, x2, x3, x4, x5, x6)            | plots a BMP or GIF map image.                        |
| Maplab (copt, ckey)                              | defines label options.                               |
| Maplev (copt)                                    | specifies land or lake plotting.                     |
| Mapmod (copt)                                    | modifies the connection of points used in CURVMP.    |
| Mapopt (copt, ckey)                              | defines map options.                                 |
| Mappol (xpol, ypol)                              | defines the map pole used for azimuthal projections. |
| Mapref (ylw, yup)                                | defines two latitudes used for conical projections.  |
| Pos2pt (x, y, &xp, &yyp)                         | converts user coordinates to plot coordinates.       |
| Project (copt)                                   | selects a projection.                                |

| <b>Routine</b>                              | <b>Meaning</b>                                 |
|---------------------------------------------|------------------------------------------------|
| Pt2pos (x, y, &xp, &yp)                     | converts plot coordinates to user coordinates. |
| Shdafr (inray, ipray, icray, n)             | shades African countries.                      |
| Shdasi (inray, ipray, icray, n)             | shades Asiatic countries.                      |
| Shdaus (inray, ipray, icray, n)             | shades Oceanic countries.                      |
| Shdeur (inray, ipray, icray, n)             | shades European countries.                     |
| Shdmap (copt)                               | shades continents.                             |
| Shdnor (inray, ipray, icray, n)             | shades states of North and Central America.    |
| Shdsou (inray, ipray, icray, n)             | shades states of South America.                |
| Shdusa (inray, ipray, icray, n)             | shades USA states.                             |
| World ()                                    | plots coastlines and lakes.                    |
| Xaxmap (xa, xe, xor, xstp,<br>cstr, nt, ny) | plots a secondary X-axis.                      |
| Yaxmap (ya, ye, yor, ystp,<br>cstr, nt, nx) | plots a secondary Y-axis.                      |

Figure A.29: Geographical Projections

### A.30 Contouring

| <b>Routine</b>                                                                          | <b>Meaning</b>                                        |
|-----------------------------------------------------------------------------------------|-------------------------------------------------------|
| Conclr (ncray, n)                                                                       | defines colours for shaded contours.                  |
| Concerv (xray, yray, n, z)                                                              | plots generated contours.                             |
| Concfl (xray, yray, zray, n, i1ray,<br>i2ray, i3ray, ntri, zlvray, nlev)                | plots filled contours of an Delaunay triangulation.   |
| Congap (xfac)                                                                           | affects the spacing between contour lines and labels. |
| Conlab (copt)                                                                           | defines a character string used for contour labels.   |
| Conmat (zmat, nx, ny, z)                                                                | plots contours.                                       |
| Conmod (xfac, xquot)                                                                    | affects the position of contour labels.               |
| ncrv = Conpts (xray, n, yray, m,<br>zmat, zlev, xpray, ypray,<br>maxpts, iray, maxcerv) | generates contours.                                   |
| Conshd (xray, nx, yray,<br>ny, zmat, zlay, n)                                           | plots shaded contours.                                |
| Conshd2 (xmat, ymat, zmat,<br>nx, ny, zlay, n)                                          | plots shaded contours.                                |
| Contri (xray, yray, zray, n,<br>i1ray, i2ray, i3ray, ntri, zlev)                        | plots contours of an Delaunay triangulation.          |
| Contur (xray, nx, yray,<br>ny, zmat, zlev)                                              | plots contours.                                       |
| Contur2 (xmat, ymat, zmat,<br>nx, ny, zlev)                                             | plots contours.                                       |

| <b>Routine</b>                                                                                                                                                                                                     | <b>Meaning</b>                                                                                                                                                                                    |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Labclr (nclr, "CONT")<br>Labdis (ndis, "CONT")<br>Labels (copt, "CONT")<br>Shdmod (copt, "CONT")<br>ncrv = Tripts (xray, yray, zray, n,<br>i1ray, i2ray, i3ray, ntri, zlev,<br>xpray, ypray, maxpts, iray, maxcrv) | defines the colour of contour labels.<br>defines the distance between labels.<br>defines contour labels.<br>sets the algorithm for shaded contours.<br>generates contours from triangulated data. |

Figure A.30: Contouring

### A.31 Image Routines

| <b>Routine</b>                | <b>Meaning</b>                                        |
|-------------------------------|-------------------------------------------------------|
| Expimg (cfil, copt)           | copies an image from memory to a file.                |
| Imgbox (nx, ny, nw, nh)       | defines a rectangle for PostScript/PDF output.        |
| Imgclp (nx, ny, nw, nh)       | defines a clipping rectangle.                         |
| Imgfin ()                     | terminates transferring of image data.                |
| Imgini ()                     | initializes transferring of image data.               |
| Imgmod (cmod)                 | selects index or RGB mode.                            |
| Imgsiz (nw, nh)               | defines an image size for PostScript/PDF output.      |
| Imgtptr (nclr)                | defines a transparency colour for images.             |
| n = Rbfpng (cbuf, nmax)       | stores an image as PNG file in a buffer.              |
| Rbmp (cfil)                   | stores an image as a BMP file.                        |
| Rgif (cfil)                   | stores an image as a GIF file.                        |
| Rimage (cfil)                 | copies an image from memory to a file.                |
| iclr = Rpixel (ix, iy)        | reads a pixel from memory.                            |
| Rpixls (cray, ix, iy, nw, nh) | reads image data from memory.                         |
| Rpng (cfil)                   | stores an image as a PNG file.                        |
| Rppm (cfil)                   | stores an image as a PPM file.                        |
| Rpxrow (cray, nx, ny, n)      | reads a row of image data from memory.                |
| Rtiff (cfil)                  | stores an image as a TIFF file.                       |
| Tiforg (nx, ny)               | defines the position of TIFF files copied with WTIFF. |
| Tifwin (nx, ny, nw, nh)       | defines a clipping window for TIFF files.             |
| Wimage (cfil)                 | copies an image from file to memory.                  |
| Wpixel (ix, iy, iclr)         | writes a pixel to memory.                             |
| Wpixls (cray, ix, iy, nw, nh) | writes image data to memory.                          |
| Wpxrow (cray, nx, ny, n)      | write a row of image data to memory.                  |
| Wtiff (cfil)                  | copies a TIFF file created by Dislin to memory.       |

Figure A.31: Image Routines

## A.32 Window Routines

| Routine                 | Meaning                                                            |
|-------------------------|--------------------------------------------------------------------|
| Clswin (id)             | closes a window.                                                   |
| Hidwin (id, copt)       | defines whether a window is visible or not.                        |
| Opnwin (id)             | opens a window for graphics output.                                |
| Pagwin (nxp, nyp)       | defines page formats for windows.                                  |
| Selwin (id)             | selects a window for graphics output.                              |
| Winapp (capp)           | defines a window or console application.                           |
| Wincbk (crout, copt)    | defines a callback routine for the windows size.                   |
| Window (nx, ny, nw, nh) | defines the position and size of windows.                          |
| Winico (cstr)           | loads an icon for the windows title bar.                           |
| id = Winid ()           | returns the ID of the currently selected window.                   |
| Winjus (copt)           | defines the position of the graphics window.                       |
| Winkey (ckey)           | defines a key that can be used for program continuation in Disfin. |
| Winmod (copt)           | affects the handling of windows in Disfin.                         |
| Winsiz (nw, nh)         | defines the size of windows.                                       |
| Wintit (cstr)           | sets the title of the currently selected window.                   |
| Wintyp (copt)           | sets the type of the graphics window.                              |
| X11mod (copt)           | enables backing store.                                             |

Figure A.32: Window Routines

## A.33 Widget Routines

| Routine                           | Meaning                                                     |
|-----------------------------------|-------------------------------------------------------------|
| Doevnt ()                         | processes pending events.                                   |
| ival = Dwgbut (cstr, ival)        | displays a message that can be answered with 'Yes' or 'No'. |
| iret = Dwgerr ()                  | returns a status for dialog widget routines.                |
| cfil = Dwgfil (clab, cfil, cmask) | creates a file selection box.                               |
| isel = Dwglis (clab, clis, isel)  | gets a selection from a list of items.                      |
| Dwgmsg (cstr)                     | displays a message.                                         |
| cstr = Dwgtxt (clab, cstr)        | prompts an user for input.                                  |
| n = Gwgatt (id, copt)             | requests widget attributes.                                 |
| n = Gwgbox (id)                   | requests the value of a box widget.                         |
| n = Gwgbut (id)                   | requests the status of a button widget.                     |
| cfil = Gwgfil (id)                | requests the value of a file widget.                        |
| x = Gwgflt (id)                   | requests the value of a text widget as real number.         |
| n = Gwggui ()                     | returns the used GUI.                                       |
| n = Gwgint (id)                   | requests the value of a text widget as integer.             |
| n = Dwglis (id)                   | requests the value of a list widget.                        |
| x = Gwgscl (id)                   | requests the value of a scale widget.                       |

| Routine                               | Meaning                                             |
|---------------------------------------|-----------------------------------------------------|
| Gwgsiz (id, &nw, &nh)                 | returns the size of widgets.                        |
| x = Gwgtbf (id, irow, icol)           | requests the value of a table cell as real number.  |
| ix = Gwgtbi (id, irow, icol)          | requests the value of a table cell as integer.      |
| xray = Gwgtbl (id, n, idx, copt)      | requests the values of table cells.                 |
| cstr = Gwgtbs (id, irow, icol)        | requests the value of a table cell as a string.     |
| cstr = Gwgtxt (id)                    | requests the value of a text widget.                |
| n = Gwgxid (id)                       | requests the windows ID of a widget.                |
| clis = Itmcat (clis, n, citem)        | concatenates an element to a list string.           |
| n = Itmctn (clis)                     | calculates the number of elements in a list string. |
| citem = Itmstr (clis, n)              | extracts an element from a list string.             |
| Msgbox (cstr)                         | displays a message.                                 |
| Swgatt (id, catt, copt)               | sets widget attributes.                             |
| Swgbgd (id, xr, xg, xb)               | changes the background colour of widgets.           |
| Swgbox (id, isel)                     | changes the selection of a box widget.              |
| Swgbut (id, ival)                     | changes the status of a button widget.              |
| Swgcb2 (id, routine)                  | connects a callback routine with a table widget.    |
| Swgcbk (id, routine)                  | connects a callback routine with a widget.          |
| Swgclr (xr, xg, xb, copt)             | sets widget colours.                                |
| Swgdrw (xf)                           | defines the height of draw widgets.                 |
| Swgfgd (id, xr, xg, xb)               | changes the foreground colour of widgets.           |
| Swgfil (id, cfil)                     | changes the value of a file widget.                 |
| Swgflt (id, x, ndig)                  | changes the value of text widgets.                  |
| Swgfnt (cfnt, npts)                   | defines widget fonts.                               |
| Swgfoc (id)                           | sets the keyboard focus.                            |
| Swghlp (cstr)                         | sets a character string for the Help menu.          |
| Swgint (id, n)                        | changes the value of text widgets.                  |
| Swgiop (i, copt)                      | sets integer options for widgets.                   |
| Swgjus (cjus, class)                  | defines the alignment of label widgets.             |
| Swglis (id, isel)                     | changes the selection of a list widget.             |
| Swgmix (char, cmix)                   | defines control characters.                         |
| Swgmrg (ival, cmrg)                   | defines widget margins.                             |
| Swgopt (copt, ckey)                   | sets a center option for parent widgets.            |
| Swgpop (copt)                         | modifies the appearance of the popup menubar.       |
| Swgpos (nx, ny)                       | defines the position of widgets.                    |
| Swgray (xray, n, copt)                | defines the width of columns in table widgets.      |
| Swgscl (id, xval)                     | changes the value of a scale widget.                |
| Swgsiz (nw, nh)                       | defines the size of widgets.                        |
| Swgspc (xspc, yspc)                   | modifies the spaces between widgets.                |
| Swgstp (xstp)                         | defines a step value for scale widgets.             |
| Swgtbf (id, x, ndig, i, j, copt)      | changes the values of table cells.                  |
| Swgtbi (id, ix, i, j, copt)           | changes the values of table cells.                  |
| Swgtbl (id, xray, n, ndig, idx, copt) | changes the values of table cells.                  |

| Routine                                       | Meaning                                                                        |
|-----------------------------------------------|--------------------------------------------------------------------------------|
| Swgtbs (id, cstr, i, j, copt)                 | changes the values of table cells.                                             |
| Swgtit (cstr)                                 | sets a title for the main widget.                                              |
| Swgtxt (id, cval)                             | changes the value of a text widget.                                            |
| Swgtyp (ctype, class)                         | modifies the appearance of widgets.                                            |
| Swgval (id, xval)                             | changes the value of progress bars.                                            |
| Swgwin (nx, ny, nw, nh)                       | defines the position and size of widgets.                                      |
| Swgwth (nwth)                                 | sets the default width of widgets.                                             |
| id = Wgapp (ip, clab)                         | creates an entry in a popup menu.                                              |
| id = Wgappb (ip, cray, nw, nh)                | uses an image as entry in a popup menu.                                        |
| id = Wgbas (ip, copt)                         | creates a container widget.                                                    |
| id = Wgbox (ip, clis, isel)                   | creates a list widget where the list elements are displayed as toggle buttons. |
| id = Wgbut (ip, cval, ival)                   | creates a button widget.                                                       |
| id = Wgcmd (ip, clab, cmd)                    | creates a push button widget for a system command.                             |
| id = Wgdlls (ip, clis, isel)                  | creates a dropping list widget.                                                |
| id = Wgdraw (ip)                              | creates a draw widget.                                                         |
| id = Wgfil (ip, clab, cfil, cmask)            | creates a file widget.                                                         |
| Wgfin ()                                      | terminates widget routines.                                                    |
| id = Wgicon (ip, clab, nw, nh, cfl)           | creates a label widget with an icon as label.                                  |
| id = Wgimg (ip, clab, cray, nw, nh)           | creates a label widget with an image as label.                                 |
| id = Wgini (copt)                             | creates a main widget and initializes widget routines.                         |
| id = Wglab (ip, cstr)                         | creates a label widget.                                                        |
| id = Wglis (ip, clis, isel)                   | creates a list widget.                                                         |
| id = Wgltxt (ip, clab, cstr, nwth)            | creates a labeled text widget.                                                 |
| id = Wgok (ip)                                | creates an OK push button widget.                                              |
| id = Wgpbar (ip, x1, x2, xstp)                | creates a progress bar.                                                        |
| id = Wgpbut (ip, clab)                        | creates a push button widget.                                                  |
| id = Wgpicon (ip, clab, nw, nh, cfl)          | creates a push button with an icon as label.                                   |
| id = Wgpimg (ip, clab, cray, nw, nh)          | creates a push button with an image as label.                                  |
| id = Wgpop (ip, cstr)                         | creates a popup menu.                                                          |
| id = Wgpopb (ip, cray, nw, nh)                | uses an image as a popup menu.                                                 |
| id = Wgquit (ip)                              | creates a Quit push button widget.                                             |
| id = Wgscl (ip, clab, xmin, xmax, xval, ndez) | creates a scale widget.                                                        |
| id = Wgsep (ip)                               | creates a separator widget.                                                    |
| id = Wgstxt (ip, nsize, nmax)                 | creates a scrolled text widget.                                                |
| id = Wgtbl (ip, nrows, ncols)                 | creates a table widget.                                                        |
| id = Wgtxt (ip, cstr)                         | creates a text widget.                                                         |

Figure A.33: Widget Routines

### A.34 Dislin Quickplots

| Routine                       | Meaning                      |
|-------------------------------|------------------------------|
| Qplbar (xray, n)              | plots a bar graph.           |
| Qplclr (zmat, n, m)           | plots a coloured surface.    |
| Qplcon (zmat, n, m, nlv)      | makes a contour plot.        |
| Qplcrv (xray, yray, n, copt)  | plots multiple curves.       |
| Qplot (xray, yray, n)         | plots a curve.               |
| Qplpie (xray, yray, n)        | plots a pie chart.           |
| Qplsca (xray, yray, n)        | makes a scatter plot.        |
| Qplscl (a, e, or, step, copt) | sets a user-defined scaling. |
| Qplsur (zmat, n, m)           | plots a surface.             |

Figure A.34: Dislin Quickplots

### A.35 Using Threads

| Routine    | Meaning             |
|------------|---------------------|
| Thrfin ()  | terminates threads. |
| Thrini (n) | enables threads.    |

Figure A.35: Using Threads

### A.36 Reading FITS Files

| Routine                | Meaning                                   |
|------------------------|-------------------------------------------|
| Fitscls ()             | closes a FITS file.                       |
| x = Filtsflt (ckey)    | returns the floatingpoint value of a key. |
| n = Fitsimg (iray)     | copies a FITS image to an array.          |
| istat = Fitsopn (cfil) | opens a FITS file for reading.            |
| cval = Fitsstr (ckey)  | returns the string value of a key.        |
| n = Fitstyp (ckey)     | returns the type of a key.                |
| n = Fitsval (ckey)     | returns the integer value of a key.       |

Figure A.36: Reading FITS Files

### A.37 MPS Logo

| Routine                       | Meaning             |
|-------------------------------|---------------------|
| Mpslogo (nx, ny, nsize, copt) | plots the MPS logo. |

Figure A.37: MPS Logo



## **Appendix B**

# **Examples**

This appendix presents some examples of the Dislin manual in Go coding. They can be found in the subdirectory `src` of the Dislin go directory.

## B.1 Demonstration of CURVE

```
package main
import "dislin"
import "math"

func main () {
    const n int = 100
    var step, fpi, x, xx, yy float64
    var xray, y1ray, y2ray [n]float64
    var ic int

    step = 360.0 / float64 (n - 1)
    fpi = 3.1415926 / 180.0
    for i:= 0; i < n; i++ {
        xray[i] = float64 (i) * step
        x = xray[i] * fpi
        y1ray[i] = math.Sin (x)
        y2ray[i] = math.Cos (x)
    }

    dislin.Metafl ("xwin")
    dislin.Disini ()
    dislin.Pagera ()
    dislin.Complx ()

    dislin.Name ("X-axis", "X")
    dislin.Name ("Y-axis", "Y")
    dislin.Labdig (-1, "X")
    dislin.Ticks (9, "X")
    dislin.Ticks (10, "Y")

    dislin.Titlin ("Demonstration of CURVE", 1)
    dislin.Titlin ("SIN (X), COS (X)", 3)
    dislin.Axspos (450, 1800)
    dislin.Axslen (2200, 1200)
    ic := dislin.Intrgb (0.95, 0.95, 0.95)
    dislin.Axsbgd (ic)

    dislin.Graf (0.0, 360.0, 0.0, 90.0, -1.0, 1.0, -1.0, 0.5)
    dislin.Setrgb (0.7, 0.7, 0.7)
    dislin.Grid (1, 1)
    dislin.Color ("fore")
    dislin.Height (50)
    dislin.Title ()

    dislin.Color ("red")
    dislin.Curve (&xray[0], &y1ray[0], n)
    dislin.Color ("green")
    dislin.Curve (&xray[0], &y2ray[0], n)
    dislin.Disfin ()
}
```

# Demonstration of CURVE

$\text{SIN}(X)$ ,  $\text{COS}(X)$

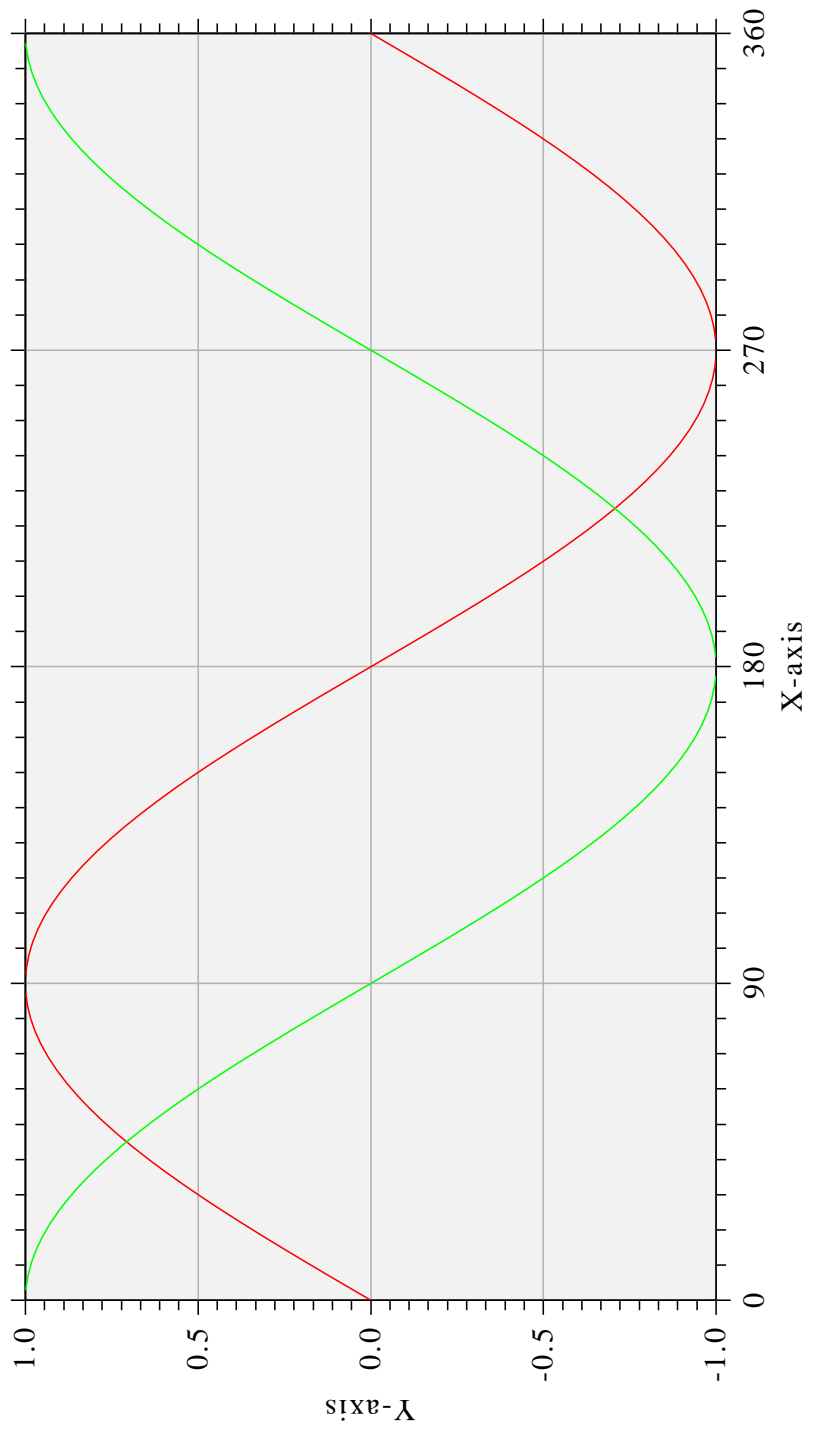


Figure B.1: Demonstration of CURVE

## B.2 Symbols

```
package main

import "dislin"

func main () {
    var ctit = "Symbols"
    var nl, ny, nxp int

    dislin.Setpag ("da4p")
    dislin.Metafl ("cons")
    dislin.Disini ()
    dislin.Complx ()
    dislin.Pagera ()

    dislin.Paghdr ("H. Michels  (", ")", 2, 0)

    dislin.Height (60)
    nl = dislin.Nlmess (ctit)
    dislin.Messag (ctit, (2100 - nl) / 2, 200)

    dislin.Height (50)
    dislin.Hsymb1 (120)

    ny = 150
    for i:=0; i < 22; i++ {
        if (i % 4) == 0 {
            ny = ny + 400
            nxp = 550
        } else {
            nxp = nxp + 350
        }

        x := float64 (i)
        nl = dislin.Nlnumb (x, -1)
        dislin.Number (x, -1, nxp - nl/2, ny + 150)
        dislin.Symbol (i, nxp, ny)
    }
    dislin.Disfin ()
}
```

# Symbols



0



1



2



3



4



5



6



7



8



9



10



11



12



13



14



15



16



17



18



19



20



21



22



23

H. Michels ( 12.01.2012, 15:51:05, DISLIN 10.2 )

Figure B.2: Symbols

## B.3 Logarithmic Scaling

```
package main

import "dislin"

func main () {
    ctit := "Logarithmic Scaling"
    clab :=[3]string{"LOG", "FLOAT", "ELOG"}

    dislin.Setpag ("da4p")
    dislin.Metafl ("cons")

    dislin.Disini ()
    dislin.Pagera ()
    dislin.Complx ()
    dislin.Axslen (1400, 500)

    dislin.Name ("X-axis", "X")
    dislin.Name ("Y-axis", "Y")
    dislin.Axsscl ("LOG", "XY")

    dislin.Titlin (ctit, 2)

    for i:= 0; i < 3; i++ {
        nya := 2650 - i * 800
        dislin.Labdig (-1, "XY")

        if i == 1 {
            dislin.Labdig (1, "Y")
            dislin.Name (" ", "X")
        }

        dislin.Axspos (500, nya)
        dislin.Messag ("Labels: " + clab[i], 600, nya - 400)
        dislin.Labels (clab[i], "XY")
        dislin.Graf (0.0, 3.0, 0.0, 1.0, -1.0, 2.0, -1.0, 1.0)

        if i == 2 {
            dislin.Height (50)
            dislin.Title ()
        }

        dislin.Endgrf ()
    }

    dislin.Disfin ()
}
```

## Logarithmic scaling

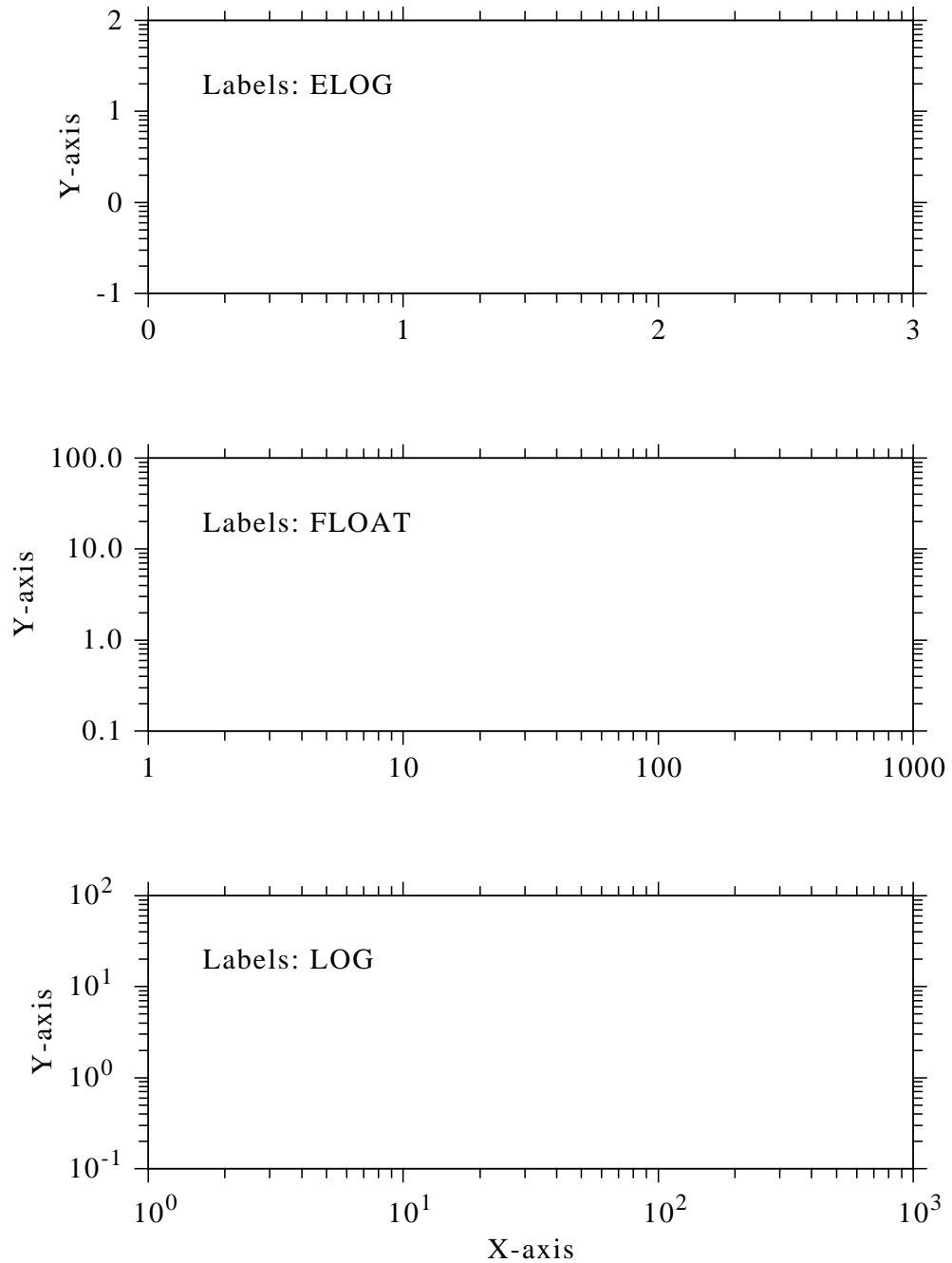


Figure B.3: Logarithmic Scaling

## B.4 Interpolation Methods

```
package main

import "dislin"

func main () {
    ctit := "Interpolation Methods"
    cpol := [6]string{"SPLINE", "STEM", "BARS", "STAIRS", "STEP", "LINEAR"}
    xray := [16]float64{0.0, 1.0, 3.0, 4.5, 6.0, 8.0, 9.0, 11.0, 12.0, 12.5,
        13.0, 15.0, 16.0, 17.0, 19.0, 20.0}
    yray := [16]float64{2.0, 4.0, 4.5, 3.0, 1.0, 7.0, 2.0, 3.0, 5.0, 2.0, 2.5,
        2.0, 4.0, 6.0, 5.5, 4.0}

    dislin.Setpag ("da4p")
    dislin.Metafl ("cons")

    dislin.Disini ()
    dislin.Pagera ()
    dislin.Complx ()

    dislin.Incmrk (1)
    dislin.Hsymb1 (25)
    dislin.Titlin (ctit, 1)
    dislin.Axslen (1500, 350)
    dislin.Setgrf ("LINE", "LINE", "LINE", "LINE")

    nya := 2700
    for i:=0; i < 6; i++ {
        dislin.Axspos (350, nya - i * 350)
        dislin.Polcrv (cpol[i])
        dislin.Marker (0)
        dislin.Graf (0.0, 20.0, 0.0, 5.0, 0.0, 10.0, 0.0, 5.0)
        nx := dislin.Nxposn (1.0)
        ny := dislin.Nyposn (8.0)
        dislin.Messag (cpol[i], nx, ny)
        dislin.Curve (&xray[0], &yray[0], 16)

        if i == 5 {
            dislin.Height (50)
            dislin.Title ()
        }

        dislin.Endgrf ()
    }

    dislin.Disfin ()
}
```



## Interpolation Methods

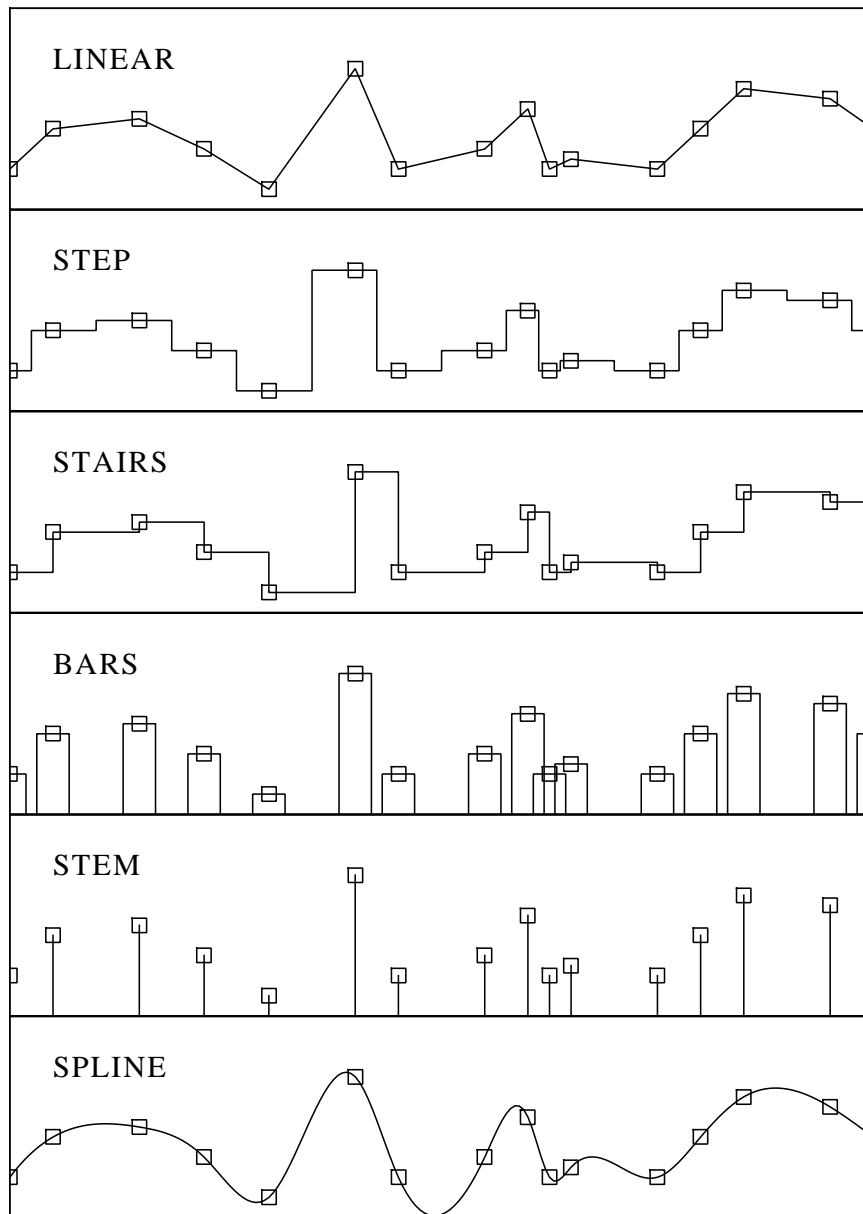


Figure B.4: Interpolation Methods

## B.5 Line Styles

```
package main

import "dislin"

func main () {
    ctit1 := "Demonstration of CURVE"
    ctit2 := "Line Styles"

    ctyp := [8]string{"SOLID", "DOT", "DASH", "CHNDSH",
                    "CHNDOT", "DASHM", "DOTL", "DASHL"}
    x := [2]float64{3.0, 9.0}
    y := [2]float64{0.0, 0.0}

    dislin.Metafl ("cons")
    dislin.Setpag ("da4p")

    dislin.Disini ()
    dislin.Pagera ()
    dislin.Complx ()
    dislin.Center ()

    dislin.Chncrv ("BOTH")
    dislin.Name ("X-axis", "X")
    dislin.Name ("Y-axis", "Y")

    dislin.Titlin (ctit1, 1)
    dislin.Titlin (ctit2, 3)

    dislin.Graf (0.0, 10.0, 0.0, 2.0, 0.0, 10.0, 0.0, 2.0)
    dislin.Title ()

    for i := 0; i < 8; i++ {
        y[0] = 8.5 - float64 (i)
        y[1] = 8.5 - float64 (i)
        nx := dislin.Nxposn (1.0)
        ny := dislin.Nyposn (y[0])
        dislin.Messag (ctyp[i], nx, ny - 20)
        dislin.Curve (&x[0], &y[0], 2)
    }

    dislin.Disfin ()
}
```

## Demonstration of CURVE

### Line styles

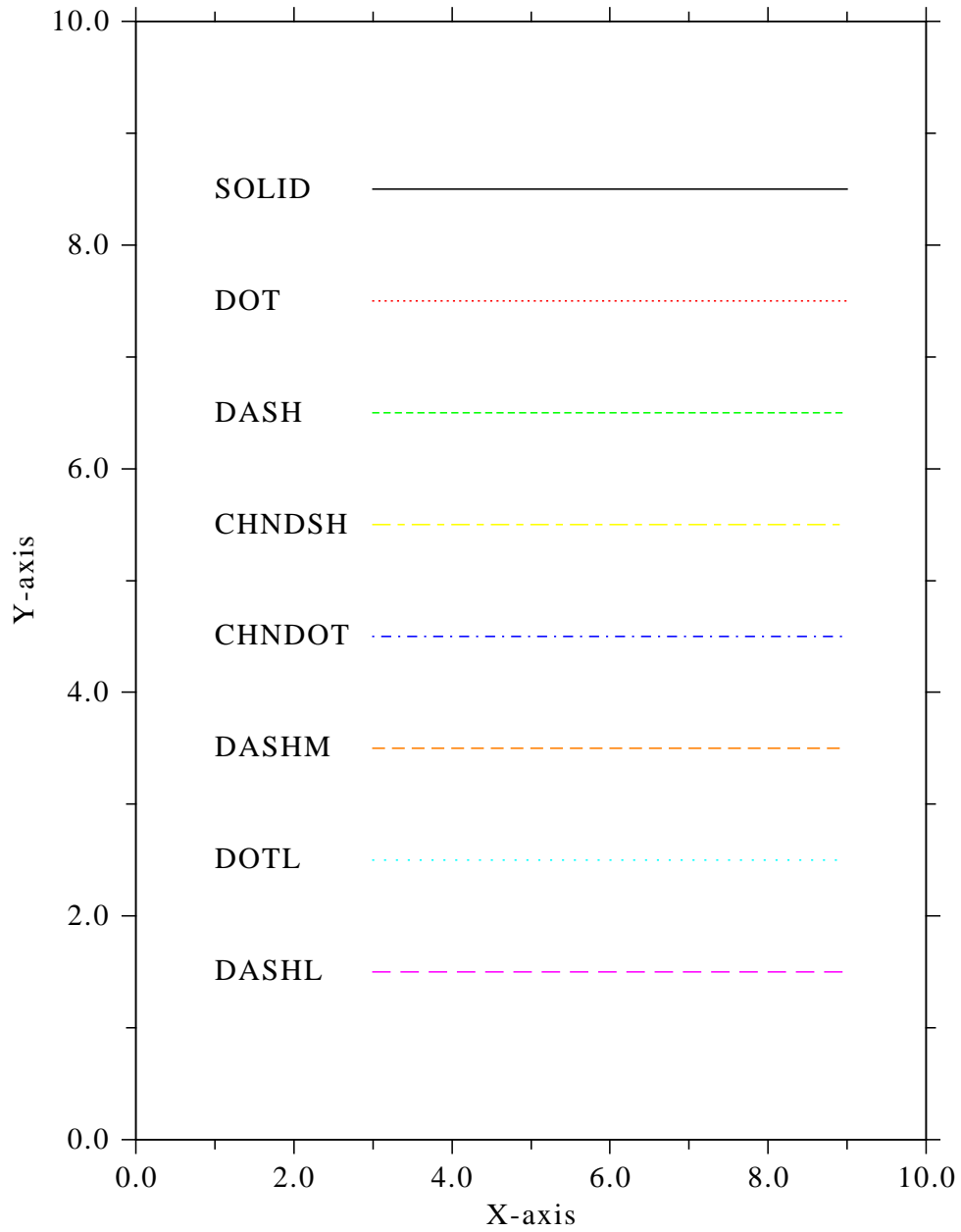


Figure B.5: Line Styles

## B.6 Legends

```
package main
import "dislin"
import "math"

func main () {
    const n int = 101
    var xray, y1ray, y2ray [n]float64
    var cbuf [14]int8

    f := 3.1415926 / 180.0
    step := 360.0 / float64 (n - 1)
    for i:= 0; i < n; i++ {
        xray[i] = float64 (i) * step
        x := xray[i] * f
        y1ray[i] = math.Sin (x)
        y2ray[i] = math.Cos (x)
    }

    dislin.Metafl ("xwin")
    dislin.Disini ()
    dislin.Complx ()

    dislin.Axspos (450, 1800)
    dislin.Axslen (2200, 1200)

    dislin.Name ("X-axis", "X")
    dislin.Name ("Y-axis", "Y")
    dislin.Labdig (-1, "X")
    dislin.Ticks (10, "XY")

    dislin.Titlin ("Demonstration of CURVE", 1)
    dislin.Titlin ("Legend", 3)

    dislin.Graf (0.0, 360.0, 0.0, 90.0, -1.0, 1.0, -1.0, 0.5)
    dislin.Title ()
    dislin.Xaxgit ()
    dislin.Chncrv ("BOTH")
    dislin.Curve (&xray[0], &y1ray[0], n)
    dislin.Curve (&xray[0], &y2ray[0], n)

    dislin.Legini (&cbuf[0], 2, 7)
    nx := dislin.Nxposn (190.0)
    ny := dislin.Nyposn (0.75)
    dislin.Leglin (&cbuf[0], "sin (x)", 1)
    dislin.Leglin (&cbuf[0], "cos (x)", 2)
    dislin.Legpos (nx, ny)
    dislin.Legtit ("Legend")
    dislin.Legend (&cbuf[0], 3)
    dislin.Disfin ()
}
```

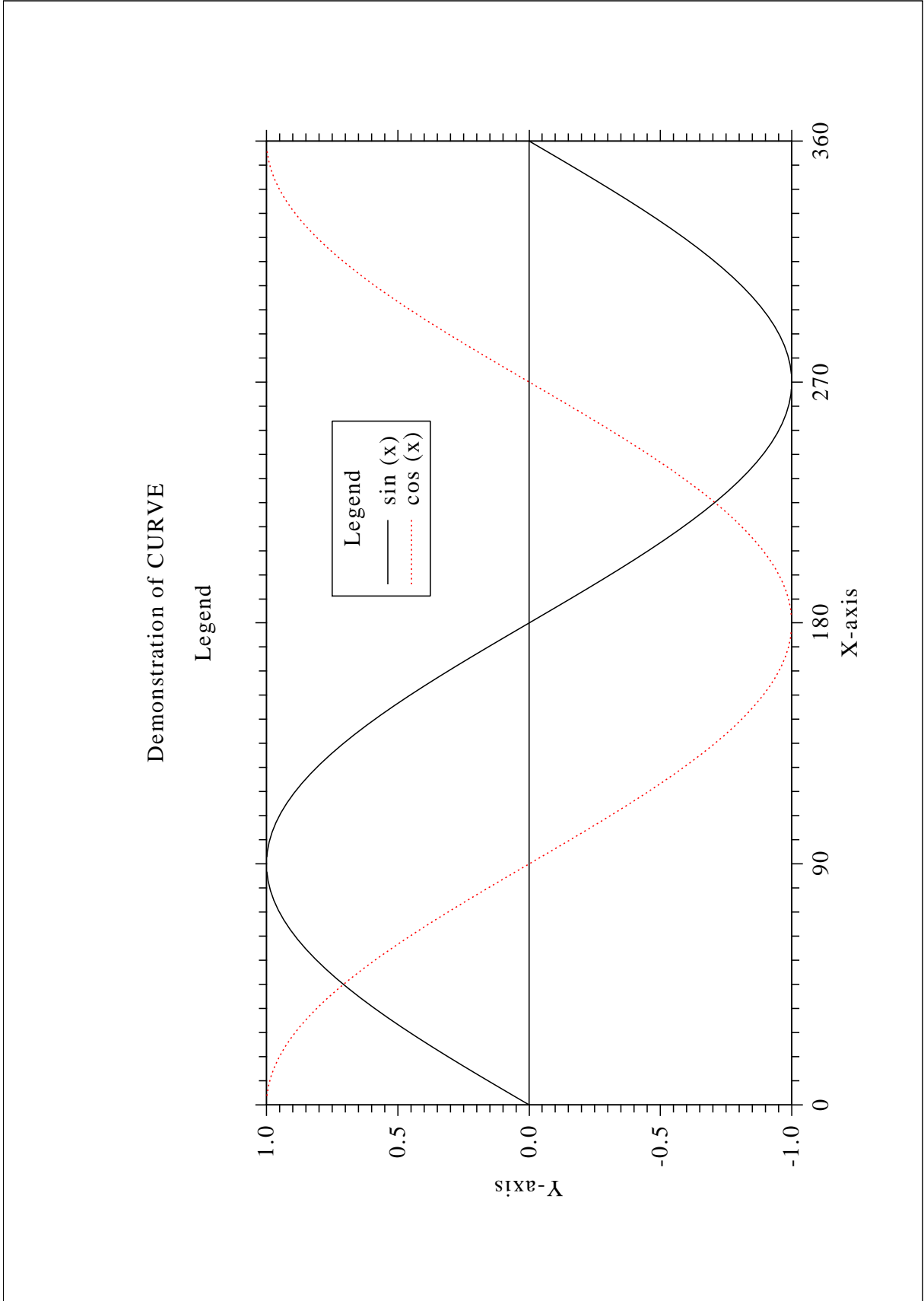


Figure B.6: Legends

## B.7 Shading Patterns (AREAF)

```
package main
import "dislin"

func main () {
    ix := [4]int32{0, 300, 300, 0}
    iy := [4]int32{0, 0, 400, 400}
    ixp := [4]int32{0, 0, 0, 0}
    iyp := [4]int32{0, 0, 0, 0}

    dislin.Metafl ("cons")
    dislin.Disini ()
    dislin.Setvlt ("small")
    dislin.Pagera ()
    dislin.Complx ()

    dislin.Height (50)
    ctit := "Shading patterns (AREAF)"
    nl := dislin.Nlmess (ctit)
    dislin.Messag (ctit, (2970 - nl)/2, 200)

    nx0 := 335
    ny0 := 350
    iclr := 0
    for i := 0; i < 3; i++ {
        ny := ny0 + i * 600

        for j := 0; j < 6; j++ {
            nx := nx0 + j * 400
            ii := i * 6 + j
            dislin.Shdpat (ii)
            iclr = iclr + 1
            iclr = iclr % 8
            if iclr == 0 {
                iclr = 8
            }
            dislin.Setclr (iclr)
            for k := 0; k < 4; k++ {
                ixp[k] = ix[k] + int32 (nx)
                iyp[k] = iy[k] + int32 (ny)
            }

            dislin.Areaf (&ixp[0], &iyp[0], 4)
            nl = dislin.Nlnumb (float64 (ii), -1)
            nx = nx + (300 - nl) / 2
            dislin.Color ("foreground")
            dislin.Number (float64 (ii), -1, nx, ny + 460)
        }
    }
    dislin.Disfin ()
}
```

Shading patterns (AREAF)

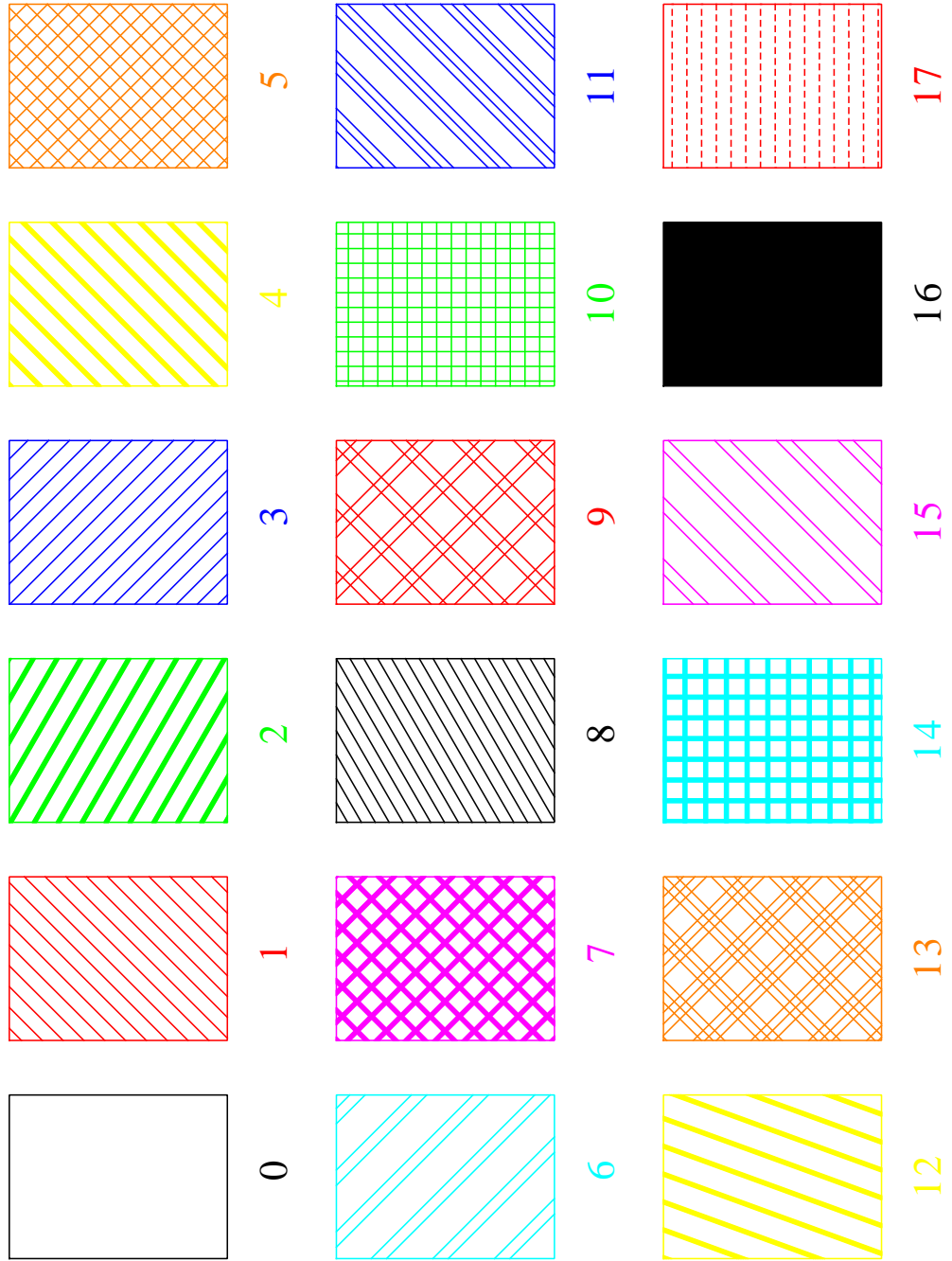


Figure B.7: Shading Patterns

## B.8 Vectors

```
package main

import "dislin"

func main () {
    ivec := [20]int{0, 1111, 1311, 1421, 1531, 1701, 1911,
        3111, 3311, 3421, 3531, 3703, 4221, 4302,
        4413, 4522, 4701, 5312, 5502, 5703}
    ctit := "Vectors"

    dislin.Metafl ("cons")
    dislin.Disini ()
    dislin.Pagera ()
    dislin.Complx ()

    dislin.Height (60)
    nl := dislin.Nlmess (ctit)
    dislin.Messag (ctit, (2970 - nl)/2, 200)

    dislin.Height (50)
    nx := 300
    ny := 400

    for i := 0; i < 20; i++ {
        if i == 10 {
            nx = nx + 2970 / 2
            ny = 400
        }

        nl = dislin.Nlnumb (float64 (ivec[i]), -1)
        dislin.Number (float64 (ivec[i]), -1, nx - nl, ny - 25)

        dislin.Vector (nx + 100, ny, nx + 1000, ny, ivec[i])
        ny = ny + 160
    }

    dislin.Disfin ()
}
```



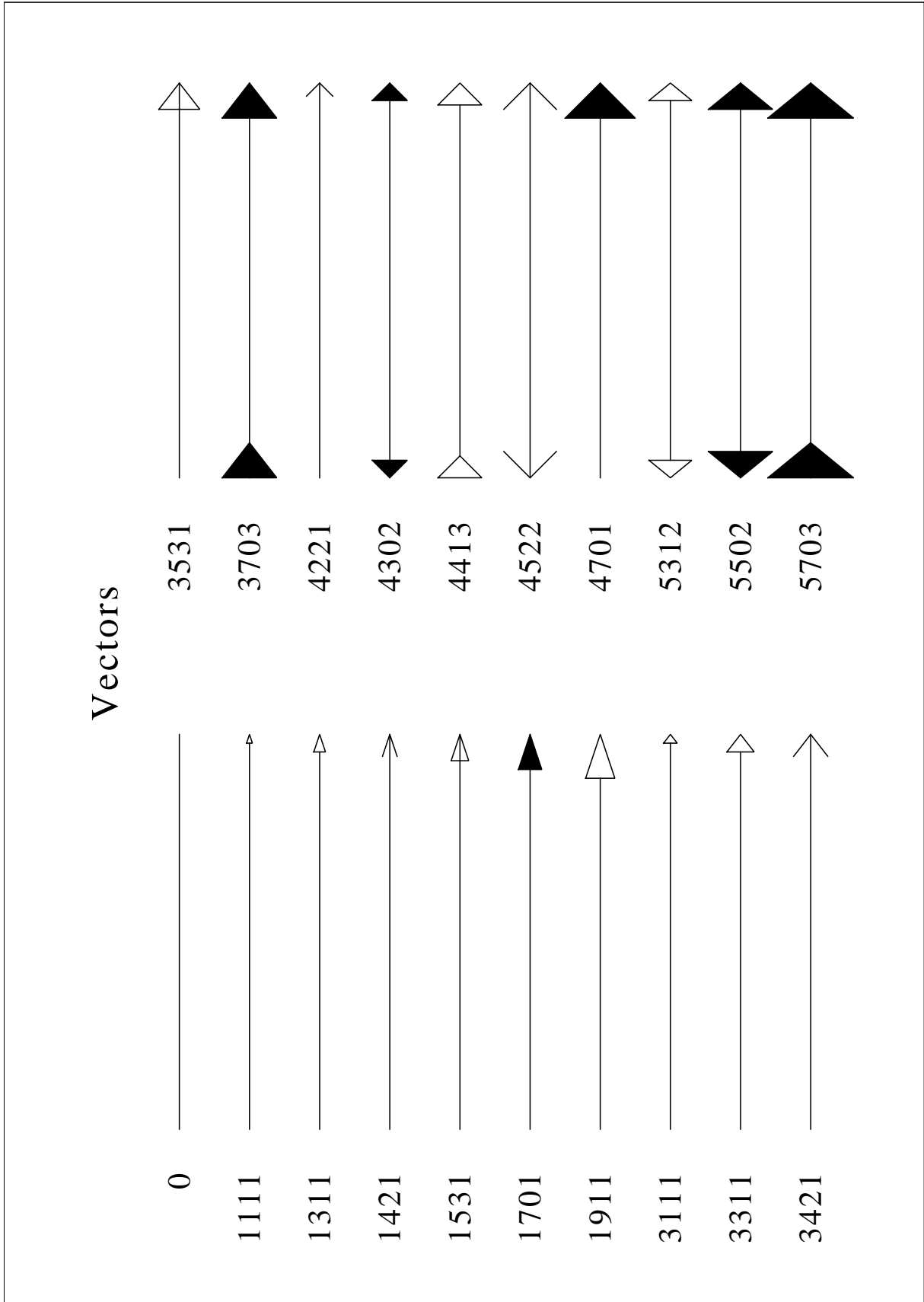


Figure B.8: Vectors

## B.9 3-D Colour Plot

```
package main

import "dislin"
import "math"

func main () {
    const n int = 50
    const m int = 50
    var zmat [n][m]float64

    ctit1 := "3-D Colour Plot of the Function"
    ctit2 := "F(X,Y) = 2 * SIN(X) * SIN (Y) "

    fpi := 3.1415927 / 180.0
    stepx := 360.0 / float64 (n - 1)
    stepy := 360.0 / float64 (m - 1)

    for i:= 0; i < n; i++ {
        x := float64 (i) * stepx
        for j:= 0; j < m; j++ {
            y := float64 (j) * stepy
            zmat[i][j] = 2 * math.Sin (x * fpi) * math.Sin (y * fpi)
        }
    }

    dislin.Metafl ("xwin")
    dislin.Disini ()
    dislin.Pagera ()
    dislin.Complx ()

    dislin.Titlin (ctit1, 1)
    dislin.Titlin (ctit2, 3)

    dislin.Name ("X-axis", "X")
    dislin.Name ("Y-axis", "Y")
    dislin.Name ("Z-axis", "Z")

    dislin.Intax ()
    dislin.Autres (n, m)
    dislin.Axspos (300, 1850)
    dislin.Ax3len (2200, 1400, 1400)

    dislin.Graf3 (0.0, 360.0, 0.0, 90.0, 0.0, 360.0, 0.0, 90.0,
        -2.0, 2.0, -2.0, 1.0)
    dislin.Crvmat (&zmat[0][0], n, m, 1, 1)
    dislin.Height (50)
    dislin.Title ()
    dislin.Disfin ()
}
```

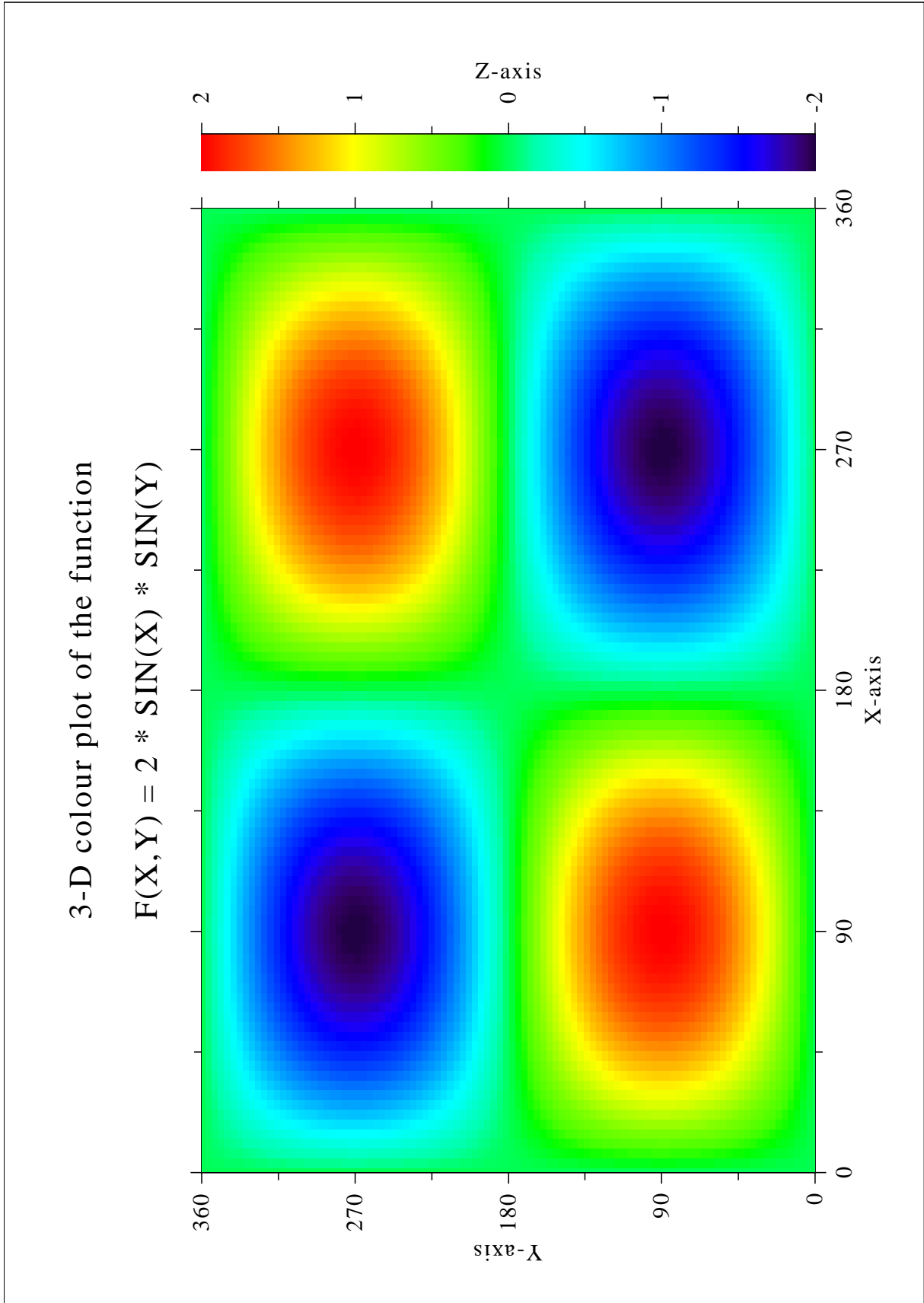


Figure B.9: 3-D Colour Plot

## B.10 Surface Plot

```
package main
import "dislin"
import "math"

func main () {
    const n int = 50
    const m int = 50
    var zmat [n][m]float64
    ctit1 := "Surface Plot of the Function"
    ctit2 := "F (X,Y) = 2 * SIN (X) * SIN (Y)"

    fpi := 3.1415927 / 180.0
    stepx := 360.0 / float64 (n - 1)
    stepy := 360.0 / float64 (m - 1)

    for i:= 0; i < n; i++ {
        x := float64 (i) * stepx
        for j:= 0; j < m; j++ {
            y := float64 (j) * stepy
            zmat[i][j] = 2 * math.Sin (x * fpi) * math.Sin (y * fpi)
        }
    }

    dislin.Metafl ("cons")
    dislin.Setpag ("da4p")
    dislin.Disini ()
    dislin.Pagera ()
    dislin.Complx ()

    dislin.Titlin (ctit1, 2)
    dislin.Titlin (ctit2, 4)

    dislin.Axspos (200, 2600)
    dislin.Axslen (1800, 1800)
    dislin.Name ("X-axis", "X")
    dislin.Name ("Y-axis", "Y")
    dislin.Name ("Z-axis", "Z")

    dislin.View3d (-5.0, -5.0, 4.0, "ABS")
    dislin.Graf3d (0.0, 360.0, 0.0, 90.0, 0.0, 360.0, 0.0, 90.0,
        -3.0, 3.0, -3.0, 1.0)
    dislin.Height (50)
    dislin.Title ()

    dislin.Color ("green")
    dislin.Surmat (&zmat[0][0], n, m, 1, 1)
    dislin.Disfin ()
}
```

Surface plot (SURMAT)

$$F(X,Y) = 2*\text{SIN}(X)*\text{SIN}(Y)$$

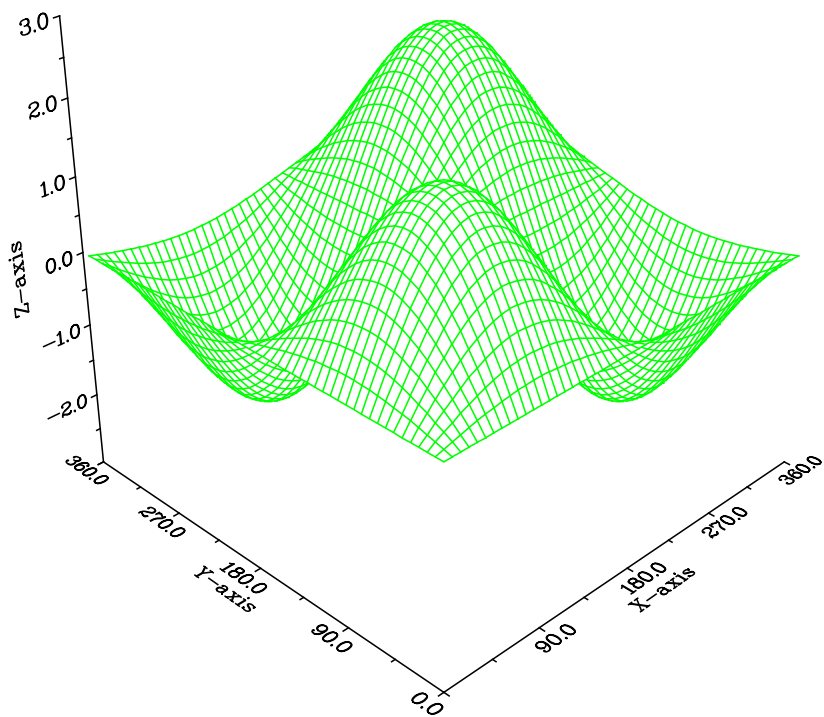


Figure B.10: Surface Plot

## B.11 Surface Plot

```
package main
import "dislin"
import "math"

func myfunc (x float64, y float64, iopt int) float64 {
    var xv float64

    if iopt == 1 {
        xv = math.Cos (x) * (3 + math.Cos (y))
    } else if iopt == 2 {
        xv = math.Sin (x) * (3 + math.Cos (y))
    } else {
        xv = math.Sin (y)
    }
    return xv
}

func main () {
    ctit1 := "Surface Plot of the Parametric Function"
    ctit2 := "[COS(t)*(3+COS(u)), SIN(t)*(3+COS(u)), SIN(u)]"
    pi := 3.1415927

    dislin.Metafl ("cons")
    dislin.Setpag ("da4p")
    dislin.Disini ()
    dislin.Pagera ()
    dislin.Complx ()

    dislin.Titlin (ctit1, 2)
    dislin.Titlin (ctit2, 4)

    dislin.Axspos (200, 2400)
    dislin.Axslen (1800, 1800)
    dislin.Name ("X-axis", "X")
    dislin.Name ("Y-axis", "Y")
    dislin.Name ("Z-axis", "Z")
    dislin.Intax ()

    dislin.Vkytit (-300)
    dislin.Zscale (-1.0,1.0)
    dislin.Surmsh ("on")

    dislin.Graf3d (-4.0,4.0,-4.0,1.0,-4.0,4.0,-4.0,1.0,-3.0,3.0,-3.0,1.0)
    dislin.Height (40)
    dislin.Title ()
    step := 2.0 * pi / 30.0
    dislin.Surfcp (myfunc, 0.0, 2.0 * pi, step, 0.0, 2.0 * pi, step)
    dislin.Disfin ()
}
```

Surface Plot of the Parametric Function  
[ $\text{COS}(t) \cdot (3 + \text{COS}(u))$ ,  $\text{SIN}(t) \cdot (3 + \text{COS}(u))$ ,  $\text{SIN}(u)$ ]

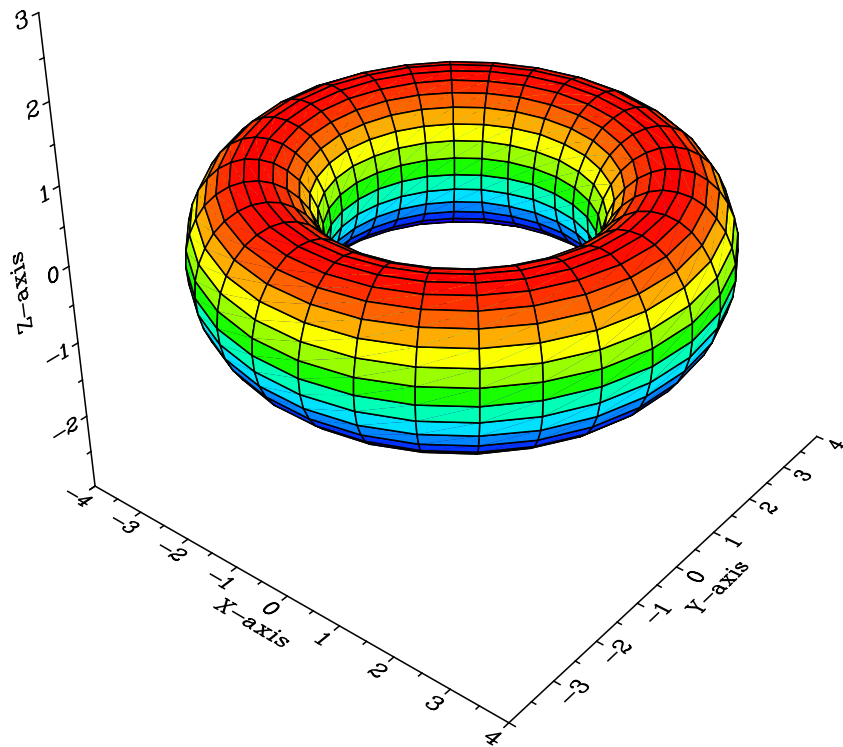


Figure B.11: Surface Plot of a Parametric Function

## B.12 Polar Plots

```
package main

import "dislin"
import "math"

func main () {
    const n int = 300
    const m int = 10
    var x1, y1 [n]float64
    var x2, y2 [m]float64

    pi := 3.1415926
    f := pi / 180.0
    step := 360.0 / float64 (n - 1)

    for i:= 0; i < n; i++ {
        a := (float64 (i) * step) * f
        y1[i] = a
        x1[i] = math.Sin (5 * a)
    }

    for i:= 0; i < m; i++ {
        x2[i] = float64 (i + 1)
        y2[i] = float64 (i + 1)
    }

    dislin.Setpag ("da4p")
    dislin.Scrmod ("revers")
    dislin.Metafl ("cons")
    dislin.Disini ()
    dislin.Complx ()
    dislin.Pagera ()

    dislin.Titlin ("Polar Plots", 2)
    dislin.Ticks (3, "Y")
    dislin.Axends ("NOENDS", "X")
    dislin.Labdig (-1, "Y")
    dislin.Axslen (1000, 1000)
    dislin.Axsorg (1050, 900)

    ic := dislin.Intrgb (0.95,0.95,0.95)
    dislin.Axsbgd (ic)
    dislin.Grafp (1.0, 0.0, 0.2, 0.0, 30.0);
    dislin.Color ("blue")
    dislin.Curve (&x1[0], &y1[0], n)
    dislin.Color ("fore")
    dislin.Htitle (50)
    dislin.Title ()
    dislin.Endgrf ()
}
```



```
dislin.Labdig (-1, "X")
dislin.Axsorg (1050, 2250)
dislin.Labtyp ("VERT", "Y")
dislin.Grafp (10.0, 0.0, 2.0, 0.0, 30.0)
dislin.Barwth (-5.0)
dislin.Polcrv ("FBARS")
dislin.Color ("blue")
dislin.Curve (&x2[0], &y2[0], m)

dislin.Disfin ()
}
```

# Polar Plots

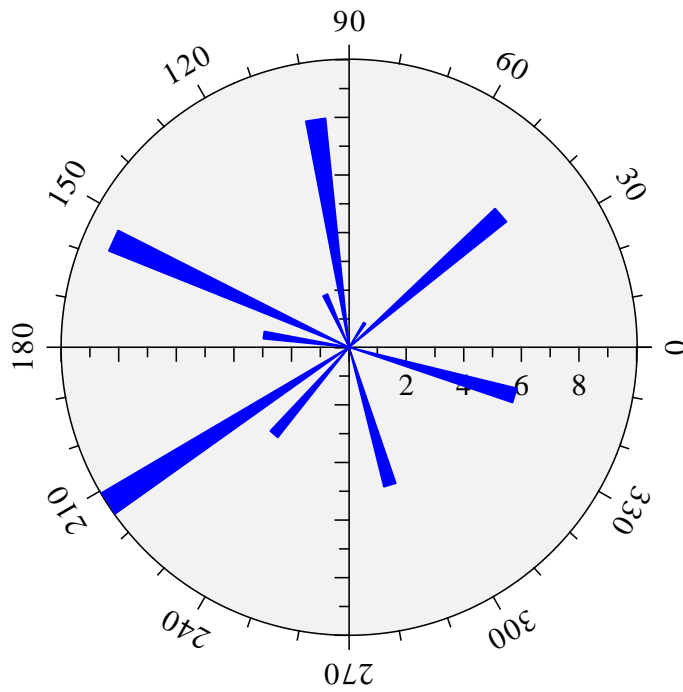
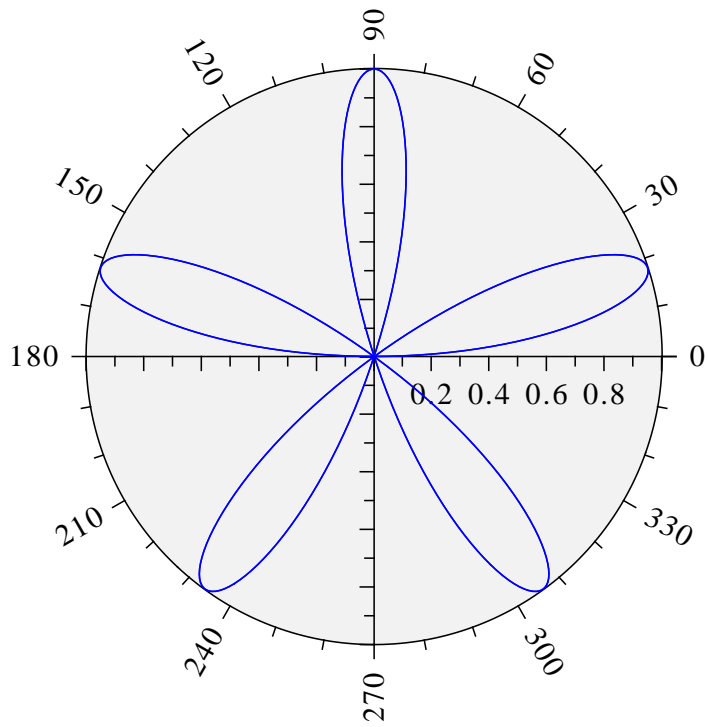


Figure B.12: Polar Plots

## B.13 Contour Plot

```
package main

import "dislin"
import "math"

func main () {
    const n int = 50
    const m int = 50
    var xray [n] float64
    var yray [m] float64
    var zmat [n][m]float64

    ctit1 := "Contour Plot"
    ctit2 := "F(X,Y) = 2 * SIN(X) * SIN(Y) "

    fpi    := 3.1415927 / 180.0
    stepx  := 360.0 / float64 (n - 1)
    stepy  := 360.0 / float64 (m - 1)

    for i:= 0; i < n; i++ {
        xray[i] = float64 (i) * stepx
    }

    for i:= 0; i < m; i++ {
        yray[i] = float64 (i) * stepy
    }

    for i:= 0; i < n; i++ {
        x := xray[i] * fpi
        for j:= 0; j < m; j++ {
            y := yray[j] * fpi
            zmat[i][j] = 2 * math.Sin (x) * math.Sin (y)
        }
    }

    dislin.Metafl ("cons")
    dislin.Setpag ("da4p")
    dislin.Disini ()
    dislin.Pagera ()
    dislin.Complx ()

    dislin.Titlin (ctit1, 1)
    dislin.Titlin (ctit2, 3)

    dislin.Intax ()
    dislin.Axspos (450, 2650)

    dislin.Name ("X-axis", "X")
    dislin.Name ("Y-axis", "Y")
}
```

```

dislin.Graf (0.0, 360.0, 0.0, 90.0, 0.0, 360.0, 0.0, 90.0)
dislin.Height (50)
dislin.Title ()

dislin.Height (30)

for i:= 0; i < 8; i++ {
    zlev := -2.0 + float64 (i) * 0.5

    if i == 4 {
        dislin.Labels ("NONE", "CONTUR")
    } else {
        dislin.Labels ("FLOAT", "CONTUR")
    }

    dislin.Setclr ( (i+1) * 28)
    dislin.Contur (&xray[0], n, &yray[0], m, &zmat[0][0], zlev)
}
dislin.Disfin ()
}

```

### Contour Plot

$$F(X,Y) = 2 * \text{SIN}(X) * \text{SIN}(Y)$$

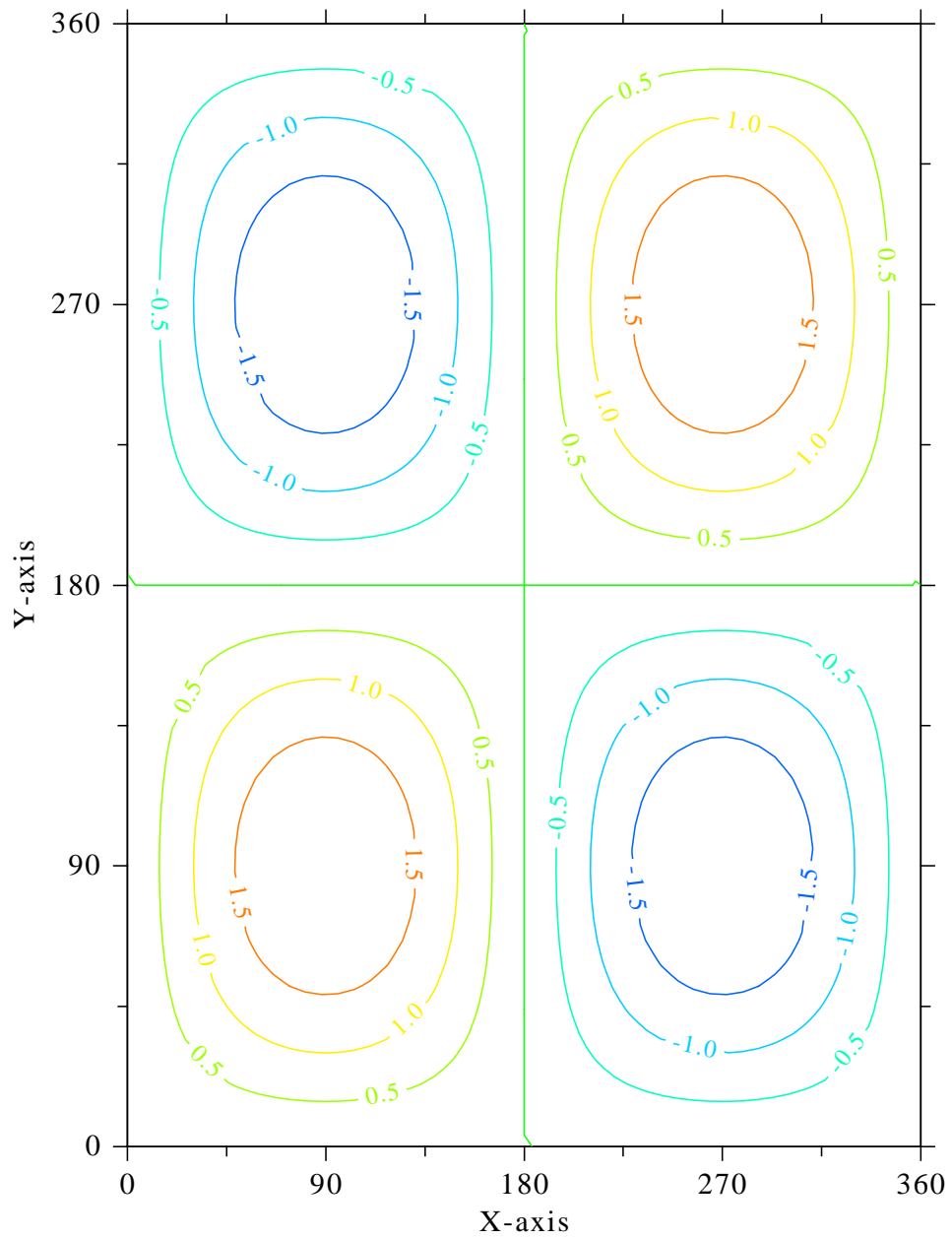


Figure B.13: Contour Plot

## B.14 Shaded Contour Plot

```
package main
import "dislin"
func main () {
    const n int = 50
    const m int = 50
    var xray [n] float64
    var yray [m] float64
    var zmat [n][m]float64
    var zlev [12] float64

    ctit1 := "Shaded Contour Plot"
    ctit2 := "F(X,Y) =(X[2$ - 1) [2$ + (Y[2$ - 1) [2$"
    stepx := 1.6 / float64 (n - 1)
    stepy := 1.6 / float64 (m - 1)
    for i:= 0; i < n; i++ {
        xray[i] = float64 (i) * stepx
    }
    for i:= 0; i < m; i++ {
        yray[i] = float64 (i) * stepy
    }
    for i:= 0; i < n; i++ {
        x := xray[i] * xray[i] - 1.0
        x = x * x
        for j:= 0; j < m; j++ {
            y := yray[j] * yray[j] - 1.0
            zmat[i][j] = x + y * y
        }
    }
    }
    dislin.Metafl ("cons")
    dislin.Setpag ("da4p")
    dislin.Disini ()
    dislin.Pagera ()
    dislin.Complx ()
    dislin.Mixalf ()

    dislin.Titlin (ctit1, 1)
    dislin.Titlin (ctit2, 3)
    dislin.Name ("X-axis", "X")
    dislin.Name ("Y-axis", "Y")
    dislin.Axspos (450, 2670)
    dislin.Shdmod ("poly", "contur")
    dislin.Graf (0.0, 1.6, 0.0, 0.2, 0.0, 1.6, 0.0, 0.2)
    for i:= 0; i < 12; i++ {
        zlev[11-i] = 0.1 + float64 (i) * 0.1
    }
    dislin.Conshd (&xray[0], n, &yray[0], m, &zmat[0][0], &zlev[0], 12)
    dislin.Height (50)
    dislin.Title ()
    dislin.Disfin ()
}
```

### Shaded Contour Plot

$$F(X,Y) = (X^2 - 1)^2 + (Y^2 - 1)^2$$

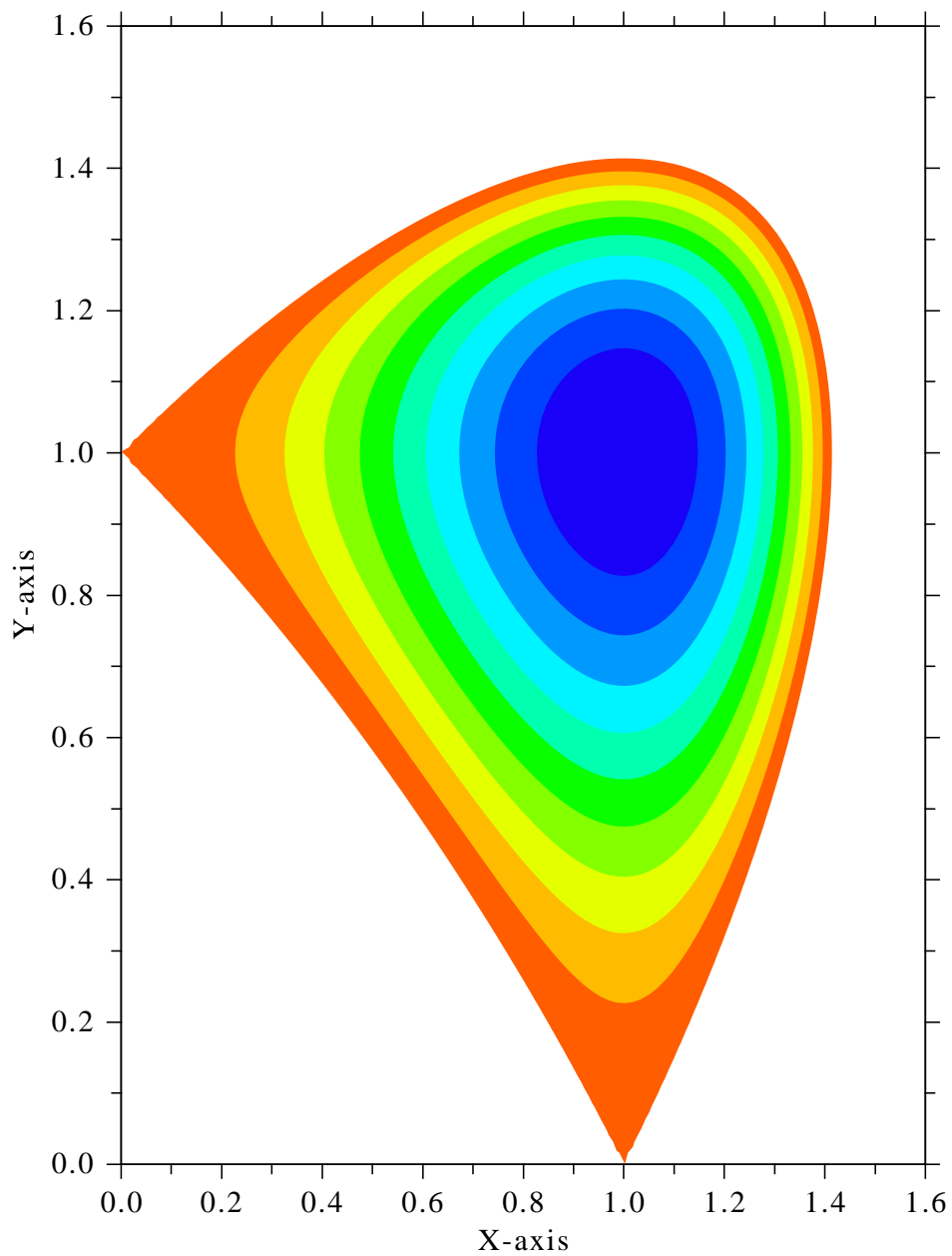


Figure B.14: Shaded Contour Plot

## B.15 Pie Charts

```
package main
import "dislin"

func main () {
    var cbuf [14]int8
    xray := [5]float64{1.0, 2.5, 2.0, 2.7, 1.8}

    ctit := "Pie Charts(PIEGRF)"

    dislin.Setpag ("da4p")
    dislin.Metafl ("cons")
    dislin.Disini ()
    dislin.Pagera ()
    dislin.Complx ()
    dislin.Chnpie ("BOTH")

    dislin.Axslen (1600, 1000)
    dislin.Titlin (ctit, 2)

    dislin.Legini (&cbuf[0], 5, 8)
    dislin.Leglin (&cbuf[0], "FIRST", 1)
    dislin.Leglin (&cbuf[0], "SECOND", 2)
    dislin.Leglin (&cbuf[0], "THIRD", 3)
    dislin.Leglin (&cbuf[0], "FOURTH", 4)
    dislin.Leglin (&cbuf[0], "FIFTH", 5)

    dislin.Patcyc (1, 7)
    dislin.Patcyc (2, 4)
    dislin.Patcyc (3, 13)
    dislin.Patcyc (4, 3)
    dislin.Patcyc (5, 5)

    dislin.Axspos (250, 2800)
    dislin.Piegrf (&cbuf[0], 1, &xray[0], 5)
    dislin.Endgrf ()

    dislin.Axspos (250, 1600)
    dislin.Labels ("DATA", "PIE")
    dislin.Labpos ("EXTERNAL", "PIE")
    dislin.Piegrf (&cbuf[0], 1, &xray[0], 5)

    dislin.Height (50)
    dislin.Title ()
    dislin.Disfin ()
}
```



## Pie Charts (PIEGRF)

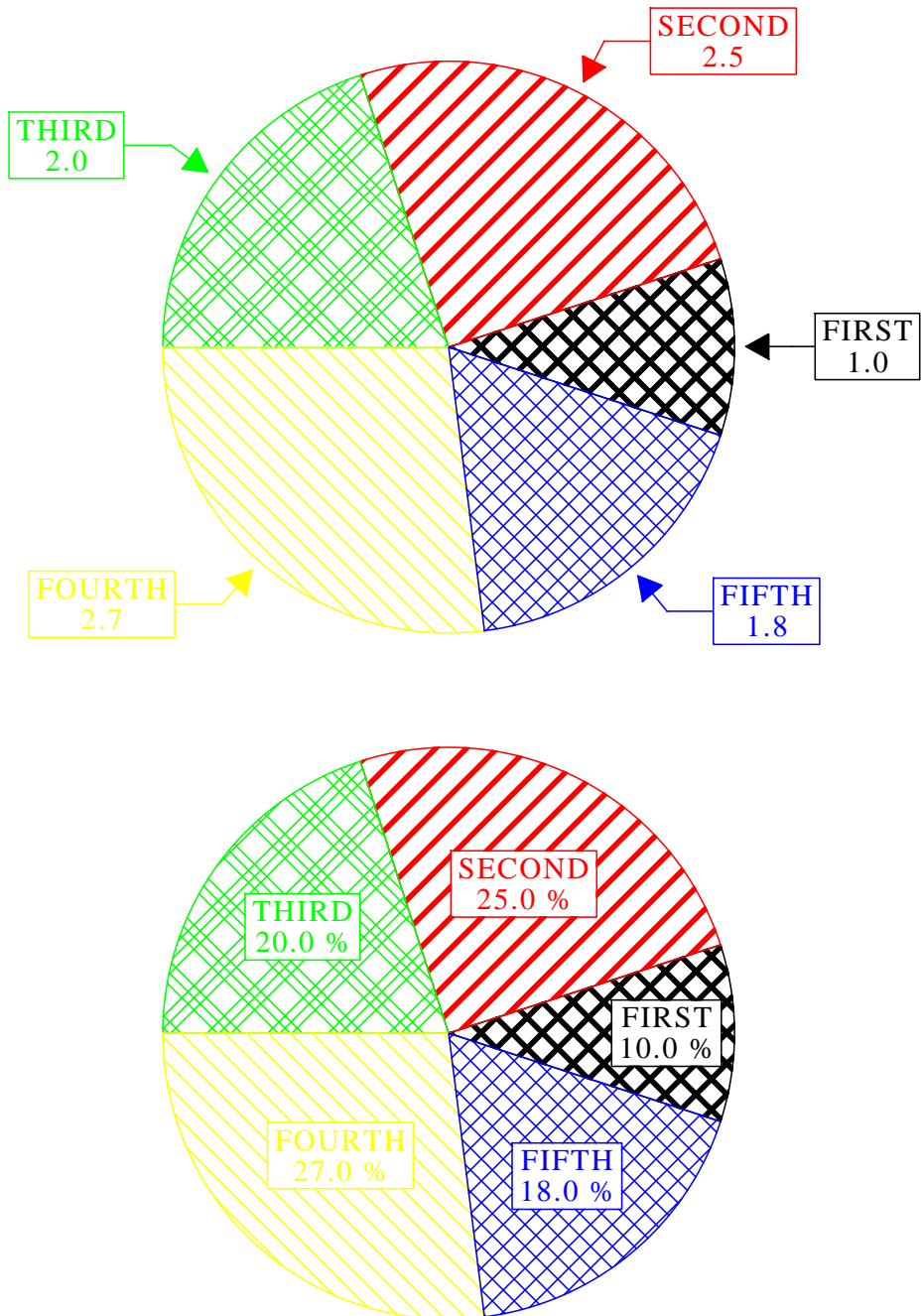


Figure B.15: Pie Charts

## B.16 World Coastlines and Lakes

```
package main

import "dislin"

func main () {
    dislin.Metafl ("cons")
    dislin.Disini ()
    dislin.Pagera ()
    dislin.Complx ()

    dislin.Axspos (400, 1850)
    dislin.Axslen (2400, 1400)

    dislin.Name ("Longitude", "X")
    dislin.Name ("Latitude", "Y")
    dislin.Titlin ("World Coastlines and Lakes", 3)

    dislin.Labels ("MAP", "XY")
    dislin.Labdig (-1, "XY")
    dislin.Grafmp (-180.0, 180.0, -180.0, 90.0, -90.0, 90.0, -90.0, 30.0)

    dislin.Gridmp (1, 1)
    dislin.Color ("green")
    dislin.World ()

    dislin.Color ("foreground")
    dislin.Height (50)
    dislin.Title ()

    dislin.Disfin ()
}
```

World coastlines and lakes

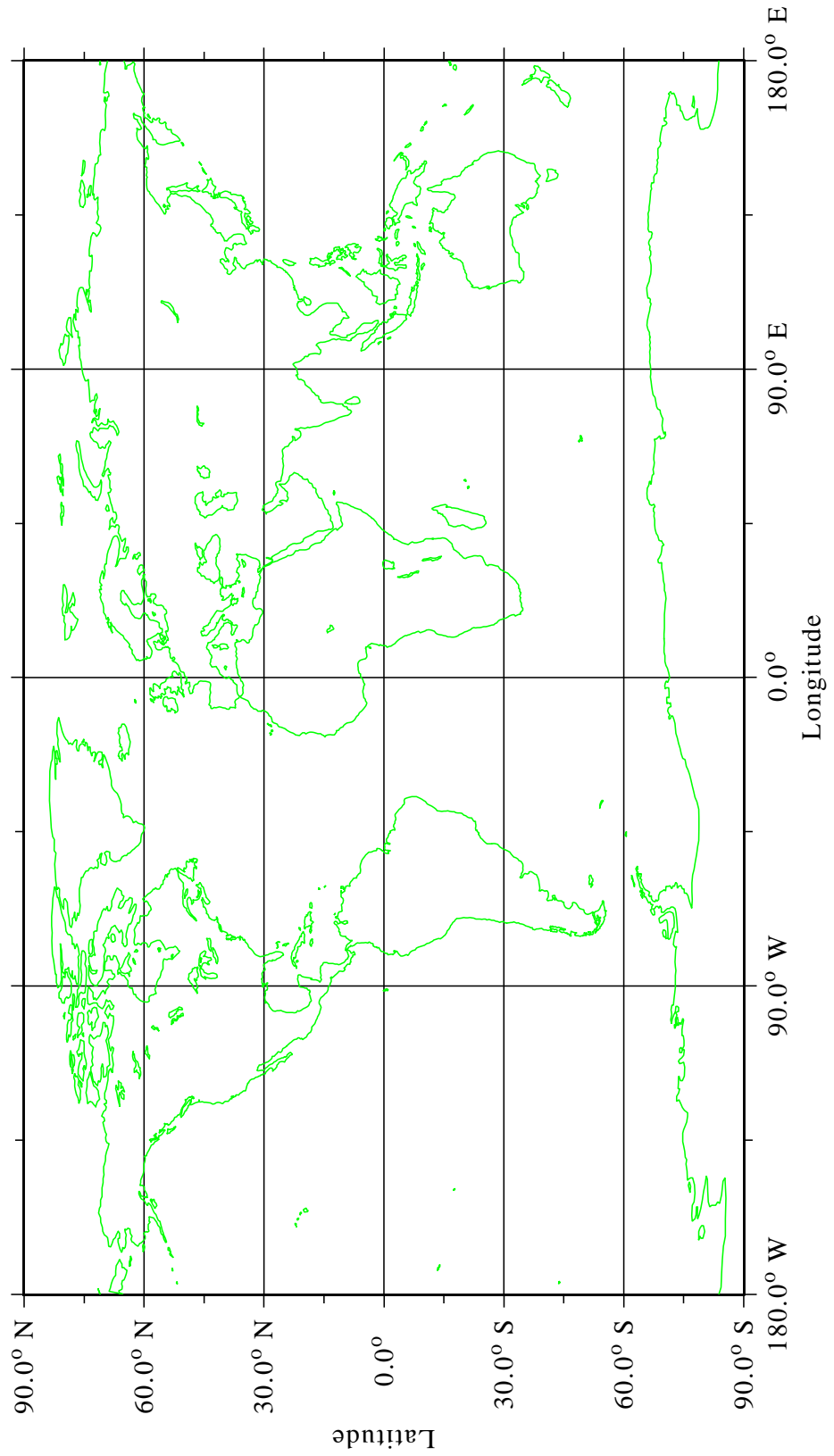


Figure B.16: World Coastlines and Lakes

