

New Dislin Features since Version 11.0

This article describes new features and options of Dislin which are added to the software since version 11.0 and not covered by the current Dislin book version 11. The current version number of Dislin is 11.5.

Chapter 4: Plotting Axis Systems and Titles

GRAFR

The routine GRAFR plots a two-dimensional axis system for a Smith chart, where the non negative impedance or admittance plane is projected to a complex reflexion coefficient plane, which is displayed in the unity radius region. The projection is done by the formula $r = (z - 1) / (z + 1)$, where z and r are complex numbers. For admittance, $1 / Z$ is used in the formula.

The call is: `CALL GRAFR (XRAY, N, YRAY, M)` level 1
or: `void grafr (const float *xray, int n, const float *yray, int m);`

XRAY is an array of non negative real values of complex impedance or admittance data. The values are plotted as labels at the X-axis.

N is the dimension of XRAY.

YRAY is an array of imaginary values of complex impedance or admittance data. The values are plotted as labels at the Y-axis (unity circle).

M is the dimension of YRAY.

Additional notes: - The type of the Smith axis system , impedance or admittance, can be defined with the routine AXSTYP.

- The conversion routine GETRCO calculates the reflection factor r for a impedance z by the formula $r = (z - 1) / (z + 1)$. The reverse transformation $z = (1 + r) / (1 - r)$ is done by GETICO. For admittance, $1 / z$ is used for the conversions above.

- Additional grid lines in a Smith chart can be plotted with the routines GRIDRE and GRIDIM.

- A similar axis system for a Smith chart can be created with the normal GRAF routine, where the scaling is defined from -1.0 to 1.0 for both axes. For that case, values must be converted by GETRCO to reflection factors before passing them to plot routines. For GRAFR, this is done automatically in plot routines.

GRIDRE

GRIDRE plots a grid line of a constant real part in a Smith chart.

The call is: `CALL GRIDRE (ZRE, ZIMG1, ZIMG2, N)` level 2
or: `void gridre (float zre, float zimg1, float zimg2, int n);`

ZRE is the constant real value of the grid line (≥ 0.0).

ZIMG1, ZIMG2 are the start and end imaginary parts of the grid line.

N is the resolution of the curve, which means the number of generated points between ZIMG1 and ZIMG2.

GRIDIM

GRIDIM plots a grid line of a constant imaginary part in a Smith chart.

The call is:	CALL GRIDIM (ZIMG, ZRE1, ZRE2, N)	level 2
or:	void gridim (float zimg, float zre1, float zre2, int n);	
ZIMG	is the constant imaginary value of the grid line.	
ZRE1, ZRE2	are the start and end real parts of the grid line (≥ 0.0).	
N	is the resolution of the curve.	

Example:

The Fortran program

```
PROGRAM SMITH
PARAMETER(N=1000, M=1000)
DIMENSION X(N), Y(N), ZIMG(11), ZRE(5)
DATA ZIMG/5.0, 2.0, 1.0, 0.5, 0.2, 0.0, -0.2, -0.5,
*      -1.0, -2.0, -5.0/
DATA ZRE/0.2, 0.5, 1.0, 2.0, 5.0/

STEP = (50.0 + 50.0) / (N - 1)
DO I=1,N
  X(I) = 1.0
  Y(I) = -50.0 + I * STEP
END DO

CALL METAFL ('PDF')
CALL DISINI ()
CALL HWFONT ()
CALL NAME ('X-axis', 'X')
CALL NAME ('Y-axis', 'Y')
CALL LABTYP ('HORI', 'POLAR')
CALL LABTYP ('VERT', 'X')
CALL GRAFR (ZRE, 5, ZIMG, 11)

CALL GRIDRE (1.0, -50.0, 50.0, M)
CALL GRIDRE (2.0, -50.0, 50.0, M)
CALL GRIDRE (3.0, -50.0, 50.0, M)
CALL GRIDRE (4.0, -50.0, 50.0, M)
CALL GRIDRE (5.0, -50.0, 50.0, M)

CALL GRIDIM (0.5, 0.001, 50.0, M)
CALL GRIDIM (1.0, 0.001, 1.0, M)
CALL GRIDIM (2.0, 0.001, 50.0, M)
CALL GRIDIM (3.0, 0.001, 50.0, M)
CALL GRIDIM (4.0, 0.001, 50.0, M)
CALL GRIDIM (5.0, 0.001, 50.0, M)

CALL CURVE (X, Y, N)
CALL DISFIN ()
END
```

produces the following figure:

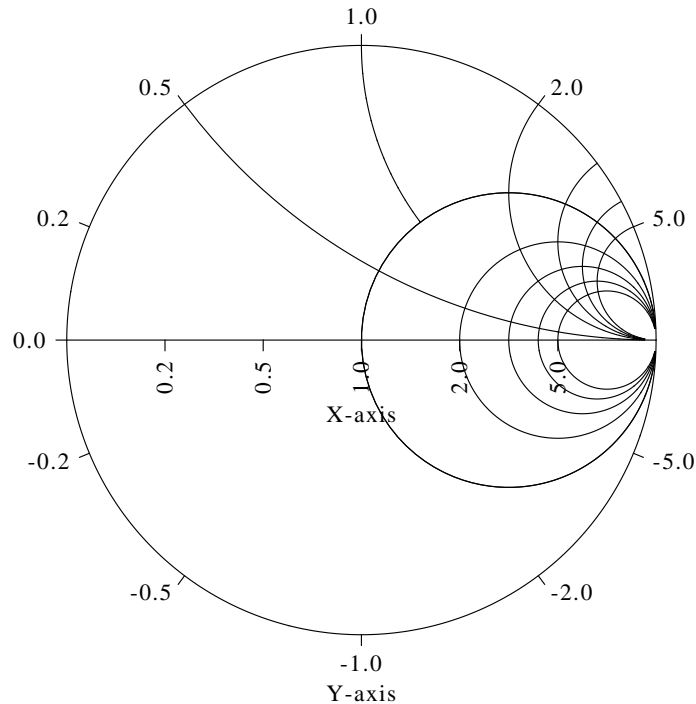


Figure 4.1: Smith Plot

Y P O L A R

This routine plots a secondary axis routine for a polar or Smith axis system.

The call is: `CALL YPOLAR (A, B, OR, STEP, CSTR, NDIST)`
or: `void ypolar (float a, float b, float or, float step, char *cstr, int ndist);`

A, B are the lower and upper limits of the axis.
OR, STEP are the first label and the step between labels.
CSTR is a character string containing the axis name.
NDIST is the distance to the polar or Smith circle in plot coordinates.

Chapter 5: Plotting Curves

L I N F I T

LINFIT plots a straight line that has the best fit to a series of data points.

The call is: `CALL LINFIT (XRAY, YRAY, N, A, B, R, COPT)` level 2, 3
or: `void linfit (const float *xray, const float yray, int n, float *a, float *b, float *r, const char *copt);`

XRAY, YRAY are arrays that contain X- and Y-coordinates.

N	the number of data points.
A, B	are the returned values of the calculated line $Y = A * X + B$.
R	is the returned correlation coefficient of the fit between -1.0 and 1.0. A value around zero means no correlation, a value near -1.0 or 1.0 means good correlation.
COPT	is a character string that can have the values 'NONE', 'ALL' and 'LINE'. 'NONE' means that just the values A, B and R are calculated. Nothing is plotted. For that case LINFIT can also be called in the levels 0 and 1. 'LINE' means that the straight line is plotted and 'ALL' that the straight line and the data points are plotted.

LEGEND

The new position values 9 and 10 are added for plotting the legend in the left and right center of the page. The number of possible legend lines is increased from 30 to 80.

LEG TYP

The option 'TABLE' is added to LEGTYP for plotting legends in table form.

LEG TBL

This routine defines the number of columns in table legends and the direction in which the legend entries are plotted.

The call is:	CALL LEGTBL (N, COPT)	level 1, 2, 3
or:	void legtbl (int n, const char copt);	
N	is the number of columns.	
COPT	is a character string that can have the values 'VERT' and 'HORI'. Default: (3, 'VERT').	

Chapter 6: Parameter Setting Routines

METAFL

The Windows Enhanced Metafile format EMF is supported by DISLIN. EMF files can be created with the keyword 'EMF' in METAFL.

AXSTYP

The new options 'IMPEDANCE' and 'ADMITTANCE' define the type of the Smith axis system plotted by GRAFR.

LABTYP

The new option ('HORI', 'POLAR') allows true horizontal labels on polar and Smith plots.

POLCRV

The new keyword 'CURVE' defines a curve interpolation between two points in a Smith chart. The limitation of the number of points for spline and parametric spline interpolation is removed.

GAPSI Z

The 'Z' option is added to GAPSIZ for enabling gaps in 3D curves.

The following mathematical symbols are added to the LaTeX symbols:

<code>\approx</code>	<code>\leqslant</code>	<code>\geqslant</code>	<code>\nless</code>	<code>\ngtr</code>	<code>\nleq</code>	<code>\ngeq</code>
<code>\nleqslant</code>	<code>\ngeqslant</code>	<code>\prec</code>	<code>\succ</code>	<code>\nprec</code>	<code>\nsucc</code>	<code>\preceq</code>
<code>\succeq</code>	<code>\npreceq</code>	<code>\nsucceq</code>	<code>\ll</code>	<code>\gg</code>	<code>\lll</code>	
<code>\ggg</code>	<code>\nsubseteq</code>	<code>\nsupseteq</code>	<code>\sqsubset</code>	<code>\sqsupset</code>	<code>\sqsubseteq</code>	
<code>\sqsupseteq</code>	<code>\doteq</code>	<code>\simeq</code>	<code>\propto</code>	<code>\angle</code>	<code>\measuredangle</code>	
<code>\circ</code>	<code>\sphericalangle</code>					

Chapter 9: Utility Routines

GETRCO

GETRCO converts a complex impedance value to a reflection factor by the formula $r = (z - 1) / (z + 1)$.

The call is: `CALL GETRCO (ZRE, ZIMG, RRE, RIMG)` level 0, 1, 2, 3
or: `void getrco (float zre, float zimg, float *rre, float *rimg);`
ZRE, ZIMG are the real and imaginary parts of z.
RRE, RIMG are the returned real and imaginary parts of r.

GETICO

GETICO converts a complex reflection factor to an impedance by the formula $z = (1 + r) / (1 - r)$.

The call is: `CALL GETICO (RRE, RIMG, ZRE, ZIMG)` level 0, 1, 2, 3
or: `void getico (float rre, float rimg, float *zre, float *zimg);`
RRE, RIMG are the real and imaginary parts of r.
ZRE, ZIMG are the returned real and imaginary parts of

CSRPOL

CSRPOL is a similar routine to CSRPTS. It returns an array of mouse positions, where help lines are plotted between the points. CSRPOL is waiting for mouse button 1 clicks and terminates if mouse button 2 is pressed.

The call is: `CALL CSRPOL (NXRAY, NYRAY, NMAX, N, IRET)` level 1, 2, 3
or: `void csrpoly (int *nxray, int *nyray, int nmax, int *n, int *iret);`
NXRAY, NYRAY are the returned coordinates of the collected mouse positions.
NMAX is the dimension of NXRAY and NYRAY and defines the maximal number of points that will be stored in NXRAY and NYRAY.
N is the number of points that are returned in NXRAY and NYRAY.
IRET is a returned status. IRET not equal 0 means that not all mouse movements could be stored in NXRAY and NYRAY.

Chapter 12: 3-D Graphics

GRFIMG

The routine GRFIMG includes a PNG, BMP, TIFF or GIF file into a 3-D plane defined by GRFINI. This routine can only be used if the output format is a raster format (screen or image file).

The call is: `CALL GRFIMG (CFIL)` level 1, 2, 3
or: `void grfimg (char *cfil);`
CFIL is a character string that contains the filename.

SETRES3D

The routine SETRES3D sets the symbol size for the 3-D symbol with the number 0 (cube) plotted by SYMB3D, CURV3D and CURV4D.

The call is: `CALL SETRES3D (XL, YL, ZL)` level 1, 2, 3
or: `void setres3d (float xl, float yl, float zl);`
XL, YL, ZL is the cube size in absolute 3-D coordinates.
Default: (0.08, 0.08, 0.08).

AUTRES3D

The routine AUTRES3D calculates the symbol size for cubes from the number of data points.

The call is: `CALL AUTRES3D (IXDIM, IYDIM, IZDIM)` level 1, 2, 3
or: `void autres3d (int ixdim, int iydim, int izdim);`
IXDIM, IYDIM, IZDIM are the number of data points in the X-, Y- and Z-directions.
Additional note: HSYM3D, SETRES3D and AUTRES3D can overwrite each other for the symbol 'cube'.

Chapter 14: Contouring

CONTUR2

The routine CONTUR2 calculates and plots contours of the function $Z = F(X,Y)$, where the functions values are located on a curvilinear grid.

The call is: `CALL CONTUR2 (XMAT, YMAT, ZMAT, N, M, ZLEV)` level 2, 3
or: `void contur2 (const float *xmat, const float *ymat, const float *zmat, int n, int m, float zlev);`
XMAT is a matrix of the dimension (N, M) containing the X-coordinates of the curvilinear grid.
YMAT is a matrix of the dimension (N, M) containing the Y-coordinates of the curvilinear grid.
ZMAT is a matrix of the dimension (N, M) containing function values.
N, M define the dimension of XMAT, YMAT and ZMAT.
ZLEV is a function value that defines the contour line to be calculated. ZLEV can be used for labels.

CONSHD2

The routine CONSHD2 plots filled contours of the function $Z = F(X,Y)$, where the functions values are located on a curvilinear grid.

The call is:	CALL CONSHD2 (XMAT, YMAT, ZMAT, N, M, ZLVRAY, NLV) level 2, 3
or:	void conshd2 (const float *xmat, const float *ymat, const float *zmat, int n, int m, <div style="text-align: right;">const float *zlvray, int nlv);</div>
XMAT	is a matrix of the dimension (N, M) containing the X-coordinates of the curvilinear grid.
YMAT	is a matrix of the dimension (N, M) containing the Y-coordinates of the curvilinear grid.
ZMAT	is a matrix of the dimension (N, M) containing function values.
N, M	define the dimension of XMAT, YMAT and ZMAT.
ZLVRAY	is an array containing the levels. For polygon filling, the levels should be sorted in such a way that inner contours are plotted last.
NLV	is the number of levels.

Chapter 15: Widget Routines

SWGOPT

The following both options are added to SWGOPT:

SWGOPT (file, 'BACKGROUND') sets an image as background for main widgets, where the parameter 'file' should contain the name of a BMP file. If the image is smaller than the size of the main widget, the image will be duplicated. Otherwise, it will be clipped.

SWGOPT ('NOFIELD' or 'FIELD', 'FILE') disables or enables a text widget created by WGFIL for the selected file name. The default value is 'FIELD'.

SWGCB2

The routine SWGCB2 accepts now also callback routines for main widgets. The callback routine is invoked when the size of the main widget has changed.