List of GIFA Commands

Version 4.0

ABS

Computes absolute value of data-sets, Works only on real data-sets (itype=0)

see also : MODULUS REAL

ABSMAX

Absolute factor to which scale refers to for displaying and plotting. Usually ABSMAX holds the largest point of the data-set, but you can set it by hand in order to do absolute plots. ABSMAX is recomputed whenever the data is changed.

You can force the ABSMAX to be recomputed by setting it to zero. You can also hampers the recomputation of absmax (which can be VERY long (on 3D for instance)) by setting it to a standard value at the end of the command line

see also : MAX SCALE DISP

ADD

ADD name_of_file -or- ADDC name_of_file

permits to add to the current data-set the content of the file name_of_file, which is in standard format. Works for both 1D, 2D and 3D data-sets.

see also ADDH, ADDDATA MULT READ

ADDBASE

ADDBASE constant

Removes a constant to the data. The default value is the value of SHIFT (computed by EVALN).

see also : EVALN SHIFT

ADDC

Equivalent to ADD
see also : ADD

ADDDATA

Add the contents of the DATA buffer to the current data-set. Equivalent to ADD but in-memory.
see also : MULTDATA MAXDATA MINDATA ADD ADDH MULT

ADDH

ADDH name_of_file
Same as ADD, with H format files.

see also : ADD ADDDATA MULT CONCAT

ADDNOISE

addnoise noise seed

add to the current data-set (1D, 2D, 3D) a white-gaussian, characterized
by its level noise, and the random generator seed.

see also: SIMUN, SIMU, EVALN

ALERT

alert text
Creates a graphic alert box displaying the text. User must click in the alert box to continue. Works only if the graphic mode has been intialzed by isuing, at least once, a graphic command.

see also : PRINT ERROR

ALGO

Context used to describe the current algorithm

Algo is a number of the form abc
where a, b and, c are the decimal digits.
c: 0 Gull and Daniel equation for Entropy
1 GIFA equation for Entropy

2 steepest descent equation is used

3 conjugate gradient is used

b: 0 single step iteration

1 line-maximization using parabolic fit.

2 : line-maximization using bracketing before parabolic fit.

a: 0 no Wu correction

1 Wu correction is applied.

see also : MAXENT gifa conjg gad

ALPHA

alpha a

This context is the angle along the OZ axis by which the cube is rotated during a 3D display with the DISP3D/REF3D set of commands

see also : BETA GAMA DISP3D

APPLY

APPLY what_to_apply

Apply a windowing function to the data-set.

You can apply :

FILTER : applies the filter used during MaxEnt iterations, depends on LB, GB, JCONS, FILTER

WINDOW : applies the window used during MaxEnt iterations.

see also : SHOW GET PUT FILTER WINDOW LB GB JCONS

APPROX

Used in the BCORR 3 module, to choose the way the baseline is estimated.

0 : with ITERMA2 times a moving average filtering on a window of
size WINMA2.

1 : with a Legendre polynomial approximation of degree DEGRE.

+10 : with a linear interpolation of signal before approximation.

+100 : with an 'elastic effect' that is a rude way to prevent
from burying the weak lines.

see also : BCORR

AR->DT

AR->DT size n

Extend the data-set up to size points long by predicting the missing data points, using linear prediction.

n = 1 : following points predicted -> forward prediction

n = 2 : previous points predicted -> backward prediction

see also : BURG DT->AR SVD->AR ORDER

AR->RT

AR->RT n

Solve the prediction-error polynomial, calculated from the autoregressive coefficients. This is the third step of the LP-SVD method.

n = 1 : forward coefficients are used to extracted forward roots

n = 2 : backward coefficients are used to extracted backward roots

n = 3 : both set of roots are computed

see also : ARLIST RT->AR AR->RT2 ORDER SVD->AR

AR->RT2

AR->RT2 n

Equivalent to AR->RT, but seems to be more stable for very large polynomial.

see also : ARLIST RT->AR ORDER SVD->AR

AR->SP

Calculate the modulus of spectrum from the autoregressive Burg coefficients. This is the so-called Burg spectrum, (sometimes unfortunately called mem spectrum). The spectrum is computed to the size of the current 1D buffer size.

see also : DT->AR ORDER

ARLIST

ARLIST n i j

list the autoregressive coefficients from entry i to entry j.

n = 1 : forward coefficients

n = 2 : backward coefficients

see also : AR->DT AR->RT RT->AR

AXIS3D

axis3d fx where x is 1, 2, 3, 12, 13, or 123

This context holds the focal length that will be used for computing a 3D display with the DISP3D/REF3D set of commands

see also : REF3D CHECK3D DISP3D DISP3D?

BACKPROD

Replaces the 2D buffer by the product of the current 2D buffer by the transpose of it

see also : DIAGONALIZE

BCOLOR

BCOLOR index window

Defines the current background color used for display. The chosen color remains until the next BCOLOR command.

index is : 1:white 2:red 3:yellow 4:green
5:cian 6:blue 7:purple 8:black

Window is : 1:1D 2:disp 3:contour 4:3D

see also : DISP1D, DISP2D, DISP3D, CDISP2D, COLOR, CCOLOR, SCOLOR

BCORR

BCORR A { R {n} list_of_points 0 }

Computes and applies a base-line correction to the current data set.

A describe the algorithm used:
1 is linear correction
2 is cubic spline correction.
3 is polynomial (and related) correction

o If A is 1 or 2 :

R is the radius around which each pivot point is averaged.

n in 2D is either f1 or f2 (dimension in which correction is applied).

list_of_points is then the list of the pivot points used for the base-line correction. The list finishes by 0. The content of th point stack is used for prompting.
Linear correction can use 1 or more pivot points. 1 point corresponds to correction of a continuous level. Spline corrections needs at least 3 points.
In any case maximum is 100 pivot points.

o If A is 3 :
This method uses a correction by estimation of the baseline and subtraction. Words in uppercase are commands.
The four steps are :

* Initial smooth of data : SMOOTH1
* First segmentation (detection of signal) : SEGM1
* Second segmentation (Polynomial approximation
  may be done on several areas) : SEGM2
* Approximation (the estimated baseline is an approximation
  of data points that are not signal) : APPROX

If you do not know how to use these corrections the next three commands should help you :

* BCORRP : Will prompt you for the related parameters, along with the name of the command..
* BCORRP? : the current choices are listed.
* BCORRP0 : the default set up is restored.
* BCORRP1 : another set up is activated

Polynomial correction is described by a set of related external contexts :
BLCW, BLCITER, BLOCBASE, BCORRP, BCORRP?, BCORRP0, BCORRP1, SMOOTH1, WINMA, LEVELHYSTE, SEGM1, BLCU, BLCV, SDS, SDB, SCS, SCB, DCFACTOR, DCITER, MORPHOB, MORPHOS, DCALGO, DCDISTANCE, SEGM2, WINDOW, WINSEGM2, LEVELSEGM2, APPROX, WINMA2, ITERM2, DEGRE, WINLIN.

see also : SMOOTH, MEDIAN, WINDOW, BLCW, BLCITER, BLOCBASE, BCORRP, BCORRP?, BCORRP0, BCORRP1, SMOOTH1, WINMA, LEVELHYSTE, SEGM1, BLCU, BLCV, SDS, SDB, SCS, SCB, DCFACTOR, DCITER, MORPHOB, MORPHOS, DCALGO, DCDISTANCE, SEGM2, WINDOW, WINSEGM2, LEVELSEGM2, APPROX, WINMA2, ITERM2, DEGRE, WINLIN, point

BCORRP

a step by step command to choose a set up for the BCORR 3 module

see also : BCORR

BCORRP0

Restores the initial configuration with a polynomial approximation for the BCORR 3 module

see also : BCORR, BCORRP, BCORRP1

BCORRP1

Enables a configuration with a dynamic clusters segmentation and a moving average approximation for the BCORR 3 module

see also : BCORR, BCORRP, BCORRP0

BCORRP?

lists the current configuration for the BCORR 3 module.

see also : BCORR BCORRP0 BCORRP1

BETA

beta b

This context is the angle along the OY axis by which the cube is rotated during a 3D display with the DISP3D/REF3D set of commands

see also : ALPHA GAMA DISP3D

BLCITER

to be used with BCORR 3

The maximum number of iterations for the whole process

see also : BCORR

BLCU

blcu real_value

internal value used by the BCORR 3 algorithm

see also : BCORR

BLCV

blcv real_value
internal value used by the BCORR 3 algorithm

see also : BCORR

BLCW

To be used with BCORR 3
When the RMS of the correction is lower than BLCW times the RMS of the data, the correction is finished

see also : BCORR

BLOCBASE

to be used with BCORR 3
All parameters in point unit are scaled in the ratio of the data set size to this parameter.

see also : BCORR

BURG

BURG size
Will perform an in-place forward linear prediction, by using the Burg method. The size of the final data-set is size. This command is equivalent to the sequence :
DT->AR AR->DT 1 size
but is computed on-place, and can thus be used on a full 2D or 3D spectrum.

see also : AR->DT DT->AR ORDER AR->SP burg_rev svd_rev burg2d burg3d

BUTTONBOX

BUTTONBOX menu_name [button_name gifa_command | *separator*] (n times) *
Permits to construct a new menu in the menu_bar (will create one if none is available). This menu will appear as *menu_name* in the menu_bar. When *menu_name* is clicked by the user, a button box will appear, with as many entry as item in the list of commands. Each entry in the list is defined as a

couple of word, the first one being the name of the button in the box, the second one being the gifa command to be executed if the button is clicked. The command will be executed exactly as if the user had typed it as the prompt level, thus the command can be a simple command, several commands on a line, of a macro name. However the WHILE, FOR, IF .. THEN, GOTO commands are not available; the IF .. any_command syntaxe is available. If a parameter is missing for the command, the user will be prompted in the text window.
The special entry *separator* is not associated to a command, but permits to enter a separator in the menu box.
The list is terminated with a star.

see also : CLOSEBUTTON FORMBOX DIALOGBOX MACRO

CALIBDI

CALIBDI distance_ref rate_ref
Sets the reference distance to a reference relaxation rate. Used when computing interspin distances from nOe build-up measurements.

see also : RELAXRATE METH

CCOLOR

ccolor n
If n = 1 color plots (command CDISP2D) are displayed in color,
if n = 0 color plots are displayed in B&W mode

see also : COLOR SCOLOR CDISP2D

CD

CD new_directory_path
Changes the current working directory.

see also : SH ls pwd more rm vi vim vip

CDISP2D

CDISP2D 0/1
When CDISP2D is 1 the window displaying the contour plot of the 2D buffer is activated on the graphic screen in 2D.

see also : DISP1D DISP2D DISP3D STPL COLOR BCOLOR CCOLOR SCALE LEVEL LOGA CLEAR SHOWC dispcont disp? super2d

CHECK3D

This command enters a interactive mode in which the user can choose the different parameters used for a 3D display. Only the cube is shown for speed convinience. DISP3D must have been previously set to 1.

see also : REF3D, DISP3D

CHEMS

Obsolete command, has been superseded by mkdbppm
This command is an help for constructing chemical shift files to be used with the SIMUNOE command.

see also : SimSpect2D SIMUNOE

CHSIZE

CHSIZE SI1 { SI2 { SI3 } }
Change size of data, zero-fill or truncate.
DO NOT change the value of OFFSET and SPECW, so EXTRACT should always be preferred on spectra (unless you know exactly what your are doing).

see also : EXTRACT MODIFYSIZE lshift rshift size

CLEAR

CLEAR 0/1
When set to 0, the 1D spectrum displayed by the current command, is put on the top of the currently displayed spectrum. Permits to add many spectrum images on the same display. When set to 1, the displat reacts in the standard fashion.

see also : COLOR, REF, UNREF

CLOSE

CLOSE file_name
Closes the file that was previously OPENed.

see also : OPEN FPRINT CONNECT

CLOSEBUTTON

This command closes the current menu bar, and all the associated opened menu box. It is equivalent to closing it with the graphical close box.

see also : BUTTONBOX

COL

Extracts a 1D column (along dim 1) from the active 2D data-set, and displays it in the 1D window. The row will be available as a 1D data set when going from 2D to 1D

see also : ROW PLANE DIAG VERT DIM rowint colint

COLOR

COLOR index window
Defines the current color used for display. The chosen color remains until the next
COLOR command.
index is : 1:white 2:red 3:yellow 4:green
5:cian 6:blue 7:purple 8:black
Window is : 1:1D 2:disp 3:contour 4:3D

see also : DISP1D, DISP2D, CDISP2D, DISP3D, BCOLOR, CCOLOR, SCOLOR

CONCAT

CONCAT file_name
Appends the content of file_name to the current data-set in memory. Work with 'H' files

see also : ADDH READH

CONFIG

Output to the terminal the actual configuration of the system dependent parameters :

-Licence being used
-Version number
-Sizes of the different static buffers
-Operating System (VMS or UNIX)
-Graphic manager (X_Windows or VWS)
-Plotter driver (HP-GL or Postscript)

see also : PLOTTER VARIABLES MEMORY size list

CONJG

Reset to default value for conjugate gradient algorithm for Maximum Entropy run.

see also : GIFA ALGO STEPMAX LAMBDA LAMBCONT

CONNECT

CONNECT file_name
The journaling of all the actions, performed by GIFA is directed to the file_name. Journaling is resumed to the standard gifa.log files with DISCONNECT.

see also : DISCONNECT LOG OPEN CLOSE

CONTEXTS

List of the internal GIFA variables (contexts):

$_ value of the next parameter present on the calling line. CANNOT BE USED WITHIN EVALUATED EXPRESSIONS.
if the following command is used :
@test 3 test.001
within the file test, $_ will be 3 the first time and test.001 the second time.
If no value is present on the calling line, the user will be prompted for the value
$ABSMAX current value of ABSMAX
$ARG True (1) if arguments are present on the calling line (within macro only)
$BUTTON 1,2 or 3 depending on which mouse button was last clicked
$CALIBDI[1] the current calibrating distance, as defined with the CALIBDI command
$CALIBDI[2] the current calibrating relaxation rate, as defined with the CALIBDI command
$CDISP2D current value of command CDISP2D
$CHI2 value of the chi2 returned by the last command : GO, LINEFIT, === RT->PK

$COL	index of the last extracted column in 2D
$COLOR	current value for COLOR
$CONFIG_OS	The system type, as returned by the CONFIG command
$CONFIG_PLOT	The plot driver, as returned by the CONFIG command
$CONFIG_WIN	The window manager, as returned by the CONFIG command
$CX	current value for CX
$CY	current value for CY
$CZ	current value for CZ
$C_ABSMAX	value of ABSMAX of the currently JOINed dataset
$C_DIM	value of DIM of the currently JOINed dataset
$C_FREQ	value of FREQ of the currently JOINed dataset
$C_FREQ1	value of FREQ_1D of the currently JOINed dataset
$C_FREQ2	value of FREQ_2D of the currently JOINed dataset
$C_FREQ3	value of FREQ_3D of the currently JOINed dataset
$C_JOINED	is if there is a currently JOINed dataset, 0 otherwise
$C_OFFSF1	value of OFFSET_1 of the currently JOINed dataset
$C_OFFSF2	value of OFFSET_2 of the currently JOINed dataset
$C_OFFSF3	value of OFFSET_3 of the currently JOINed dataset
$C_SIZEF1	value of SI1 of the currently JOINed dataset
$C_SIZEF2	value of SI2 of the currently JOINed dataset
$C_SIZEF3	value of SI3 of the currently JOINed dataset
$C_SPECWF1	value of SPECW_1 of the currently JOINed dataset
$C_SPECWF2	value of SPECW_2 of the currently JOINed dataset
$C_SPECWF3	value of SPECW_3 of the currently JOINed dataset
$C_TYPE	value of ITYPE of the currently JOINed dataset
$DIM	current value of DIM
$DISP1D	current value of command DISP1D
$DISP2D	current value of command DISP2D
$DISP3D	current value of command DISP3D
$DIST	the result of the last DIST command
$FREQ	main frequency (1H) of the spectrometer
$FREQ_1D	frequency in 1D
$FREQ_1_2D	frequency in 2D in F1
$FREQ_1_2D	frequency in 3D in F1
$FREQ_2_2D	frequency in 2D in F2
$FREQ_2_2D	frequency in 3D in F2
$FREQ_3_2D	frequency in 3D in F3
$GB1	value of gb in F1
$GB2	value of gb in F2

$GB3	value of gb in F3
$ITYPE_1D	value of itype in 1D
$ITYPE_2D	value of itype in 2D
$ITYPE_3D	value of itype in 1D
$LB1	value of lb in F1
$LB2	value of lb in F2
$LB3	value of lb in F3
$LEVEL	current value of command LEVEL
$LICENCE	The licence as returned by the CONFIG command
$LOGA	current value of command LOGA
$MAX[1]	value of last computed max computed with the command MAX
$MAX[2]	value of last computed min computed with the command MAX
$MEM_MAX	The larger data set available, as returned by the CONFIG command
$MEM_PRO_1D	The size of the protected 1D area, as returned by the CONFIG command
$MEM_PRO_2D	The size of the protected 2D area, as returned by the CONFIG command
$MEM_PRO_3D	The size of the protected 3D area, as returned by the CONFIG command
$NAME	name of the current data-set
$NAR	number of A.R. coefficients as listed by ARLIST
$NOISE	value of the noise, as given by the NOISE command
$NPK1D	The number of entries in the 1D peak table
$NPK2D	The number of entries in the 2D peak table
$NPK3D	The number of entries in the 3D peak table
$NPOINT	The number of entries in the POINTX,Y buffer
$NRT	Number of root as listed by RTLIST
$NSVD	Number of root as listed by SVDLIST
$OFFSET_1D	spectral offset in 1D
$OFFSET_1_2D	spectral offset in 2D in F1
$OFFSET_1_3D	spectral offset in 3D in F1
$OFFSET_2_2D	spectral offset in 2D in F2
$OFFSET_2_3D	spectral offset in 3D in F2
$OFFSET_3_3D	spectral offset in 3D in F3
$ORDER	current value for ORDER
$PH0	0th order of the last phase correction
$PH1	1st order of the last phase correction
$PK1D_A[i]	amplitude of the ith entry in the 1D peak table
$PK1D_F[i]	frequency of the ith entry in the 1D peak table
$PK1D_P[i]	phase of the ith entry in the 1D peak table
$PK1D_W[i]	width of the ith entry in the 1D peak table
$PK2D_A[i]	amplitude of the ith entry in the 2D peak table

$PK2D_F1F[i] F1 frequency of the ith entry in the 2D peak table
$PK2D_F1W[i] F1 width of the ith entry in the 2D peak table
$PK2D_F2F[i] F2 frequency of the ith entry in the 2D peak table
$PK2D_F2W[i] F2 width of the ith entry in the 2D peak table
$PK3D_A[i] amplitude of the ith entry in the 3D peak table
$PK3D_F1F[i] F1 frequency of the ith entry in the 3D peak table
$PK3D_F1W[i] F1 width of the ith entry in the 3D peak table
$PK3D_F2F[i] F2 frequency of the ith entry in the 3D peak table
$PK3D_F2W[i] F2 width of the ith entry in the 3D peak table
$PK3D_F3F[i] F3 frequency of the ith entry in the 3D peak table
$PK3D_F3W[i] F3 width of the ith entry in the 3D peak table
$PLANE[1] axis of the last extracted plane in 3D
$PLANE[2] index of the last extracted plane in 3D
$PLOTAXIS[1] current value for the unit used in PLOTAXIS
$PLOTAXIS[2] current value for the tick distance in x axis
$PLOTAXIS[3] current value for the tick distance in y axis
$PLOTOFFSET[1] current value for plot offset on X axis
$PLOTOFFSET[2] current value for plot offset on Y axis
$POINTX[i] X coordinates of ith clicked point (with POINT or MULTIPOINT or MONOPOINT
$POINTY[i] Y coordinates of ith clicked point (with POINT or MULTIPOINT or MONOPOINT
coordinates are in index unit, absolute on the spectrum.
$POINTY[1] after a MONOPOINT in 1D, ranges between 0 and 1 depending of the position in the 1D window.
$RANDOM a random variable in the range 0..1 with equiprobable distribution
$RANDOMG a random variable with normal law, unit variance and zero mean
$RANDOMZ same as $RANDOM but resets the random series
$RELAX the result of the last RELAXRATE or SLOPE commands
$RCRYST the result of the last RCRYST command CALIBDI
$ROW index of the last extracted row in 2D
$SCALE current value of the context SCALE
$SCOLOR current value of the context SCOLOR
$SHIFT value of the offset, as given by the SHIFT command
$SI1_1D size of the 1D buffer.
$SI1_2D size in F1 of the 2D buffer
$SI1_3D size in F1 of the 3D buffer
$SI2_2D size in F2 of the 2D buffer
$SI2_3D size in F2 of the 3D buffer
$SI3_3D size in F3 of the 3D buffer
$SIGN current value of command SIGN

$SOMREC	value returned by the last SOMREC command
$SPECW_1D	spectral width in 1D
$SPECW_1_2D	spectral width in 2D in F1
$SPECW_1_2D	spectral width in 3D in F1
$SPECW_2_2D	spectral width in 2D in F2
$SPECW_2_3D	spectral width in 3D in F2
$SPECW_3_3D	spectral width in 3D in F3
$UNIT	current value for UNIT
$VAR_MAX	The total number of user variable available, as returned by the CONFIG command
$VERSION	The current version, as returned by the CONFIG command
$VHEIGHT	current value of command VHEIGHT
$ZOOM	1 if in ZOOM mode
$ZOOM_1D[1]	left coordinate of the 1D zoom window
$ZOOM_1D[2]	right coordinate of the 1D zoom window
$ZOOM_2D[1]	lower F1 coordinate of the 2D zoom window
$ZOOM_2D[2]	left F2 coordinate of the 2D zoom window
$ZOOM_2D[3]	upper F1 coordinate of the 2D zoom window
$ZOOM_2D[4]	right F2 coordinate of the 2D zoom window
$ZOOM_3D[1]	"left" F1 coordinate of the 3D zoom window
$ZOOM_3D[2]	"right" F1 coordinate of the 3D zoom window
$ZOOM_3D[3]	"left" F2 coordinate of the 3D zoom window
$ZOOM_3D[4]	"right" F2 coordinate of the 3D zoom window
$ZOOM_3D[5]	"left" F3 coordinate of the 3D zoom window
$ZOOM_3D[6]	"right" F3 coordinate of the 3D zoom window

see also : VARIABLES MACRO FUNCTIONS

CONTROLS

Control structures are available in the macros file :
GOTO - one line IF - multiple line IF - FOR loop - WHILE loop
macros can be called from within macros, up to 9 levels of nesting.
No recursivity is allowed.

see also : MACRO GOTO IF FOR WHILE VARIABLES CONTEXTS FUNCTIONS ERROR SETPATH SETPROMPT

CX

CX x

Size in centimeters in the x-axis used for the graphic windows and for the plot. Used by PLOT and STPL

see also : CY CZ PLOT STPL

CY

CY y

Size in centimeters of the y-axis used for the graphic windows and for the plot. Used by PLOT and STPL.

see also : CX CZ PLOT STPL

CZ

CZ z

Size in centimeters of the z-axis used for 3D display.

see also : PLOT3D DISP3D CY CX

DBCLOSE

DBCLOSE array_name

Closes the db file previously associated to the array *array_name* by a previous DBOPEN command.

see also : DBOPEN

DBOPEN

DBOPEN file_name array_name

associates the array *array_name* with the db file *file_name*. Each operation on the array will be propagated to the file : access, change, delete, exist(), nextlm() etc... Any db file can be accessed.

see also : DBCLOSE VARIABLES FUNCTIONS

DCALGO

DCALGO n

Selects the which kind of data will be used for the first segmentation step of BCORR 3 when Dynamic clustering is used.
+1 : the algorithm runs on the data set.
+10 : the algorithm runs on the first derivative of the data set.
+100 : the algorithm runs on the second derivative of the data set.

see also : BCORR

DCDISTANCE

Selects the norm that will be used for the first segmentation step of BCORR 3 when Dynamic clustering is used.
0 : Norm 1.
1 : Euclidian distance.

see also : BCORR

DCFACTOR

Selects the scaling that will be used for the first segmentation step of BCORR 3 when Dynamic clustering is used.

see also : BCORR

DCITER

Selects the maximum number of iterations that will be used for the first segmentation step of BCORR 3 when Dynamic clustering is used.

see also : BCORR

DEBUG

When this context is set to 1 some command generate some debuging information. Used mainly by developers.

see also : VERBOSE REFMACRO TIMER

DIAG

DIAG {direc}
In 2D, extracts the diagonal of the 2D and put into the 1D buffer
In 3D, extracts one diagonal plane of the 3D cube, chosen with the direc parameter
and put it into the 2D buffer

see also : DIM ROW COL PLANE VERT

DIAGONALIZE

Diagonalize the current 2D buffer (must be square and real), puts the eigenvalues
into the 1D buffer, sorted in decreasing order; and puts the eigenvector matrix
into the 2D buffer

see also : BACKPROD

DIALOGBOX

DIALOGBOX dialog_name ...
... [field_name field_type [field_list [enum_list] field_var default_value] | *separator*] (n times) *
Permits to construct a dialog box with editable fields. The box will appear in a window called *dialog_name* . The Gifa program will be blocked as long as the user has not closed the window. It will have two buttons at the button :
Ok closes the form and resume the execution of Gifa
Cancel closes the form, abort the command and generate an error

The dialog is also composed en fields which appear one on each line in the window. A dialog can hold as many field as needed. Each field has :
* a name *field_name* that will show on the left side of the field
* a type *field_type* which is chosen in the following list : (message, string, int, real, file, enum)
message type of field have no other field, this kind of entries serves to put plain text in the dialog
* All other type have a *field_var* entry which determines the name of the Gifa vairable in which the user input will be stored, and the *default_value* entry which determines the value of that variable when creating the form.
string int real and **file** types of field have no other entry. They appear in the window as editable string. When the user hits return there, the content of the string is copied to the Gifa variable *field_var*.
file entries will have a small arrow which permits to open a standard file dialog window.
enum have an additional entry : *enum_list*.. This kind of field will appear as a pop up menu permitting to choose between the entries defined in the list : *enum_list.* This list is a single blank separated word, each item being separated with commas.

The special entry *separator* is not associated to a variable, but permits to enter a separator in the dialog.

The list is terminated with a star.

see also : FORMBOX BUTTONBOX CLOSEBUTTON MACRO

DIM

DIM n (n=1 or 2)

DIM is a context which equals 1 if you are processing one dimensional NMR spectra, 2 if you are processing two dimensional spectra and 3 if you are processing three dimensional spectra.

Since 1D 2D and 3D data-sets are kept in different buffers, you can switch between this 3 modes freely, keeping the data-sets protected.

see also : COL ROW PLANE DIAG VERT size

DISCONNECT

Journaling is resumed to the standard gifa.log files when previously redirected with the CONNECT command.

see also : CONNECT LOG OPEN CLOSE

DISJOIN

disconnect the previously connected file with the JOIN command

see also : JOIN LISTFILEC dataset

DISP1D

DISP1D 0/1

When DISP1D is set to one, the window displaying the content of the 1D buffer is opened. The window is closed if DISP1D is 0.

see also : DISP2D CDISP2D DISP3D STPL COLOR BCOLOR CCOLOR SCALE CLEAR REF UNREF SHOWC disp? super1d

DISP2D

DISP2D 0/1

When DISP2D is set to one, the window displaying the content of the 2D buffer is opened. The window is closed if DISP2D is 0.

see also : DISP1D CDISP2D DISP3D STPL COLOR BCOLOR CCOLOR SCALE LEVEL LOGA CLEAR REF UNREF disp?

DISP3D

DISP3D n

When DISP3D is set to 1, a graphic window is opened which will be used by the REF3D command to display 3D vues of the current3D data-set. The size of the window is set to max(cx,cy,cz) in each dimension.

see also : DISP1D DISP2D CDISP2D REF3D CHECK3D DISP3D? PLOT3D ALPHA BETA GAMA ZNOT CX CY CZ OFFSET3D AXIS3D SCALE3D disp3d? disp3d_form

DIST

This command will issue the distance value as computed from the reference distance, set by CALIBDI, and the current relaxation rate, set by RELAXRATE

see also : CALIBDI RELAXRATE METH RCRYST

DSA

DSA shift factor.

Adds to the current data set the same data set multiplied by factor and shifted left by shift points. Useful for suppression of water signal at zero frequency.

DT->AR

Calculates the autoregressive coefficients directly from the data using the Burg method.

see also : AR->DT BURG ARLIST ORDER

DT->SVD

DT->SVD size

Calculates the Toeplitz rectangular matrix from the data and perform the singular value decomposition. size is the size of data used in the formation of Toeplitz matrix. This is the first step of the LP-SVD spectral analysis method. The same singular value decomposition can be used to perform forward and backward analysis.

see also : SVD->DT SVDLIST SVDCLEAN1 SVD->AR

DUMP

Lists the values of all the variables presently defined within GIFA.
Each value is shown with its associated context : context 20 is the global context, higher context are associated to higher call levels.

see also : SET

ELSE

part of the IF command
see also : IF

ELSIF

part of the IF command
see also : IF

EM

EM lb {lb' {lb"}}
Multiply data by an exponential, and set LB. In 2D, and 3D a parameter is used for each direction.

see also : LB GM GB SIN SQSIN TM JMULT FILTER

ENDFOR

part of the FOR command
see also : FOR

ENDIF

part of the IF command
see also : IF

ENDWHILE

part of the while command

see also : WHILE

ERROR

ERROR string

Generates an error, and issue the string to the user. Permits to handle errors in command files.

see also ALERT PRINT

ESCALE

The Entropy expression during Maximum Entropy run is computed as follow :

A = Escale * Sum(F(i))

P(i) = F(i)/A

S = -Sum(log(P(i)) * P(i))

Escale should be set to 1.0 for normal operation

see also : CONJG GIFA MAXENT

EVALN

EVALN x1 y1 {x2 y2}

evaluates the noise level as well as the overall offset of the data, over a area of the data. The results are stored in the NOISE and SHIFT contexts This command is called automatically whenever a data set is read. The command will prompt for the last selected region with the POINT command

see also SHIFT NOISE

EXIT

EXIT yes/no

Exit this program. You can choose to keep or to remove the log file

QUIT and BYE are also equivalent. As well as ^Z (F10) on VMS systems.

EXPLIM

This context holds the limit for the exponential function used in the Gull & Daniel algorithm, as well as in the GIFA algorithm.

A function exp2(x) is used which is

- equal to exp(x) if x <= explim
- equal to exp(explim)*(x+1.0-explim) if x > explim

i.e. a linear extension of the exponential function is used fo x > explim.
see also MAXENT

EXTRACT

EXTRACT x1 y1 {x2 y2}

Permits to extract a portion of the data currently displayed. Data can then be processed as a regular data-set. EXTRACT changes the value of OFFSET and SPECW accordingly. The command will prompt for the last selected region with the POINT command

see also POINT ZOOM CHSIZE lshift rshift

FILTER

Determines the filter used during Maximum Entropy processing, Filter can take values: 0, 1, or 2. Default value is 1.
0: is no filtering at all
1: is filtering done only if one of the following is non 0 : LB GB or JCONS
The filter is constructed automatically from the value of those parameters
2: is filtering always, the filter used is the one currently loaded (can be checked with SHOW FILTER , and changed with PUT FILTER)

see also SHOW GET PUT WINDOW NCHANNEL

FLIP

Switch from a itype=1 2D data-set to a itype=2 by exchanging the imaginary parts. Useful only for processing phase modulated 2D data-sets :

```
Gifa> FT F2 FLIP FT F1 REAL F1
```

will process a phase modulated data-set.

see also : FLOP FT MODULUS ITYPE ft_phase_modu

FLOP

Performs the inverse of FLIP.
See FLIP.

see also : FLIP FT MODULUS ITYPE ft_phase_modu

FLUSH

Flushes the modified blocks of the cache memory system associated to the currently connected file onto the disk.

see also : JOIN LISTFILEC dataset FLUSHCACHE

FLUSHCACHE

Flushes all the modified of the cache memory system onto the associated files.

see also : FLUSH JOIN

FOR

FOR var = n TO p { STEP k }
... GIFA commands
ENDFOR

Control structure permitting to loop over a set of commands. The variable var will be created if does not yet exist. By default of STEP is 1, but any value can be used (including negative values in which case n should be larger than p). n, p and k should always be integer numbers.

FOR and ENDFOR should appear alone on one line, eventually followed by a comment.

see also : IF WHILE GOTO CONTROLS

FORGET

FORGET plot_file_name
Permits to remove a plot file from the internal list of opened plots without issuing the PAGE command which also sends the plot to the plotter. However the plotter code for the ejection of the paper sheet is not inserted at the end of the plot_file.

see also : PLOT PAGE

FORMBOX

FORMBOX form_name gifa_command ...

... [field_name field_type [field_list [enum_list] field_var default_value] | *separator*] (n times) *

Permits to construct a form (dialog box with editable fields). The box will appear in a window called *form_name* . The program continues after having created the form, whitch will stay on screen as long as the user dos not close it exolicitely. It will have three buttons at the button :

Apply executes the command *gifa_command* and the form will remain on screen

Ok executes the command *gifa_command* and closes the form

Cancel does not execute any command and closes the form

The command will be executed exactly as if the user had typed it as the prompt level, thus the command can be a simple command, several commands on a line, of a macro name. However the WHILE, FOR, IF .. THEN, GOTO commands are not available; the IF .. any_command syntaxe is available. If a parameter is missing for the command, the user will be prompted in the text window.

The form is also composed en fields which appear one on each line in the window. A form can hold as many field as needed. Each field has :

* a name *field_name* that will show on the left side of the field

* a type *field_type* which is chosen in the following list : (message, string, int, real, file, enum)

message type of field have no other field, this kind of entries serves to put plain text in the form

* All other type have a *field_var* entry which determines the name of the Gifa vairable in which the user input will be stored, and the *default_value* entry which determines the value of that variable when creating the form.

string int real and **file** types of field have no other entry. They appear in the window as editable string. When the user hits return there, the content of the string is copied to the Gifa variable *field_var*.

file entries will have a small arrow which permits to open a standard file dialog window.

enum have an additional entry : *enum_list*.. This kind of field will appear as a pop up menu permitting to choose between the entries defined in the list : *enum_list.* This list is a single blank separated word, each item being separated with commas.

The special entry *separator* is not associated to a variable, but permits to enter a separator in the form.

The variables used by the editable entries can be global variables, in which case they should have been created as global before calling the FORMBOX command. If this is not the case, they are created in a special context, available only to the command *gifa_command* when executed with the **Apply** or **Ok** buttons. This special context is completely local to the form. This permits to have several forms, using the same variables syntax, without interaction.

The list is terminated with a star.

see also : DIALOGBOX BUTTONBOX CLOSEBUTTON MACRO

FPRINT

FPRINT file_name string

Outputs the string to the file. file must have previously be OPENed (see OPEN).

see also : OPEN CLOSE PRINT fprintf

FREEZE

freeze { p }

FREEZE will freeze the specified graphic window in its current state, and will open a new window which will become the active window. There is no way to kill the frozen window, but to exit GIFA. Useful for comparing many graphics.

Depending on which display mode are active, no parameters may be needed, or the user may have to choose which window to freeze : D (density) C (contour) 3 (3D)

see also : DISP1D, DISP2D, DISP3D, CDISP2D

FREQ

FREQ freq_H1 freq1 { freq2 { freq3 } }

The context FREQ holds the basic frequency of the spectrometer (in MHz). freq_H1 is meant to be the basic frequency of the spectrometer (1H freq) and is not used in the program. freq2 (and freq1 in 2D) are the freq associated to each dimension (different if in heteronuclear mode). Values are in MHz.

see also : SPECW OFFSET calib

FREQLIS

FREQLIST center excursion n

Will list the index in the peak table which hold entries with frequency values in a given range of frequencies (centered on *center* with radius *excursion*). The thus selected entries will be then selected with the RTSELECT command.

see also : RTSELECT RTLIST

FT

FT { axis }

Performs in-place complex Fourier Transform on the current data-set; Data-set must be Complex.

All FT commands work in 1D, 2D or 3D.

Here is a complete overview of FT routines : C stands for Complex, R stands for Real

FIDs		Spectra
C	---FT--->	C
C	<--IFT---	C
R	--RFT-->	C
R	<--IRFT--	C
C	-FTBIS->	R
C	<-IFTBIS-	R
R	Does not exist	R

see also : FLIP FLOP MODULUS REAL ft_seq ft_sim ft_sh ft_tppi ft_sh_tppi ft_phase_modu ft_n+p corect1st easy2d easy3d proc2d proc3d

FTBIS

FTBIS { axis }

Performs complex-to-real Fourier Transform on data

see also : FT

FUNCTIONS

the following functions and operators are available within evaluated expression :

Operators :

o the regular 4 operations, modulo and power

+ - / * % ^

o arithmetic comparison

== != < > <= >=

o string comparison

s= s!

o logical operators

| *(or)* & *(and)* ! *(not)*

o string concatenation

// *(direct concatenation)* ; *(concatenation with a blank)*

functions :

o the mathematical functions

sqrt(x) cos(x) sin(x) atan(x) log(x) exp(x) abs(x) int(x) power2(n)
max(x,y) min(x,y)

o the alphanumeric functions :

toupper(st) tolower(st) *(transformations)*

sp(i) *(makes blank strings)* len(st) *(length of the string)*

index(st1,st2) (locates st2 in st1) subst(st,i,j) *(extracts (i:j) from st)*

head(st) *(extract the 1st word)* tail(st) *(everything but the 1st word)*

headx(st,char) tailx(st,char) *(same as head & tail but char is the separator)*

o file access

eof(file) *(true if End of File file is reached)*

o GIFA internal parameters

va1d(i) val2d(i,j) val3d(i,j,k) *(values in the 1D, 2D and 3D buffers)*

itoh(index,dim,axis) htoi(hertz,dim,axis)itop(index,dim,axis) ptoi(ppm,dim,axis)

htop(hertz,dim,axis) ptoh(ppm,dim,axis) *(unit conversions)*

nextlm(array,index) *(next entry in associative or dbm array)*

exist(var_name) *(true if var_name exists)*

dbm(var_name) *(true if var_name is a dbm array)*

see also : VARIABLES CONTEXTS CONTROLS MACRO

GAMA

gama g

This context is the angle along the OX axis by which the cube is rotated during a 3D display with the DISP3D/REF3D set of commands

see also : DISP3D ALPHA BETA

GB

Constant used for Gaussian deconvolution during MaxEnt. Set also by the GM command

see also : GM LB EM FILTER

GET

GET buffer_name

Moves the content of another buffer, back to the current buffer with buffer_name equal to:

-DATA: get the content of the Data buffer

-LINEFIT: get the simulated spectrum obtained form the current peak table

-WINDOW: get actual window used to compute the chisquare

-FILTER: get filter used for Deconvolution
-LAMB: get evolution of Lambda during MaxEnt iteration
-ENT: get evolution of Entropy during MaxEnt iteration
-CHI: get evolution of ChiSquare during MaxEnt iteration
-SUM: get evolution of Sum of point of Image during MaxEnt iteration
-CONV: get evolution of Convergence during MaxEnt iteration
-STEP: get evolution of Step during MaxEnt iteration
-RESIDUE get residue of the spectrum after a MaxEnt run
-AMIBE: get the Amoeba map used during the last integration (in 2D only)

see also : PUT SHOW APPLY

GETC

syntaxe depends on the dimentionnality of the currently joined data-set (from) and on the current context dim (to) :
1D to 1D : GETC left right
2D to 1D : GETC axis index left right
3D to 1D : GETC axis index1 index2 left right
2D to 2D : GETC left1 left2 right1 right2
3D to 2D : GETC axis index left1 left2 right1 right2
3D to 3D : GETC left1 left2 left3 right1 right2 right3

Loads from the currently connected file, the area of data as described.

see also : PUTC JOIN dataset LISTFILEC

GETHEADER

GETHEADER param_name value
Display the value of a parameter associated to the currently JOINed data-set. Any parameters can be used. Gifa already uses the following parameters (as listed with the dataset command) :
Frequency, Freq1, Freq2, Freq3, Specw1, Specw2, Specw3, Offset1, Offset2, Offset3, Type, Dim.
But paameters created with PUTHEADER can also be used.
The value of the last read parameter is set into the Gifa $c_header variable.

see also : JOIN NEWFILEC dataset PUTHEADER

GM

GM gb1 { gb2 { gb3 } }
Multiply data by an Gaussian, and set GB.

see also : GB EM LB SIN SQSIN TM JMULT FILTER

GOTO

GOTO label
Will branch the flow a macro to the label. Labels are alphanumeric strings, and appear preceded by an = sign in macro text. Labels should be alone on the line. GOTO cannot be used in interactive mode, but only in macros.

see also : IF CONTROLS

GRID

GRID direc plot_file
This command draws a set of horizontal (if direc=F2 or F12) and vertical (if direc=F1 or F12) lines. A CX by CY space is spanned. The grid lines are plotted as defined by the plotaxis? command,. The plot is sent to plot_file.

see also PLOT PLOTAXIS PLOTAXIS?

HELP

This is the GIFA Version 4.0 program.

Type help Name_of_item to get info about a specific command. H can be used instead of HELP.
Use apropos topic, to find commands name related to a specific topic.
Use help primer for a first introduction.

see also : apropos

HPHASE

HPHASE x y {dim}
Equivalent to PHASE, but the phase rotation is computed for real spectra, using the properties of the Hilbert transform. HPHASE x y is equivalent to the sequence : IFTBIS CHSIZE *2 FT PHASE x y REAL

see also : PHASE FT REAL

IF

Used for tests in command.
2 syntaxes are available :

IF test THEN
..commands on several lines
{ ELSIF test2 THEN
..commands } (eventually many exclusive tests)
{ ELSE (default case)
..commands }
ENDIF
The different commands will executed conditionnally on the value of the tests. A non-zero value is considered as true.
Permits to construct complex tests in command files.

IF test remaining_of_the_line
execute the remaining_of_the_line conditionally on the value of test. It will be executed only if test is true (non zero). The remaining of the line can span several lines by using the line continuation sign : \
This form can be used at the prompt level, as well as in call-backs, for instance in graphic buttons. It is called the one-line IF.

Tests available are :
for numeral : == != < > <= >=
for strings : s= s!
for combinations: & | !
as well as several tests functions and variables : exist() eof() $arg $c_joined, etc...

IF(in the first syntaxe) ELSIF ELSE and ENDIF should appear alone on one line, eventually followed by a comment. There is no limitation for the one-line IF.

see also FUNCTIONS GOTO FOR WHILE CONTROLS

IFT

Perform complex inverse Fourier Transform on data
see also : FT

IFTBIS

Perform inverse of complex-to-real Fourier Transform on data
see also : FT

INCREM

Constant used to increment lambda during MaxEnt iteration(0.1 .. 1)

see also : LAMBDA LAMBCONT MAXENT

INITINPROGRESS

INITINPROGRESS n
Presets for n iterations, the progress bar, of the form :
`In Progress : 0%....25%....50%....75%....100%`
The progress bar is then updated with the INPROGRESS command.

see also : INPROGRESS PRINT

INPROGRESS

INPROGRESS i
Displays the progress of the operation in the progress bar, inited with the INITINPROGRESS command.

see also : INITINPROGRESS PRINT

INT1D

A very crude 1D graphic integrator. The data is replaced by the running sum of the previous data. Better integrals will be obtain with base-line corrected spectra. The curve can then be optimized with BCORR, ADDBASE, etc...

see also : INTEG

INTEG

INTEG factor slope thres { radius }

INTEG computes the volume of the peak detected by PEAK. You need to use the peak-picker PEAK before to use INTEG. INTEG uses the methods described for the PARIS method. It first evaluate the extension of the current peak, using 3 criteria :

- factor : the extension stop whenever the level goes below inten/factor (where inten is the intensity of the peak)
- slope : the extension stop whenever the the slope get larger than slope/point (0 means whenever it goes up)
- thres the extension stop whenever the level goes below thres.

In 2D, an additional parameter is the maximum extension radius for each peak, and the extensions are stored in an amoeba file.

INTEG uses the baseline and noise information held in SHIFT and NOISE, which are computed automatically by EVALN

see also : MSKINTEG SIGN_PEAK ZERO_QU PEAK SHIFT NOISE INT1D SOMREC mskread mskwrite

INVF

INVF {Fx}

Process data-sets by multiplying by -1 1 point every 2 points. Equivalent to taking the conjugated on complex data-sets, or hyperconjugated on hypercomplex data-sets. If applied on a complex FID, inverses the final spectrum obtained after Fourier transform.

see also : REVF ITYPE FT REVERSE

IRFT

Perform inverse real-to-complex Fourier Transform on data

see also : FT

ITER

Number of iterations used by all the iterative modules of GIFA : MaxEnt with MAXENT or MAXENTCONT . But also LINEFIT and AUTOPHASE see also : MINITER

ITERMA2

ITERMA2 value
internal value for BCORR 3 algorithm

see also : BCORR

ITYPE

ITYPE is a context which describes the type of data in the image buffer.

For 1D if Itype is 1 then the data-set is considered as complex (with real and imaginary parts interleaved), if Itype is 0, the data-set is considered as real.

For 2D data-sets, itype takes values 0 (real) 1 (complex in dim 2, real in dim 1), 2 (complex in dim1, real in dim 2) and 3 (complex in both dimensions).

For 3D data-sets, itype takes values 0 (real) 1 (complex in dim 3, real in dim 1 and 2), 2 (complex in dim 2, real in dim 1 and 3), 4 (complex in dim 1 real in dim 2 and 3) and the sums for the combinations.

Itype is normally handled automatically by the program. Changing the value of Itype DOES NOT CHANGE the data, only what the program believes they are. When the itype is wrong, use another command (example FT instead of RFT) or make them real (command REAL).

Results of FT, IFT, RFT, IFTBIS PK->DT, SIMU, SIMUN etc... are complex. Results from IRFT, FTBIS, MODULUS, REAL (!) etc... are real. Image from Maximum Entropy Iteration are real.

Linear Prediction package works only on real FIDS. To make real FID complex,use the sequence RFT IFT.

When displaying complex data sets, only the real part is shown on the screen.

see also : REAL FT MODULUS FLIP FLOP

IVALUE

Constant used to set the initial value of the image (1e-3 .. 1e3) Default value is 1.0

see also MAXENT

JCONS

JCONS J1 {J2) k

Permits to set up the J (J1 and J2) constant for J-deconvolution; the J entered should be half the J of the coupling you want to deconvoluate. k is either A or P if you want Antiphase or Inphase J-deconvolution.

see also JMULT FILTER

JMULT

JMULT J1 {J2) k

Set up the J (J1 and J2) constant for J-deconvolution; the J entered should be half the J of the coupling you want to deconvoluate. k is either A or P if you want Antiphase or Inphase J-deconvolution. The data are actually multiplied by the filter function. Used mostly to check the effect of the filter one the Data.

<u>DO NOT</u> use JMULT before launching a J-deconvolution Maximum Entropy run.

see also : JCONS FILTER EM GM SIN SQSIN TM

JOIN

JOIN filename

Connects the cache memory system to the file `filename' in standard format (created with WRITE or NEWFILEC). The file can then accessed randomly with GETC and PUTC commands

see also : DISJOIN dataset LISTFILEC GETC PUTC NEWFILEC GETHEADER PUTHEADER FLUSH FLUSHCACHE SHOWC dataset proc2d proc3d super1d super2d

LAMBCONT

Tells the Maximum Entropy minimizer which lambda controlling should be used.

If LAMBCONT is 0 no controlling is done and Lambda is kept at its initial value may be used for hand made lambda control.

If LAMBCONT is 1 lambda is computed such as dQ.dS = 0.0 if conv<1 (dC dS angle larger than pi/2)

If LAMBCONT is 2 lambda is computed such as cos(dQ,-dC) = cos(dQ.dS) if lambcont.eq.2

see also : LAMBDA LAMBSP ALGO MAXENT

LAMBDA

Used to set the value for Lambda when no lambda controlling is used.

see also : LAMBCONT LAMBSP ALGO MAXENT

LAMBSP

Constant used to increment lambda (1 .. 20) The larger the value of LAMBSP, the faster is the advancement of the Maximum Entropy run .
A too large value for LAMBSP may hamper the correct convergence of the program, a value too small may stop the convergence. Typical values are within 3..5

see also : LAMBDA LAMBCONT ALGO MAXENT

LB

Constant used for exponential deconvolution. The value is loaded from the last EM command. A non-zero value indicates the program to do Lorentzian deconvolution during a Maximum Entropy run (if FILTER is 1).

see also : EM GM GB FILTER

LEVEL

LEVEL n
number of levels to be plotted by the plot command in 2D and in the contour plot graphic window, and in 3D contour. Default is 8; maximum is 64.

see also : CDISP2D CDISP3D PLOT PLOT3D

LEVELHYSTE

LEVELHYSTE value
internal value for BCORR 3 algorithm

see also : BCORR

LEVELSEGM2

LEVELSEGM2 value

internal value for BCORR 3 algorithm

see also : BCORR

LINEFIT

linefit line_shape
where line_shape is Lorentz / Gauss used to determine the fitted line-shape (Lorentzian or Gaussian).
Starts the line fitter, the parameter of the line will be modified to the best fit of the current spectrum using the Levenberg-Marquardt method. The fit is estimated with a chi squared value, computing on the displayed region, using the value of the noise. Optimum fit is btained for a chi squared equal to 1. This value is return in the $CHI2 variable. The program will fit the currently displayed region, using the values in the peak table as starting values, and returning the fitted values in the same peak table. Uses the values of TOLERANCE and ITER for the run. All the values in the peak-table will be used, but only the line in the displayed region will be adjusted. Unknown line shapes will be assumed to be Lorentzian.

see also : NOISE LINEFITD SHOW GET PK->DT show_fit plot_fit

LINEFITD

linefitd { peak_to_fit line_shape amplitude frequency line_width } *n times* 0

Start the line fitter, fitting only the selected parameters to fit. The command will prompt for the index of the peak to fit, for the parameters to fit for that peak. Several peaks can be fitted at once, the list is finished with a 0.
All questions are answered with yes/no, but *line_shape* which is to be Lorentz/Gauss for Loretzian or Gaussian line-shapes.
In all cases, the command LINEFITD %% fits all the peaks, retaining the current line-shape of each peaks.
All the values in the peak-table will be used, but only the line with the selected parameters will be adjusted. Unknown line shapes will be assumed to be Lorentzian.

see also : LINEFIT SHOW GET PK->DT pkadjust

LISTFILEC

Lists to the user the name of the currently connected file with the JOIN command

see also : JOIN DISJOIN dataset

LOG

The LOG file is a text file journaling all the actions performed during the session in the GIFA program. This file is called gifa.log. When exiting the program, you are asked whether you want to keep this file for latter use.

see also : CONNECT DISCONNECT

LOGA

LOGA n with n>=1

Controls the way the levels are computed in the Contour Plot graphic mode.

If LOGA = 1, then the lowest level is ABSMAX/(SCALE*LEVEL), and all the levels above are equidistant with the same quantity.

If LOGA >1, the lowest level is the same, but each level is computed from the previous one : L(i+1) = L(i)*LOGA; in a exponential manner. For instance, LOGA = 2 corresponds to the standard BRUKER output.

see also : CDISP2D PLOT

MACRO

Macros are text files holding regular GIFA commands. Loops, tests, controls, input/output and variables are available within macro.

Macros are called form the user interface exactly as native commands. Parameters can be passed to macros. Macro are called by typing their name (case sensitive). It is also possible to force for a macro call by prefixing the name with @. Permitting thus to have macros having the same name as native commands.

When called, macro files are first searched in a list of directories, held in an internal path context (The GifaPath). By default the path consists in the current working directory (as given by pwd), then in the user macro directory : $HOME/macro, and finally in the global macro directory : /usr/local/gifa/macro. This PATH can be changed with the SETPATH command, and examined with the $GIFAPATH context.

All the macros in the standard distribution are in the /usr/local/gifa/macro directory, some other macro can be found on the ftp server.

see also : VARIABLES CONTEXTS FUNCTIONS OPEN CONTROLS SETPATH SETPROMPT BUTTONBOX FORMBOX DIALOGBOX

MAX

Finds the maximum and minimum points of the current data-set. Load the values used by MINIMAX.

see also : MAX ABSMAX

MAXDATA

Compare the content of the current buffer with the content of the DATA buffer, and leave in memory the largest of the 2 values.
Usefull for projections or symetrisation macros.

see also : MINDATA ADDDATA MULTDATA

MAXENT

MAXENT SI1 { SI2 }
Start Maximum Entropy iteration. The user is prompted for the size of the reconstructed image.

see also : ALGO MAXENTCONT ITER NDISP gifa conjg gad

MAXENTCONT

Continue Maximum Entropy iteration. All the parameters may be modified before CONTinuing, but the current spectrum as hold in the working buffer should not be modified.

see also : MAXENT ITER

MEDIAN

MEDIAN { y } x index
Executes a median filter on the data-set (1D or 2D).a window of x points (or y by x in 2D) is moved along the data set, the point are ordered, and the indexth point is taken as the new point for the data set.
see also : SMOOTH

MEMORY

The GIFA program holds all the data in one single large memory buffer. This buffer is used for several purposes, and may be divided into smaller pieces. The size of this buffer is displayed when entering the program, or with the CONFIG command. There is no way to use a larger data-set than this memory size but recompiling. This buffer is used for 1D as well as 2D and 3D operations. In all cases the whole buffer is available. However, when moving back and forth between 1D, 2D or 3D mode, or when using memory intensive commands, only partial regions of the buffers are protected. The size of these protected regions are given with CONFIG command. The idea is that you can eat-up all the available memory for a single data-set if you which, but that certain operations will be forbidden on it.

For instance, you can handle a full memory 2D data-set if you do not wish to do 3D. However, when working in 3D, if you zero-fill a 2D plane extracted from the 3D, over the protected area size for 2D, you will destroy a part of the 3D buffer. You will sometime get the question : 'This will overflow the xx buffer, Ok?' when there is any risk of destroying one of the buffer. This question is not asked in macro execution, where you are supposed to know what you are doing.

See documentation for a more detailed description of the memory set-up.

see also : CONFIG VARIABLES

MESSAGE

message text

This command ermits to build interactive macros. If no argument is present on the calling line when MESSAGE is executed, the text will be issued to the user when the next data entry will be needed; if an argument is present, nothing will happend.

If the macro is called from a menu button, then the text will appear in the data entry dialog box.

see also : PRINT VARIABLES ALERT SETPROMPT

METH

METH 0/1/2

When set to 0, indicates that the current build-up curve is taken from 2 simple hydrogens, when set to 1, indicates that there is 1 methyle group, when set to 2, 2 methyle groups. Used by DIST

see also : RELAXRATE DIST CALIBDI

MINDATA

Compare the content of the current buffer with the content of the DATA buffer, and leave in memory the smallest of the 2 values.

Usefull for projections or symetrisation macros.

see also : MAXDATA ADDDATA MULTDATA

MINIMAX

MINIMAX n1 n2

Load the value of the minimum and the maximum points that will be used for peak-picking. These value are loaded by the MAX command.

see also : PEAK MAX

MINITER

Secondary iteration number for two steps minimisation in GIFA.
Number of iterations used for the line-minimization during one step of the MaxEnt iteration.
In a MaxEnt run, the number of FT's per iteration is MINITER + 2
Also used by other iterative process LINEFIT AUTOPHASE

see also : ITER

MINUS

Sets to zero the positive parts of the signal

see also : ONE ZERO PLUS ABS

MODIF_AMOEBA

MODIF_AMOEBA crdf1 crdf2 nopeak

Assign the pixel at coordinates crdf1 crdf2 to the amoeba of the peak nopeak. The pixel coordinates crdf1 and crdf2 are given in index.
Setting a nopeak of zero removes that pixel from the amoeba definition.

see also : MSKINTEG INTEG mskread mskwrite

MODIFYSIZE

MODIFYSIZE si1 si2 {si3}

Used mostly for hacking

Permits to modify the leading sizes of a 2D or a 3D data-set, provided the product of the sizes : si1*si2{*si3} is equal to the product of the old ones.
Does not actually modify the data.

see also : size CHSIZE

MODULUS

MODULUS is used to calculates the modulus of the spectrum.
In 2D, will only works for fully Hypercomplex data-sets (itype == 3) or for data-sets complex in F2 (itype == 1) not for data-set complex in F2 (itype == 2). In this last case, you should use FLOP before doing MODULUS.

see also : FLIP FLOP REAL ABS

MONOPOINT

Permits to click once on the data-set. The top value on the point stak : context variables, $POINTX[1-], $POINTY[1] and $BUTTON are set accordingly. If $NPOINT is zero (after a POINT_CLEAR) it is set to 1, unchanged otherwise.
Useful for macros.

see also : ZM POINT_PUSH point

MORPHOB

internal value for BCORR 3 algorithm

see also : BCORR

MORPHOS

internal value for BCORR 3 algorithm

see also : BCORR

MSKINTEG

integrates the current 2D data buffer using the last defined amoeba
mask.

The results are put into the peak list.

see also : MODIF_AMOEBA SIGN_PEAK ZERO_QU INTEG PUT GET mskread mskwrite

MULT

MULT x

Multiply data-set by a scalar.

see also : MULTDATA ADD ADDDATA

MULTDATA

Multiplies point by point, the content of the current working buffer with the content of the DATA buffer. Permits to realize convolution product. Works in 1D, 2D, in real, complex and hypercomplex modes.

see also : ADDDATA MIDATA MAXDATA MULT

MULTIPOINT

Permits to click several time on the data-set. Similar to POINT, but no output is generated. The values of the pseudo-variables $NPOINT, $POINTX[1], $POINTY[1] and $BUTTON are set accordingly. Useful for macros.

see also : MONOPOINT POINT

MUNSET

munset list_of_variable_name *

Removes all the variables which names are found in list_of_variable_name from the variable table. The list is terminated by a * sign. Note that no $ sign are needed.

see also : SET UNSET VARIABLES tunset

NCHANNEL

Context which holds the number of channels for multi-channel Maximum Entropy If NCHANNEL is equal to n>1, then the filter buffer is supposed to hold n different versions of the filter function (thus filter_size = n*size_of_data). n versions of the spectrum are reconstructed from the data, corresponding to the n different filters. To be active, FILTER should be set to 2, and the filter buffer loaded "by hand" with the PUT FILTER command. You can then manipulate the MaxEnt spectrum in 2D mode with the following sequence :

Gifa> DIM 2 CHSIZE 1 n*size_of_one_spectrum PUT ROW 1 MODIFYSIZE n %

see also : PUT FILTER SHOW MAXENT

NDISP

NDISP n

During a MaxEnt process, the display will refreshed every n iteration. n=0 means never.

see also : MAXENT

NEWFILEC

NEWFILEC file-name dim (parameters)(xdim) spec_freq itype

dim is 1D 2D or 3D

(parameters) is (size, offset, specw, freq)

parameters are entered in the order : F1 {F2 {F3 } }

Creates an empty data file, as described by the parameters. The file
can then be filled with PUTC or with macros. e.g.

newfilec data_set 2D 128 0 6000 100.13 256 0 4000 400.13 400 1

for a 1H-13C experiment on a 400MHz machine.

A file connected with NEWFILEC can be JOINed and DISJOINed. However a GETC is not possible (it is a Write only acces mode).

see also : JOIN GETHEADER PUTHEADER dataset macro/proc2d

NEWS

NOISE

Contains the level of noise in the data-set. When loading data (1 or 2D) the noise level is evaluated automatically from the last 10th of the data. Can also be set with EVALN.

Used by INTEG and by Maximum Entropy run.

see also : EVALN SHIFT

OFFSET

Permits to specify the offset of the right-most (upper right most in 2D) point of the data set. The value for offset are changed by EXTRACT

see also : SPECW size calib

OFFSET3D

offset3d dx dy dz

This context holds the shift by which the (0,0,0) point will be shifted from the lower left corner of the DISP3D window during a 3D display with the DISP3D/REF3D set of commands. A value of 0 corresponds to the lower left corner; a value of 1 to the upper right corner; a value of 0.5 corresponds to the center of the window.

see also : DISP3D PLOT3D CHECK3D

ONE

Put the value one (1.0) in the current buffer

see also : ZERO PLUS MINUS

OPEN

OPEN file_name

Opens the file called file_name for reading or writing. Writing is performed with the FPRINT command. Reading is done through the <file_name syntax.

e.g. read <file_name

read the file which name is found as the next line of the file file_name

set line = <file_name

set the variable line to the contents of the next line of the file file_name

OPENing an already OPENed file will rewind the file to the beginning.

see also : CLOSE FPRINT DBOPEN

ORDER

ORDER n

This context determines the order for all linear prediction operations. Should be an integer, typically one third of data size for spectral analysis, smaller values for simple operations such as compacting. You should set it before starting any linear prediction calculation.

see also : DT->AR DT->SVD BURG

OUTPUT

1 create numeric output of the data for each Maximum Entropy iteration
Mostly used for debugging purpose.

see also : MAXENT

OVERIT

Controls Maximum Entropy stop points. Overiteration is on then
CHI2 = 1.0 is no more the stopping criterion; and only the iteration
counter is considered.
0 inhibits over-iteration, 1 permits it.

see also : MAXENT MAXENTCONT ITER

PAGE

PAGE plot_file_name

This command appends code at the end of plot_file_name for page ejection; closes the plot file; and send it to the plotter.

see also : PLOT PLOTTER FORGET

PCOLOR

PCOLOR color_index plot_file_name

This command permit to change the color mode when plotting on a color Postscript printer. On HPGL PCLOR and PEN are equivalent.

color_index is : 1:black 2:red 3:yellow 4:green
5:cian 6:blue 7:purple 8:white

see also : PLOT COLOR PEN

PEAK

PEAK output {radius}

Will generate a peak-table from the current data set, output can be either a file-name or *screen. Peaks are located if their intensity lies within the margin defined by the MINIMAX command.

In 2D, PEAK will prompt you for a "packing radius". Peaks which lies within a distance equal or smaller than this packing radius (in points) will be ignored in favor of the larger peak in this radius.

The command SHOWPEAK will show you on the display where the peaks have been found. The command PKLIST will list the peak table contents.

This is the peak-picker form the PARIS integrator.

see also : MINIMAX SIGN_PEAK ZERO_QU MAX PKLIST PK->DT SHOW INTEG LINEFIT PKCLEAN PKCLEAR PKREAD PKWRITE PKSELECT PKRM PKRESET POINT->PK PKSYM PKPROJ SETPEAK pp

PEN

Pen n plot_file_name

Selects the pen number n on the plotter. On a postscript laser printer, this corresponds to different line width and dash patterns.

see also : PLOT PAGE PCOLOR

PH

PH is an interactive 1D phasing module. A button window pops-up which permits the user to adjust the 0th and 1st order phase. The 1st order pivot point is set by clicking in the 1D Spectrum window.

If you exit by clicking in Ok the phase correction is applied, and the phase values becomes then the default values for the PHASE command. If you exit with Cancel, no phase correction is applied and the original data is restored.

see also : PHASE HPHASE ZM apsl ph2dc ph2dr

PHASE

PHASE X Y {dim)

Permits to modify the phase of the current displayed spectrum. You have to enter zero-order (X) and first-order (Y) corrections, and dimension (dim) in 2D. Values are in degree, and first-order correction uses center point as pivot.

For 1D data an interactive phase command is available : PH.

see also : PH HPHASE apsl

PK->DT

Computes a pseudo-FID from the peaks parameters s held in the PEAK() table. Permits to display a spectrum computed by LP analysis.

see also : RT->PK PEAK SHOW

PKCLEAN

pkclean min_int

Will remove from the pak table entries with intensity less than min_int

see also : PKCLEAR PKSELECT PEAK PKRESET RT->PK

PKCLEAR

This command remove all entry from the current peak table.

see also : PKSELECT PKRM PKCLAN PEAK PKRESET

PKLIST

pklist i j

List the contents of the current peak table, from entry i to entry j.

see also : PEAK PKSELECT INTEG RT->PK

PKPROJ

PKPROJ axis alogrithm tolerance

Propagates the contents of the 2D peak-table on the 1D peak-table. All the 2D peaks will generate an entry in the 1D table. If two (or n) peaks are less than tolerance (in Hz) apart during the projection,

they will be merged into one peak. The intensity of the resulting peak will be either the mean value ot the max value of the initial peak, depending on the chosen algorithm (Mean or Skyline).
PKPROJ does not modify the spectral parameters (spectral width, frequences and offsets), so, if the 1D spectral parameters are different from the 2D parameters, strange results can be obtained.

see also: PEAK PKSYM

PKREAD

PKREAD file_name yes/no
Reads a peak list file created with PKWRITE into internal peak list. Depending on the option, the read values are appended to the current value (no) or the peak table is cleared before reading the file.

see also : PKWRITE PEAK PKRESET

PKRESET

This command recompute peak intensities from the current data set and the current peak table, giving results equivalent to PEAK. Usefull when either the data-set or the peak table has changed. Note that PKREAD reads in the whole peak table, including intensities, thus PKRESET permits to recompute intensities form a stored table.

see also : PEAK PKREAD PKCLEAR PKLIST

PKRM

PKRM index
Remove the peak index in the current peak table.

see also : PKRESET PEAK PKCLEAN PKCLEAR PKSELECT pkrmi

PKSELECT

PKSELECT index()
Select the peaks parameters whose index() are given. To end selection, enter the index 0.

see also : PKRESET PEAK PKCLEAN PKCLEAR PKRM

PKSYM

PKSYM param tolerance

Symmetrize in the 2D peak-table by either removing (param=0) entries in the peak table that do not have a symmetrical counterpart, or by adding (param=1) a symmetrical peak in the table. The presence or absence of a symmetrical peak is checked within a window of tolerance (in Hz) around the peak.
PKSYM does not modify the spectral parameters (spectral width, frequences and offsets), so, if they significally different on the both axis, strange results can be obtained.

see also: PEAK PKPROJ

PKWRITE

PKWRITE file_name
Writes the contents of the peak table on a file. The table can then be read with the PKREAD command.

see also : PKREAD PEAK

PLANE

PLANE axis index
In 3D only.
Permits to select a plane from the 3D buffer and copies it to the 2D buffer. Axis is either F1, F2 or F3 as the axis orthogonal to the chosen plane. Thus F1 selects plane F23, etc...

see also : VERT ROW COL DIAG DIM planeint vertint

PLOT

plot plot_file_name
Plots the current spectrum on the plotter, using the current value of contexts (see PLOT?).
If in 1D, plot the 1D spectrum
If in 2D, plot the contour-plot, as displayed with the CDISP2D command.
If in 3D, plot the currently selected plane, as displayed with the CDISP2D command.
will create a plot file named file_name, which can later be plotted by the command Gifa> PAGE plot_file_name
If your GIFA program is working in postscript (check with the CONFIG command), and if you have a postscript previewer (IBM, Silicon Graphics, some X-stations) then you can display the contents of your plot-file before sending it to the plotter.

The filename *s
will send the plot to the screen rather than to a file or the plotter.
The file name *PL will create a temporary plot file that will be deleted after plotting.

see also : PAGE CX CY PLOTOFFSET PLOTAXIS PLOTAXIS? GRID TITLE PLOTLINE PLOTLINES PLOTTEXT PLOTTEXTS PLOTPEAKS DISP1D CDISP2D STPL PLOT3D LEVEL PEN PCOLOR PLOTTER easyplot plot? plot_fit plotcolordisp plotn

PLOT3D

plot3d plot_file_name
Equivalent to the PLOT command, will generate a 3D plot, equivalent to the display obtained with the DISP3D command.

see also : PAGE PLOT DISP3D CHECK3D disp3d_form

PLOTAXIS

PLOTAXIS direc output
This command plots a set of axis on its output.
The axis are 0.5 cm below the current plotoffset.
If direc=F1 a vertical axis is drawn
If direc=F2 a horizontal axis is drawn
If direc=F12 both axes are drawn
PLOTAXIS uses the current SPECW and OFFSET for the axis calibration.

see also : PLOTAXIS? PLOT PAGE PLOT GRID

PLOTAXIS?

This command prompts you for different parameters that will be used by PLOTAXIS : unit (may be different of the UNIT command) and separations on x and y axes. The definition is also used for the GRID command.

see also : PLOTAXIS PLOT TITLE GRID

PLOTLINE

PLOTLINE X1 Y1 X2 Y2 plot_file
Draws a line from the X1 Y1 point to the X2 Y2 point (in cm, relative to PLOTOFFSET).

see also : PLOTLINES PLOTTEXT PLOTTEXTS PLOT

PLOTLINES

PLOTLINES X1 Y1 X2 Y2 plot_file

Draws a line from the X1 Y1 point to the X2 Y2 point (in index unit, projected on the CX x CY area, as used by the PLOT command).

see also : PLOTLINE PLOTTEXT PLOTTEXTS PLOT PLOTPEAKS

PLOTOFFSET

PLOTOFFSET x y

Values are used to offset the (0,0) origin of the plot at (x,y). Permits to draw several plots on the same page, or to leave room for axes. Default values are 1 1 (in cm).

see also : CX CY PLOT

PLOTPEAKS

will plot on the plot file, the contents of the internal peak table, thus permitting to plot the result of the PEAK command.

see also : PEAK SHOW PLOTLINE PLOTLINES PLOTTEXT PLOTTEXTS PLOT

PLOTTER

PLOTTER H(P-GL) / p(ostscript)

Permits to select which kind of output will be generated ans using all the plot commands. HP-GL or postscript code can be selected. If the gifaplot shell script has been correctly st-up, the correct output should be sent to the correct plotter with no user intervention.

see also : PLOT PAGE CONFIG

PLOTTEXT

PLOTTEXT string X Y plot_file

Sends the string "string" to the plot_file, at the X, Y position (in cm, relative to PLOTOFFSET) see also : PLOTLINE PLOTLINES PLOTTEXT PLOT

PLOTTEXTS

PLOTTEXTS string X Y plot_file

Sends the string "string" to the plot_file, at the X, Y position (in index unit, projected on the CX x CY area, as used by the PLOT command)

see also : PLOTLINE PLOTLINES PLOTTEXTS PLOT

PLUS

Sets to zero the negative part of the data set.

see also : MINUS ZERO ONE

POINT->PK

This command will add the point in the point stack (set by the user with the last point or MONOPOINT command), into the current peak table.

see also : MONOPOINT PEAK PKLIST SETPEAK POINT_PUSH point

POINT_CLEAR

clears the point stack

see also : POINT_PUSH point

POINT_INPUT

POINT_INPUT F1 {F2}

A point with coordinates F1 (in 1D, or F1,F2 (in 2D) is put on the top of the point stack.

see also : POINT_PUSH point

POINT_POP

remove one entry from the point stack.

see also : POINT_PUSH MONOPOINT point

POINT_PUSH

Add one entry into the point stack.

The point stack contains spectral coordinates, which can be used within macros, and are also used by certain commands (typically BCORR).

Whithin macros, the content of the point stack can be examined with the context variables : $npoint; $pointx[i]; $pointy[i]

The POINT_CLEAR command empties the stack.
The MONOPOINT command sets the top entry on the stack.
The POINT_PUSH command push the entry on the top, thus duplicating the entry, and adding an entry in the stack.
The POINT_INPUT command permits to enter a value with determined values
The POINT_POP command pops one entry in the stack.
The macro *point* permits fill the stack by clicking on the spectra
The macro *point_dump* permits to dump the content of the stack

see also : POINT_INPUT POINT_POP POINT_CLEAR MONOPOINT BCORR POINT->PK point point_dump

PRIMER

The GIFA program is a multi-purpose NMR program. It is designed for the processing and the display of 1D and muti-dimensional NMR data-sets.

The program includes sophisticated signal processing methods such as Maximum Entropy processing, Linear prediction, line fitting, complete control language, or 2D peak integrator as well as more classical processing, displaying and plotting of NMR experiments. It can handle the complete study of a set of NMR multidimensionnal data-sets, from the raw data to plotting, visualisation, and finally give some help to the assignment process.

In contrast with most NMR programs, Gifa tends to do as much data processing as possible in memory. Thus a typical processing session consists in :

* transfering the raw data-set from the spectrometer to the processing computer and converting it to a gifa-compatible format, for instance with the ux2gifa shell script.
* loading the data-set in memory with the command READ
* apodizing the data set with EM (exponential), GM (Gaussian), TM (trapezoid), SIN (shifted sine bell) or SQSIN (squared shifted sine-bell).

* changing the size of the data set (zero filling or truncature) with CHSIZE.
* realizing the Fourier transform in the acquisition dimension with ft_sim (simultaneous acquisition) or ft_seq (sequential acquisition); and in the non-classical dimension with ft_tppi (TPPI acq.), ft_sh (States-Haberkorn acq.), ft_sh_tppi (mixed mode) or ft_phase_modu (phase modulation) (there are als the basic commands : FT, RFT, FTBIS, IFT, IRFT, IFTBIS, REVF, INVF).
* correcting the phase with PH (interactive), ph_2dc and ph2dr (interactive in 2D), PHASE or HPHASE (batch); or taking the modulus with MODULUS.
* eventually correcting the base line with BCORR (see help for that command)
* contour plot as well as density plot are available for 2D's. There is even an on-screen stacked plot available.
* rows and columns from 2D's can be examined : rowint, colint (interactive); ROW, COL, DIAG; as well as planes and lines from 3D's planeint, vertint, PLANE, VERT, DIAG.

You should also know that :
o 1D, 2D and 3D processing can be selected
o plots are available on postscript as well as HP-GL
o there is a peak-picker in 1D, 2D and 3D, as well as a line-fitter (in 1D so far)
o MaxEnt and Linear Prediction are available with a comprehensive set of commands
o you can write your own commands with the command langage
o you can even set-up your own graphic interface with the BUTTONBOX command.

Try also :
* HELP a_given_command / for instance : help ft
* apropos a_given_topic / for instance : apropos transform
* reading the manual

PRINT

PRINT string
This command permits to issue a string to the user. Permits interactivity in command file.

see also : FPRINT ERROR ALERT SHOWTEXT PLOTTEXT printf sprintf fprintf

PROJ

PROJ n a

Permits to perform a projection of 2D data-set. Projections are done either along F1 (n=1) or F2 (n=2) axes, with either skyline (a=s) or mean (a=m) technic.

see also : DIAG DIM proj3d

PUT

PUT parameter {n}

Moves the content of the current buffer to an other buffer
With parameter equal to:
-DATA: load the data to be used for MaxEnt processing in 1D

-FILTER: load the filter used for Deconvolution. If NCHANNEL is greater than 1, then which channel you want to put. eg. PUT FILTER 3. PUT FILTER 0 will consider the current data set as the multichannel filter, and will load the whole filter. Useful when associated with GET FILTER to store filters as files.

-WINDOW: load the window to be used for MaxEnt processing

in 2D only
-ROW n : load the 1D buffer in the ROW n
-COL n : load the 1D buffer in the COL n
-AMOEBA : the mask definition to be used by the next MSKINTEG command

in 3D only
-PLANE Fx n : load the 2D buffer in the plane Fx n

see also : GET SHOW APPLY

PUTC

syntaxe depends on the dimentionnality of the currently joined data-set (to) and on the current context dim (from) :
1D to 1D : PUTC left right
1D to 2D : PUTC axis index left right
1D to 3D : PUTC axis index1 index2 left right
2D to 2D : PUTC left1 left2 right1 right2
2D to 3D : PUTC axis index left1 left2 right1 right2
3D to 3D : PUTC left1 left2 left3 right1 right2 right3

Writes into the currently connected file, the area of data as described.

see also : GETC JOIN dataset LISTFILEC

PUTHEADER

PUTHEADER param_name value

Set the value of a parameter associated to the currently JOINed data-set. If the parameter already exists, the value is changed, otherwise it is created. Any parameters can be used. Gifa already uses the following parameters (as listed with the dataset command) :

Frequency, Freq1, Freq2, Freq3, Specw1, Specw2, Specw3, Offset1, Offset2, Offset3, Type, Dim.

But you can use any other parameter name if you wish.

see also : JOIN NEWFILEC dataset GETHEADER

QUIT

QUIT yes/no

Equivalent to EXIT and BYE

see also : LOG

RCRYST

rcryst n

Will compute the discrepancy (computed as a R factor) between the current experimental nOe build-up curves, and the build-up curve as analysed from the last LP analysis.

see also : CALIBDI RELAXRATE DIST

READ

READ file_name

Reads the file as the new data set in standard format .

Same as READC

see also : WRITE READH READL READM READS READT READV READZ build2d build3d

READC

Same as READ

READH

READH file_name

Will read ft-nmr compatible files.

see also : READ WRITEH

READL

Load NMR1/NMR2 data-set

see also : READ WRITEL

READM

READM file_name

Loads a file created with the program MATLAB, using the -ascii option; or with the WRITEM command.

see also : READ WRITEM

READS

READS filename

Read from the disk in standard compact format. This format is compatible among incompatible platforms. It is ASCII format and can even be E_Mailed.

see also : READ WRITES READT

READT

READT filename

Read from the disk in text format. The data are in formatted values, 1 entry per line. Useful for entering data from an other program.

see also : READ WRITET READS

READV

Load Varian data set.

see also : READ

READZ

READZ filename
Reads from the disk in compacted file format.

see also : READ WRITEZ ORDER

REAL

REAL {Fx}
This command extract the real part of the current the data set considered as complex.
On 2D and 3D you can choose on which axis the REAL operation will be taken. Data size can thus be reduced either by 2 , 4 or even 8.

see also : ITYPE FLIP FLOP MODULUS

REF

This command will force a new computing of the display, as if a display command had been issued. This is useful for cleaning the spectrum after any command that draw on the spectrum, such as POINT, SHOWLINE, etc...

see also : UNREF, DISP1D, DISP2D, CDISP2D, REF3D REFMACRO

REF3D

This command will display a 3D view of the current 3D data-set, in the window previously opened with the DISP3D command. The parameters used for the display can be chosen with the DISP3D? and the CHECK3D commands.

see also : DISP3D CHECK3D

REFMACRO

When this context is set to 1 the program refreshes the on-screen display during macro execution, this permits to follow the data during the program execution, but slows down the macro execution.

When equal to 0, the display is refreshed only at the end of the macro execution. Default value is 0.

see also : CLEAR REF UNREF

RELAXRATE

Compute the relaxation rate (in arbitrary unit) from the last LP analysis.

see also : DIST CALIBDI RCRYST METH SLOPE RT->PK ORDER

RESIDUE

Returns into the current buffer the result of the substraction of the DATA buffer, with the reconstruction of the mock data from the spectrum held into the current buffer. To be used after a MaxEnt run to see that part of the signal not analysed by the MaxEnt process.

see also : SHOW GET MAXENT

REVERSE

REVERSE {Fx}
reverse the order of the current data-set (i.e. first points are last, last points are first).
If dataset is complex, REVERSE will reverse the complex vector (2 by 2).

see also : TRANSPOSE REVF INVF

REVF

REVF {Fx}
Processes FID data-sets by multiplying by -1 2 points out of 4. Permits to preprocess Bruker FIDs in Dim 2 (Bruker trick) before RFT, or permits to bring back zero frequency in the center for some other data formats

see also : INVF ITYPE FT REVERSE

RFT

RFT {Fx}
Perform real-to-complex Fourier Transform on data

see also : FT

ROTATE

ROTATE 0 / 1

When ROTATE=1, plots are rotated by 90 degrees, useful for plotting 1D spectra on A4 Postscript plotter or on a A3 HP-GL plotter

see also : PLOT

ROW

ROW n

Extract the nth 1D row (along F2) from the 2D data-set, and put it in the 1D buffer. The row will be available as a 1D data set when going from 2D to 1D

see also : COL PLANE VERT DIM rowint colint

RT->AR

RT->AR n

Compute back the autoregressive coefficients from a given set of roots.

n = 1 : forward coefficients are calculated from forward roots

n = 2 : backward coefficients are calculated from backward roots

n = 3 : both coefficient sets are calculated

see also : AR->RT, RTLIST, RTPIV, RTCLEAN, RTMATCH, RTSELECT, RTREFLECT, RTINV, RT->PK, ORDER

RT->PK

RT->PK size n delay

Calculate the peaks parameters from the roots by a least-square method, computed on the *size* first points of the current data-set.

This is the fourth (and last) step of the LP-SVD method.

n = 1 : forward roots

n = 2 : "matched" roots

delay is the dead-time delay of the first point .in seconds. (the error on the time zero) see also : RT->AR, AR->RT, RTLIST, RTPIV, RTCLEAN, RTMATCH, RTSELECT, RTREFLECT, RTINV, ORDER, PK->RT

RTCLEAN

RTCLEAN n

Keep only the roots whose modulus is greater than one. Useful to clean up root-set computed from a backward analysis, before inverting.

n = 1 : forward roots

n = 2 : backward roots

see also : RT->AR, AR->RT, RTLIST, RTPIV, RTMATCH, RTSELECT, RTREFLECT, RT->PK, RTINV, ORDER

RTINV

RTINV n

Calculate the inverse conjugate of all the polynomial roots, thus converting backward and forward polynomial. To be used after a backward analysis.

n = 1 : forward roots are inversed, and their inverses are put in the backward root array.

n = 2 : backward roots are inversed, and their inverses are put in the forward root array.

see also : RT->AR, AR->RT, RTLIST, RTPIV, RTCLEAN, RTMATCH, RTSELECT, RTREFLECT, RT->PK, ORDER

RTLIST

RTLIST n i j

List the roots of prediction-error polynomial from entry i to entry j.

n = 1 : forward roots

n = 2 : backward roots

n = 3 : matched roots

see also : RT->AR, AR->RT, RTPIV, RTCLEAN, RTMATCH, RTSELECT, RTREFLECT, RTINV, RT->PK

RTMATCH

RTMATCH nsignals

"Match" the forward and backward roots to eliminate the roots dues to noise. Those matched roots can then be used by RT->PK, with the "matched forward & backward" roots. nsignals is the number of true roots expected.

see also : RT->AR, AR->RT, RTLIST, RTPIV, RTCLEAN, RTSELECT, RTREFLECT, RTINV, RT->PK, ORDER

RTPIV

RTPIV alpha

Will remove all the forward roots with frequency higher than 1/alpha (in -PI +PI unit). Used to remove spurious frequency when doing build-up curve analysis.

see also : RT->AR, AR->RT, RTLIST, RTCLEAN, RTMATCH, RTSELECT, RTREFLECT, RTINV, RT->PK

RTREFLECT

RTREFLECT n

Calculate the inverse conjugate of roots lying outside the unity circle. The modified roots are let in the same root array.

n = 1 : forward roots

n = 2 : backward roots

see also : RT->AR, AR->RT, RTLIST, RTPIV, RTCLEAN, RTMATCH, RTSELECT, RTINV, RT->PK, ORDER

RTSELECT

RTSELECT n index()

Select the roots whose index() are given. To end selection, enter the index 0.

n = 1 : forward roots

n = 2 : backward roots

n = 3 : matched roots

see also : RT->AR, AR->RT, RTLIST, RTPIV, RTCLEAN, RTMATCH, RTREFLECT, RTINV, RT->PK, ORDER

RZOOM

This command permits, when in zoom mode in 2D, to jump to a zoom window symmetric relatively to the diagonal of the one currently used.

see also : ZOOM ZM

SCALE

SCALE x

Changes display scale factor. Default value is 1. The data-set is multiplied by SCALE before display. The larger the SCALE, the higher the spectrum will be.

Used by all graphical output, either on screen or on plots.

see also : DISP1D, DISP2D, DISP3D, CDISP2D, ZM, PLOT, LEVEL

SCALE3D

SCALE3D x

This context is used to reduce / expand the drawing during a 3D display, usually a value smaller than 1 is needed to fit the whole cube within the graphic window.

see also : DISP3D PLOT3D

SCOLOR

scolor index

Determines the color used by the SHOW command

index is : 1:white 2:red 3:yellow 4:green
5:cian 6:blue 7:purple 8:black

see also : COLOR SHOW SHOPEAK

SEGM1

SEGM1 n

choose the way the first segmentation is done in BCORR 3 .

0 : without.

1 : with a standard deviation algorithm on the data set.

2 : with a standard deviation algorithm on the first derivative
of the data set.

3 : with thresholds on the data set, and on the first and second derivatives.
4 : with a dynamic clusters algorithm.
+10 : with a morphological filtering after the segmentation.

see also : BCORR

SEGM2

choose the way the second segmentation is done in BCORR 3. This feature is only useful with a polynomial approximation.
0 : without.
1 : interactive.
2 : automatic.
When you choose an interactive second segmentation, you have to choose one or several areas with the WINDOW command to allow the program to cut the correction in these areas if needed. The automatic segmentation uses a window, WINSEGM2, and a threshold, LEVELSEGM2.

see also : BCORR

SET

SET variable_name = value
SET static_variable := value

Assigns the value of the variable variable_name to value. Creates the variable if it does not already exist. Note that you should not use a $ sign before variable_name. value can be any kind of value : number, string, read from a file, another variable, a calling parameter ($_) or an evaluated expression.
If value is missing, or if $_ is used and no parameter are available, the user is prompted for input.
When the variable is created during the execution of a macro, the first syntax creates a local/volatile variable which value cannot be accessed by subsequent called macros, and that will be removed at the end of the macro. The second syntax creates a global/static variable that can be accessed by any macros, and that will remain until explicitly removed with the UNSET command.

see also : UNSET MUNSET DUMP VARIABLES FUNCTIONS

SETPATH

SETPATH new_path_string
Permits to change the current GifaPath used for calling macros.

The new_path_string contents all the directories in which macros file will searched for, the list is blank separated. The content of the current GifaPath is held into the $GIFAPATH context. So the following is a very typical use for adding an address

SETPATH ("new_directory"; $GIFAPATH)

See also : MACRO SETPROMPT

SETPEAK

SETPEAK index F1 { F2 { F3 } }

Permit to modify directly the content of the peak table. The number of entries depends whether you are in 1D, 2D or 3D.
The peak entry at *index* is modified in the current peak table. Only coordinates are given, intensity is taken form the data-set, widthes and other parameters are lost.
This is not the way for adding new peaks in the peak table, use POINT_INPUT POINT->PK instead.

See also : PEAK POINT->PK PKRESET

SETPROMPT

SETPROMPT new_prompt_string

Permits to modify the prompt used in the text window.

See also : MACRO

SETVAL

setval i { j { k } } x
Will set the value of the data point to x. The number of coordinates of the point depends of dim. In dim 2 or 3, coordinates are F1 F2 or F1 F2 F3. Can be usefully used when associated to the functions valnd() to change data point value.

see also : SET FUNCTIONS

SH

SH string
Will pass the string to the operating system. For instance, used by macros such as pwd, ls, more, etc...
SH alone will enter the Operating system

see also : CD ls pwd more rm vi vim vip

SHIFT

This context holds the systematic baseline shift of the current data-set,
computed automatically by EVALN.
Used by INTEG.

see also : EVALN NOISE ADDBASE

SHOW

SHOW parameter
Displays graphically extra information without disturbing the content of the working buffer. Uses the current definition of SCOLOR for drawing.

With parameter equal to:
-WINDOW: display actual window used to compute the chisquare
-FILTER: display filter used for Deconvolution
-LAMB: display evolution of Lambda during MaxEnt iteration
-ENT: display evolution of Lambda during MaxEnt iteration
-CHI: display evolution of Lambda during MaxEnt iteration
-STEP: display evolution of Lambda during MaxEnt iteration
-SUM: display evolution of Lambda during MaxEnt iteration
-CONV: display evolution of Lambda during MaxEnt iteration
-FT: display evolution of Lambda during MaxEnt iteration
-RESIDUE: displqy the residue of the spectrum after a MaxEnt run
-CURRENT: moves the current displayed image to the other window
-LINEFIT: the result of the last line-fitting computation
-AMOEBA: the mask used by the last INTEG command

see also : GET PUT APPLY SHOWTEXT SHOWLINE SHOWPEAKS SHOWPEAK SCOLOR SHOWC

SHOWC

SHOWC scale

display the currently JOINed file using the current zoom window (redefined in ppm and used on the JOINed data-set)

The current parameters are used for the display (VHEIGH, LEVEL, etc..) but not SCALE which is passed to the command. In 2D the display is only showed in the controur window. The command is not implemented in 3D yet.

This command is used by the super1d / super2d macros to display several spectra on screen.

see also : JOIN SHOW SCOLOR super1d super2d

SHOWLINE

showline x1 y1 x2 y2

Will draw a line on the displayed spectrum. Coordinates are in index values. The line will be drawn accordingly to the zoom mode. When in 2D with density plot and contour plot both on, the contour plot will be used for display. Uses the current definition of SCOLOR for drawing.

see also : SHOWTEXT PLOTLINES PLOTLINE SHOWPEAKS SHOWPEAK SCOLOR

SHOWPEAK

showpeak index

will display on the current graphic, the indexth peak of the peak table. Uses the current definition of SCOLOR for drawing.

see also : PEAK PLOTPEAKS SHOWPEAKS SCOLOR

SHOWPEAKS

will display on the current graphic, the contents of the internal peak table, thus permitting to visualize the result of the PEAK command.Uses the current definition of SCOLOR for drawing.

see also : PEAK PLOTPEAKS SHOWPEAK SCOLOR

SHOWTEXT

SHOWTEXT string X Y

Displays the string "string" on the displayed spectrum, at the X, Y position. Coordinates are in index values. When in 2D with density plot and contour plot both on, the contour plot will be used for display. Uses the current definition of SCOLOR for drawing.

see also : PLOTTEXT PLOTTEXTS SHOWLINE SHOWPEAKS SHOWPEAK SCOLOR

SIGN

SIGN -1 / 0 / 1

Tells the program how to display 2D data-sets. Used by DISP2D, CDSIP2D and PLOT commands

if 1, 2D program will display only positive levels,

if -1, 2D program will display only negative levels,

if 0, 2D program will display both positive and negative levels,

see also : DISP2D CDISP2D PLUS MINUS

SIGN_PEAK

SIGN_PEAK 1 / -1

Indicates to PEAK and to INTEG to search for positive or negative peaks.

see also : PEAK INTEG

SIMU

Simulate data, works in 1D in 2D and in 3D

You will be prompted for input. SIMU memorizes the last used values so SIMU %% is a valid syntax.

see also : SIMUN SimSpect2D READ ADDNOISE

SIMUN

(1D) simun ampl linecoord linewidth phase

(2D) simun flg ampl linecoordf1 linecoordf2 linewidthf1 \
linewidthf2 phasef1 phasef2

(3D) simun ampl linecoordf1 linecoordf2 linecoordf3 \

linewidthf1 linewidthf2 linewidthf3 phasef1 phasef2 phasef3

Simulate data, works in 1D in 2D and in 3D
You will be prompted for input. One line is simulated at each run,
and is added to the data. The line coordinates are entered in PPM,
Hz or Index, according to the unit value. The line widthes are in Hz.
The 1D and 3D data are complex in all dimensions, the 2D data can be
phase or amplitude modulated, depending on flg :
flg = 0 : the data are simulated in phase modulation
flg = 1 : the data are simulated in amplitude modulation

SIMUN memorizes the last used values
so SIMUN %% is a valid syntax.

see also : ZERO SIMU READ ADDNOISE SimSpect2D

SIMUNOE

This command is now obsolete, it has been superseded by the SimSpect2D macro which realizes the same action.

This command permits to simulate a 2D data set from the result of a program computing NOESY intensities (such as CORMA).

It will prompt you for the filename of the Corma output file, the filename of the chemical shift file (generated with the command CHEMS), the name of the output file, holding information on simulated peaks and for number of data points and spectral windows you wish to simulate.

It generates a time domain data-set which can be subsequently processed as a regular FID.

see also : SimSpect2D CHEMS SIMU

SIN

SIN x { axis }

SIN is used to multiply data by a trigonometric function varying continuously from pure sine to pure cosine. It works for 1D 2D and 3D spectra. You have to enter first the value of the parameter x varying continuously from 0 (pure cosine) to 0.5 (pure sine). x corresponds to the position of the maximum of the filter in your window. If you are working on a 2D or a 3D spectrum, there is another parameter which determines on which axis (axes) the apodization will be applies.

see also : SQSIN EM GM TM JMULT FILTER

SLOPE

slope n

Will compute a relaxation rate on the current nOe build-up curve by using the initial slope method, computed from the first n points of the data-set.

see also : RELAXRATE METH

SMOOTH

SMOOTH z { z2 { z3 }}

where z { z2 { z3 }} is the size of the smoothing window

Permits to smooth a data set by the moving-average technic. Works in 1D as well as in 2D and 3D.

see also : MEDIAN

SMOOTH1

SMOOTH1 n

to be used with BCORR 3

controls the smoothing of data for the first segmentation

0 : no smoothing.

+1 : moving average of the data set on a window of WINMA points.

+10 : hysteresis smoothing of the data set with the value LEVELHYSTE.

see also : BCORR

SNAPSHOT

SNAPSHOT file_name

Permits to store the current graphic display to a UIS file (VMS/VWS only) The file can then be displayed with the render command (see VWS documentation)

SPECW

SPECW x { y { z }}

Permits to enter the value for the spectral width of the current data-set. One parameter will be needed for each dimension of the data-set.

When reading a file the spectral width is set to 2000 * 3.1416 if no parameter block is available.
The value for spectral width are changed by EXTRACT

see also : size OFFSET EXTRACT calib

SPIN

SPIN n
Shifts the color table of 2D density image n times. Nice but probably useless. (Only VWS)

SQSIN

SQSIN x { axis }

SQSIN is used to multiply data by the square of a trigonometric function varying continuously from pure squared sine to pure squared cosine. It works for 1D 2D and 3D spectra. You have to enter first the value of the parameter x varying continuously from 0 (pure square cosine) to 0.5 (pure square sine). x corresponds to the position of the maximum of the filter in your window. If you are working on a 2D or a 3D spectrum, there is another parameter which determines on which axis (axes) the apodization will be applies.

see also : SIN EM GM TM JMULT FILTER

STDY

Parameter for STPL.
Each new line during STPL is shifted by STDY centimeters on the Y axis. Default is 0.1

see also : STPL STPL?

STEP

Constant used when Maximum Entropy run with no line-minimization.
Algo 1, (1E-6..0.5) Algo 2 or 3 (0.1 .. 10)
These algorithms are usually highly inefficient.!

see also : MAXENT STEPMAX ALGO

STEPMAX

This context holds the maximum extension of the search during the line-minimization. Default value is 1.2 for GIFA algorithm, 100 for CONJG.

see also : STEP ALGO MAXENT

STPL

STPL output

This command permits to draw stacked-plots of 2D data-sets. The plot is determined by the value of the following parameters:

CX : length in centimeters of a line.

CY : maximum height of a peak in centimeters.

SCALE : scaling applied to the data before plotting, if scale=1 then the larger peak on the surface will be CY high, if scale>1 then clipping will occur.

STDY : each new line is offsetted by STDY centimeters in the Y axis.

STSKIP : a line every STSKIP line will drawn.

STSHIFT : each new line will shifted right by STSHIFT points, if negative then the shift is done on the left. May be fractional.

STSKEW : determine how the horizontal line are skewed during plot, if negative skewed to the right, to the right if positive.

output is either a file name or *s for on screen display.

The command STPL? prompts you for all those parameters

see also : STPL? STDY STSHIFT STSKEW STSKIP CX CY PLOT PAGE

STPL?

Prompts for all the parameters of the STPL command (see STPL)

see also : STPL STDY STSHIFT STSKEW STSKIP

STSHIFT

Parameter for STPL.

STPL will shift each new by stshift points to the right during the plot. May be negative, null or positive; may be fractional.

Default is 0.5.

see also : STPL STPL?

STSKEW

Determines how the horizontal lines are skewed during a stacked plot. If 0 no skewing occurs, if positive the plot is skewed to the right, if negative, the plot is skewed to the left.
For instance, a value of -1.0 will skew the horizontal by 45° to the left. STSKEW gives nice effects in conjunction with STSHIFT.

see also : STPL STPL?

STSKIP

Parameter for STPL.
STPL will skip stskip lines during the plot. Default is 1.

see also : STPL STPL?

SUMCONS

When set to 1, the program tries to minimize the value of the sum of all the points of spectrum during the Maximum Entropy run.
Should be set whenever a window 1 n is used, or when Antiphase J-deconvolution is used.

see also : MAXENT ALGO

SUMREC

SUMREC x1 y1 x2 y2
Realizes the integration under a given area, value is corrected for baseline shift (if any computed with EVALN)

see also : INTEG EVALN sumrec_1d

SVD->AR

SVD->AR n
Compute the autoregressive coefficients from the singular decomposition. This is the second step of the LP-SVD method.
n = 1 : forward autoregressive coefficients
n = 2 : backward autoregressive coefficients

n = 3 : both coefficients

see also : SVDLIST SVDCLEAN1 SVDLIST SVD->DT ORDER DT->SVD svd_rev

SVD->DT

Compute back a new data-set from a truncated set (cf SVDCLEAN1) of singular values.

see also : SVDLIST SVDCLEAN1 SVDLIST SVD->AR ORDER DT->SVD

SVDCLEAN1

SVDCLEAN1 nsignals flag

Keep only the first n signals singular values and put the other to 0. Use it only if you have a good insight of the number of signals in the spectrum. The flag (yes/no) determines wether you want to remove the noise power associated with the removed root.

see also : SVDCLEAN2 SVDLIST SVD->DT SVD->AR DT->SVD ORDER cadzow

SVDCLEAN2

SVDCLEAN2 level flag

Keep only the singular values larger than level and put the other to 0. The flag (yes/no) determines wether you want to remove the noise power associated with the removed root.

see also : SVDCLEAN1 SVDLIST SVD->DT SVD->AR DT->SVD ORDER

SVDLIST

SVDLIST i j

List the singular values from index i to j.

see also : SVDCLEAN1 SVD->AR SVD->DT

SWA

SWA {Fx}

Consider the left part of the data as real and the right part as imaginary, and swap them to transform the data set to the internal representation of complex data; i.e. odd points real and even points imaginary. This is the opposite of USWA.

see also : USWA REAL FLIP FLOP

SYM

SYM n

Sumetrizes the current 2D data set using either a mean value (n=1) or a smallest value (n=2) algorithm. n=3 corresponds to an algorithm which smootly switch from one algorithme to the other depending on the difference between the 2 values.

see also : TRANSPOSE MINDATA

TIMER

When this context is set to 1, then the time needed to execute each command is given. If the command TIMER 1 is given while the value of timer is already 1, the timing is issued, and the clock reseted to 0. Within macros, only the overall execution time of the macro is given, unless an explicit TIMER 1 or TIMER 0 is given.

see also : VERBOSE DEBUG REFMACRO SH

TITLE

TITLE string_of_characters plot_file

Draw the string at x=0, y=cy+1.0 on the designed plot_file.

see also : PLOT PAGE PLOTTEXT PLOTTEXTS

TM

TM n1 n2 {Fx}

This commands applies a trapezoid filter function to the data-set. The function raises from 0.0 to 1.0 from the first point to point n1. The function then stays to 1.0 until point n2, from which it goes down to 0.0 at the last point.

If in 2D or 3D then Fx tells on which axis to apply the filter.

see also : SIN SQSIN GM EM FILTER

TOLERANCE

TOLERANCE x
This context holds the value used by all minimisation routines within GIFA as
stopping criterium : line fitting, LineMinimisation in MaxEnt, etc...
Typically a small value (1e-2 .. 1e-4)

see also : ITER MINITER

TRANSPOSE

Transposes the 2D matrix or planes of the 3D cube. The sizes of the matrix must be a power of two for this command to be used. After transposition, the two dimensions are completely permuted

see also : SYM

UNIT

UNIT value

The context UNIT will tel some other commands (mostly POINT) to choose the unit used. UNIT can take the values INDEX (number of channel) HERTZ (based on SPECW and OFFSET), PPM (HERTZ divided by FREQ) and SECOND (usefull only for FID). All other values are invalid.

see also : PKLIST SPECW OFFSET FREQ calib point

UNREF

When this command is issued, the command line will not refresh the current display. Typically used at the end of a command line in macr :

```
read file  unref    ; read the file but does not display it
```

This is the inverse of REF

see also : CLEAR REF UNREF

UNSET

unset variable_name

Remove the variable variable_name from the variable table. Note that no $ sign are needed.

see also : MUNSET SET tunset

USWA

USWA {Fx}

Take a complex data set in internal representation (i.e. odd points real and even points imaginary), and put the real part on the left side, and the imaginary part on the right side of the data set. This is the opposite of SWA.

see also : SWA FLIP FLOP REAL

VARIABLES

There are two kinds of variables in the macro language :

o User variables that are handled in macros

o Contexts that hold the different internal states of the program

In all cases, variable names are case insensitive.

User variables :

can be set :

```
set a = 3
```

evaluated :

`$a` (directly) `($a+3)` (or in expressions)

Variables are usually lcal to a macro (or a formbox), unless specified :

```
set b := 4
```

creates a variable $b that can be evaluated anywhere, (static and global)

Associative array can be built with the syntaxe :

```
set array["abc"] = 5
```

Associative arrays can be linked to dbm data file (see DBMOPEN)

User variables can also be UNSET (rmoved from the memory)

The total number of user vatiables is limited, as given by the CONFIG command

On the other hands, contexts can only be evaluated, but not set:

```
if ($dim != 2) ....
```

see CONTEXTS for the list of all the context variables

see also : CONTEXTS FUNCTIONS SET UNSET MUNSET CONFIG DBOPEN

VERBOSE

When this context is set to 1 the program switches to a more verbose mode. Specially, text output will be generated during macro execution, during MaxEnt run and during LP execution. When verbose equals to 0, no text is generated.

see also : DEBUG

VERT

vert i j

In 3D mode, extract a column orthogonal to the last displayed plane.
The column is taken at coordinates i and j in this plane.

see also : PLANE COL ROW DIM vertint planeint

VHEIGHT

This context is used when displaying or plotting 1D spectra, it determines the level at which the zero value will be displayed in the window. Values are in %, i.e. 0 corresponds to the bottom line and 1 to the top. Default value is 0.3

see also : DISP1D PLOT

WHILE

WHILE test
..commands
ENDWHILE

Commands in the WHILE .. ENDWHILE structure will be executed as long as the test holds true. Can be used only in Macros.

WHILE and ENDWHILE should appear alone on one line, eventually followed by a comment.

see also : IF FOR GOTO FUNCTIONS CONTROLS

WINDOW

WINDOW {Fx} x y or WINDOW {Fx} 0

Define the window (with the starting point and the ending point)on which data is actually used for the iteration. Data outside this window(displayed as 0 during the interactive input) are just ignored for the processing. Window can be entered several time, the result being cumulative.
Entering 0 to reset the whole window available.

see also : SHOW PUT APPLY

WRITE

WRITE file_name

Writes the current data set to a file in standard format.

see also : READ WRITEC WRITEH WRITEL WRITEM WRITES WRITEZ

WRITEC

Same as WRITE

see also : READC WRITE

WRITEH

WRITEH file_name

Writes the current data set to a file in a standard format compatible with ft-nmr file format.

see also : READH WRITE

WRITEL

WRITEL file_name

Write a NMR1/NMR2 compatible data-set

see also : READL WRITE

WRITEM

WRITEM file_name

Creates an ascii files which can than be read with the MATLAB program, or ith the READM command. Does not work for 3D data-set.

see also : WRITE READM

WRITES

WRITES filename

Writes to the disk in standard compact format. This format is compatible among incompatible platforms. It is ASCII format and can be shared in a network by different machines, and can even be E_Mailed.

see also : READS WRITE WRITET

WRITET

WRITET filename

Write to the disk in text format. The data are in formatted values, 1 entry per line. Useful for sending data to an other program.

see also : READT WRITE WRITES

WRITEZ

WRITEZ filename

Write from the disk in compacted file format. Uses ORDER for compaction.

see also : READZ WRITE

ZERO

Put the value zero (0.0) in the current buffer. Quite a destructive command !

see also : ONE PLUS MINUS

ZERO_QU

ZERO_QU 0 or 1

Controls the algorithm for INTEG 2D peak integration. When ZERO_QU is 0, normal integration is done, When 1, the amoeba is computed one the Absolute value of the data-set, and the integration is done over this amoeba in normal mode. Useful when integrating NOESY peaks with Zero-Quanta contribution. (Zero-quanta integrates to 0)

see also : PEAK INTEG SIGN-PEAK

ZM

Creates a graphic box that permits to control the value of SCALE and the zoom state.

To zoom in the display, simply click on the graphic window with the left and middle buttons pressed, and draw a rectangle on the display. You can redraw as many time as wished this rectangle. To zoom into the selected region, simply click in this rectangle with the left and right button of the mouse. You can also use the Zoom in and Zoom out buttons in the control box.

The four little arrows can be used to move around the zoom region. Each arrow will move the selected region by half the size of the zoom box.

There are five controls to adjust the vertical scale used for displaying the spectra. The Reset button will reset the display to the default, (`ABSMAX 0 SCALE 1 ZOOM 0`) where the largest peak in the screen in full size. The four other buttons permit to modify the scale value, raising or lowering the SCALE context by factor of 2 and of 1.2.

The ZM box also displays the coordinate of the cursor when a button is depressed, the unit used for display depends on UNIT

see also : ZOOM SCALE MONOPOINT UNIT point

ZNOT

znot value

This context holds the focal length that will be used for computing a 3D display with the DISP3D/REF3D set of commands. A large value (1000) will produce a nearly orthometric projection, a small value (10 or less) will result in large conic distortions.

see also : DISP3D DISP3D?

ZOOM

ZOOM n x1 y1 (1D) or x1 x2 y1 y2 (2D) or x1 y1 x2 x3 y2 y3 (3D)

if n=0, zoom mode is off.
if n=1, zoom mode is on, you are then prompted for the coordinates (row,col) of the lower left point and upper right point for defining the 2D zoom window.
In 1D, you are only prompted for left and right borders.
In 3D, you first do a 1D selection on one axis, and then a 2D selection on the orthogonal plane.
If a region was already selected on the display with the POINT command, this region is proposed as the default region.

see also : RZOOM ZM point zoom3di