
Subject: Re: New User seeks book
Posted by [grunes](#) on Tue, 14 Mar 1995 18:41:40 GMT
[View Forum Message](#) <> [Reply to Message](#)

In article <3k4h6i\$In0@netnews.upenn.edu> stauffer@psych.upenn.edu (Michael George Stauffer) writes:

> I'm new to IDL, and am hoping to find a book to help me
> learn how to program(in IDL). The manuals seem to have most of the
> necessaary info, but it's taking me a while and I'd like to get up to speed
> so I can work on some actual projects. I've called a few local (Philly)
> bookstores, but no luck.

See the section below.

> BTW, what's P-Wave?
PV-Wave, you mean. Read the FAQ.

For the first question, RSI probably intends that you will go through their interactive on-line help. The only problem is that, as with most on-line help and hypertext systems, you pretty much have to know the exact name of a feature before you can look it up.

For the names of features, you are welcome to my incomplete, probably incorrect, and out of date notes that I took a long time ago. The good part is that it is old enough that it mostly represents the common core of IDL and PV-Wave:

-----CUT HERE-----

IDL: notes by mitch grunes from Introduction to IDL, RSI (Research Systems, Inc. (303) 399-1326

VARIABLE: named locations for storing informat
name=letter+up to 14 characters from a-z,digit,_,\$,

TYPE	EXAMPLE
unsigned byte	1B
(short) integer (16 bit)	1
long integer (32 bit))	1L
single precision floating point	1. 1E0 (must contain '.')
double precision floating point	1D0
complex floating point	complex(1,0)
string (length=0-32767)	'1'
undefined (no defined value)	undefined
structure (composites)	{type_name,x:5,y:2,a:'xy'}
mixed expressions similar to fortran: 3/2	gives 1
	3/2. gives 1.5

ORGANIZATION	EXAMPLE
scalar	1
vector (1 dimensional array)	[1,2,3]

array [[1,2],[3,4]]
 up to 7 dimensions
 subscripts
 start at 0 x(0)
 * means ala x(2,*)
 ranges x(0:4)
 * means end x(3:*)
 array subscripts x([1,2,3])
 dimension matches subscript
 values clipped to legal subscript values
 a+b (and similar ops) will have length = to the smaller of
 the lengths of a and b
 a(i,j)=b will place array b into a starting at indices i,j
 matrix interpretation: (col,row)
 structure
 ,=array concatenation a,[1 2]

OPERATORS

executed left to right within precedence, order changed by ()
 precedence
 1: ^=power
 2: *=multiply #=matrix multiply /=divide MOD=modulus
 3: +=add -=subtract or negate <=minimum >=maximum NOT=logical not
 4: EQ NE LE LT GE GT (arithmetic comparisons give 0 or 1)
 5: AND OR XOR

General

user's library provided with IDL
 your own functions and procedures
 operators and most functions apply to arrays too

Statement types

assignment: variable=expression
 INCONSISTANCY: if expression if of byte type, variable is integral
 Note: the parenthesized expression (variable=expression) has a value.
 procedure call: <procedure_name>[,arguments][,keyword=expression]
 arguments in procedures and functions passed by reference unless are
 expressions
 keywords may be abbreviated to shortest unique string
 keyword order with respect to each other and positional parameters unimportant
 /keyword is same as /keyword=1
 includes user procedures (including those provided in user library)
 and system procedures.
 executive commands: .<command>
 FOR variable=low_expression,hi_expression[,incr] DO statement
 if low_expression>hi_expression, no execution
 IF expression THEN statement [ELSE] statement
 WHILE expression DO statement
 REPEAT statment UNTIL expression
 CASE expr OF
 expr: statement ... END

GOTO,label
compound statement: BEGIN statement,...,statement END
PRO procedure_name,parameters
FUNCTION function_name,parameters
COMMON block_name,variables
END ends a procedure or function
RETURN value returns function value

Syntax

upper/lower case: ignored
Special first characters of line:
. <command> executive command
? help request
\$ <command> operating system command
@ <file> redirect file and command input to <file>.pro

Keys

^Z suspend IDL (resume with fg)
^D EXIT IDL, close files
<up arrow> recall previous command, enter line editing mode
<esc>? help: print this list
<esc>H same as <esc>?
<up,down,left,right arrow> move around in recall buffer
^N same as <up arrow>
^A start of line
^D end of line, EOF if current line empty
^E end of line
^R redraw current line
^B or R13 back 1 word
^F or R15 forward 1 word
<esc>I insert/overstrike toggle
<backspace> delete previous character
 same as backspace
^W delete word
<esc> delete previous word
^U kill to end of line
^text search for text, recall prev line if text blank

Continuation character: \$ at end of line

; means rest of line is a comment

SYSTEM PROCEDURES AND FUNCTIONS

Array manipulation

HISTOGRAM(ar) count intensity distributions.
result vector indices range from 0 to max value.
type is long integer
MAX(ar[,max_subscript]) maximum element
MIN... minimum...
MEDIAN(Ar[,width]) median filter for byte arrays
N_ELEMENTS(x) number of elements in x
REBIN(ar,d1,d2...) resample to given dimensions.
dimensions multiple or factor of those of ar.

/SAMPLE	use nearest neighbor or average instead of bilinear interpolate
REFORM(array,d1,d2...)	reform
ROTATE(mat,direction)	rotate and/or transpose matrix
SHIFT(ar,s1,s2...)	rotate element positions. (<0=left!)
TOTAL(ar)	sum
SMOOTH(ar,n)	NOTE: result is float when applied to integer smooth using n-sized box car average
SORT(ar)	sort
TRANPOSE(ar)	transpose
WHERE(ar[,count])	subscripts of non-zero elements
INVERT(matrix[,status])	returns vector invert
LUDCMP,a,index,d	LU decomposition of square matrix
LUBKSB,a,index,b	solve linear equations, given LU decomposition
MPROVE,a,alud,index,b,x	iterated improvement of LUDKSB
SVD,a,w[,u,v]]	Singular value decomposition
SVBKSB,u,w,v,b,x	solve simultaneous equations Ax=b using SVD
TRED2,a[,d[,e]]	reduction by householder's method of a real symmetric matrix to tridiagonal form.
TQLI,d,e,z	eigenvalues and eigen vectors of real, symmetric tridiagonal matrix
TRIDAG,a,b,c,r,u	solution of tridiagonal linear set of equations
ZROOTS,a,roots[,polish]	roots of complex polynomial
CONVOL(ar,kernal[,scale_factor])	convolve array ar(n) with kernal(m) result(i)=sum(over j) a(i-j)*b(j) for i=<size of kernal>-1,..., <size of ar>-1 result(i)=0 for i=0...<size of kernal>-2 Also works for arrays. Final result is divided by scale_factor.
FFT(array,d)	fast fourier transform note: d=-1,1 for forward,reverse most efficient when dimension is power of 2
convert type	
BYTE(x[,offset[,d1,d2...]])	convert to byte
BYTSCCL(array)	scale by min,max and convert to byte
TOP=!D.N_COLORS-1	sets top scaling to 255
MIN=min	uses min instead of data minimum
MAX=max	similar for max
FIX(x[,offset[,d1,d2...]])	convert to integer
LONG(x[,offset[,d1,d2...]])	convert to long integer
FLOAT(x[,offset[,d1,d2...]])	convert to single precision
DOUBLE(x[,offset[,d1,d2...]])	convert to double precision
COMPLEX(real[,imaginary])	convert to complex
COMPLEX(x,offset,d1[,d2...])	convert to complex
STRING(x1,x2...)	convert to string
FORMAT='(<fortran format>')	use fortran formatting

create zero-filled array of size (d1,d2...)

BYTARR(d1,d2...)	byte
e.g. a=bytearr(10) creates array with indices from 0 to 9	
INTARR(d1,d2...)	short integer
LONARR(d1,d2...)	long integer
FLTARR(d1,d2...)	single precision
DBLARR(d1,d2...)	double precision
COMPLEXARR(d1,d2...)	complex
STRARR(d1,d2...)	string
/NOZERO	keyword to suppress zero-fill

create array filled by index

BINDGEN(d1,d2...)
INDGEN(d1,d2...)
LINDGEN(d1,d2...)
FINDGEN(d1,d2...)
DINDGEN(d1,d2...)
CINDGEN(d1,d2...)
SINDGEN(d1,d2...)

create array, other

MAKE_ARRAY(d1,d2...)	general purpose
REPLICATE(value,d1,d2...)	fill with value
RANDOMN(seed,d1,d2...)	gaussian randomn noise, sigma=1. seed initially set to time of day.
RANDOMNU(seed,d1,d2...)	same, but uniform distribution

files

OPENR,unit,file	open for read
file/keyword used to specify non-default format	
OPENW,unit,file	open for write
OPENU,unit,file	open for update
var=ASSOC(unit,arr[,offset]) associate file with array structure	
defines records to be of same type and dimension as arr	
then, var(0) refers to record 0=first record	
offset=# of bytes to skip in file	
GET_LUN,unit	reserve unit
GET_KBRD(wait)	read one character
READ,var1,var2...	formatted input from keyboard
type and dimension determined by previous definition	
free format assumed	
READF,unit,var1,var2...	similar, from an opened unit
fortran FORMAT='(...)' available	
READU,unit,var1,var2...	unformatted input
FORRD,unit,var1,var2...	unformatted input
POINT_LUN,unit,N	position unit: 0=first byte
PRINT,var1,var2...	formatted output
fortran FORMAT='(...)' available	
PRINTF,unit,var1,var2...	formatted output
WRITEU,unit,var1,var2...	unformatted output
FORWRT,unit,var1,var2...	unformatted output

EOF(unit)	end of file?
FLUSH[,unit1,unit2...]	flush
FSTAT(unit)	get info
CLOSE[,unit1,unit2...]	close
FREE_LUN[,unit1,unit2...]	de-allocate
FINDFILE(file_spec)	find files matching spec
graphics	
SET_PLOT,device	specify graphics device
ERASE[,background_color]	erase screen of current device
XYOUTS,x,y,string	write text
USERSYM,[x,]y	set plotting symbol
PLOT,[x,]y	plot vector
default assumptions	
minimum y-axis value=0, unless some y values<0	
maximum y-axis value=maximum data element	
connect points with solid lines	
...	
assumptions changed by call parameters or system variables	
see also plotting symbols	
see also graphics keyword !P	
PLOT_IO,[x,]y	same, x linear, y logarithmic
PLOT_OI,[x,]y	same, x logarithmic, y linear
PLOT_OO,[x,]y	same, x logarithmic, y logarithmic
OPLOT,[x,]y	plot vector over old axis
PLOT_IO,[x,]y	plot with linear-log axis
PLOT_OI,[x,]y	plot with log-linear axis
PLOT_OO,[x,]y	plot with log-log axis
PLOTS,[x,],y[,z]	plot point
PLOTS,x1,y1,x2,y2	plot vector
in either case coordinate system is from last call to PLOT (data units)	
AXIS[[[,x],y],z]	draw additional axis
CONTOUR,z[,x,y]	contour plots
POLYFILL,x[,y[,z]]	irregular polygon fill
POLYSHADE,vertices,polygons	shaded surface from polygons
SET_SHADING	set light source shading parameters
SHADE_SURF,z[,x,y]	shaded surface from gridded data
SURFACE,z[,x,y]	surface from array, remove hidden lines
CURSOR,x,y[,wait]	get cursor position
DEVICE	device specific functions
/HELVETICA	selects HELVETICA hardware font if !P.FONT=0
EMPTY	empty graphics output buffer
help	
HELP[,x1,x2...]	print info on current session.
information includes type,structure, value (for scalars) of variables.	
may also request information on:	
/BREAKPOINTS,/DEVICE,/FILES,/KEYS,	

/MEMORY,/RECALL_COMMANDS,/ROUTINES,
 /STRUCTURES,/SYSTEM_VARIABLES,
 /TRACEBACK

image processing

DILATE(ar,structure[,x,y]) morphologic dilation
 ERODE(ar,structure[,x,y]) morphologic erosion
 POLY_2D(ar,c,c[,interp[,d1,d2]]) polynomial image warp
 result(i,j)=input(u,v),
 where $u = \sum_{s,t} c(s,t) * x^s * y^t$
 $v = \sum_{s,t} d(s,t) * x^s * y^t$
 interp, if present and NE 0 selects
 bilinear interpolation in place of
 nearest neighbor.
 missing=p pixel values whose u,v are outside image,
 else extrapolated
 POLYFILLV(x,y,sx,sy[,run_length]) subscripts of pixels inside polygon
 ROBERTS(ar) image edge enhancement
 $\text{abs}(\text{ar}(i,j) - \text{ar}(i+1,y+1)) + \text{abs}(\text{ar}(i+1,j) - \text{ar}(i,j+1))$
 SOBEL(image) image edge enhancement
 sum of absolute value of convolution with
 $\begin{matrix} -1 & -2 & -1 & & -1 & 0 & 1 \\ 0 & 0 & 0 & \text{and} & -2 & 0 & 2 \\ 1 & 1 & 1 & & -1 & 0 & 1 \end{matrix}$
 TVCRS[,ON_OFF] cursor on/off?
 TVCRS,x,y position image cursor
 TV,image[,x,y[,channel]] display image, scaled, lower left at
 x,y. Order is bottom up unless set
 !order=1
 TV,image[,position] display image unscaled
 TV,image,cell display in 0-origin cell # of size
 determined by image array
 TVSCL,... same, but scale first by min,max
 TVLCT,v1,v2,v3[,start] load display color tables, starting with index 'start'
 /HLS,/HSV use HLS or HSV color tables instead of RGB
 TVRD(x,y,nx,ny[,channel]) read display memory
 TVZOOM controls hardware zoom

mathematical (scalar)

ISHFT(i,n) shift i by n (<0 for left!).
 ABS(x) absolute value
 SQRT(x) square root
 IMAGINARY(x) imaginary part
 CONJ(x) complex conjugate
 SIN(x) circular functions: use radians
 ASIN(x)
 COS(x)
 ACOS(x)
 TAN(x)
 ATAN(x)

SINH(x)	hyperbolic functions
COSH(x)	
TANH(x)	
EXP(x)	e**x
ALOG(x)	natural log
ALOG10(x)	base 10 log
BESELI(x,n)	I Bessel function
BESELJ(x,n)	J Bessel function
BESELY(x,n)	Y Bessel function
ERRORF(x)	error function
GAMMA(x)	gamma function
GAUSSINT(x)	integral of gaussian
FINITE(x)	number finite?
programming	
BREAKPOINT[,file],index	set/clear breakpoints
BYTEORDER,variable1,...	convert short and long integers between host and network byte order
CHECK_MATH(print_flag[,message_inhibit])	check and clear accumulated math err
DEFINE_KEY,key[,value]	define function key
DEFSYSV,name,value[,read_only]	define new system variable
EXECUTE(string)	compile,execute string
EXIT	exit IDL, close files
FINITE(x)	true if x finite
KEYWORD_SET(x)	test if arg defined and .ne.0
N_PARAMS()	number of non-keyword parameters
N_TAGS(x)	number of structure tags in x
ON_ERROR,n	error recovery handling method
ON_IOERROR,label	I/O error handler
SIZE(x)	size and type of expression
STOP[,x1....xn]	exit program or batch file
STRMESSAGE(err#)	error message text formats
TAG_NAMES(x)	names of tags in structure
WAIT,sec	delay
operating system commands	
CD[,directory]	print or change directory
ENVIRONMENT()	unix environment strings
GETENV(name)	environment string translation
SETENV(environment_expression)	add or change environment string
e.g. SETENV IDL_DEVICE <device> graphic output device	
<device>=NULL(no graphic output),HP(HPGL),PS(Postscript),SUN(SunView),	
TEK(Tektronix terminal), X(X-Window system: not yet on Suns)	
SPAWN[,command[,result]]	spawn child process
SYSTEMTIME(x)	current system time
saving and restoring IDL session	
JOURNAL[,arg]	keep log of interactive session to file
RESTORE[,filename]	load IDL save file
SAVE[,var1...varn]	save variables to file
FILE='filename'	specify file other than idlsave.dat

string processing

STRCOMPRESS(string)	compress or remove whitespace
STRTRIM(string[,flag])	remove leading and trailing blanks
STRLEN(string)	length
STRLOWCASE(string)	convert to lower case
STRUPCASE(string)	convert to upper case
STRMID(string,stcol,length)	substring
STRPOS(str,substr[,pos])	search for substring in string
STRPUT(dest,source[,position])	copy source into destination

windows

WINDOW[,index]	create (index=0 to 9) window 0 may be created automatically
WDELETE[,index]	delete
WSET[,index]	select window
WSHOW[,index[,show]]	expose or hide window
WMENU(strings)	display menu, return response

SYSTEM VARIABLES

Name	Type	Description
!EDIT_INPUT	integer	enables keyboard line editing
!C	long	plot cursor position
!ERR	long	last error code
!JOURNAL	long	unit # of journal output or 0 (read-only)
!ORDER	long	image transfer order (0=bottom-up,1=reverse)
!QUIET	long	0=print informational message, else don't
!PI	float	pi (read-only)
!DPI	double	pi (read-only)
!DTOR	float	pi/180 (read-only)
!RADEG	float	180/pi (read-only)
!VERSION	string	type and version of IDL (read-only)
!PROMPT	string	IDL interactive prompt
!ERR_STRING	string	text of last error (read-only)
!MSG_PREFIX	string	prefix string for error messages
!DIR	string	main IDL directory
!PATH	string	search path for libraries,include files, executive commands, and procedures list of directories, seperated by colons
!D	structure	plotting device (read-only)
!D.N_COLORS	integer	# of color indices available on current graphics device
!P	structure	main plot system parameters
!P.FONT	integer	-1=vector drawn text (default) 0=hardware font
!P.LINESTYLE		
!P.PSYM		
!P.SUBTITLE		
!P.TITLE		
!P.MULTI		vector: (0)=# of plots already on page (1) # of plots horizontally (2) # of plots vertically

default value=scalar 0, meaning 1 plot/page

!P.T matrix generalized transformation matrix: see graphics
and plotting keyword P3T, plotting keyword SAVE

!P.THICK line thickness: see graphics keyword THICK

!X structure axis structure for x

!Y structure axis structure for y

!Z structure axis structure for z

GRAPHICS KEYWORDS

(Proc: c=CURSOR,p=PLOTS,f=POLYFILL,s=TVCRS,x=XYOUTS,

*=corresponds to system variable field)

Name	Proc	Description
------	------	-------------

ALIGNMENT	x	alignment of text baseline
-----------	---	----------------------------

ALIGNMENT=.5 centers text

CLIP	px*	clip rectangle for graphics output
------	-----	------------------------------------

COLOR	fpx*	color index for text,line,polygon fills
-------	------	---

DATA	cpfsx	keyword flag if clipping coordinates are in data coordinates
------	-------	--

DEVICE	cpfsx	coordinates are in device coordinates
--------	-------	---------------------------------------

FILL_PATTERN	f	hardware dependent fill index
--------------	---	-------------------------------

FONT	x*	graphics text font #
------	----	----------------------

LINESTYLE	fp*	line draw style
-----------	-----	-----------------

LINE_FILL	f	fill polygons with lines,not solid or pattern
-----------	---	---

NOCLIP	px*	clipping of vectors and vector-drawn text
--------	-----	---

NORMAL	cpfsx	coordinates in normalized (0-1) coordinate system
--------	-------	---

ORIENTATION	fx	angle in deg from horiz of text baseline and polygon fill lines
-------------	----	---

PATTERN	f	fill pattern pixel array
---------	---	--------------------------

SIZE	x	character size as factor of normal char size
------	---	--

SPACING	f	spacing in cm between parallel line fills
---------	---	---

SYMSIZE	p	size of line marking symbols (1=normal)
---------	---	---

T3D	fpx*	keyword flag indicating generalized transformation matrix !P.T is to be used
-----	------	---

TEXT_AXES	x	plane of vector drawn text for 3D plotting
-----------	---	--

THICK	fpx*	line thickness: overrides !P.THICK
-------	------	------------------------------------

WIDTH	x	returns width of text string to designated variable
-------	---	---

Z	fpxs	z coordinate if Z not present in call
---	------	---------------------------------------

PLOTTING KEYWORDS

(Proc: a=AXIS,c=CONTOUR,o=OPLOT,p=PLOT,s=SURFACE,

*=corresponds to system variable field)

Name	Proc	Description
------	------	-------------

(rotations)

AX	s	(deg) angle of rotation about x towards viewer
----	---	--

AZ	s	counterclocksize angle of rotation about z
----	---	--

XAXIS	a	AXIS procedure applies to X axis, define location
-------	---	---

YAXIS	a	AXIS procedure applies to Y axis, define location
-------	---	---

(colors)

BACKGROUND	all*	background color when erasing
------------	------	-------------------------------

BOTTOM	s	color for bottom surface
--------	---	--------------------------

COLOR all* color index to draw data,axes,annotation
C_COLORS c color for contours

(characters)

CHARSIZE all* overall character size for annotation
[XYZ]CHARSIZE all* size of characters for axis
C_CHARSIZE c character size for contours
SYMSIZE all size of symbols for PSYM
C_ANNOTATION c label drawn on each contour
FONT all* graphics text font
SUBTITLE all* subtitle underneath X axis
TITLE all* main centered title above plot window
[XYZ]TITLE all* axis title

(lines)

C_LINestyle c line style for contours
C_THICK c line thickness for contours
THICK po* thickness of lines between points
LINestyle ops* line style:
2=dashed line
3=dot/dash pattern
PSYM all* symbol for data points instead of line draw:
4=diamond
5=triangle
FOLLOW c if present and <>0, use line following contours,
not cell drawing
SPLINE c contour paths interpolated by cubic splines.
implies FOLLOW. May sometimes cross. Not generally
needed or suggested when dimensions>15
SPLINE=.005 gives default length for SPLINE
SKIRT s draw skirt around given Z value

(contour levels)

C_LABELS c which contour levels to label
LEVELS c vector of contour levels
NLEVELS c # of equally spaced contour levels
PATH_FILENAME c name of file of contour positions

(style)

HORIZONTAL s keyword flag to draw only lines across plot
perpendicular to line of site
LOWER_ONLY s only draw lower surface of object
UPPER_ONLY s only draw upper surface of object
[XYZ]MINOR all* # of minor tick marks
[XYZ]TICKNAME all* annotations of each tick, up to 30 elements in array
TICKLEN all* length of tics, as fraction of window size
[XYZ]TICKS all* # of major tic intervals to draw
[XYZ]TICKV all* data value for each tick, up to 30 elements in array

[XYZ]TYPE ap logarithmic axis if <>0
 ZAXIS acs specifies Z-axis existence for CONTOUR, position
 for SURFACE
 ZVALUE poc specify Z coordinate in normalized units (0-1)
 of axis and data output from PLOT, OPLOT, CONTOUR

NSUM po # of data points to average when plotting
 POLAR po keyword, <>0 means make polar plots
 [XYZ]STYLE all* specifies axis tic value rounding, selection of
 a box axis, etc.

(clipping, windows, units)

CLIP all* rectangle coordinates to clip graphics output
 DATA all keyword flag indicating CLIP coordinates are
 in data units
 DEVICE all POSITION and CLIP coordinates are in device units
 [XYZ]MARGIN all* 2 element array for left(bottom), right(top) of
 plot window in character size units
 [XYZ]RANGE all* range of axis: 2 element vector
 POSITION all* direct specification of plot window
 MAX_VALUE c ignore data points over this value
 NOCLIP cop* keyword flag to surpress plot clipping
 YNOZERO ap do not set y axis min to 0, even if YRANGE and
 !Y.RANGE not specified and all Y are non-negative.
 NODATA all draw axes, titles, annotation, not data
 NOERASE all* do not erase screen or page
 NORMAL all keyword flag: CLIP and/or POSITION coordinates
 are in normalized coordinate system, from 0 to 1
 T3D all* flag to use generalized transformation matrix !P.T
 SAVE as save 3d to 2d transformation matrix from SURFACE
 in !P.T

USER'S LIBRARY ROUTINES (in main user library directory)

Name	Description
------	-------------

(colors)

ADJCT	adjust color table contrast function using mouse
C_EDIT	interactive color table creation, HSV or HLS systems using 3 bars
COLOR_EDIT	interactive color table creation, HSV or HLS systems using color wheel and bars
HIST_EQUAL_CT,image	histogram equalization of color table from region. With no parameters, mark region with mouse
HLS	hue/lightneww/saturation color system color tables sprals through HLS space
HSV	hue/saturation/value color system color tables spirals thru single ended HSV cone
HSV_TO_RGB	convert HSV color system to red/green/blue
LOADCT[,index]	load standard color tables (0-15; obtain menu with no index)

MODIFYCT save modified color tables in standard color table file.
 permanently replaces colors used by LOADCT
 MULTI replicates current color table, enhancing contrast
 ONLY_8BIT initialize sun display for systems without monochrome planes
 PALETTE interactively construct color tables using RGB
 PSEUDO color tables based on LBH (Lightness,Hue,Brightness) color system
 RGB_TO_HSV convert RGB system to HSV
 STRETCH,lo,hi linear color table contrast enhancement to match min=lo, max=hi
 TEK_COLOR load tektronix 4115 default color map
 (documentation)
 DOC_LIBRARY,'name' documents users' library procedures; name=* for all
 (filter)
 DIGITAL_FILTER calculate coefficients of non-recursive digital filter
 HANNING window function for FFT filtering
 (fit)
 CURVEFIT non-linear least squares fit
 FUNCT sample function for CURVEFIT
 GAUSSFIT fits sum of gaussian and quadratic
 POLY_FIT(x,y,n,yfit...) polynomial least squares fit
 result contains n+1 fit coefficients
 POLYFITW polynomial weighted least squares fit
 REGRESS multiple linear regression
 SURFACE_FIT polynomial fit to a surface
 SVDFIT general least squares fit using SVD
 COSINES sample function used by SVDFIT
 (graphics)
 DEMO_HP produce HP-GL demo plots
 ERRPLOT overplot error bars
 MENUS crudge menu facility used by ADJCT
 OPLOTBAR overplot bar graphs
 OPLOTERR overplot data with error bars
 PLOT_FIELD plot 2d field using arrows
 PLOTBAR bar graph
 PLOTERR plot data with error bars
 POLYCONTOUR fill countour paths with solid colors
 SCALE3D scales 3D unit cube into display area
 SET_SCREEN set plot window size and position (VMS compatibility)
 SET_VIEWPORT set plot window size and position (VMS compatibility)
 SET_XY set X and Y limits (VMS compatibility)
 SURFR duplicate rotation, translation and scaling of SURFACE
 T3D various 3D transformations
 THREED plot 2D array using pseudo 3D plot
 VEL plot vector flow field with streamlines
 VELOVECT plot vector flow field
 ZOOM zoom mouse-selected portion of display
 (images)
 DEFROI define region of interest by mouse
 FLICK flicker between 2 images

HIST_EQUAL	histogram equalization
IMAGE_CONT	overlay image and its contour plot
LEEFILT	Lee filter for images
MOVIE	display loop of images
POLYWARP	polynomial spatial warp, determine coeff
PROFILE	extract values along a line
PROFILES	interactively plot row or column image lines
RDPIX	interactively read pixels
READ_SRF	read sun raster format files
ROT	rotates/magnifies image using nearest neighbor sampling
	Slower than ROTATE, but can be any angle
ROT_INT	rotates/magnifies image using bilinear interpolation
SHOW3	displays images with combination image, using SURFACE and CONTOUR
WRITE_SRF	write sun raster format files
(interpolation)	
BILINEAR	bilinear interpolaton
CONGRID	resample image to specified dimensions
INTERPOL	linearly interpolate from vectors
SPLINE	cubic spline
(math functions)	
CHEBYSHEV	forward/reverse Chubyshev polynomial expansion
CORRELATE(vec1,vec2)	simple correlation coefficient
	-1 or 1=perfect correlation, 0=uncorrelated
DERIV	derivative using 3 point lagrangian interpolation
JULDAY	julian day number from month/day/year
POLY	polynomial
POLY_AREA	area of polygon
STDEV	standard deviations
(matrices and vectors)	
CROSSP	vector cross product
DETERM	determinant of square matrix
DIST	create matrix with each element set to its distance to origin
EXTRAC	emulate obsolete EXTRAC system function
HILBERT	hilbert matrix
REVERSE	reverse order of vector or array
(system)	
POPD	pop PUSHHD/POPD stack and make popped name current directory
PUSHHD	push directory name to PUSHHD/POPD stack
PRINTD	print PUSHHD/POPD directory name stack
SCRABBLE	solves Scrabble puzzles
SETUP_KEYS	define escape sequence keys for common keyboards
Executive Commands	

Name Function

.RUN <files> compile and run user procedure
 if no <files>, take from keyboard
 .PRO is the normal file suffix
 user procedures end with END command

-t sends listing to terminal

-l sends listing to name.lst file

Note: procedures and functions will be automatically compiled if their source code.pro file is in a !PATH directory

.RNEW same, but erases main program variables first

.CON continues execution of stopped program

.GO executes previously compiled main program

.SKIP skip next statement, single step

.S or .STEP execute single step

COORDINATE SYSTEMS

Data: last PLOT call

Image Display: (x,y) x=rightwards, y=upwards; (0,0) to (ncol-1,nrow-1)

COLOR SYSTEMS

RGB=Red Green Blue

all 0 to 255 on 8 bit display, 0,0,0=black, 255,255,255=white

HLS=Hue Lightness Saturation (Otswald color system)

hue: 0=red,120=green,240=blue

lightness,saturation: 0-1

HSV=Hue Saturation Value

value: 0-1

LHB=Lightness hue brightness

COLOR TABLE INFORMATION IS KEPT IN

COMMON colors,r_orig,g_orig,b_orig,r_curr,g_curr,b_curr

~_orig=previous value, before user library function call

~_curr=current value, after user library function call

set plot=postscript

cat printfile to printer from unix

...

To print image from <filename>

z=intarr(25,15)

openr,1,'<filename>'

readu,1,z

set_plot,'ps'

device,filename='temp.ps'

!order=1 so images written bottom done

tvsc1,rebin(z,500,300)

device,/close_file

set_plot,'sun'

unix: cat temp.ps>/dev/ttya

Signal Processing

Butterworth (nth-order) Butterworth filter (in frequency domain)

$$1/(1+f/f_0)^{2n}$$

f_0 =half strength frequency, n might be 2.

triangular kernal

1 2 1

2 4 2

1 2 1

(opinions expressed are mine alone)

Mitchell R Grunes (grunes@nrlvax.nrl.navy.mil)

Allied-Signal Tech. Serv. / Naval Research Lab
