

## 4. Displaying Data as Line Plots

- Customizing a line plot with more than 60 keywords, examples:

```
(x_vec = FINDGEN(200)/20  y_vec = sin(x_vec))
```

```
PLOT, y_vec, CHARSIZE=1.5, PSYM=1, $  
  XRANGE=[100,200]
```

; 1 argument (y: subrange of y\_vec), PSYM-PlotSYMBOL, see online help

```
PLOT, x_vec, y_vec, TICKS=10, XTICKFORMAT='(F8.2)'
```

; 2 arguments! (x + v), x axis customized

; (F8.2)- 8 characters, with 2 places after the decimal point

- Plotting multiple datasets

```
y1 = y_vec/EXP(x_vec)
```

```
OPLOT, x_vec, y1, COLOR = 150, IINestyle = 2. $  
  THICK = 2
```

- Customising the axes

- axis types:

[XYZ]STYLE (XSTYLE,YSTYLE,ZSTYLE) keywords:

each option encode in a bit, s. online help

**PLOT, FINDGEN(180), XSTYLE=(1 +8)**

; exact range + no x-box

- Customising the axes:

- tick intervals:

[XYZ]TICKS : number of major tick intervals

[XYZ]MINOR : number of minor tick intervals

TICKLEN and [XYZ]TICKLEN:

tick length between -1 and 1

- Use of multiple axes: *AXIS* procedure

```
PLOT, x_vec, y_vec, ystyle=8      ;ystyle=8: no box  
AXIS,10, /YAXIS,YRANGE=[-5,5], $  
    /SAVE, COLOR=160  
    ;draw an additional y-axis + save new data coord.  
OPLOT,x_vec, 8*y1, LINESTYLE=2, COLOR=160
```

- IDL has 3 coordinate systems !:

DATA : established by PLOT,SURFACE

DEVICE : system of the graphics device(pixels)

NORMAL: range from 0 ...1 in the plot window

- Use the system variables !X, !Y (+ !Z) to change the default values für the axis settings.

Structure fields correspond to keywords:

**!X.STYLE = 1** ;new default: exact axis range

**!Y.RANGE = [0, 5]** ;new default y range

- Drawing lines (or plotting points) with *PLOTS*, *xcoord\_vec*, *ycoord\_vec*, */data*  
*xcoord\_vec*, *ycoord\_vec*: providing the x-v coordinates of the points to be connected.

- Example:

`PLOT,x_vec,y_vec ;establish data coordinates!`

`x=[4,6,6,4,4]`

`y=[0.35,0.35,0.5,0.5,0.35]`

`PLOTS, x, y,/DATA, color=200 ; plot a box`

- Annotation keywords:

TITLE, [XYZ]TITLE, CHARSIZE, [XYZ]CHARSIZE

Create a line plot with a title, a y-title and a larger charsize (e.g.: 2):

Explicit labels for tick marks with keywords:

[XYZ]TICKNAME ; set to astring array

[XYZ]TICKFORMAT ; define a function for tick labels

- Axes with date/time labels
    - New with IDL 5.4: TIMEGEN
      - returns an array of time values (double "Julian dates" )
      - contains several keywords to provide specific date/time data generation
- time = TIMEGEN(200, UNITS = 'Seconds', \$  
START = JULDAY(04, 23, 2002,10,20,30))**  
;time vector starting on April, 23rd, 1 0:20,...  
;also try UNITS = 'Days' ...



- Axes with date/time labels, example:  
dummy = LABEL\_DATE(DATE\_FORMAT=['%I:%S '])  
;Label\_date specifies axis format ("Minutes:Seconds")

```
PLOT, time, y_vec, $  
XTICKUNITS = 'Time', $ ;specify axis type  
XTICKFORMAT = 'LABEL_DATE', $ ;use internally  
;stored result of LABEL_DATE  
XTICKINTERVAL = 0.5 ;specify tick interval
```

(see file w9-timeaxis.pro)

- Adding text to any graphics with XYOUTS:

```
PLOT, x_vec, y1
```

```
XYOUTS, 5, 0.3, 'TEXT', CHARS=3 ;/DATA default!
```

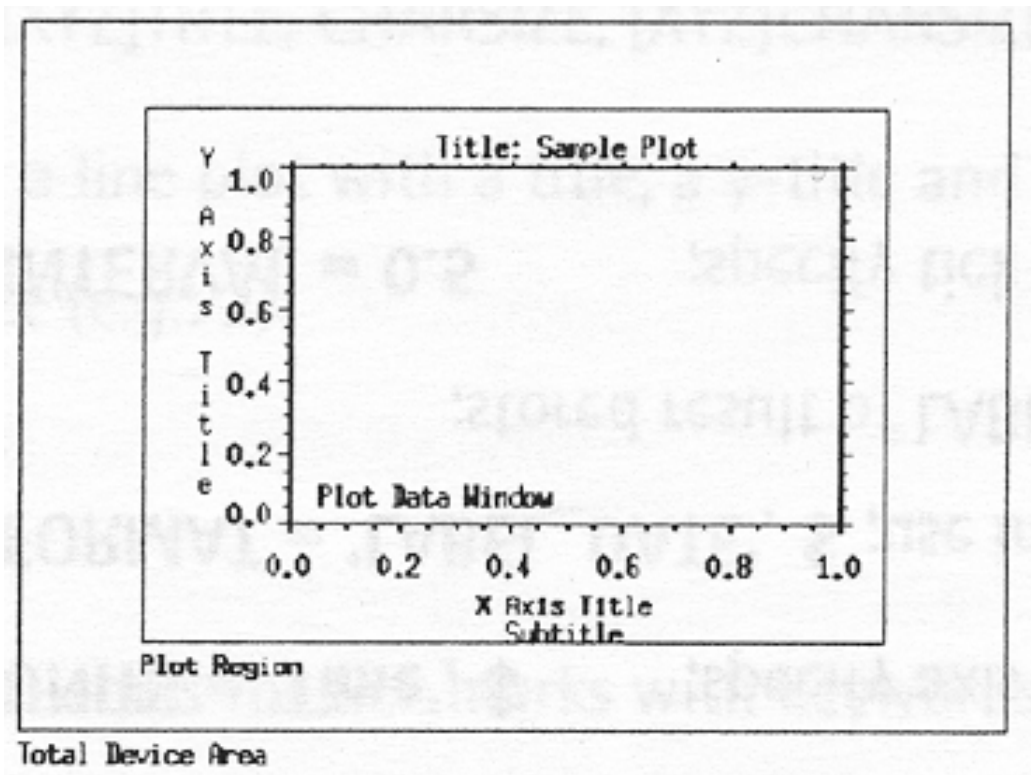
(line6)

Exercise:

Position text in the center of the plot window:

```
XYOUTS, ... , ... .' ...'. INORMAL,ALIGNMENT=0.5
```

- Positioning a PLOT (a SURFACE ... ) in the window:



- Plot Data Window in normalized coord. . .  
PLOT, x\_vec, y\_vec, POSITION=[0.2, 0.2, 0.8, 0.9]  
!P.POSITION=[0.2,0.2,0.8,0.9] ; or system variable !P  
;coord. of lower left + upper right corner!

- Multiple plots in one display window

**!P.MULTI=[0,3,2]**

    ; IP.MULTI(1)=3:        3 columns

    ; !P.MULTI(2)=2:        2 rows

...                    ;create some plots

**!P.MULTI=0**              ; reset

- PLOTTING in 3D space, example:

**SURFACE,DIST(40),/NODATA,/SAVE**

; /SAVE: save 3d to 2d-transformation matrix

; /NODATA don't plot dummy data

**PLOT, x\_vec, y\_vec. /T3D,ZVALUE=1.0,/NOERASE**

; /T3D : use transformation matrix !P.T to create  
a planar plot at ZVALUE= 1.0 ( norm. coord.)

- IDL library routines for setting up

a transformation matrix similar to SURFACE

**SURFR, AX=45**

**PLOT, x\_vec , y\_vec, CHARSIZE=2,/T3d**

**; ZVALUE=0 default !**

- Change your 3D coordinate system with T3D

**T3D, / YZEXCH** ;exchange y and z axis

**PLOT, x\_vec, y\_vec, /T3D, /NOERASE, ZVALUE=1**

exchange of x and y with keyword !XYEXCH ,  
exchange of x and z with keyword !XZEXCH

- library routine to plot  $z=f(x,y)$  in a 3d box:

**plot\_3dbox, x\_vec, y\_vec, (x\_vec+y\_vec), \$**  
**/xz\_plane, /yz\_plane**

- LIVE\_PLOT: Interactiv tool created with the IDL object graphics system:

                  ;example: plot with 2 (or more) curves

*X=*FINDGEN(200)

**LIVE\_PLOT, sin(x/10), cos(x/20)** ;select objects and  
                                  ;open the properties dialogs

**yNew = 0.01 \* x \* sin(x/10)**                   ;add another curve:

**LIVE\_OPLOT, yNew**

                          ;;add other graphics objects. e.g.

**live\_text, 'Live\_Tool Test'**

(wg\_livetest.pro)



## 4. Procedures and Functions

- Call a procedure or function (IDL or user-written):  
***pro\_name,p1,p2...., KEYWORD=pk1***  
***result=func\_name(p1,p2,...KEYWORD=pk 1]***

p1,p2.... : positional parameters (optional),  
must appear in a particular order

pk1.... : optional keyword variable,  
- appear in arbitrary order  
- shortcuts can be used

- IDL procedure (function) file template,

(filename with extension ".pro" ,e.g. pro\_name.pro) :

***PRO pro\_name, p1,p2,.....,KEYWORD=pk1....***

***(FUNCTION func\_name,p1,p2,... ,KEYWORD=pk1..... )***

      ;p1,p2...       positional parameters

      ;pk1....       keyword parameter

.....

.....       ;IDL Code

***(RETURN, result       ;functions only)***

***END***

- Primitive example (file "quad.pro"):

```
FUNCTION quad,x  
  r = x^2+x+400  
  RETURN, r  
END
```

- ;call "quad" within an IDL PRINT- command

- Parameter passing:
  - variables are passed by reference, they can be modified in the calling routine: input and output parameters
  - constants, subscripted variables and structure tags are passed by value: input parameters
    - Change "quad.pro": overwrite the input argument

- **Compiling and Debugging of IDL routines**
  - A routine is compiled automatically before the first execution, when the file "*name.pro*" is in the IDL path or in the current working directory.
  - Compile a procedure or function:
    - a) *IDIDEMenu: Run->Compile...*
    - b) 'Executive' command on the command line  
**.COMPILE name**

- Setting the IDL Path:

*a) IDLDEMenu: file->Preferences*

*b) IDL command:*

**!path = expand\_path(' +/export/home/wg:') + !path**

- Debugging of procedures + functions:
  - a) see *IDiDE* Menu *Run* and toolbar buttons
    - Run->Step Info* (execute 1 statement)
    - Run->Set Breakpoint* (stop execution + ...)
    - .....
  - b) Use IDL executive commands, examples:
    - `.step` ; execute 1 statement
    - `.step 10` ; execute 10 statements
    - `.skip 10` ; skip over 10 statements
    - .....

- Control statements in IDL programs:
  - *IF ( a EO b) THEN ... ELSE*
  - *FOR i=0,9 DO ...*
  - *WHILE ( NOT EOF (lun) ) DO ...*
  - *REPEAT ... UNTIL (b GT a)*
  - *CASE test OF ...*
  - *SWITCH test OF ... (new!)*



- Statement blocks with BEGIN and end with  
END,ENDIF,ENDFOR... :

*IF ( true ) THEN BEGIN*

*. .*

*. .*

*ENDIF ELSE BEGIN*

*. .*

*. .*

*ENDELSE*

- *CASE test* of ; 'test' : IDL variable  
0: ...  
1: *BEGIN*  
.....  
*END*  
*ELSE:*  
*ENDCASE*

- *GOTO, STOP* ; 'stop' : IDL label  
...  
*STOP:*  
.....

## 4.2 Using keywords and optional parameters

- Determine the number of parameters used in a call:  
*number=N\_PARAMS( )*
- Determine if a keyword is defined:
  - a) *defined=N\_ELEMENTS(KEYWORD-VARIABLE)*  
*; returns the number of elements*
  - b) *defined=KEYWORD\_SET(KEYWORD-VARIABLE)*  
*; Used with toggle keywords:*  
*; defined=1 (TRUE), =0 (FALSE)*

- Simple example: function "multip"

```
FUNCTION multip, value, times, ADD=add  
  if {N_ELEMENTS{add} GT 0} then begin  
    value = value+add  
  endif  
  if N_PARAMS() eq 1 then begin  
    RETURN, value*2  
  endif  
  RETURN, value*times  
END
```

- Exercise:

Write an IDL-procedure, that draws a shaded surface with a wire mesh overplotted. IF desired, make a contour at the top of the plot and select a color index for the surface and the contour plots:

***PRO mysurface, data, CONT=cont, COLOR=color***

- Keyword inheritance with the formal keyword parameter `"_EXTRA"`

(used e.g for wrapper function)

simple example:

```
pro myplot, data, _EXTRA=extra
```

```
HELP, extra, /str ; examine whats behind
```

```
; _EXTRA
```

```
PLOT, data, _EXTRA=extra
```

```
end
```

- Call the procedure "myplot" with different keywords of IDL's *PLOT* command
- Exercise: Insert the *\_EXTRA* -keyword into *MYSURFACE.PRO*  
- or into IDL's library routine *SHOW3.PRO*

(copy show3.pro to your own directory !)