IDL Tutorials

Day 2

Henry (Trae) Winter
winter@physics.montana.edu
http://solar.physics.montana.edu/winter/idl_tutorial/2003/index.html

Today's Topics

If you didn't get everything from yesterday,

DON'T PANIC!

Refer to notes, use help screens, ask me (That's what I'm here for!)

- Review journal procedure
- Using Functions and Procedures
- Review xdoc
- Macros or Scripts
- Plotting to screen and to files
- Using the Save & Restore procedures
- Review structures

Journal Procedure

- The journal procedure allows you to record everything you entered in the command line, into a text file
- Syntax:

  IDL>journal, “text_filename.ext” ;opens the file and starts recording

  IDL >stuff ;command line statement

  IDL >more stuff ; “ “

  IDL >journal ; closes the file and stops recording
- Remember: IDL is not case sensitive but unix is:

  IDL > help, var = IDL > Help, Var = IDL > HELP,VAR

  IDL > journal, “text_filename.ext”

  ≠

  IDL > journal, “Text_Filename.EXT”
- The journal file can be opened with a text editor such as emacs or nedit:

  computer_name> emacs text_filename.ext

  computer_name> nedit text_filename.ext

Functions and Procedures

- IDL is based on two types of basic program: Functions and Procedures
- Function syntax

  result=function_name(parameter, KEYWORD=SET, …)

  Examples: size, n_elements, sin, cos, reverse, where

- Procedure Syntax

  procedure_name, param1, param2, KEYWORD=SET, …

  Examples: help, print, open, printf, plot, set_plot, device, save,restore

- The exact syntax for procedures or functions can be found using either the help menu, xdoc, or the doc_library procedure
- Functions can be placed anywhere that you put a value or variable

xdoc

- xdoc allows you to graphically browse the library of Solarsoft routines
- Just type in xdoc at the IDL command prompt.
- If you know the name of the procedure/function you can type it in the field
- Usually you don't know the name of a particular procedure so ssw makes it "easy" for you to search
- xdoc example

Commonly used Procedures

- print,x,y,"printing stuff" ; Prints variables, the results of functions

  >print, sin(x)

- plot ; Plot variables to the current graphics device (usually the screen)

  >plot, x, sin(x), TITLE="Sine Plot", XTITLE=" X ",$

  YTITLE="Sine of X"

- set_plot; Tells idl where to send plots to. Screen (Xdisplay) is the default

  >set_plot, "x" ;sends the output to the x display

  >set_plot, "ps" ;sends the output to a postscript file with default name of idl.ps. Postscript files can be read with ghostview.

  unix_prompt> gv idl.ps

- device ; Changes the properties of the device currently enabled by set_plot. For an example see example.pro or the help menu

Commonly used Functions

- You have already seen functions to create arrays (intarr, dblarr, fltarr, etc.)
- Many math operators are functions

  IDL>y=sin(x) IDL>y=cos(x) IDL> y=exp(x) ;Natural exponent

- where allows you to pick out the index of an array that meets some criterion.

  IDL>index=where(new_array eq 15.)

  More on Boolean operators later

- replicate makes copies of variables, arrays, structures and so on…

  IDL>stars=replicate(stars,15)

- dialog_pickfile allows you to interactively pick out file names

  IDL>file_name=dialog_pickfile()

Macros (Scripts)

- Simplest form of program
- In the editor window write commands just like you would at the command prompt
- At the very end type END so that IDL knows when the program is over
- Save the program with some unique name
- At the command prompt type either:

  >.run program_name

  >.rnew program_name

- See example.pro for an example

Save & Restore

- Save procedure allows you to save variables to a file

Example:

save, variable1, variable1,FILE=“file.sav”

- Restore allows you to restore saved variables to a session

Example:

restore, “file.sav”