Analysis Methods in Atmospheric and Oceanic Science

AOSC 652

Introduction to IDL

Week 8, Day 1

17 Oct 2016

Introduction to IDL

Interactive Data Language was created in the mid to late 1970's for the space program.

First IDL license sold to NASA's Goddard Space Flight Center in 1979.

IDL is a nice tool for data visualization and is useful for processing large amounts of data.

Like MATLAB, has built in library of functions.

To open the IDL help utility: at the IDL command line type ? and hit enter

AOSC 652: Analysis Methods in AOSC

First, need to modify .cshrc file.

Near the top of your list of "setenv" commands, please add

setenv IDL_DIR /usr/local/exelis/idl83 **May not need to do this!**
setenv IDL_PATH +/usr/local/exelis/idl83/lib

after the line LM_LICENSE_FILE...

setenv IDL_STARTUP /homes/metogra/userid/idl/idl_startup_aosc652.pro

Next, in your home directory, create a sub-directory called *idl*

Copy the files:
~tcanty/IDL/idl_startup_aosc652.pro
~tcanty/IDL/read_file.pro
into your new idl directory

source your .cshrc file for these changes to take effect.

AOSC 652: Analysis Methods in AOSC

You should now be able to run IDL.
To do so, type idl at the command prompt.

Type: x=4 and hit return
Type: y=2 and hit return

To see the value of x, type: print, x

Let's try some basic math.

Type: print, x/y

Hopefully, the result will print out as 2

Now set x to equal 5 and repeat x/y

AOSC 652: Analysis Methods in AOSC

You should now be able to run IDL.
To do so, type idl at the command prompt.

Note: we'll provide an overview of syntax here

Between now and Wed, please visit

http://www.exelisvis.com/docs/GOTO.html
http://www.exelisvis.com/docs/FOR.html
http://www.exelisvis.com/docs/FOREACH.html
http://www.exelisvis.com/docs/IF___THEN___ELSE.html
http://www.exelisvis.com/docs/WHILE___DO.html

to study IDL syntax

AOSC 652: Analysis Methods in AOSC

To find out information regarding variable type, use the IDL help command

```
help, x

X           FLOAT     =       5.00000

help,y

Y           INT       =       2
```

To set a variable as double precision, place a "d" after the number

```
x=5.0d

help, x

X           DOUBLE    =       5.0000000
```

AOSC 652: Analysis Methods in AOSC

To convert between variable types, use:

z1 = double (x)

z2 = float (x)

z3 = long (x)

for z1, z2, and z3 to represent value of x in either single precision, double precision, or integer

That's right … IDL uses LONG: http://www.exelisvis.com/docs/LONG.html for integer conversion

AOSC 652: Analysis Methods in AOSC

Generally, you will be creating figures and doing calculations as part of an IDL program, not at the command line.

Please copy ~tcanty/AOSC652/2016/WEEK_08/test*.pro

```
;Test1.pro
      X=5.
      Y=2.
      Z1=X/Y
      Z2=X^2/Y

      print,format='("X=",F7.2," Y =", F7.2)',x,y
      print,format='("X/Y =",F7.2," X^2/Y =",F7.2)',Z1,Z2
end
```

There are a couple of ways to run a program in IDL

.run will compile and run the code
.rnew will erase all main variables in memory and compile and run the code

AOSC 652: Analysis Methods in AOSC

Generally, you will be creating figures and doing calculations as part of an IDL program, not at the command line.

Please copy ~tcanty/AOSC652/2016/WEEK_08/test*.pro

```
;Test1.pro
      X=5.
      Y=2.
      Z1=X/Y
      Z2=X^2/Y

      print,format='("X=",F7.2," Y =", F7.2)',x,y
      print,format='("X/Y =",F7.2," X^2/Y =",F7.2)',Z1,Z2
end
```

There are a couple of ways to run a program in IDL

To run the program, type: .run test1

AOSC 652: Analysis Methods in AOSC

Now we'll look at how to do loops

Without exiting IDL,

.run test2.pro

This program demonstrates how to set up arrays:

```
X2=[5.,6.,7.,8.,9.,10.0]
```

and loops

```
for i=1,x do begin
      ........
endfor
```

AOSC 652: Analysis Methods in AOSC

Now we'll look at how to do loops

Without exiting IDL,

.run test2.pro

This program demonstrates how to set up arrays:

```
X2=[5.,6.,7.,8.,9.,10.0]

and loops
    for i=1,x do begin
        ........
    endfor
```

First problem:

In Fortran and Matlab, our indices go from *1* to *N*

In IDL, indices go from *0* to *N-1*

AOSC 652: Analysis Methods in AOSC

Now we'll look at how to do loops

Without exiting IDL,

.run test2.pro

This program demonstrates how to set up arrays:

```
X2=[5.,6.,7.,8.,9.,10.0]

and loops
    for i=1,x do begin
        ........
    endfor
```

Second problem:

IDL will retain information from variables used in previous programs or from the command line if you do not use .rnew.

A closer look at how to do loops

```
x1=fltarr(5)

Y1=[1. ,2. ,3. ,4., 5.]
Y2=[6. ,7. ,8. ,9., 10.]

for i=0,4 do begin
     x1(i)=y1(i)*y2(i)
endfor
```


AOSC 652: Analysis Methods in AOSC

A closer look at how to do loops

```
x1=fltarr(5)

Y1=[1. ,2. ,3. ,4., 5.]
Y2=[6. ,7. ,8. ,9., 10.]

for i=0,4 do x1(i)=y1(i)*y2(i)
```

AOSC 652: Analysis Methods in AOSC

A closer look at how to do loops

Y1=[1. ,2. ,3. ,4., 5.]
Y2=[6. ,7. ,8. ,9., 10.]

x1=y1*y2

AOSC 652: Analysis Methods in AOSC

You can save your current session in IDL

To create an IDL save file,

save,file='idl1.sav'

Now, exit and restart IDL

Type:
restore,file='idl1.sav‘

AOSC 652: Analysis Methods in AOSC

Input/Output

Copy the file
~tcanty/AOSC652/2016/WEEK_08/random_integers_1_10000.dat

Let's take a look at test3a.pro

```
filename='   '
read,' Enter filename ', filename

openr,1,filename
readf,1,ncol,nhead
cdum=' '
for i=0,nhead–2 do begin
      readf,1,cdum
endfor
```


AOSC 652: Analysis Methods in AOSC

Input/Output

Copy the file
~tcanty/AOSC652/2016/WEEK_08/random_integers_1_10000.dat

Let's take a look at test3a.pro

```
count=0
variables=fltarr(ncol,1.e6)
variables(*,*)= –999.00

tmp=fltarr(ncol)
while (not eof(1)) do begin
      readf,1,tmp
      variables(0:ncol-1,count)=tmp
      count=long(count)+1
endwhile
      variables=variables(*,0:count–1)
close,1
```

Input/Output

Copy the file
~tcanty/AOSC652/2016/WEEK_08/random_integers_1_10000.dat

Let's take a look at test3a.pro

```
openw,2,'sorted_integers_10000.dat'

printf,2,'3,3'
printf,2,'Sequence, Integer Value, Sorted Integer'
printf,2,'Integers 1 to 10000, Randomly Sorted Using Random Number
Generator Seed –99'

for i=0L,sz–1 do begin
      printf,2,variables(0,i),variables(1,i),sorted_data(i),format='(3(1x,I7))'
endfor

close,2
```


Input/Output

To sort data in IDL, use the "sort" command

```
xsort=sort(variables(1,*))
```

"sort" outputs an array of indices that allows you to re-order the original array.

```
sorted_data=variables(1,xsort)
```

AOSC 652: Analysis Methods in AOSC

Input/Output

Now take a look at test3b.pro

What is the major difference between test3a.pro and test3b.pro?

AOSC 652: Analysis Methods in AOSC

Input/Output

Now take a look at test3b.pro

What is the major difference between test3a.pro and test3b.pro?

Earlier, we added a program to your an idl directory called read_file.pro.

Just as load_header_data.m read files into MATLAB, read_file.pro reads "Ross & Tim" formatted files into an IDL program.

read_file,filename,variables,headers

"variables" contains the data contained in the file
"headers" contains the names of the data in each column

AOSC 652: Analysis Methods in AOSC

Manipulating data:

Please copy

~tcanty/AOSC652/2016/WEEK_08/plot_temp.pro
~tcanty/AOSC652/2016/WEEK_08/beltsville.dat
~tcanty/AOSC652/2016/WEEK_08/beltsville_10yr_mean_sd.dat

What does the program test4.pro do?

AOSC 652: Analysis Methods in AOSC

Manipulating data:

The “where” command, like the “find” command in MATLAB, allows you to find the indices of an array that satisfies a given argument.

```
p1=where(month eq 12 and tave ne -999.00)

dec_mean=mean(tave(p1))
dec_sdev=stddev(tave(p1))
```

AOSC 652: Analysis Methods in AOSC

Manipulating data:

The "where" command, like the "find" command in MATLAB, allows you to find the indices of an array that satisfies a given argument.

```
p1=where(month eq 12 and tave ne -999.00)

dec_mean=mean(tave(p1))
dec_sdev=stddev(tave(p1))
```

What does the rest of this code do?

AOSC 652: Analysis Methods in AOSC

Data visualization:

First,

Copy ~tcanty/IDL/symbols.pro
~tcanty/IDL/oploterror.pro
~tcanty/IDL/colorbar1.pro

into your idl directory.

These programs will make it easier to plot different types of symbols, error bars, and a color bar.

Copy ~tcanty/IDL/generic.pro into your working directory

This program can be thought of as a customizable "stncl" file that you can use to plot data.