

Analysis Methods in Atmospheric and Oceanic Science

AOSC 652

Getting to know FORTRAN:
Input/Output, Data Sorting, Simple Statistics
Day 1

19 Sep 2016

IF Statements

Old style:

if (i.gt.2) j=k+1

! Page 144 of reading

IF Statements

New style:

```
if (i.gt.2) then
    j=k+1
endif
```

IF Statements

Another new style:

```
if (i.gt.2) then
    j=k+1
else
    j=0
endif
```

IF Statements

Yet another new style:

```
if (i.gt.2) then
    j=k+1
else if(i.ge.0.and.i.le.2) then
    j=3
else
    j=0
endif
```

See:

<http://www.cs.mtu.edu/~shene/COURSES/cs201/NOTES/chap03/else-if.html>

or

http://en.wikibooks.org/wiki/Fortran/Fortran_control

for more details

DO Statements

Old style:

```
do 10 ilat=1,nlat      ! Page 100 of reading
    ozone_pt=ozone_zmm_array(ilat)
    if (ozone_pt.le.0.) ozone_pt=-999.0
10  continue
```

DO Statements

New style:

```
do ilat=1,nlat
    ozone_pt=ozone_zmm_array(ilat)
    if (ozone_pt.le.0.) ozone_pt=-999.0
enddo
```

Nested DO Statements

```
do ilat=1,nlat
  do ilon=1,nlon
    ozone_pt=ozone_glb_array(ilat,ilon)
    if (ozone_pt.le.0.) ozone_pt=-999.0
  enddo
enddo
```


Nested DO Statements

```
do ilat=1,nlat
  do ilon=1,nlon
    ozone_pt=ozone_glb_array(ilat,ilon)
    if (ozone_pt.le.0.) ozone_pt=-999.0
    if(ilon.eq.nlon) then
      j=1
      write(*,*)'back where we started'
    endif
  enddo
enddo
```

Nested DO Statements

```
do ilat=1,nlat
  do ilon=1,nlon
    ozone_pt=ozone_glb_array(ilat,ilon)
    if (ozone_pt.le.0.) ozone_pt=-999.0
    if(ilon.eq.nlon) then
      ilat=1
      write(*,*)'back where we started'
    endif
  enddo
enddo
```

AOSC 652: Analysis Methods in AOSC

Subroutines and functions

Copy files:

`~rjs/aosc652/week_04/demo_main.f`

`~rjs/aosc652/week_04/demo_sub_average.f`

`~rjs/aosc652/week_04/demo_fn_avrage.f`

to your directory.

How can we combine `demo_main.f`, `demo_sub_average.f`, and `demo_fn_avrage.f` into a FORTRAN executable?

AOSC 652: Analysis Methods in AOSC

Subroutines and functions

Copy files:

`~rjs/aosc652/week_04/demo_main.f`

`~rjs/aosc652/week_04/demo_sub_average.f`

`~rjs/aosc652/week_04/demo_fn_avrage.f`

to your directory.

How can we combine `demo_main.f`, `demo_sub_average.f`, and `demo_fn_avrage.f` into a FORTRAN executable?

Have a look at `~/pgif77`

AOSC 652: Analysis Methods in AOSC

Subroutines and functions

Copy files:

```
~rjs/aosc652/week_04/demo_main.f  
~rjs/aosc652/week_04/demo_sub_average.f  
~rjs/aosc652/week_04/demo_fn_avrage.f
```

to your directory.

How can we combine **demo_main.f**, **demo_sub_average.f**, and **demo_fn_avrage.f** into a FORTRAN executable?

Have a look at `~/.pgif77`

```
pgf77 -c Msave -Mbounds -traceback demo_main.f  
pgf77 -c Msave -Mbounds -traceback demo_sub_average.f  
pgf77 -c Msave -Mbounds -traceback demo_fn_avrage.f  
pgf77 -o program_name.e demo_main.o demo_sub_average.o demo_fn_avrage.o
```

AOSC 652: Analysis Methods in AOSC

Simple Programming

Copy file:

`~rjs/aosc652/week_04/program_random_number.f`

to your directory.

Code contains a FORTRAN function that given a “seed” (user must input *negative integer*), returns a random number between 0.0 and 1.0

Variable `nheads` tracks number of successive “heads” \Rightarrow
heads \equiv random_number \leq 0.5

Variable `c_coin` records whether flip was ‘heads’ or ‘tails’

Using `program_random_number.f` as starting point, add code that:

- a) keeps track of how many heads in a row have been tossed
- b) assigns proper value to variable `c_coin`

Challenge: determine “seed” that leads to **3 successive “heads”**
on **first 3 flips!**

AOSC 652: Analysis Methods in AOSC

More programming

Copy files:

`~rjs/aosc652/week_04/sort_numbers.f`

`~rjs/aosc652/week_04/piksrt.f`

`~rjs/aosc652/week_04/random_integers*.dat`

to your directory.

Insert the subroutine contained in file `piksrt.f` into the end of file `sort_numbers.f`

Modify subroutine `piksrt.f` such that it works properly, given the *nomenclature for the call* to this subroutine in `sort_numbers.f`

Run `sort_numbers.e` on the `random_integers*.dat` files to test that the code works (note: code will take a long time to sort 300,000 or 1,000,000 integers, so begin working with the smaller files and stop at the 100,000 file)

AOSC 652: Analysis Methods in AOSC

Please note Assignment 4a is due on Thursday @ noon:

Straight forward assignment that should not take much time to complete

Designed to provide “practice” in FORTRAN

Fluency in FORTRAN is important for remainder of this course

Press et al.

Available on line at <http://www.nrbook.com/a/bookfpdf.php>

Must download and install plugin for Abode reader in order to open these files on you home PC or MAC:

http://plugin.fileopen.com/?type=Filter&name=FOPN_foweb

On mac mini, try to read using Safari

Have handed out copies of this week’s Press readings in case you have trouble accessing the files