Computational Statistics

Programming – Assignment 4b
Dr. Mark Bourassa

Topics:
WRITE to a file
Review of READ and WRITE Formats
Creating file names in a program
Plotting data with xmgrace

Turn your graphics into the TA

Reading and Writing Data

- Reading Data:
  - Reading data without considering the format
    - Examples:

      ```
      READ(7,*) junk
      READ(7,*) x1, y2, number_of_tornados
      ```

    - In these examples, ‘7’ is the IO stream, and ‘*’ is the format, which in this case means ignore formatting.
  - Reading data while considering the format
    - Examples:

      ```
      READ(7,'(12X,F4.1,2X,F4.1)') tmax1, tmin1
      ```

    - Where the character string in quotes indicated the format
- Writing data to a file
  - ‘WRITE’ commands are similar in structure to ‘READ’
  - `WRITE(7,'(  2(2X,F4.1)  )') tmax1, tmin1`

IO Formatting

- What is meant by something like (A12,F4.1,2X,F4.1)?
- A: Characters. The number following the A is the number of characters. If no number is specified there can be any number of characters (which makes reading harder).
- X: skip (or output) a space. The number before X is the number of spaces.
- /: skip a line. The number before X is the number of lines.
- I: Integer.
  - The number following the I is the upper limit of the number of digits in the integer. For example, I4 would be up to 4 digits.
    - Output is left justified.
    - Unused digits are left blank (not filled with zeros).
  - The number before the I is the number of integers.
  - Example: print, 9999, 888, 77,6 using 4I4 results in
    9999 888 77   6

IO Formatting

- F: floating point output (numbers with decimals). E.g., aFb.c
  - The first number after the F (b in the example) is the number of digits, including the decimal place.
  - The number after the decimal (c in the above example) is the number of digits after the decimal place.
    - The value of c must be at least one less than the value of b.
    - Example: write 4.1234567 via F6.3 outputs a blank followed by 4.123
  - The number before the F is the number of floating point numbers
    - Example output 4.12, 4.123 via 2F6.3 outputs a blank followed by 4.120 4.123
- When you get output that looks like **** **** it means that the numbers are too large to fit in the specified format.
  - Example: 1234 output via I3 would be ***

Assignment 4b

- To plot the data we have to write it to file. As we process each case, we want to open a file, write the data, and close the file. You are familiar with opening and closing files, and writing is very similar to reading. The complication is that the easiest approach is to name the file on the fly. That is, for each case we will create a character string that is the file name.
- 2) Edit as4.f90
- 3) Add code to write the data being correlated to a file. The following code could be added after the case1 correlation.

```
OPEN(7, file=’case0.dat’)
DO i_pts=1, n1_pts
   WRITE(7, fmt=’(F6.3, 2X, F6.3)’) tmin1(i_pts), tmin2(i_pts)
ENDDO
CLOSE(7)
```

- 4) Declare a new character variable, named file_out, that is between 10 and 80 characters long.

Variable File Names

- Modify your DO LOOP that adds noise as follows.

```
WRITE( file_out, fmt='(A4,I1,A4)') 'case', (i_wt1-1)*n_wt2 + i_wt2, '.dat'
OPEN(7, file=file_out)
DO i_pts = 1, n_pts
  call random_number(rand_out)
  tmin1_mod(i_pts) = tmin1(i_pts) + wt1 * standev_tmin1 * gaussian(rand_out)
  call random_number(rand_out)
  tmin2_mod(i_pts) = tmin2(i_pts) + wt2 * standev_tmin2 * gaussian(rand_out)
  WRITE(7, fmt='(F6.3,2X,F6.3)') tmin1_mod(i_pts), tmin2_mod(i_pts)
ENDDO
close(7)
```

- The WRITE on the first line above writes the string to the variable file_out. Note that this only works for case numbers that are single digits because of the I1 in the format. To use large case numbers you can use an IF statement that checks the size of the case number, and selects a write with the appropriate format.

Plotting the Data With xmgrace

- Plot the data for each case. Use the tool xmgrace to make the plots. It is not a high quality plotter, but it is easy to use. At the UNIX prompt, type xmgrace and press return. Make the following changes to the settings.
- To get the data ‘into’ the program, import the data file:
    - Data – Import
        - Files: Select file
        - OK
        - Ignore error message (close the error window)
        - Close the import window

  To make it look good:
    - Plot options – set appearance – Main
        - Symbol properties
            - Type: circle
            - Size: 24 (or something like that)
            - Color: black
        - Line properties
            - Type: none
        - Apply
    - Change to symbols menu
        - Color fill color: black
        - Color fill pattern: solid black
        - ACCEPT