1 Worksheet 4: File Input and Output

In this worksheet you will be introduced to file I/O in Fortran 90. Scientific computing tasks often produce huge amounts of data that need to be stored on disk and later used for analysis, such as plotting. For this worksheet you will need to read carefully sections 1.6.4 (by now you should be well familiar with most of 1.6) and 2.3. We will also use a simple GUI program for data visualization, `xmgrace`.

1.1 Experimenting with the Series for $\mathrm{erf}(x)$

Notice that there are several executables you can play with in the directory `/classes/phy201/`. These programs will give compare the value found by the series expansion we have been using in our worksheets with the "correct" result (we find this one using a ready made Fortran 90 program by Alan Miller, found in `erf_precise.f90`, which will be made available with other sources later).

Play with these functions and use the utility `script` to record your experimenting session. Comment on any interesting results you observe. Here is how to use `script`:

```
> script erf_calculator.txt
> /classes/phy201/erf_calculator_real.x
  ... Play with a few numbers here ...
> exit
> gedit erf_calculator.txt &
  ... Look at the session file and then print it ...
> enscript erf_calculator.txt
```

1.2 Plotting the Error Function

Your assignment this time is to write a program that will `USE` the module you developed previously and write to a file the values of $\mathrm{erf}(x)$ for x being in the interval [`x_min`, `x_max`]. The user should enter the precision ε , the number of points on the plot n_points, and `x_min` and `x_max`.

The manual contains enough information about the proper way of opening and closing files, as well as writing to them. Use a `DO` loop to loop over different values of x:

```
DO i=1, n_points
  x= ... ! Calculate x here
  erf=ErfSeries(x) ! Call your function
  WRITE(...) x, erf
  ... ! Make sure there were no I/O errors
END DO
```

After you make sure your program works (be careful for the case when $x = 0$, you may run into trouble...), vizualize (plot) the error function by using the program `xmgrace`:

> `xmgrace` *`your_output_file`*`.dat &`

The `.dat` extension is standard for output data files. The program is graphics menu driven and simple, so you should be able to easily make your plot look nice. Make sure the axis and the curves are labeled properly.