AME 20214
Homework 2
Due: Thursday, 11 September 2014, in class

Consider the mathematics problem on the 20214 homepage,

$$\frac{dy}{dt} = -y, \qquad y(0) = 1.$$

By inspection, the exact solution is $y(t) = e^{-t}$.

Now, using standard approximations of the derivative, $dy/dt \sim \Delta y/\Delta t$, we can write a discrete approximation of the differential equation as

$$\frac{y_{n+1} - y_n}{\Delta t} = -y_n, \qquad y_0 = 1.$$

Solving for y_{n+1}, we get

$$y_{n+1} = y_n - \Delta t y_n, \qquad y_0 = 1.$$

If this formula is repeated iteratively within a computer code, we get a discrete approximation for $y(t)$.

1. (10) Download the `Fortran` source code, identical to that on the 20214 primary homepage,

   `http://www.nd.edu/~powers/ame.20214/euler.f90`,

   compile and execute it, so as to reproduce the results on the homepage. Plot $y(t)$ for $t \in [0, 1]$ for the exact solution and the numerical approximation with $\Delta t = 0.1$.

2. (70) Run the code for several different values of Δt and generate a log-log plot of the magnitude of the error in your prediction of y at $t = 1$. That is, plot

   $$|y_{exact}(t = 1) - y_{num}(t = 1)| \qquad \text{versus} \qquad \Delta t.$$

   For small values of Δt, you will probably wish to suppress the printing of intermediate values and only print the final values at $t = 1$. See how small a value of Δt you can use before roundoff error corrupts the solution. Note, the exact value is

   $$y_{exact}(t = 1) = e^{-1} = 0.367879441171442321595523770161460867445 8...$$

3. (20) Repeat the previous exercise with double the precision and generate the equivalent plot. To adjust the precision from its default value is a tricky task, and several generalizations must be made to `euler.f90`. The many generalizations have already been made and are found in the modified file

   `http://www.nd.edu/~powers/ame.20214/euler2.f90`

   As written, the code `euler2.f90` should give identical results as `euler.f90`. The precision in `euler2.f90` can be doubled by simply changing the value of `p` from `4` to `8`. For sufficiently small Δt, the error should be significantly lower than for the previous exercise. See how small a value of Δt you can effectively use before roundoff error corrupts the solution.

   Adjusting the precision in `Fortran` can be a messy chore, fraught with machine-dependencies. The text of C&S has an extended discussion of the `kind` specification, see pp. 67-84, whose difficulty reflects well the complexities of machine precision.

For this homework, there is a *three page maximum.* All plots must adhere to the standards described in the course notes. Though not required, you may wish to compare on your own the computation time for the equivalent `MATLAB` program with small Δt to convince yourself that compiled languages have real advantages for problems requiring high accuracy.