PHYSICS 3800. Assignment 2.
Ordinary Differential Equations.

1. (5 pts.) Write each of the following ODE's as an equivalent system of first-order ODE's. Let $y_1 = u$, $y_2 = \ldots$ etc. and write out your expressions in terms of the y_i's.
   (a) Van der Pol equation.
   $$u'' = u'(1 - u^2) - u$$
   (b) Blasius equation:
   $$u''' = -uu''$$
   (c) Newton's Second Law of Motion for a coupled two-body system:
   $$u_1'' = -GMu_1/(u_1^2 + u_2^2)^{3/2}$$
   $$u_2'' = -GMu_2/(u_1^2 + u_2^2)^{3/2}$$

2. (5 pts.) Predictor-Corrector for the SHO.
   Convert the `spring-midpoint-fn.cpp` (or `spring-midpoint-fnc.c` or `spring-midpoint-fnf.f90`) code (see website) to implement the Predictor Corrector method derived in class,
   $$\begin{aligned} y^* &= y_i + hf(y_i, t_i) \\ y_{i+1} &= y_i + \frac{h}{2}\Big[f(y^*, t_{i+1}) + f(y_i, t_i)\Big], \end{aligned}$$
   to solve the SHO (with $k/m = 1$, $x = 0$ and $v = 5$ initially) from t =0 to 10. Plot the absolute error (numerical solution minus analytical solution) as a function of time from $t = 0$ to 10 using a time step of 0.01. Does the solution appear to be stable? What is the error at $t = 10$? What is the error at $t = 10$ if the time step is reduced by a factor of ten to 0.001 (and does it make sense)? Please attach hard copies of your code and the graph.

3. (5 pts.) Adaptive time step fourth order Runge-Kutta (RK4).
Consider the simple differential equation,

$$y' = -y$$

with $y(0.5) = e^{-0.5}$. For this question, feel free to use Python, C, Fortran, Mathematica etc. (include code printout with submission) to carry out the tedious calculations. It may be easiest to modify `spring-midpoint-fn.cpp` (or equivalent) again.

(a) Evaluate $y(0.7)$ using RK4 with $h_1 = 0.2$ to at least 10 significant figures.

(b) Evaluate $y(0.7)$ using two steps of RK4 with $h_2 = 0.1$ to at least 10 significant figures.

(c) If Δ_1 is the difference between the results of parts (a) and (b), estimate the stepsize h_0 required to achieve a desired error of $\Delta_0 = 10^{-7}$, using the formula (see class notes and Numerical Recipes handout),

$$h_0 = h_1 \left| \frac{\Delta_0}{\Delta_1} \right|^{0.2}.$$

(d) Confirm your result for h_0 by calculating $y(0.5 + h_0)$ with one step of RK4 and comparing to the exact solution $y(0.5 + h_0) = e^{-(0.5+h_0)}$.

4. (5 pts.) The Leap Frog Method to solve ODE's.

(a) Starting from the Central Difference approximation for derivatives, derive the Leap Frog algorithm to solve a set of first-order ODE's $\dot{\mathbf{y}} = \mathbf{f}(\mathbf{y}, t)$:

$$\mathbf{y}_{i+1} = \mathbf{y}_{i-1} + 2\Delta t \mathbf{f}(\mathbf{y}_i, t).$$

Argue that the neglected terms (error) are of order $(\Delta t)^2$ after N time steps (compared to $\mathcal{O}(\Delta t)$ for the Euler method)

(b) Following the method used in class for the Euler Method, show in this case that the growth factor g satisfies the equation

$$g = a \pm \sqrt{a^2 + 1}$$

where $\delta y_i = g \delta y_{i-1}$, $\delta y_{i+1} = g^2 \delta y_{i-1}$ and $a = \Delta t (\partial f / \partial y_i)$.
(c) Assume a is small (and real) and show that one of the solutions will always give $g^2 > 1$. This means that the method is unstable for both growth ($a > 0$) and decay ($a < 0$).