MAE 3210 - Spring 2020 - Homework 6

Homework 6 is due **online** through Canvas by 11:59PM on Monday, April 20.

IMPORTANT REMARKS (Please read carefully):

- You are required to submit code for all functions and/or subroutines built to solve these problems, which is designed to be easy to read and understand, in your chosen programming language, **and which you have written yourself**. The text from your code should both be copied into a single PDF file submitted on canvas. **Your submitted PDF must also include responses to any assigned questions, which for problems requiring programming should be based on output from your code**. For example, if you are asked to find a numerical answer to a problem, the number itself should be included in your submission.

1. (a) Develop an algorithm which, for a given function of two variables $f(x, y)$, interval bounds a and b with $a < b$, and c and d with $c < d$, and input integer $n \geq 1$, does the following:

   (i) If n is odd, it applies the multiple-application trapezoidal rule in each dimension to approximate $I = \int_c^d \left(\int_a^b f(x, y)\, dx \right) dy$.

   (ii) If n is even, it applies the multiple-application Simpson's 1/3 rule in each dimension to approximate $I = \int_c^d \left(\int_a^b f(x, y)\, dx \right) dy$.

   (b) Suppose the temperature T (°C) at a point (x, y) on a 16 m^2 rectangular heated plate is given by

$$T(x, y) = x^2 - 3y^2 + xy + 72,$$

   where $-2 \leq x \leq 2$ and $0 \leq y \leq 4$ (here x and y are measured in meters about a reference point at $(0, 0)$). Determine the average temperature of the plate:

   (i) Analytically, to obtain a true value.

   (ii) Numerically, using the algorithm you developed in question 1(a) above, and plot the true percent relative error ϵ_t as a function of n for $1 \leq n \leq 5$. Provide some interpretation of the results.

2. Write code for two separate algorithms to implement (a) Euler's method and (b) the standard 4th order Runge-Kutta method, for solving a given first-order **one-dimensional** ODE. Design the code to solve the ODE over a prescribed interval with a prescribed step size, taking the initial condition at the left end point of the interval as an input variable.

3. The drag force F_d (N) exerted on a falling object can be modeled as proportional to the square of the objects downward velocity v (m/s), with a constant of proportionality c_d (kg/m).

   (a) Assume that a falling object has mass $m = 100$ (kg) with a drag coefficient of $c_d = 0.25$ kg/m, and let $g = 9.81$ (m/s^2) denote the constant downward acceleration due to gravity near the surface of the earth. Starting from Newton's second law, explain the derivation of the following ODE for the downward velocity $v = v(t)$ of the falling object:

   $$\frac{dv}{dt} = 9.81 - 0.0025v^2. \tag{1}$$

   (b) Suppose that this same object is dropped from an initial height of $y_0 = 2$ km. Determine when the object hits the ground by solving the ODE you derived in question 3(a) using

      (i) Euler's method.

      (ii) the standard 4th order Runge-Kutta method.

   HINT: Note that, with the velocity v oriented downward, the height $y = y(t)$ satisfies $\frac{dy}{dt} = -v$. You are asked to find the final time t_f when the height y of the falling object reaches zero, i.e. when $y(t_f) = 0$. There are two ways to solve this problem.

   A. You can use your algorithm for solving one-dimensional ODEs (Euler and Runge-Kutta 4) from question 2 to solve the ODE (1) to find $v = v(t)$ (at discrete time points) with initial condition $v(0) = 0$. Then, you can use your one-dimensional ODE algorithms, again, to solve $\frac{dy}{dt} = -v$ with initial condition $y(0) = 2000$ m, and try to identify when $y(t_f) = 0$.

   B. Alternatively, you can use your algorithm for solving two-dimensional ODEs (Euler and Runge-Kutta 4) from question 4 to solve the coupled ODE system

   $$\begin{aligned}\frac{dy}{dt} &= -v \\ \frac{dv}{dt} &= 9.81 - 0.0025v^2,\end{aligned}$$

with initial condition $y(0) = 2000$, $v(0) = 0$. Then, try to identify when $y(t_f) = 0$.

4. Write code for two separate algorithms to implement (a) Euler's method and (b) the standard 4th order Runge-Kutta method, for solving a given first-order **two-dimensional** system of ODEs. Design the code to solve the system of ODEs over a prescribed interval with a prescribed step size.

5. The motion of a damped mass spring is described by the following ODE

$$m\frac{d^2x}{dt^2} + c\frac{dx}{dt} + kx = 0, \tag{2}$$

where x = displacement from equilibrium position (m), t = time (s), m = mass (kg), k = stiffness constant (N/m) and c = damping coefficient (N·s/m).

   (a) Rewrite the 2nd order ODE (2) as a two-dimensional system of first order ODEs for the displacement $x = x(t)$ and velocity $v = v(t)$ of the mass attached to the spring.

   (b) Assume that the mass is $m = 10$ kg, the stiffness $k = 12$ N/m, the damping coefficient is $c = 3$ N·s/m, the initial velocity of the mass is zero ($v(0) = 0$), and the initial displacement is $x = 1$ m ($x(0) = 1$). Solve for the displacement and velocity of the mass over the time period $0 \leq t \leq 15$, and plot your results for the displacement $x = x(t)$,

      (i) using Euler's method with step size $h = 0.5$, and then with step size $h = 0.01$.

      (ii) using the standard 4th order Runge-Kutta method with step size $h = 0.5$, and then with step size $h = 0.01$.

   (c) Assume that the mass is $m = 10$ kg, the stiffness $k = 12$ N/m, the damping coefficient is $c = 50$ N·s/m, the initial velocity of the mass is zero ($v(0) = 0$), and the initial displacement is $x = 1$ m ($x(0) = 1$). Solve for the displacement and velocity of the mass over the time period $0 \leq t \leq 15$, and plot your results for the displacement $x = x(t)$,

      (i) using Euler's method with step size $h = 0.5$, and then with step size $h = 0.01$.

      (ii) using the standard 4th order Runge-Kutta method with step size $h = 0.5$, and then with step size $h = 0.01$.