MATH 131: Numerical Methods for scientists and engineers - Assignment 6

Homework Assignment 6 due by 11:59 PM, April 21, 2021. Test your answers in Matlab.
For this assignment you are allowed 5 submissions per exercise.
Save each code as an .m file. We will use those codes later in class.

Matlab Grader

1. (20 points) Write a Matlab function called `AB4` that solves a differential equation using four-step Adams-Bashforth method. To generate the first w_1, w_2, w_3, use Runge-Kutta 4 (rk4) method. You can use the code you generated from Homework 5. Your code should look like:

```
function [y,t] = ab4(f,t0,tf,alpha,N)
```

   where N is the number of intervals used, so that $\Delta t = \frac{tf-t0}{N}$. Note that the output should be y, a **column** vector that contains the evaluation of the solution at all time steps and t, a **column** vector of the time variable.

   The AB4 method is given by:

$$w_0 = \alpha, \quad w_1 = \alpha_1, \quad w_2 = \alpha_2, \quad w_3 = \alpha_3$$
$$w_{j+1} = w_j + \frac{\Delta t}{24}\left[55f(t_j, w_j) - 59f(t_{j-1}, w_{j-1}) + 37f(t_{j-2}, w_{j-2}) - 9f(t_{j-3}, w_{j-3})\right]$$

2. (20 points) Write a Matlab function called `euler_system` that solves a system of two IVPs:

$$y_1' = f(t, y_1, y_2), \quad y_1(t_0) = \alpha_1$$
$$y_2' = g(t, y_1, y_2), \quad y_2(t_0) = \alpha_2$$

   over the interval $t_0 \le t \le t_f$ using Euler's method. The header should look like

```
function [y,t] = euler_system(f,g,t0,tf,alphas,N)
```

   where N is the number of intervals used, so that $\Delta t = \frac{tf-t0}{N}$ and alphas represents a row vector with the initial conditions. Note that the output should be y, a matrix for which the first **column** contains the evaluation of the solution for the first differential equation at all time steps and the second **column** contains the evaluation of the solution for the second differential equation at all time steps etc., and t a column vector of the time variable.

Catcourses

3. (20 points) Consider the initial value problem $y' = -21y, \quad 0 \le t \le 1, \quad y(0) = 10$

   (a) Solve the IVP using Euler's method (from Assignment 5) with a step-size of 0.1. Plot the solution versus time along with the exact solution on the same plot.

   (b) Looking at your graph from a, does your solution approach the exact solution? Why or why not?

   (c) What is the minimum number of time steps needed to have your solution approach the exact solution/not blow up? Explain how you arrived at that number and then plot the solution in blue on top of your figure from a).

   (d) Now use your RK4 code from Assignment 5 to solve the IVP with the same step size from c). Plot the solution on the same graph from parts a-c in green. How does the RK4 solution approximate the true solution? Better or worse than Euler's Method? Why?

4. (20 points) Consider the damped spring problem: $mx''(t) + dx'(t) + kx(t) = 0$ where $x(t)$ represents the position of the spring at time (t), m represents the mass of the spring, k is the spring constant, and d the coefficient for damping.

   (a) Write this 2nd order differential equation as a system of first-order differential equations.

   (b) Solve the system of first-order differential equations using the `euler_system` code you created for Problem 2. Consider a spring of mass 1kg, a spring constant of 4N/m, and a damping coefficient of 0.5 Ns/m. To start the simulation assume the spring is pulled back 5 cm (x(0)=-0.05 m) and released. Solve over the time interval from 0 to 10 seconds with N=10000.

   (c) Write some Matlab code to determine the time at which the spring no longer reaches a distance of greater than 1 cm from the center. What time does this occur at? *Hint: Look at the* `find` *function in Matlab (type 'help find' for more info)*

5. (20 points) Consider the numerical scheme

$$w_0 = \alpha, \quad w_1 = \alpha_1$$
$$w_{j+1} = -w_{j+1} + 3w_j - w_{j-1} + \Delta t F(t_{j+1}, t_j, t_{j-1}, w_{j+1}, w_j, w_{j-1})$$

   (a) Classify this numerical scheme. Is this a multi-step method or a single-step method? If multi-step, how many steps? Is the method implicit or explicit?

   (b) Is the method stable? If so, is it strongly stable or weakly stable? Show all work!