Math 686: Chapter 2 Homework – Spring 2020

Due: Tuesday, February 4, 2020

1. Derive equation (2.16) in the textbook using the techniques discussed in class

2. Exercise 2.1 in the textbook.

3. Consider $y' = f(t, y)$ where

$$f(t,y) = -(1+y^4)y + g(t), \quad y(0) = 1, \quad 0 \le t \le 1,$$

where

$$g(t) = e^{-t}\left[-10\sin(10t) + e^{-4t}(\cos(10t))^5\right].$$

This equation has exact solution $y_{exact}(t) = e^{-t}\cos(10t)$.

(a) Solve this IVP numerically by implementing Euler's method.

(b) Solve this IVP numerically by implementing Huen's method.

(c) Solve this IVP numerically by implementing 2-step Adams-Bashforth method defined in equation (2.6). In this case, use Huen's method to get the method started (that is, initial condition for y_0, Huen's method for y_1 and A-B thereafter).

For parts (a), (b) and (c), document the error in the solution by taking the infinity norm of $y - y_{exact}$ (where y represents your approximate solution vector over all discrete time points and y_{exact} is the corresponding exact solution vector at those same time points) and analyzing this error and its dependence on the number of intervals N used to discretize $[0, 1]$. A suitable report of this information might include a table showing the three methods and their error associated with different values of N (and/or h) where $h = 1/N$ is the time-step size. You should explore the error for $N = 10, 20, 40, 80, 160, 320, 640, 1280, 2560, 5120$.

For one choice of time-step size (i.e. for some choice N) you should plot these three solutions on a graph (please do not submit 9x3 plots of the solutions!) along with the exact solution (hopefully they all look fairly similar).

While your numerical solutions should approximate the exact solution, your goal here is to demonstrate precisely *how well* they approximate the solution – that is, you should be able to demonstrate that the methods that you have implemented (not just the methods in theory) show convergence properties expected theoretically. A numerical observation of a specific convergence property is often a good way to debug your code (e.g. if your high order scheme is only showing low order convergence that is a clue that something is not behaving as expected and you might not be getting the correct solution).