COSC 4364 Spring 2018

Assignment 7

Due date: April 5. e-mail your answers and code.

Problem	Points		
	a)	b)	Total
1	4	4	8
2	8		8
3	8		8
4	30		30
5	40		40
Total			94

Motivations are important, even if not explicitly requested in a problem description. Omitting motivations for answers will result in deductions of points, possibly significant deductions, even if (numeric) answers are correct. On the other hand, if your approach to solving the problem is well described and would lead to a correct result minor mistakes will only incur minor point deductions. For the Matlab problems the output of the code together with a discussion of the results and a script for how to run the code need to be submitted as well. The discussion carries a significant weight towards the problem credit

Problem 1. (8 points) For the problems below determine the error in the solution at t=10 and t=20 because of an error of ε in the initial condition..

a) $x' = x$, correct initial condition $x(0)=c$; initial condition with error $x(0)=c+\varepsilon$.
b) $x' = -x$, correct initial condition $x(0)=c$; initial condition with error $x(0)=c+\varepsilon$.

Problem 2. (8 points) Calculate an approximate value of x(t) at t=0.1 using one step of the Taylor series method of order three for the ODE below

$$x'' = x^2e^t+x', \text{ with the initial conditions } x(0) = 1,\ x'(0) = 2.$$

Problem 3. (8 points) Rewrite the set of equations below as an autonomous system of first order differential equations.

$$\begin{aligned} x'' - 2xzx' &= 3x^2yt^2 \\ y'' - e^y y' &= 4xt^2z \\ z'' - 2tz' &= 2te^{xz} \end{aligned}$$

with the initial conditions $x(1) = 3$, $y(1) = 3$, $z(1) = 2$, $x'(1) = -7$, $y'(1) = 2$, $z'(1) = 3$.

Matlab problems

Problem 4. (30 points) Write your own non-adaptive 4th order Runge-Kutta routine (RK4) in Matlab and apply it to

$x_1' = -3x_2$
$x_2' = (1/3)\, x_1$
with initial conditions $x_1(0) = 0$ and $x_2(0) = 1$ on the interval [0,4].

The true solution for this problem is x_1=3cos(t) and x_2 = sin(t). Compute the errors in x_1 and x_2 by comparing the output of your RK4 routine to the true values for 100 equally spaced points in [0,4] and make a plot of the errors. Try different steps sizes for the Runge-Kutta procedure and make a polynomial interpolation to the Runge-Kutta points for the error computation at the 100 equally spaced points. Use your own interpolation function.

What step size is required for a maximum absolute error of 10^{-5}?
How many steps are taken by the method for this maximum absolute error?
What execution time was required?

Problem 5. (40 points) Write your own non-adaptive Adams-Bashforth-Moulton routine in Matlab and apply it to the equations in Problem 4 with the same initial conditions and on the same interval. Compute and plot the errors at the same points as in Problem 4. Try different steps sizes. For the error computation use interpolation using the function values and times generated by the Adams-Bashforth-Moulton procedure as nodes when computing the errors compared to the true solution.

What step size is required for a maximum absolute error of 10^{-5}?
How many steps are taken by the method for this maximum absolute error?
What execution time was required?