COMPUTER SCIENCE 349A
ASSIGNMENT #7

DUE NEVER
(0 points)

1. **(0 points)** Construct the Taylor polynomial approximations of order $n = 3$ for both $f(x_0 + h)$ and $f(x_0 + 2h)$ expanded about x_0 (with their remainder terms written as $O(h^4)$). Derive a numerical differentiation formula for approximating $f'(x_0)$ by setting $-3f(x_0) + 4f(x_0 + h) - f(x_0 + 2h)$ equal to the appropriate linear combination of the above approximations and solving for $f'(x_0)$. What is the order of the error of your approximation formula? Your answer whould be $O(h^d)$ for some integer d.

2. **(0 points)**

   (a) Give the iterative formula for the Taylor method of order $n = 2$ for approximating the solution of the initial-value problem

   $$y'(x) = 2 + (x - y(x))^2, \quad y(1) = 1.$$

   Note: an iterative formula is of the form $y_{i+1} =$ (some function of y_i, x_i and h).

   (b) Complete the specification of the following MATLAB M-file `taylor2.m` so that it will compute an approximate solution to the above initial-value problem on $[1, 1.25]$ using a step size of $h = 0.01$ and the Taylor method of order $n = 2$. Instead of using one-dimensional arrays to store the values of x_i and y_i, this M-file uses only two (scalar) variables, x and y (that is, y is initialized to y_0, and all successive computed approximations $y_1, y_2, \ldots$ are also stored as y). Similarly, x is initialized to 1 and then is incremented by h at each step.

   ```
   function  taylor2
   fprintf(values of x        approximation y\n)
   x = 1 ;
   y = 1 ;
   h = 0.01 ;
   fprintf ( %8.2f ,x), fprintf (%19.8f \n,y)
   for  i  =  ____________________
        y = ______________________________ ;
        x = _________________________ ;
        fprintf ( %8.2f ,x), fprintf (%19.8f \n,y)
   end
   ```

   Run this function M-file `taylor2`.

   (c) Use the fact that the exact solution to this initial-value problem is $x + \tan(x - 1)$ to compute the global truncation error at $x = 1.25$.

3. **(0 points)** Consider the initial-value problem $y'(x) = 2 + y(x)/x$, with $y(1) = 3$.

(a) Compute the approximation y_1 to $y(1.01)$ using the second-order Runge-Kutta method

$$y_{i+1} = y_i + \frac{h}{2}\left(f(x_i, y_i) + f(x_i + h, y_i + h \cdot f(x_i, y_i))\right)$$

with $h = 0.01$. Show all of your work.

(b) If the above Runge-Kutta method is used to approximate the solution of the above initial-value problem with a fixed step size of $h = 0.01$ on $[1, 1.04]$, then the following results are obtained (with the value y_1 that you computed in (a) omitted):

x_i	y_i	$y(x_i)$
1.00	3.0000000	3.0000000
1.01		3.0500997
1.02	3.1003961	3.1003974
1.03	3.1508892	3.1508911
1.04	3.2015765	3.2015791

(Here the values $y(x_i)$ denote the exact solution.) Use only the information in the above table to answer the following. Use at least 8 significant digits (i.e. 7 decimal places) in your computations. Show all of your work.

What is the global truncation error at $x = 1.04$?

What is the local truncation error at $x = 1.04$? To do this, as was done in Handout #35 for Eulers method, compute a value v by using the above Runge-Kutta formula with $x_i = 1.03$ and $h = 0.01$, but with y_i replaced by the exact value $y(x_i) = y(1.03)$. Then the local truncation error at $x = 1.04$ is $|y(1.04) - v|$. Note that this value should be smaller than the global truncation error.

Note: the exact solution to this initial-value problem is $y(x) = x\ln(x) + 2x$. This was used to compute the above values of $y(x_i)$.

4. **(0 points)** (a) Use the MATLAB ode solver `ode45` to solve the following initial-value problem:

$$y'(x) = -y(x) - 2e^{-2x}\sin(3x), \quad y(0) = 0.2,$$

on $[0, 3]$. Print all values of x and y for which ode45 computes an approximate solution. The solver `ode45` is a variable stepsize (or adaptive) Runge-Kutta method that uses two Runge-Kutta formulas of orders 4 and 5; it is similar to the adaptive Runge-Kutta methods in Section 25.5 of the textbook.

In their simplest form, all MATLAB ode solvers can be invoked as follows:

```
[x,y] = solver( 'f', xspan, y0 )
```

where 'f' is a string containing the name of the function $f(x, y)$, `xspan` is a vector containing the interval of integration, and `y0` is the initial condition. Thus, if you have defined a MATLAB function

```
function z = f(x, y)
z = (1/x)*(y*y+y);
```

the initial-value problem

$$y'(x) = \frac{y^2 + y}{x}, \quad y(1) = -2,$$

can be solved on the interval $[1, 3]$ by entering

```
[x, y] = ode45(' f ', [1 3], -2 )
```

The results will be stored in the (column) vectors x and y (and will be output to the screen, if you don't put a semi-colon(;) at the end of this statement).

In this form, `ode45` automatically selects the initial step size and all subsequent step sizes, and attempts to compute the solution so that the global truncation error has a relative error less than 10^{-3}.

NOTE: solve the problem given at the beginning of this question, not this latter sample problem.

(b) For the sample initial-value problem in (a), if one enters

```
ode45( ' f ', [1, 3], -2)
```

without any output parameters on the left hand side of the `=`, then MATLAB produces a graph of the computed solution. Generate and print a graph of the computed solution to the initial-value problem

$$y'(x) = -y(x) - 2e^{-2x}\sin(3x), \quad y(0) = 0.2.$$

As in part (a), compute you solution in the range $[0, 3]$.