College of Engineering and Computer Science
Mechanical Engineering Department

California State University Northridge

Mechanical Engineering 309
Numerical Analysis of Engineering Systems

Spring 2014 Number: 15237 Instructor: Larry Caretto

Programming Assignment Seven – Numerical Solution of Ordinary Differential Equations[1]

Overview, algorithms, and tasks

This assignment asks you to program two algorithms with second order global error for solving a system of N differential equations of the form $dy_m/dt = f(t,\mathbf{y})$, m = 1, N. The first algorithm, known as the Huen algorithm, is shown below. You should program this algorithm in MATLAB.

$$y^0_{m,i+1} = y_{m,i} + hf_m(t_i, \mathbf{y}_i)$$
$$t_{i+1} = t_i + h$$
$$y_{m,i+1} = \frac{y_{m,i} + y^0_{m,i+1} + hf_m(t_{i+1}, \mathbf{y}^0_{m,i+1})}{2}$$

The second is the modified Euler algorithm, which you should program in VBA.

$$y_{m,i+1/2} = y_{m,i} + \frac{h}{2} f_m(t_i, \mathbf{y}_i)$$
$$y_{m,i+1} = y_{m,i} + hf_m(t_i + h/2, \mathbf{y}_{i+1/2})$$
$$t_{i+1} = t_i + h$$

Program both algorithms should as functions with the following input parameters: initial t value, final t value, the number of steps to be used, and an array with the initial values of **y**. The functions should determine the number of equations to be solved as the size of the input initial **y** array. The functions should return the values of the **y** array at the final time. Both ODE solver functions should use a separate function that calculates all the derivatives. This function should not be part of the function that solves the ODEs. The derivative-calculation function should be a vector-valued function that returns a one-dimensional array containing the values of f_m, m = 1, N. MATLAB naturally return vectors as function results. In VBA your function to compute the derivatives should be declared as follows:

```
Function calcDerivs(y() as Double, N as Long) As Variant
```

This function should use Dim()/Redim statement for a type Double variable, f, which can then be used to compute the individual derivatives, f(k). At the end of the function the statement `calcDerivs = f` will return a variant array to your ODE solver. You can equate the return from this function to a type variant variable that is declared as a scalar in the ODE solver.

In order to separate the derivative calculation routine from the ODE solver itself, you have to pass the name of the derivative-calculation routine to the ODE solver. In MATLAB, you should pass a function handle for the routine that computes the derivatives to the ODE solver function. In VBA, you should pass the name of the function that calculates the derivatives as a string to the ODE solver and use the Application.Run procedure to evaluate the derivative function.

The differential equations to be solved

In this assignment, you should solve the three ordinary differential equations shown below.

[1] Assignment due Monday, May 5, at 11:59 pm. May be submitted by Wednesday, May 7, at 11:59 pm with a 30% penalty. No grade for assignments submitted after this date.

Jacaranda (Engineering) 3314 Mail Code Phone: N/A
Email: lcaretto@csun.edu 8348 Fax: 818.677.7062

$$\frac{dy_1}{dt} = -y_1 + \sqrt{y_2} - y_3 e^{2t} \qquad \frac{dy_2}{dt} = -2y_1^2 \qquad \frac{dy_3}{dt} = -3y_1 y_2$$

Solve these equations for an initial value of t = 0 and a final value of t = 1. The initial y values for the three variables are: $y_1(0) = y_2(0) = y_3(0) = 1$. Your solution should produce a value of each y_k for the final t = 1. The exact solution to this set of equations is $y_k = e^{-kt}$. Use these exact solutions to compute the RMS error for all three results at t = 1.

$$\epsilon_{rms} = \sqrt{\frac{\sum_{k=1}^{N}\left(y_{k,numeric} - y_{k,exact}\right)^2}{N}}$$

1. Using MATLAB

a. Run the program you write for a start time of zero an end time of 1 with constant step sizes, h, from 0.1 to 10^{-8}. Plot the RMS error as a function of step size.

b. Use the MATLAB ODE solver, ode45, discussed in class. Compare the results of this function to the exact results, using the RMS error. See the documentation for the "Stats" option for this solver and use this option to get the number of steps used and the number of calls to the derivative-evaluation function. Note that this solver uses an adjustable step size to efficiently solve the problem for a given desired error. The default value for ode45 can be changed by using the options for this solver.

2. Using EXCEL VBA

a. Run the program you write for a start time of zero an end time of 1 with constant step sizes, h, from 1 to 10^{-8}. Plot the RMS error as a function of step size.

Submission Requirements

You should submit only two files for this assignment. The first is a Word file that contains all your MATLAB results and discusses the questions below. The second is an Excel file with your Excel results.

1. What is the order of the truncation error for the Huen method according to your results? Is there any point where roundoff error becomes important for this method?
2. What is the truncation error for the modified Euler method according to your results? Is there any point where roundoff error becomes important for this method?
3. For both the Huen method and the modified Euler method determine the number of steps that are required to obtain the same RMS error that you found with the MATLAB solver. How many steps did ode45 require to obtain this error?
4. For both the Huen and the modified Euler method runs that you identified in the previous question as having an error similar to that in the MATLAB solver, compare the number of times the derivative evaluation function was called to the number times the derivative evaluation function was called by the MATLAB solver.