California State University Northridge

College of Engineering and Computer Science
Mechanical Engineering Department

Mechanical Engineering 501A
Seminar in Engineering Analysis

Fall 2017 Number: 15966 Instructor: Larry Caretto

November 22 Homework Solutions

Consider the following differential equation: $\frac{dy}{dx} = \frac{2\sqrt{y - \ln x}}{x} + \frac{1}{x}$ **with y = 0 at x = 1**

Verify that the exact solution is y = (ln x)^2 + ln x.

Compute the numerical solution of this equation and the corresponding error using the following methods and step sizes and compute the errors at the final step:

- **Euler's method with h = 0.1**
- **Heun's method with h = 0.2**
- **Fourth-order Runge-Kutta method with h = 0.4**
- **Fourth-order Runge-Kutta method with h = 0.1**

We can verify the exact solution by substituting it into the left and right-hand side of the differential equation. This gives

$$\frac{dy}{dx} = \frac{d}{dx}\left[(\ln x)^2 + \ln x\right] = \frac{2\ln x}{x} + \frac{1}{x} = RHS\,?$$

$$RHS = \frac{2\sqrt{y - \ln x}}{x} + \frac{1}{x} = \frac{2\sqrt{(\ln x)^2 + \ln x - \ln x}}{x} + \frac{1}{x} = \frac{2\ln x}{x} + \frac{1}{x}$$

So, the exact solution satisfies the differential equation; it also satisfies the initial condition that y(1) = 0.

The Euler method algorithm, $y_{m+1} = y_m + hf(x_m,y_m)$, is implemented for h = 0.1, with the results shown in the table below. For each row after the first row, the new y value is the old y value plus the derivative in the last column of the previous row times the step size, h = 0.1.

x	y	Exact y	Error	f
1	0	0	0	1
1.1	0.1	0.104394	0.004394	1.033604
1.2	0.20336	0.215563	0.012202	1.075080
1.3	0.310868	0.331199	0.020331	1.108056
1.4	0.421674	0.449686	0.028012	1.131276
1.5	0.534802	0.569867	0.035065	1.146178
1.6	0.649419	0.690907	0.041488	1.154469
1.7	0.764866	0.812195	0.047328	1.157625
1.8	0.880629	0.933280	0.052651	1.156832

Huen's method uses the following algorithm: $y_{m+1} = y_m + (k_1 + k_2)/2$, where $k_1 = hf(x_m, y_m)$ and $k_2 = hf(x_{m+1}, y_m + k_1)$. This algorithm is implemented in the table below for h = 0.2. The values of k_1 and k_2 are shown. The average of these two values in each row is added to the y value in that row to get the y value in the next row.

x	y	Exact y	Error	k1	k2
1	0	0	0	0.2	0.210987
1.2	0.205493	0.215563	0.010069	0.217408	0.226854
1.4	0.427624	0.449686	0.022062	0.229118	0.233033
1.6	0.65870	0.690907	0.032207	0.233598	0.233739
1.8	0.892368	0.933280	0.040912		

The classic Runge-Kutta algorithm uses four derivative function evaluations per step to compute $y_{m+1} = y_m + (k_1 + 2k_2 + 2k_3 + k_4)/6$, where $k_1 = hf(x_m, y_m)$, $k_2 = hf(x_m+h/2, y_m + k_1/2)$, $k_3 = hf(x_m+h/2, y_m + k_2/2)$, and $k_4 = hf(x_{m+1}, y_m + k_3)$. This algorithm is implemented for h = 0.4 and h = 0.1 in the two tables below. In each table, the value of y and k_1 to k_4 in a given row are used to compute the new value, y_{m+1} that is shown in the next row.

x	y	Exact y	Error	k1	k2	k3	k4
1	0	0	0	0.4	0.421974	0.446205	0.475006
1.4	0.435227	0.449686	0.014459	0.465287	0.472411	0.474404	0.474362
1.8	0.907441	0.93328	0.025839	0.473502	0.468639	0.467914	

x	y	Exact y	Error	k1	k2	k3	k4
1	0	0	0	0.1	0.101863	0.104053	0.107909
1.1	0.10329	0.104394	0.001104	0.107151	0.109701	0.110534	0.112915
1.2	0.21338	0.215563	0.002183	0.112705	0.114535	0.114872	0.116413
1.3	0.328035	0.331199	0.003164	0.116348	0.117546	0.117697	0.118649
1.4	0.445616	0.449686	0.00407	0.118624	0.119337	0.119404	0.119926
1.5	0.564954	0.569867	0.004913	0.119915	0.120263	0.120289	0.120492
1.6	0.685206	0.690907	0.005701	0.120487	0.120563	0.120568	0.120534
1.7	0.805753	0.812195	0.006441	0.120532	0.120404	0.120398	0.120188
1.8	0.926141	0.93328	0.007139	0.120187	0.119907	0.119894	

Comparing the final errors for the first three methods, where the total number of derivative evaluations, a measure of the work in the algorithm, is the same, shows that the error decreases from 0.052651 to 0.040912 to 0.025839 as we go from the Euler to the improved Euler to the Runge-Kutta. This example shows that the error decreases for higher order of the error at an equivalent amount of work. When we cut the step size in the classical Runge-Kutta from 0.4 to 0.1, a factor of 0.24, the error is reduced from 0.025839 to 0.007139, a factor of 0.276; this is not the fourth-order error that we would expect.