M348-53610: Scientific Computing
Homework # 10
Handout: 04/05/2016, Tuesday
Due: 04/19/2016, Tuesday

Submission. Please make your homework neat and stapled. You have to submit your homework in ECJ 1.204 before **3:00 PM** on the due date. Note that *no late homework will be accepted without compelling reasons.*

---

1 To be Graded

Problem 1. Use Taylor's method of order two to approximate the solutions for each of the following initial-value problems.

$$(a)\ y' = te^{3t} - 2y, \quad 0 \le t \le 1, \quad y(0) = 0, \quad \text{with } h = 0.5$$

$$(b)\ y' = 1 + (t - y)^2, \quad 2 \le t \le 3, \quad y(2) = 1, \quad \text{with } h = 0.5$$

$$(c)\ y' = -ty + 4ty^{-1}, \quad 0 \le t \le 1, \quad y(0) = 1, \quad \text{with } h = 0.25$$

Problem 2. Use the Modified Euler method to approximate the solutions to each of the following initial-value problems.

$$(a)\ y' = te^{3t} - 2y, \quad 0 \le t \le 1, \quad y(0) = 0, \quad \text{with } h = 0.5$$

$$(b)\ y' = 1 + (t - y)^2, \quad 2 \le t \le 3, \quad y(2) = 1, \quad \text{with } h = 0.5$$

Problem 3. Use the Midpoint method to approximate the solutions to each of the following initial-value problems.

$$(a)\ y' = te^{3t} - 2y, \quad 0 \le t \le 1, \quad y(0) = 0, \quad \text{with } h = 0.5$$

$$(b)\ y' = 1 + (t - y)^2, \quad 2 \le t \le 3, \quad y(2) = 1, \quad \text{with } h = 0.5$$

Problem 4 (Programming Assignment)**.** A full description of the Runge-Kutta method of order four is provided in Algorithm 5.2 of Burden and Faires (also attached here in case you do not have the textbook).
(a) Implement the Runge-Kutta method of order four in Matlab, C++ or any other program language that you prefer. Include the full program in your submission.
(b) Use the code developed in (a) to solve the initial value problem

$$y' = e^{t-y}, \quad 0 \le t \le 1, \quad y(0) = 1$$

with $h = 0.01$. In the $t - y$ coordinates, plot your numerical solution together with the true solution

$$y(t) = \ln(e^t + e - 1).$$

You can use solid line to plot the true solution and dashed line to plot the approximation.

Problem 5 (Programming Assignment)**.** A full description of the Runge-Kutta-Fehlberg method of order four is provided in Algorithm 5.3 of Burden and Faires (also attached here in case you do not have the textbook).
(a) Implement the Runge-Kutta method of order four in Matlab, C++ or any other program language that you prefer. Include the full program in your submission.
(b) Use the code developed in (a) to solve the initial value problem

$$y' = e^{t-y}, \quad 0 \le t \le 1, \quad y(0) = 1$$

with tolerance $TOL = 10^{-4}$, $hmax = 0.025$, and $hmin = 0.005$. In the $t - y$ coordinates, plot your numerical solution together with the true solution

$$y(t) = \ln(e^t + e - 1).$$

You can use solid line to plot the true solution and dashed line to plot the approximation.

2 Reading Assignments

- Review Sections 5.2-5.5 of Burden & Faires or Sections 6.2, 6.4, 6.5 and 6.9 of Epperson.

gives the values in Table 5.7. Note the decreased error throughout the range over the Midpoint and Modified Euler approximations. □

Table 5.7

t_i	$y(t_i)$	Heun's Method	Error
0.0	0.5000000	0.5000000	0
0.2	0.8292986	0.8292444	0.0000542
0.4	1.2140877	1.2139750	0.0001127
0.6	1.6489406	1.6487659	0.0001747
0.8	2.1272295	2.1269905	0.0002390
1.0	2.6408591	2.6405555	0.0003035
1.2	3.1799415	3.1795763	0.0003653
1.4	3.7324000	3.7319803	0.0004197
1.6	4.2834838	4.2830230	0.0004608
1.8	4.8151763	4.8146966	0.0004797
2.0	5.3054720	5.3050072	0.0004648

Runge-Kutta methods of order three are not generally used. The most common Runge-Kutta method in use is of order four in difference-equation form, is given by the following.

Runge-Kutta Order Four

$$
\begin{aligned}
w_0 &= \alpha,\\
k_1 &= hf(t_i, w_i),\\
k_2 &= hf\left(t_i + \frac{h}{2}, w_i + \frac{1}{2}k_1\right),\\
k_3 &= hf\left(t_i + \frac{h}{2}, w_i + \frac{1}{2}k_2\right),\\
k_4 &= hf(t_{i+1}, w_i + k_3),\\
w_{i+1} &= w_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4),
\end{aligned}
$$

for each $i = 0, 1, \ldots, N-1$. This method has local truncation error $O(h^4)$, provided the solution $y(t)$ has five continuous derivatives. We introduce the notation k_1, k_2, k_3, k_4 into the method is to eliminate the need for successive nesting in the second variable of $f(t, y)$. Exercise 32 shows how complicated this nesting becomes.

Algorithm 5.2 implements the Runge-Kutta method of order four.

ALGORITHM
5.2

Runge-Kutta (Order Four)

To approximate the solution of the initial-value problem

$$y' = f(t, y), \quad a \le t \le b, \quad y(a) = \alpha,$$

at $(N + 1)$ equally spaced numbers in the interval $[a, b]$:

INPUT endpoints a, b; integer N; initial condition α.

OUTPUT approximation w to y at the $(N + 1)$ values of t.

Step 1 Set $h = (b - a)/N$;
$t = a$;
$w = \alpha$;
OUTPUT (t, w).

Step 2 For $i = 1, 2, \ldots, N$ do Steps 3–5.

Step 3 Set $K_1 = hf(t, w)$;
$K_2 = hf(t + h/2, w + K_1/2)$;
$K_3 = hf(t + h/2, w + K_2/2)$;
$K_4 = hf(t + h, w + K_3)$.

Step 4 Set $w = w + (K_1 + 2K_2 + 2K_3 + K_4)/6$; (*Compute* w_i.)
$t = a + ih$. (*Compute* t_i.)

Step 5 OUTPUT (t, w).

Step 6 STOP. ■

Example 3 Use the Runge-Kutta method of order four with $h = 0.2$, $N = 10$, and $t_i = 0.2i$ to obtain approximations to the solution of the initial-value problem

$$y' = y - t^2 + 1, \quad 0 \le t \le 2, \quad y(0) = 0.5.$$

Solution The approximation to $y(0.2)$ is obtained by

$$w_0 = 0.5$$

$$k_1 = 0.2f(0, 0.5) = 0.2(1.5) = 0.3$$

$$k_2 = 0.2f(0.1, 0.65) = 0.328$$

$$k_3 = 0.2f(0.1, 0.664) = 0.3308$$

$$k_4 = 0.2f(0.2, 0.8308) = 0.35816$$

$$w_1 = 0.5 + \frac{1}{6}(0.3 + 2(0.328) + 2(0.3308) + 0.35816) = 0.8292933.$$

The remaining results and their errors are listed in Table 5.8. ■

Table 5.8

t_i	Exact $y_i = y(t_i)$	Runge-Kutta Order Four w_i	Error $\lvert y_i - w_i \rvert$
0.0	0.5000000	0.5000000	0
0.2	0.8292986	0.8292933	0.0000053
0.4	1.2140877	1.2140762	0.0000114
0.6	1.6489406	1.6489220	0.0000186
0.8	2.1272295	2.1272027	0.0000269
1.0	2.6408591	2.6408227	0.0000364
1.2	3.1799415	3.1798942	0.0000474
1.4	3.7324000	3.7323401	0.0000599
1.6	4.2834838	4.2834095	0.0000743
1.8	4.8151763	4.8150857	0.0000906
2.0	5.3054720	5.3053630	0.0001089

Because of the penalty in terms of function evaluations that must be paid if the steps are repeated, q tends to be chosen conservatively. In fact, for the Runge-Kutta-Fehlberg method with $n = 4$, a common choice is

$$q = \left(\frac{\varepsilon h}{2|\tilde{w}_{i+1} - w_{i+1}|}\right)^{1/4} = 0.84\left(\frac{\varepsilon h}{|\tilde{w}_{i+1} - w_{i+1}|}\right)^{1/4}.$$

In Algorithm 5.3 for the Runge-Kutta-Fehlberg method, Step 9 is added to eliminate large modifications in step size. This is done to avoid spending too much time with small step sizes in regions with irregularities in the derivatives of y, and to avoid large step sizes, which can result in skipping sensitive regions between the steps. The step-size increase procedure could be omitted completely from the algorithm, and the step-size decrease procedure used only when needed to bring the error under control.

ALGORITHM
5.3

Runge-Kutta-Fehlberg

To approximate the solution of the initial-value problem

$$y' = f(t, y), \quad a \le t \le b, \quad y(a) = \alpha,$$

with local truncation error within a given tolerance:

INPUT endpoints a, b; initial condition α; tolerance TOL; maximum step size $hmax$; minimum step size $hmin$.

OUTPUT t, w, h where w approximates $y(t)$ and the step size h was used, or a message that the minimum step size was exceeded.

Step 1 Set $t = a$;
$w = \alpha$;
$h = hmax$;
$FLAG = 1$;
OUTPUT (t, w).

Step 2 While $(FLAG = 1)$ do Steps 3–11.

Step 3 Set $K_1 = hf\,(t, w)$;

$$K_2 = hf\left(t + \tfrac{1}{4}h, w + \tfrac{1}{4}K_1\right);$$
$$K_3 = hf\left(t + \tfrac{3}{8}h, w + \tfrac{3}{32}K_1 + \tfrac{9}{32}K_2\right);$$
$$K_4 = hf\left(t + \tfrac{12}{13}h, w + \tfrac{1932}{2197}K_1 - \tfrac{7200}{2197}K_2 + \tfrac{7296}{2197}K_3\right);$$
$$K_5 = hf\left(t + h, w + \tfrac{439}{216}K_1 - 8K_2 + \tfrac{3680}{513}K_3 - \tfrac{845}{4104}K_4\right);$$
$$K_6 = hf\left(t + \tfrac{1}{2}h, w - \tfrac{8}{27}K_1 + 2K_2 - \tfrac{3544}{2565}K_3 + \tfrac{1859}{4104}K_4 - \tfrac{11}{40}K_5\right).$$

Step 4 Set $R = \frac{1}{h}|\frac{1}{360}K_1 - \frac{128}{4275}K_3 - \frac{2197}{75240}K_4 + \frac{1}{50}K_5 + \frac{2}{55}K_6|$.
(*Note:* $R = \frac{1}{h}|\tilde{w}_{i+1} - w_{i+1}|$.)

Step 5 If $R \le TOL$ then do Steps 6 and 7.

Step 6 Set $t = t + h$; (*Approximation accepted.*)

$$w = w + \tfrac{25}{216}K_1 + \tfrac{1408}{2565}K_3 + \tfrac{2197}{4104}K_4 - \tfrac{1}{5}K_5.$$

Step 7 OUTPUT (t, w, h).

Step 8 Set $\delta = 0.84(TOL/R)^{1/4}$.

Step 9 If $\delta \le 0.1$ then set $h = 0.1h$
else if $\delta \ge 4$ then set $h = 4h$
else set $h = \delta h$. (*Calculate new h.*)

Step 10 If $h > hmax$ then set $h = hmax$.

Step 11 If $t \ge b$ then set $FLAG = 0$
else if $t + h > b$ then set $h = b - t$
else if $h < hmin$ then
set $FLAG = 0$;
OUTPUT ('*minimum h exceeded*').
(*Procedure completed unsuccessfully.*)

Step 12 (*The procedure is complete.*)
STOP. ■

Example 1 Use the Runge-Kutta-Fehlberg method with a tolerance $TOL = 10^{-5}$, a maximum step size $hmax = 0.25$, and a minimum step size $hmin = 0.01$ to approximate the solution to the initial-value problem

$$y' = y - t^2 + 1, \quad 0 \le t \le 2, \quad y(0) = 0.5,$$

and compare the results with the exact solution $y(t) = (t + 1)^2 - 0.5e^t$.

Solution We will work through the first step of the calculations and then apply Algorithm 5.3 to determine the remaining results. The initial condition gives $t_0 = 0$ and $w_0 = 0.5$. To determine w_1 using $h = 0.25$, the maximum allowable stepsize, we compute

$$k_1 = hf(t_0, w_0) = 0.25\left(0.5 - 0^2 + 1\right) = 0.375;$$

$$k_2 = hf\left(t_0 + \frac{1}{4}h, w_0 + \frac{1}{4}k_1\right) = 0.25\left(\frac{1}{4}0.25, 0.5 + \frac{1}{4}0.375\right) = 0.3974609;$$

$$k_3 = hf\left(t_0 + \frac{3}{8}h, w_0 + \frac{3}{32}k_1 + \frac{9}{32}k_2\right)$$

$$= 0.25\left(0.09375, 0.5 + \frac{3}{32}0.375 + \frac{9}{32}0.3974609\right) = 0.4095383;$$

$$k_4 = hf\left(t_0 + \frac{12}{13}h, w_0 + \frac{1932}{2197}k_1 - \frac{7200}{2197}k_2 + \frac{7296}{2197}k_3\right)$$

$$= 0.25\left(0.2307692, 0.5 + \frac{1932}{2197}0.375 - \frac{7200}{2197}0.3974609 + \frac{7296}{2197}0.4095383\right)$$

$$= 0.4584971;$$

$$k_5 = hf\left(t_0 + h, w_0 + \frac{439}{216}k_1 - 8k_2 + \frac{3680}{513}k_3 - \frac{845}{4104}k_4\right)$$

$$= 0.25\left(0.25, 0.5 + \frac{439}{216}0.375 - 8(0.3974609) + \frac{3680}{513}0.4095383 - \frac{845}{4104}0.4584971\right)$$

$$= 0.4658452;$$