Homework 4 Solutions

Igor Yanovsky (Math 151B TA)

Section 5.9, Problem 2(a): Use the Runge-Kutta method for systems to approximate the solutions of first-order differential equation

$$\begin{array}{ll} u_1' = u_1 - u_2 + 2, & u_1(0) = -1, \\ u_2' = -u_1 + u_2 + 4t, & u_2(0) = 0; \\ \quad 0 \le t \le 1; \quad h = 0.1, & \end{array} \tag{1}$$

and compare the result to the actual solution

$$\begin{aligned} u_1(t) &= -\frac{1}{2}e^{2t} + t^2 + 2t - \frac{1}{2}, \\ u_2(t) &= \frac{1}{2}e^{2t} + t^2 - \frac{1}{2}. \end{aligned}$$

Solution: For a system of two differential equations

$$\begin{array}{ll} u_1' = f_1(t, u_1, u_2), & u_1(a) = \alpha_1, \\ u_2' = f_2(t, u_1, u_2), & u_2(a) = \alpha_2, \\ \quad a \le t \le b, & \end{array} \tag{2}$$

the fourth-order Runge-Kutta method is

$$\begin{aligned} k_{1,1} &= hf_1\big(t_i, w_{1,i}, w_{2,i}\big), \\ k_{1,2} &= hf_2\big(t_i, w_{1,i}, w_{2,i}\big), \\ \\ k_{2,1} &= hf_1\left(t_i + \frac{h}{2}, w_{1,i} + \frac{k_{1,1}}{2}, w_{2,i} + \frac{k_{1,2}}{2}\right), \\ k_{2,2} &= hf_2\left(t_i + \frac{h}{2}, w_{1,i} + \frac{k_{1,1}}{2}, w_{2,i} + \frac{k_{1,2}}{2}\right), \\ \\ k_{3,1} &= hf_1\left(t_i + \frac{h}{2}, w_{1,i} + \frac{k_{2,1}}{2}, w_{2,i} + \frac{k_{2,2}}{2}\right), \\ k_{3,2} &= hf_2\left(t_i + \frac{h}{2}, w_{1,i} + \frac{k_{2,1}}{2}, w_{2,i} + \frac{k_{2,2}}{2}\right), \\ \\ k_{4,1} &= hf_1\big(t_i + h, w_{1,i} + k_{3,1}, w_{2,i} + k_{3,2}\big), \\ k_{4,2} &= hf_2\big(t_i + h, w_{1,i} + k_{3,1}, w_{2,i} + k_{3,2}\big), \\ \\ w_{1,i+1} &= w_{1,i} + \frac{1}{6}\big(k_{1,1} + 2k_{2,1} + 2k_{3,1} + k_{4,1}\big), \\ w_{2,i+1} &= w_{2,i} + \frac{1}{6}\big(k_{1,2} + 2k_{2,2} + 2k_{3,2} + k_{4,2}\big), \end{aligned} \tag{3}$$

where $u_1(t_i) \approx w_{1,i}$, $u_2(t_i) \approx w_{2,i}$.

NOTE THAT THE BOOK USES THE "i,j" NOTATION INCONSISTENTLY. COMPARE, FOR EXAMPLE, FORMULAS (5.48)-(5.52) WITH ALGORITHM 5.7 ON PAGE

316.

Running the code gives:
$N = 10, h = 1.0000000e-001, t = 1.00,$
$w1 = -1.1944446208e+000, w2 = 4.1944446208e+000,$
$u1 = -1.1945280495e+000, u2 = 4.1945280495e+000,$
$err1 = 8.3428635596e-005, err2 = 8.3428635596e-005.$

Running the code with $h = 0.1$ and $h = 0.05$, gives the convergence rate of $p \approx 3.88$ for each of the two ODEs.
Running the code with $h = 0.01$ and $h = 0.005$, gives the convergence rate of $p \approx 3.99$ for each of the two ODEs. This verifies that our algorithm is indeed fourth-order accurate.

Section 5.9, Problem 3(b): Use the Runge-Kutta for Systems Algorithm to approximate the solution of the following higher-order differential equation

$$\begin{aligned}
&t^2 y'' - 2ty' + 2y = t^3 \log t, \\
&y(1) = 1, \quad y'(1) = 0, \\
&1 \le t \le 2, \\
&\text{with } h = 0.1,
\end{aligned} \tag{4}$$

and compare the result to the actual solution

$$y(t) = \frac{7}{4}t + \frac{1}{2}t^3 \log t - \frac{3}{4}t^3. \tag{5}$$

Solution: We will convert the second order differential equation (4) into a system of two first order differential equations.

Let

$$\begin{aligned}
u_1(t) &= y(t), \\
u_2(t) &= y'(t).
\end{aligned}$$

Then,

$$\begin{aligned}
u_1' &= y' = u_2, \\
u_2' &= y'' = \frac{t^3 \log t + 2ty' - 2y}{t^2} = \frac{t^3 \log t + 2tu_2 - 2u_2}{t^2}
\end{aligned}$$

is a first order system of two differential equations with initial conditions

$$\begin{aligned}
u_1(1) &= y(1) = 1, \\
u_2(1) &= y'(1) = 1.
\end{aligned}$$

Exact solution (5) for the system above can be written as

$$\begin{aligned}
u_1(t) &= y(t) = \frac{7}{4}t + \frac{1}{2}t^3 \log t - \frac{3}{4}t^3, \\
u_2(t) &= y'(t) = \frac{7}{4} + \frac{3}{2}t^2 \log t - \frac{7}{4}t^2.
\end{aligned}$$

Running the Runge-Kutta fourth-order code for systems, gives

$N = 10, h = 1.0000000e-001, t = 2.00,$
$w1 = 2.7258237314e-001, w2 = -1.0911211887e+000,$
$u1 = 2.7258872224e-001, u2 = -1.0911169166e+000,$
$err1 = 6.3491028489e-006, err2 = 4.2720262261e-006.$

Running the code with $h = 0.1$ and $h = 0.05$, gives the convergence rate of $p \approx 3.97$ for the first ODE and $p \approx 4.02$ for the second ODE.
Running the code with $h = 0.01$ and $h = 0.005$, gives the convergence rate of $p \approx 4.00$ for each of the two ODEs. This verifies that our algorithm is indeed fourth-order accurate.

Theorem 5.20, (i): Suppose the initial-value problem

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

is approximated by a one-step difference method in the form

$$w_0 = \alpha, \quad w_{i+1} = w_i + h\phi(t_i, w_i, h).$$

Suppose also that a number $h_0 > 0$ exists and that $\phi(t, w, h)$ is continuous and satisfies a Lipschitz condition in the variable w with Lipschitz constant L. Then the method is stable.

Section 5.10, Problem 1: To prove Theorem 5.20, part (i), show that the hypothesis imply that a constant $K > 0$ exists such that

$$|u_i - v_i| \leq K|u_0 - v_0|, \quad \text{for each } 1 \leq i \leq N, \tag{6}$$

whenever $\{u_i\}_{i=1}^N$ and $\{v_i\}_{i=1}^N$ satisfy the difference equation $w_{i+1} = w_i + h\phi(t_i, w_i, h)$.

Solution: We have

$$\begin{aligned} u_{i+1} &= u_i + h\phi(t_i, u_i, h), \\ v_{i+1} &= v_i + h\phi(t_i, v_i, h). \end{aligned} \tag{7}$$

Subtracting the second equation in (7) from the first, we obtain

$$u_{i+1} - v_{i+1} = u_i - v_i + h(\phi(t_i, u_i, h) - \phi(t_i, v_i, h)).$$

According to one of the hypothesis of Theorem 5.20, $\phi(t, w, h)$ is continuous and satisfies a Lipschitz condition in the variable w with Lipschitz constant L:

$$|\phi(t, w_1, h) - \phi(t, w_2, h)| \leq L|w_1 - w_2|.$$

Thus,

$$\begin{aligned} |u_{i+1} - v_{i+1}| &= |u_i - v_i + h(\phi(t_i, u_i, h) - \phi(t_i, v_i, h))| \\ &\leq |u_i - v_i| + h|\phi(t_i, u_i, h) - \phi(t_i, v_i, h)| \\ &\leq |u_i - v_i| + hL|u_i - v_i| \\ &= (1 + hL)|u_i - v_i| \\ &= (1 + hL)^{n+1}|u_0 - v_0|. \end{aligned}$$

Thus, $|u_i - v_i| \leq K|u_0 - v_0|$, where $K = (1 + hL)^n$.

Section 5.10, Problem 2:
For the Adams-Bashforth and Adams-Moulton methods of order four,
a) Show that if $f = 0$, then

$$F(t_i, h, w_{i+1}, \ldots, w_{i+1-m}) = 0.$$

b) Show that if f satisfies a Lipschitz condition with constant L, then a constant C exists with

$$|F(t_i, h, w_{i+1}, \ldots, w_{i+1-m}) - F(t_i, h, v_{i+1}, \ldots, v_{i+1-m})| \leq C \sum_{j=0}^{m} |w_{i+1-j} - v_{i+1-j}|.$$

Solution: **a)** The **Adams-Bashforth** fourth-order method can be expressed as

$$\begin{aligned}
&w_{i+1} = w_i + F(t_i, h, w_{i+1}, w_i, w_{i-1}, w_{i-2}, w_{i-3}),\\
&F(t_i, h, w_{i+1}, w_i, w_{i-1}, w_{i-2}, w_{i-3})\\
&\quad = \frac{h}{24}\big[55f(t_i, w_i) - 59f(t_{i-1}, w_{i-1}) + 37f(t_{i-2}, w_{i-2}) - 9f(t_{i-3}, w_{i-3})\big].
\end{aligned}$$

If $f = 0$, then $F(t_i, h, w_{i+1}, w_i, w_{i-1}, w_{i-2}, w_{i-3}) = 0$. ✓.

The **Adams-Moulton** fourth-order method can be expressed as

$$\begin{aligned}
&w_{i+1} = w_i + F(t_i, h, w_{i+1}, w_i, w_{i-1}, w_{i-2}),\\
&F(t_i, h, w_{i+1}, w_i, w_{i-1}, w_{i-2})\\
&\quad = \frac{h}{24}\big[9f(t_{i+1}, w_{i+1}) + 19f(t_i, w_i) - 5f(t_{i-1}, w_{i-1}) + f(t_{i-2}, w_{i-2})\big].
\end{aligned}$$

If $f = 0$, then $F(t_i, h, w_{i+1}, w_i, w_{i-1}, w_{i-2}) = 0$. ✓.

b) Function f satisfies a Lipschitz condition with constant L:

$$|f(t, w_1) - f(t, w_2)| \leq L|w_1 - w_2|.$$

Thus, for the **Adams-Bashforth** fourth-order method, we have

$$\begin{aligned}
&|F(t_i, h, w_{i+1}, w_i, w_{i-1}, w_{i-2}, w_{i-3}) - F(t_i, h, v_{i+1}, v_i, v_{i-1}, v_{i-2}, v_{i-3})|\\
&= \frac{h}{24}\Big|\big[55f(t_i, w_i) - 59f(t_{i-1}, w_{i-1}) + 37f(t_{i-2}, w_{i-2}) - 9f(t_{i-3}, w_{i-3})\big]\\
&\qquad -\big[55f(t_i, v_i) - 59f(t_{i-1}, v_{i-1}) + 37f(t_{i-2}, v_{i-2}) - 9f(t_{i-3}, v_{i-3})\big]\Big|\\
&\leq \frac{55h}{24}|f(t_i, w_i) - f(t_i, v_i)| + \frac{59h}{24}|f(t_{i-1}, w_{i-1}) - f(t_{i-1}, v_{i-1})|\\
&\quad + \frac{37h}{24}|f(t_{i-2}, w_{i-2}) - f(t_{i-2}, v_{i-2})| + \frac{9h}{24}|f(t_{i-3}, w_{i-3}) - f(t_{i-3}, v_{i-3})|\\
&\leq \frac{55h}{24}L_1|w_i - v_i| + \frac{59h}{24}L_2|w_{i-1} - v_{i-1}| + \frac{37h}{24}L_3|w_{i-2} - v_{i-2})| + \frac{9h}{24}L_4|w_{i-3} - v_{i-3}|.
\end{aligned}$$

Let $C = \max\left(\frac{55h}{24}L_1, \frac{59h}{24}L_2, \frac{37h}{24}L_3, \frac{9h}{24}L_4\right)$. Then,

$$\begin{aligned}
&|F(t_i, h, w_{i+1}, w_i, w_{i-1}, w_{i-2}, w_{i-3}) - F(t_i, h, v_{i+1}, v_i, v_{i-1}, v_{i-2}, v_{i-3})|\\
&\leq C|w_i - v_i| + C|w_{i-1} - v_{i-1}| + C|w_{i-2} - v_{i-2})| + C|w_{i-3} - v_{i-3}|\\
&= C\sum_{j=1}^{4} |w_{i+1-j} - v_{i+1-j}|. \quad ✓
\end{aligned}$$

Similarly, for the **Adams-Moulton** fourth-order method, we have

$$\begin{aligned}
&|F(t_i, h, w_{i+1}, w_i, w_{i-1}, w_{i-2}) - F(t_i, h, v_{i+1}, v_i, v_{i-1}, v_{i-2})|\\
&\leq C\sum_{j=0}^{3} |w_{i+1-j} - v_{i+1-j}|. \quad ✓
\end{aligned}$$

Section 5.10, Problem 4:
Consider the differential equation

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

a) Show that

$$y'(t_i) = \frac{-3y(t_i) + 4y(t_{i+1}) - y(t_{i+2})}{2h} + \frac{h^2}{3} y'''(\xi_i),$$

for some ξ, where $t_i < \xi_i < t_{i+2}$.

d) Analyze the following method

$$w_{i+2} = 4w_{i+1} - 3w_i - 2hf(t_i, w_i) \tag{8}$$

for consistency, stability, and convergence.

Solution: **a)** We want to show that

$$y'(t_i) - \frac{1}{2h}\big[- 3y(t_i) + 4y(t_{i+1}) - y(t_{i+2})\big] = \frac{h^2}{3} y'''(\xi_i). \tag{9}$$

Expanding $y(t_{i+1})$ and $y(t_{i+2})$ in Taylor's series about t_i, we obtain

$$\begin{aligned}
& y'(t_i) - \frac{1}{2h}\big[- 3y(t_i) + 4y(t_{i+1}) - y(t_{i+2})\big] \\
&= y'(t_i) + \frac{3}{2h} y(t_i) \\
&\quad - \frac{4}{2h}\big[y(t_i) + hy'(t_i) + \frac{h^2}{2} y''(t_i) + \frac{h^3}{6} y'''(\xi_{i1})\big] \\
&\quad + \frac{1}{2h}\big[y(t_i) + 2hy'(t_i) + \frac{4h^2}{2} y''(t_i) + \frac{8h^3}{6} y'''(\xi_{i2})\big] \\
&= \frac{h^2}{3} y'''(\xi_i),
\end{aligned}$$

where $t_i < \xi_{i1} < t_{i+1}$, $t_i < \xi_{i2} < t_{i+2}$, and $t_i < \xi_i < t_{i+2}$.

d) The method is consistent if the local truncation error $\tau_i(h) \to 0$ as $h \to 0$. The method in (8) has the following truncation error (in formulas below, we again expand $y(t_{i+1})$ and $y(t_{i+2})$ in Taylor's series about t_i, similar to part (a)):

$$\begin{aligned}
\tau_{i+2}(h) &= \frac{y(t_{i+2}) - 4y(t_{i+1}) + 3y(t_i)}{h} + 2f(t_i, y(t_i)) \\
&= \frac{y(t_{i+2}) - 4y(t_{i+1}) + 3y(t_i)}{h} + 2y'(t_i) = \frac{2h^2}{3} y'''(\xi_i),
\end{aligned}$$

and since $\tau_{i+2}(h) \to 0$ as $h \to 0$, the method in (8) is consistent. ✓

For a multistep method to be stable, it has to satisfy the root condition. A multistep method is said to satisfy the root condition if all roots λ_i of the characteristic polynomial $P(\lambda)$ (for a general form of $P(\lambda)$ see equation (5.57) in the book) are such that $|\lambda_i| \le 1$, and if $|\lambda_i| = 1$, then λ_i is simple.
The characteristic polynomial of the following multistep method

$$w_{i+2} - 4w_{i+1} + 3w_i = -2hf(t_i, w_i) \tag{10}$$

is

$$P(\lambda) = \lambda^2 - 4\lambda + 3,$$

which has roots

$$\lambda_1 = 1, \; \lambda_2 = 3.$$

Thus the method in (8) does not satisfy the root condition, and therefore is unstable. ✓
Thus, the method is not convergent. ✓

Note that a quick way of writing a characteristic polynomial is to associate the coefficient a_0 to the leftmost grid point in the method's stencil. In the example above, the leftmost grid point has an index i, and therefore, $a_0 = 3$, $a_1 = -4$.

Section 5.10, Problem 8:
Consider the problem $y' = 0$, $0 \le t \le 10$, $y(0) = 0$, which has the solution $y \equiv 0$. If the difference method of Exercise 4 is applied to the problem, then

$$\begin{aligned} w_{i+1} &= 4w_i - 3w_{i-1}, \quad \text{for } i = 1, 2, \ldots, N-1, \\ & \quad w_0 = 0, \quad \text{and} \quad w_1 = \alpha_1. \end{aligned}$$

Suppose $w_1 = \alpha_1 = \varepsilon$, where ε is a small rounding error. Compute w_i exactly for $i = 2, 3, 4, 5, 6$ to find how the error ε is propagated.

Solution: The reason for this exercise is to show that a method, even though it is consistent, will not produce reasonable results unless it is stable. We have

$$\begin{aligned} w_0 &= 0, \\ w_1 &= \varepsilon, \\ w_2 &= 4w_1 - 3w_0 = 4\varepsilon, \\ w_3 &= 4w_2 - 3w_1 = 16\varepsilon - 3\varepsilon = 13\varepsilon, \\ w_4 &= 4w_3 - 3w_2 = 52\varepsilon - 12\varepsilon = 40\varepsilon, \\ w_5 &= 4w_4 - 3w_3 = 160\varepsilon - 39\varepsilon = 121\varepsilon, \\ w_6 &= 4w_5 - 3w_4 = 484\varepsilon - 120\varepsilon = 364\varepsilon. \end{aligned}$$

Thus, the error grows fast with each iteration for this multistep method due to method's instability.