Math 5604 Homework 4

Jacob Hauck

March 8, 2024

Problem 1.

Consider

$$y' = \left(1 - 2t^3\right) y^2, \quad t > 0; \qquad y(0) = 1. \tag{1}$$

(a) In order to apply the second-order Taylor series method, we need to use the ODE (1) to find y'' in terms of y:

$$y'' = \frac{\mathrm{d}}{\mathrm{d}t}\left[\left(1 - 2t^3\right) y^2\right] = -6t^2 y^2 + 2\left(1 - 2t^3\right) y y' = -6t^2 y^2 + 2\left(1 - 2t^3\right)^2 y^3.$$

Then the second-order Taylor series method is given by

$$\begin{cases} y^{n+1} = y^n + k\left(1 - 2t_n^3\right)\left(y^n\right)^2 + \frac{k^2}{2}\left[-6t_n^2\left(y^n\right)^2 + 2\left(1 - 2t_n^3\right)^2\left(y^n\right)^3\right], & n = 0, 1, 2, \dots \\ y^0 = 1. \end{cases}$$

This method is implemented in `ts2.m`.

(b) The recursive rule for the two-step Adams-Bashforth method is given by

$$y^{n+1} = y^n + k\left[\frac{3}{2} f\left(t_n, y^n\right) - \frac{1}{2} f\left(t_{n-1}, y^{n-1}\right)\right], \quad n \geq 0,$$

where, in our case, $f(t, y) = \left(1 - 2t^3\right) y^2$. We use the forward Euler method to obtain y^1, as the forward Euler method has second-order local truncation error. Thus, our scheme is

$$\begin{cases} y^{n+1} = y^n + k\left[\frac{3}{2}\left(1 - 2t_n^3\right)\left(y^n\right)^2 - \frac{1}{2}\left(1 - 2t_{n-1}^3\right)\left(y^{n-1}\right)^2\right] & n = 1, 2, 3, \dots \\ y^1 = y^0 + k\left(1 - 2t_0^3\right)\left(y^0\right)^2 \\ y^0 = 1. \end{cases}$$

This method is implemented in `ab2.m`.

(c) The recursive rule for the trapezoidal method is given by

$$y^{n+1} = y^n + k\left[\frac{1}{2} f\left(t_n, y^n\right) + \frac{1}{2} f\left(t_{n+1}, y^{n+1}\right)\right], \quad n \geq 0,$$

where, in our case, $f(t, y) = \left(1 - 2t^3\right) y^2$. Then our scheme is given implicitly by

$$\begin{cases} y^{n+1} = y^n + \frac{k}{2}\left[\left(1 - 2t_n^3\right)\left(y^n\right)^2 + \left(1 - 2t_{n+1}^3\right)\left(y^{n+1}\right)^2\right] & n = 0, 1, 2, \dots \\ y^0 = 1. \end{cases}$$

In order to solve the implicit equation for y^{n+1}, we can equivalently use Newton's method to find the root of

$$f_n(y) = y - y^n - \frac{k}{2}\left[\left(1 - 2t_n^3\right)\left(y^n\right)^2 + \left(1 - 2t_{n+1}^3\right) y^2\right], \quad n = 0, 1, 2, \dots$$

We will need f_n' to use Newton's method:

$$f_n'(y) = 1 - k(1 - 2t_{n+1}^3)y.$$

This method is implemented in `tp.m` and uses the implementation of Newton's method in `newton.m`.

(d) The recursive rule for the midpoint method is given by

$$y^{n+1} = y^n + kf\left(t_n + \frac{k}{2}, \frac{y^n + y^{n+1}}{2}\right), \quad n \geq 0,$$

where, in our case, $f(t, y) = \left(1 - 2t^3\right) y^2$. Then our scheme is given implicitly by

$$\begin{cases} y^{n+1} = y^n + k\left(1 - 2\left(t_n + \frac{k}{2}\right)^3\right)\left(\frac{y^n + y^{n+1}}{2}\right)^2 & N = 0, 1, 2, \ldots \\ y^0 = 1. \end{cases}$$

To solve the implicit equation for y^{n+1}, we can equivalently use Newton's method to find the root of

$$f_n(y) = y - y^n - k\left(1 - 2\left(t_n + \frac{k}{2}\right)^3\right)\left(\frac{y^n + y}{2}\right)^2, \quad n = 0, 1, 2, \ldots$$

To use Newton's method, we need f_n':

$$f_n'(y) = 1 - \frac{k}{2}\left(1 - 2\left(t_n + \frac{k}{2}\right)^3\right)(y^n + y).$$

This method is implemented in `mp.m` and uses the implementation of Newton's method in `newton.m`.

(e) To compare the above methods with the exact solution of (1), we first need to determine the exact solution. Using separation of variables, we have

$$\frac{y'}{y^2} = 1 - 2t^3 \implies -y^{-1} = t - \frac{t^4}{2} + C, \quad \text{some C} \in \mathbf{R}.$$

Since $y(0) = 1$, it follows that $C = -1$, so

$$y(t) = \frac{1}{\frac{t^4}{2} - t + 1}$$

is the exact solution of the (1).

Using the code in `problem1_calculations.m`, we run the above four methods with various step sizes and compute the error at $t = 2$. The results can be found in `p1_output.txt` and are summarized in Table 1.

(f) The code to plot the errors in Table 1 on a log-log plot can be found in `problem1_calculations.m`. The resulting plot is given in Figure 1.

Observations

- By comparison with the reference line k^2, we see that all four methods have second-order error, which agrees with the theoretical predictions we had for these methods.
- For the largest step size, there is a significant deviation in the error trend for the second-order Taylor series (TS2) and two-step Adams-Bashforth (AB2) methods. This may be due to the fact that these methods are explicit and less stable than the implicit midpoint (MP) and trapezoidal (TP) methods.

	TS2		TP		AB2		MP	
k	Error	Rate	Error	Rate	Error	Rate	Error	Rate
1/4	2.3989e-03	-	6.2704e-03	-	1.8370	-	1.1923e-02	-
1/8	4.3209e-03	-0.8489	1.4774e-03	2.0854	6.9392e-03	8.0483	2.9800e-03	2.0004
1/16	8.8105e-04	2.2940	3.6416e-04	2.0204	1.8133e-03	1.9361	7.4426e-04	2.0014
1/32	1.9901e-04	2.1463	9.0720e-05	2.0050	4.4886e-04	2.0142	1.8601e-04	2.0004
1/64	4.7427e-05	2.0691	2.2660e-05	2.0012	1.1053e-04	2.0217	4.6499e-05	2.0001
1/128	1.1585e-05	2.0333	5.6638e-06	2.0003	2.7368e-05	2.0139	1.1624e-05	2.0000
1/256	2.8636e-06	2.0163	1.4158e-06	2.0000	6.8058e-06	2.0076	2.9061e-06	2.0000

Table 1: Numerical errors and convergence rates at $t = 2$

Problem 2.

Adams-Bashforth

One step of the three-step Adams-Bashforth method is given by

$$y^{n+3} = y^{n+2} + k\left(b_2 f(t_{n+2}, y^{n+2}) + b_1 f(t_{n+1}, y^{n+1}) + b_0 f(t_n, y^n)\right),$$

where b_0, b_1, and b_2 are chosen to minimize the order of the local truncation error. Hence, we can find these values by calculating the local truncation error and minimizing its order. Assume that $y^n = y(t_n)$, $y^{n+1} = y(t_{n+1})$, and $y^{n+2} = y(t_{n+2})$. Then the local truncation error is given by

$$y^{n+3} - y(t_{n+3}) = y(t_{n+2}) + k\left(b_2 y'(t_{n+2}) + b_1 y'(t_{n+1}) + b_0 y'(t_n)\right) - y(t_{n+3}).$$

Using Taylor expansion about t_n gives

$$\begin{aligned}
y^{n+3} - y(t_{n+3}) &= y(t_n) + 2ky'(t_n) + \frac{4k^2}{2}y''(t_n) + \frac{8k^3}{6}y'''(t_n) + \mathcal{O}(k^4) \\
&\quad + kb_2y'(t_n) + 2k^2b_2y''(t_n) + \frac{4k^3}{2}b_2y'''(t_n) + \mathcal{O}(k^4) \\
&\quad + b_1ky'(t_n) + b_1k^2y''(t_n) + \frac{k^3}{2}b_1y'''(t_n) + \mathcal{O}(k^4) \\
&\quad + b_0ky'(t_n) - y(t_n) - 3ky'(t_n) - \frac{9k^2}{2}y''(t_n) - \frac{27k^3}{6}y'''(t_n) + \mathcal{O}(k^4) \\
&= (b_2 + b_1 + b_0 - 1)ky'(t_n) + \left(2b_2 + b_1 - \frac{5}{2}\right)k^2y''(t_n) + \left(2b_2 + \frac{1}{2}b_1 - \frac{19}{6}\right)k^3y'''(t_n) + \mathcal{O}(k^4).
\end{aligned}$$

Thus, the minimum possible order of the local truncation error is 4, which occurs only if

$$\begin{aligned}
b_0 + b_1 + b_2 &= 1 \\
2b_2 + b_1 &= \frac{5}{2} \\
2b_2 + \frac{1}{2}b_1 &= \frac{19}{6}.
\end{aligned}$$

Solving the last two equations for b_1 and b_2 gives $b_1 = -\frac{4}{3}$, and $b_2 = \frac{23}{12}$. Substituting into the first equations gives $b_0 = \frac{5}{12}$. Therefore, the three-step Adams-Bashforth method is

$$y^{n+3} = y^{n+2} + \frac{k}{12}\left(23f(t_{n+2}, y^{n+2}) - 16f(t_{n+1}, y^{n+1}) + 5f(t_n, y^n)\right).$$

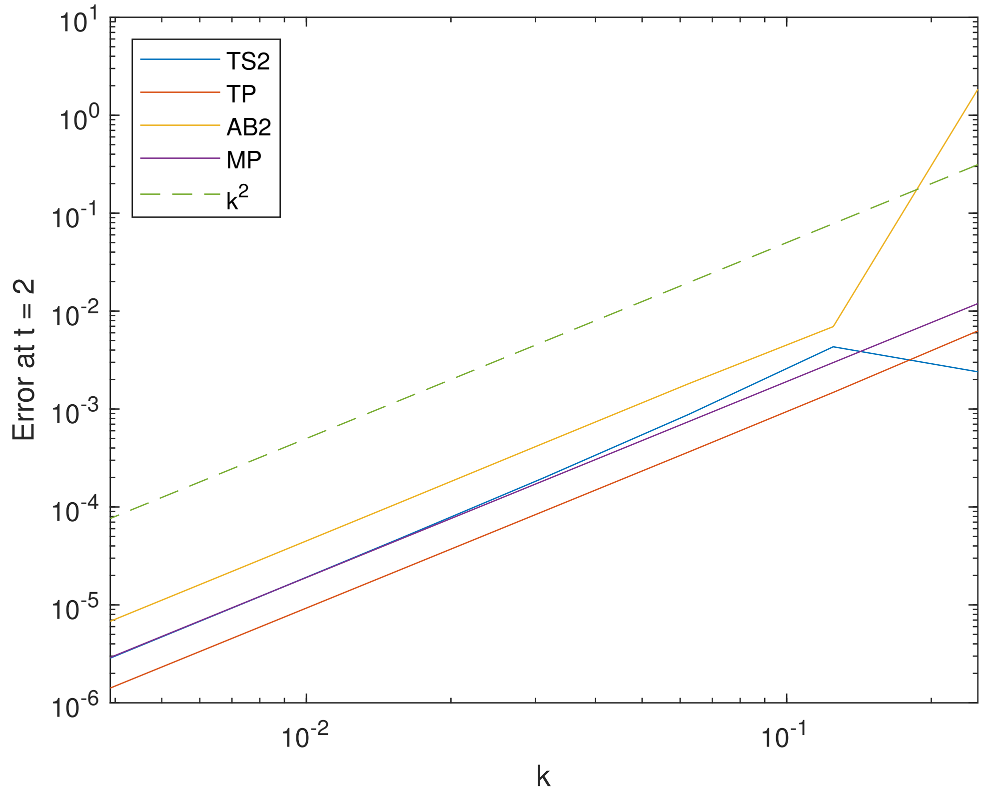

Figure 1: Numerical errors at $t = 2$ versus time step

Adams-Moulton

One step of the three-step Adams-Moulton method is given by

$$y^{n+3} = y^{n+2} + k\left(b_3 f(t_{n+3}, y^{n+3}) + b_2 f(t_{n+2}, y^{n+2}) + b_1 f(t_{n+1}, y^{n+1}) + b_0 f(t_n, y^n)\right),$$

where b_0, b_1, b_2, and b_3 are chosen to minimize the order of the local truncation error. Hence, we can find these values by calculating the local truncation error and minimizing its order. Assume that $y^n = y(t_n)$, $y^{n+1} = y(t_{n+1})$, and $y^{n+2} = y(t_{n+2})$. Then the local truncation error is given by

$$y^{n+3} - y(t_{n+3}) = y(t_{n+2}) + k\left(b_3 f(t_{n+3}, y^{n+3}) + b_2 y'(t_{n+2}) + b_1 y'(t_{n+1}) + b_0 y'(t_n)\right) - y(t_{n+3}).$$

Using Taylor expansion about t_n gives

$$\begin{aligned}
y^{n+3} - y(t_{n+3}) &= y(t_n) + 2ky'(t_n) + \frac{4k^2}{2}y''(t_n) + \frac{8k^3}{6}y'''(t_n) + \frac{16k^4}{24}y''''(t_n) + \mathcal{O}(k^5) \\
&\quad + kb_3\left(f(t_{n+3}, y^{n+3}) - f(t_{n+3}, y(t_{n+3}))\right) \\
&\quad + kb_3y'(t_n) + 3k^2b_3y''(t_n) + \frac{9k^3}{2}b_3y'''(t_n) + \frac{27k^4}{6}b_3y''''(t_n) + \mathcal{O}(k^5) \\
&\quad + kb_2y'(t_n) + 2k^2b_2y''(t_n) + \frac{4k^3}{2}b_2y'''(t_n) + \frac{8k^4}{6}b_2y''''(t_n) + \mathcal{O}(k^5) \\
&\quad + b_1ky'(t_n) + b_1k^2y''(t_n) + \frac{k^3}{2}b_1y'''(t_n) + \frac{k^4}{6}b_1y''''(t_n) + \mathcal{O}(k^5) \\
&\quad + b_0ky'(t_n) - y(t_n) - 3ky'(t_n) - \frac{9k^2}{2}y''(t_n) - \frac{27k^3}{6}y'''(t_n) - \frac{81k^4}{24}y''''(t_n) + \mathcal{O}(k^5) \\
&= kb_3\left(f(t_{n+3}, y^{n+3}) - f(t_{n+3}, y(t_{n+3}))\right) \\
&\quad + (b_3 + b_2 + b_1 + b_0 - 1)ky'(t_n) \\
&\quad + \left(3b_3 + 2b_2 + b_1 - \frac{5}{2}\right)k^2y''(t_n) \\
&\quad + \left(\frac{9}{2}b_3 + \frac{4}{2}b_2 + \frac{1}{2}b_1 - \frac{19}{6}\right)k^3y'''(t_n) \\
&\quad + \left(\frac{27}{6}b_3 + \frac{8}{6}b_2 + \frac{1}{6}b_1 - \frac{65}{24}\right)k^4y''''(t_n) + \mathcal{O}(k^5).
\end{aligned}$$

Taking the absolute value of both sides and assuming that f is L-Lipschitz in y, we get

$$\begin{aligned}
(1 - kL|b_3|)|y^{n+3} - y(t_{n+3})| \leq\ & |(b_3 + b_2 + b_1 + b_0 - 1)ky'(t_n)| \\
& + \left|\left(3b_3 + 2b_2 + b_1 - \frac{5}{2}\right)k^2y''(t_n)\right| \\
& + \left|\left(\frac{9}{2}b_3 + \frac{4}{2}b_2 + \frac{1}{2}b_1 - \frac{19}{6}\right)k^3y'''(t_n)\right| \\
& + \left|\left(\frac{27}{6}b_3 + \frac{8}{6}b_2 + \frac{1}{6}b_1 - \frac{65}{24}\right)k^4y''''(t_n)\right| + \mathcal{O}(k^5).
\end{aligned}$$

Thus, if k is sufficiently small, we can divide by $1 - kL|b_3|$, in which case we see that the minimum possible order of the local truncation error $y^{n+3} - y(t_{n+3})$ is 5, which is achieved only if

$$\begin{aligned}
b_0 + b_1 + b_2 + b_3 &= 1 \\
b_1 + 2b_2 + 3b_3 &= \frac{5}{2} \\
\frac{1}{2}b_1 + \frac{4}{2}b_2 + \frac{9}{2}b_3 &= \frac{19}{6} \\
\frac{1}{6}b_1 + \frac{8}{6}b_2 + \frac{27}{6}b_3 &= \frac{65}{24}.
\end{aligned}$$

Solving the last three equations using Gaussian elimination gives $b_1 = -\frac{5}{24}$, $b_2 = \frac{19}{24}$, and $b_3 = \frac{9}{24}$. Then $b_0 = \frac{1}{24}$, and the three-step Adams-Moulton method is given by

$$y^{n+3} = y^{n+2} + \frac{k}{24}\left(9f(t_{n+3}, y^{n+3}) + 19f(t_{n+2}, y^{n+2}) - 5f(t_{n+1}, y^{n+1}) + f(t_ny^n)\right).$$

Appendix

Listing 1: Code for second-order Taylor series method (problem 1 (a))

```
function [t, y] = ts2(k)
% Second order Taylor series method for
%   y' = (1-2t^3)y^2,    y(0) = 1
% on the interval [0, 2]
%
% Parameters
% ----------
%   k: the step size
%
% Returns
% -------
%   [t, y]
%   t: n + 1 vector of sample time points
%   y: n + 1 vector of sample solution values

% Initialization
n = ceil(2 / k);
t = linspace(0, 2, n + 1);
y = zeros(1, n + 1);

% initial condition
y(1) = 1;

% Taylor series iteration
for i = 1:n
    dy = (1 - 2*t(i)^3) * y(i)^2;
    ddy = -6*t(i)^2*y(i)^2 + 2*(1 - 2*t(i)^3)*y(i)*dy;
    y(i + 1) = y(i) + k*dy + k^2/2*ddy;
end
```

Listing 2: Code for second-order Adams-Bashforth method (problem 1 (b))

```
function [t, y] = ab2(k)
% Two-step Adams-Bashforth method for
%   y' = (1-2t^3)y^2,    y(0) = 1
% on the interval [0, 2]
%
% Parameters
% ----------
%   k: the step size
%
% Returns
% -------
%   [t, y]
%   t: n + 1 vector of sample time points
%   y: n + 1 vector of sample solution values

% Initialization
n = ceil(2 / k);
t = linspace(0, 2, n + 1);
y = zeros(1, n + 1);

% initial condition
y(1) = 1;

% First step using Forward Euler (locally second order)
y(2) = y(1) + k*(1 - 2*t(1)^3)*y(1)^2;

% Adams-Bashforth iteration
for i = 1:(n-1)
    y(i + 2) = y(i + 1) + k*(3/2 * (1 - 2*t(i+1)^3)*y(i+1)^2 - 1/2 * (1 - 2*t(i)^3)*y(i)^2);
end
```

Listing 3: Code for trapezoidal method (problem 1 (c))

```
function [t, y] = tp(k)
% Trapezoid method for
%   y' = (1-2t^3)y^2,    y(0) = 1
% on the interval [0, 2]
%
% Parameters
% ----------
%   k: the step size
%
% Returns
% -------
%   [t, y]
%   t: n + 1 vector of sample time points
%   y: n + 1 vector of sample solution values

% Initialization
n = ceil(2 / k);
t = linspace(0, 2, n + 1);
y = zeros(1, n + 1);

% initial condition
y(1) = 1;

% Trapezoid rule iteration
for i = 1:n
    f = @(y_) y_ - y(i) - k*((1 - 2*t(i)^3)*y(i)^2 + (1 - 2*t(i + 1)^3)*y_^2)/2;
    f_prime = @(y_) 1 - k*(1-2*t(i + 1)^3)*y_;
    y(i + 1) = newton(f, f_prime, y(i), 100, 1e-9, 0, 0);
end
```

Listing 4: Code for midpoint method (problem 1 (d))

```
function [t, y] = mp(k)
% Midpoint method for
%   y' = (1-2t^3)y^2,    y(0) = 1
% on the interval [0, 2]
%
% Parameters
% ----------
%   k: the step size
%
% Returns
% -------
%   [t, y]
%   t: n + 1 vector of sample time points
%   y: n + 1 vector of sample solution values

% Initialization
n = ceil(2 / k);
t = linspace(0, 2, n + 1);
y = zeros(1, n + 1);

% initial condition
y(1) = 1;

% Midpoint rule iteration
for i = 1:n
    f = @(y_) y_ - y(i) - k*((1 - 2*(t(i) + k/2)^3) * ((y(i) + y_)/2)^2);
    f_prime = @(y_) 1 - k*((1 - 2*(t(i) + k/2)^3) * (y(i) + y_)/2);
    y(i + 1) = newton(f, f_prime, y(i), 100, 1e-9, 0, 0);
end
```