Math 5604 Homework 4

Jacob Hauck

March 6, 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:

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

Then the second-order Taylor series method is given by

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

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

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

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

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

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

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

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.

Problem 2.

	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 $t = 2$

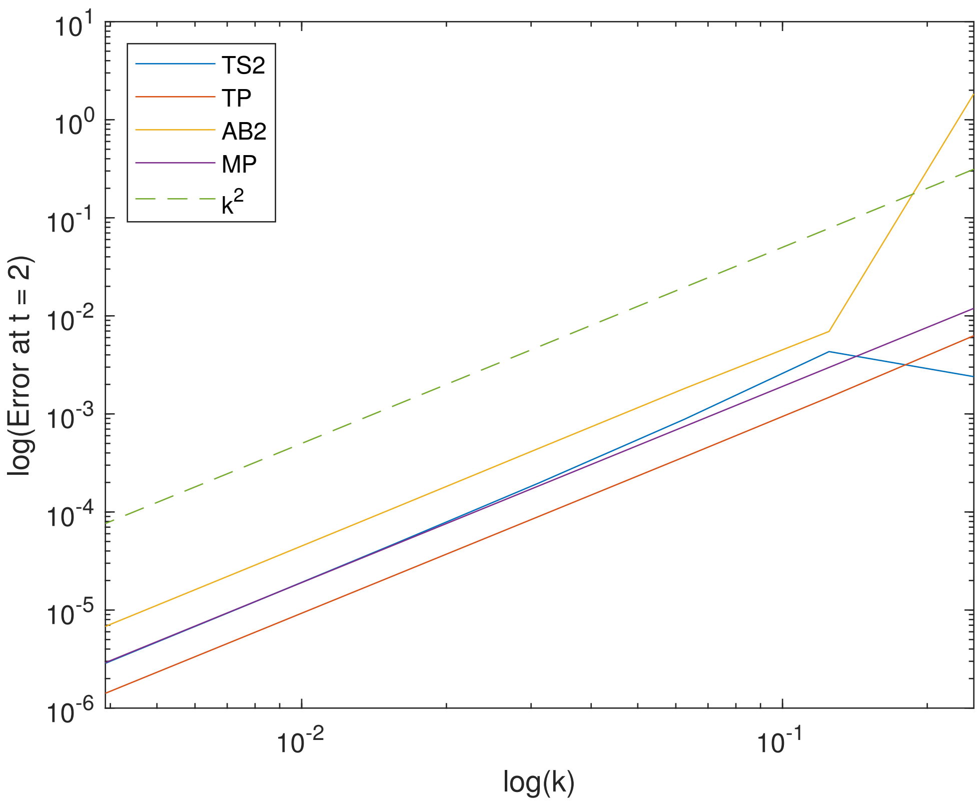

Figure 1: Numerical errors at $t = 2$ versus time step