(5.5) Multistep Methods

Consider the initial-value problem for the ordinary differential equation:
\[
y'(t) = f(t, y), \quad a \leq t \leq b, \quad y(a) = a.
\]

Let \( y(t) \) be the unique solution. In Sections 5.2, 5.3 and 5.4, one-step numerical methods: Euler Method, Taylor Methods of Order \( n \) and Runge-Kutta Methods of Order \( n \) are studied. These methods compute the current step \( y_i \) based on the information given by the previous step \( y_{i-1} \). Can earlier steps \( y_0, y_1, \ldots, y_{i-2} \) also be used to generate \( y_i \)? The methods studied in this section compute \( y_i \) using the information on \( m \) previous steps \( y_i, y_{i-1}, \ldots, y_{i-(m-1)} \). Let \( h = \frac{b-a}{N} \) for a given positive integer \( N \) and define \( t_i = t_{i-1} + h = a + ih \).

1. Multistep Methods:

   **Derivation:** Recall by the Fundamental Theorem of Calculus
   \[
y(t_{i+1}) - y(t_i) = \int_{t_i}^{t_{i+1}} f(t, y(t))\,dt.
   \]
   A difference equation can be designed based on the following equation:
   \[
y_{i+1} = y_i + \int_{t_i}^{t_{i+1}} f(t, y(t))\,dt \quad \text{where } y_i \approx y(t_i).
   \]
   Now consider approximating the function \( f(t, y(t)) \) by an \( i \)th degree interpolation polynomial \( P_i(t) \) using \( i+1 \) points
   \[
   (t_0, f(t_0, y_0)), (t_1, f(t_1, y_1)), \ldots, (t_i, f(t_i, y_i))
   \]
   or an \( (i+1) \)th degree interpolating polynomial \( P_{i+1}(t) \) using \( i+2 \) points
   \[
   (t_0, f(t_0, y_0)), (t_1, f(t_1, y_1)), \ldots, (t_i, f(t_i, y_i)), (t_{i+1}, f(t_{i+1}, y_{i+1})).
   \]

   Let \( m > 1 \). An \( m \)–step method for solving an initial-value problem uses the following difference equation to compute the approximation \( y_{i+1} \) at the points \( t_{i+1} \)
   \[
y_{i+1} = a_{m-1}y_i + a_{m-2}y_{i-1} + \ldots + a_0y_{i-(m-1)} + h \left[ b_m f(t_{i+1}, y_{i+1}) + b_{m-1} f(t_{i}, y_{i}) + \ldots + b_0 f(t_{i-(m-1)}, y_{i-(m-1)}) \right]
   \]
   linear combination of \( y_{i+1}, y_{i} \) using \( m+1 \) slopes
   \[
   \text{for } i = m-1, m, \ldots, N-1, \text{ where } a_0, a_1, \ldots, a_{m-1} \text{ and } b_0, b_1, \ldots, b_m \text{ are constants and the starting values }
   \]
   \[
y_0 = a, \quad y_1 = a_1, \ldots, \quad y_{m-1} = a_{m-1}
   \]
   are specified (these values can be computed using an one-step method).

**Explicit and Implicit Method:**
A multistep method is **explicit** if \( b_m = 0 \) and is **implicit** if \( b_m \neq 0 \).

**The local truncation error for a multistep method:**
\[
\tau_{i+1}(h) = \frac{1}{h} \left( y(t_{i+1}) - a_{m-1} y(t_i) - \ldots - a_0 y(t_{i-(m-1)}) \right) - \left[ b_m f(t_{i+1}, y(t_{i+1})) + \ldots + b_0 f(t_{i-(m-1)}, y(t_{i-(m-1)})) \right]
\]

For example, an explicit 2-step method:
\[
\begin{cases}
y_0 = \alpha \\
y_1 \text{ is computed by an one-step method}
\end{cases}
\]
\[
y_i = a_1y_1 + a_0y_0 + h[b_1f(t_{i-1}, y_{i-1}) + b_0f(t_{i-2}, y_{i-2})] \text{ for } i \geq 2
\]

and an implicit 2-step method:
\[
\begin{cases}
y_0 = \alpha \\
y_1 \text{ is computed by an one-step method}
\end{cases}
\]
\[
y_i = a_1y_1 + a_0y_0 + h[b_2f(t_i, y_i) + b_1f(t_{i-1}, y_{i-1}) + b_0f(t_{i-2}, y_{i-2})] \text{ for } i \geq 2
\]

2. Adams–Bashforth Multistep Explicit Methods:

a. 2-step Explicit Method:
At the \( i \)th iteration, points \( (t_{i-1}, f(t_{i-1}, y_{i-1})) \) and \( (t_i, f(t_i, y_i)) \) are used to form \( P_1(t) \):
\[
P_1(t) = f(t_i, y_i) \frac{t - t_{i-1}}{t_i - t_{i-1}} + f(t_{i-1}, y_{i-1}) \frac{t - t_i}{t_{i-1} - t_i}
\]
\[
= \frac{1}{h} \left( f(t_i, y_i)(t - t_{i-1}) - f(t_{i-1}, y_{i-1})(t - t_i) \right)
\]
\[
\int_{t_i}^{t_{i+1}} P_1(t) dt = \int_{t_i}^{t_{i+1}} \frac{1}{h} \left( f(t_i, y_i)(t - t_{i-1}) - f(t_{i-1}, y_{i-1})(t - t_i) \right) dt
\]
\[
= \frac{1}{h} \left( \frac{f(t_i, y_i)}{2} (t - t_{i-1})^2 - \frac{f(t_{i-1}, y_{i-1})}{2} (t - t_i)^2 \right)_{t_i}^{t_{i+1}}
\]
\[
= \frac{1}{2h} \left( f(t_i, y_i)(4h^2 - h^2) - f(t_{i-1}, y_{i-1})(h^2 - 0) \right)
\]
\[
= h\left( \frac{3}{2} f(t_i, y_i) - \frac{1}{2} f(t_{i-1}, y_{i-1}) \right)
\]
\[
\begin{cases}
y_0 = \alpha, \ y_1 = \alpha_1 \\
y_{i+1} = y_i + h\left( \frac{3}{2} f(t_i, y_i) - \frac{1}{2} f(t_{i-1}, y_{i-1}) \right), \text{ for } i = 1, 2, \ldots, N - 1.
\end{cases}
\]

Local truncation error:
\[
\tau_{i+1}(h) = \frac{1}{h} \left( y'(t_{i+1}) - y(t_i) \right) - \left( \frac{3}{2} f(t_i, y(t_i)) - \frac{1}{2} f(t_{i-1}, y(t_{i-1})) \right)
\]
\[
= \frac{1}{h} \left( y(t_{i+1}) - y(t_i) \right) - \left( \frac{3}{2} y'(t_i) - \frac{1}{2} y'(t_{i-1}) \right)
\]
\[
= \frac{1}{h} \left( y(t_{i+1}) - y(t_i) \right) - \frac{1}{2} \left( y'(t_i) + \frac{y''(t_i)}{2} h + \frac{y'''(t_i^*)}{6} h^2, t_i^* \text{ in } (t_i, t_{i+1}) \right)
\]
\[
\frac{3}{2} y'(t_i) - \frac{1}{2} y'(t_{i-1}) = \frac{3}{2} y'(t_i) - \frac{1}{2} \left( y'(t_i) + \frac{y''(t_i)}{1!} (-h) + \frac{y'''(t_i^*)}{2!} h^2 \right), \text{ where } t_i^* \text{ in } (t_{i-1}, t_i)
\]
\[
= y'(t_i) + \frac{h}{2} y''(t_i) - \frac{h^2}{4} y'''(t_i^*)
\]
\[
\tau_{i+1}(h) = y'(t_i) + \frac{y''(t_i)}{2} h + \frac{y'''(t_i^*)}{6} h^2 - \left( y'(t_i) + \frac{h}{2} y''(t_i) - \frac{h^2}{4} y'''(t_i^*) \right)
\]
\[
= h^2 \left( \frac{y''(t_i^*)}{6} + \frac{y'''(t_i^*)}{4} \right) = \frac{5}{12} y'''(t_i^*) h^2 \text{ where } t_i^* \text{ in } (t_{i-1}, t_{i+1})
\]
\[ \tau_{i+1}(h) = \frac{5}{12} y'''(t^*_i) h^2, \text{ where } t^*_i \text{ is in } (t_{i-1}, t_{i+1}). \]

b. **3-step Explicit Method:**

In a similar way, when we use the interpolating polynomial \( P_2(t) \) at \((t_{i-2}, f(t_{i-2}, y_{i-2}))\), \((t_{i-1}, f(t_{i-1}, y_{i-1}))\) and \((t_i, f(t_i, y_i))\) we can derive the following 3-step explicit method:

\[
y_0 = \alpha, \ y_1 = \alpha_1, \ y_2 = \alpha_2 \\
y_{i+1} = y_i + h \left( \frac{23}{12} f(t_i, y_i) - \frac{16}{12} f(t_{i-1}, y_{i-1}) + \frac{5}{12} f(t_{i-2}, y_{i-2}) \right),
\]

for \( i = 2, 3, \ldots, N - 1 \).

The corresponding local truncation error is:

\[ \tau_{i+1}(h) = \frac{3}{8} y^{(4)}(t^*_i) h^3, \text{ where } t^*_i \text{ is in } (t_{i-2}, t_i). \]

c. **4-step Explicit Method:**

Also in a similar way, when we use the interpolating polynomial \( P_3(t) \) at \((t_{i-3}, f(t_{i-2}, y_{i-3}))\), \((t_{i-2}, f(t_{i-2}, y_{i-2}))\), \((t_{i-1}, f(t_{i-1}, y_{i-1}))\) and \((t_i, f(t_i, y_i))\) we can derive the following 4-step explicit method:

\[
y_0 = \alpha, \ y_1 = \alpha_1, \ y_2 = \alpha_2, \ y_3 = \alpha_3 \\
y_{i+1} = y_i + h \left( \frac{55}{24} f(t_i, y_i) - \frac{59}{24} f(t_{i-1}, y_{i-1}) + \frac{37}{24} f(t_{i-2}, y_{i-2}) - \frac{9}{24} f(t_{i-3}, y_{i-3}) \right),
\]

for \( i = 3, 4, \ldots, N - 1 \).

The corresponding local truncation error is:

\[ \tau_{i+1}(h) = \frac{251}{720} y^{(5)}(t^*_i) h^4, \text{ where } t^*_i \text{ is between } (t_{i-3}, t_i). \]

3. **Adams-Moulton Multistep Implicit Methods:**

**Derivation:** Implicit methods are derived by using \((t_{i+1}, f(t_{i+1}, y(t_{i+1})))\) as additional interpolation point in the approximation of the integral \( \int_{t_i}^{t_{i+1}} f(t, y(t)) dt \)

a. **1-step Implicit Method:**

\[
y_0 = \alpha \\\ny_{i+1} = y_i + h \left( \frac{1}{2} f(t_{i+1}, y_{i+1}) + \frac{1}{2} f(t_i, y_i) \right)
\]

Local truncation error:

\[ \tau_{i+1}(h) = -\frac{1}{12} y'''(t^*_i) h^2, \text{ where } t^*_i \text{ is in } (t_i, t_{i+1}). \]

\[ P_1(t) = f(t_{i+1}, y_{i+1}) \frac{t - t_i}{t_{i+1} - t_i} + f(t_i, y_i) \frac{t - t_{i+1}}{t - t_{i+1}} = \frac{1}{h} \left( f(t_{i+1}, y_{i+1})(t - t_i) - f(t_i, y_i)(t - t_{i+1}) \right) \]
\[
\int_{t_i}^{t_{i+1}} P_1(t) dt = \frac{1}{h} \left( f(t_{i+1}, y_{i+1}) \left( \frac{t-t_i}{2} \right)^2 - f(t_i, y_i) \left( \frac{t-t_{i+1}}{2} \right)^2 \right)_{t_i}
= \frac{1}{2h} \left( f(t_{i+1}, y_{i+1}) (h^2 - 0) - f(t_i, y_i) (0 - h^2) \right)
= h \left( \frac{1}{2} f(t_{i+1}, y_{i+1}) + \frac{1}{2} f(t_i, y_i) \right)
\]

b. 2-step Implicit Method:
\[
y_0 = a, \ y_1 = \alpha_1 \\
y_{i+1} = y_i + h \left( \frac{5}{12} f(t_{i+1}, y_{i+1}) + \frac{8}{12} f(t_i, y_i) - \frac{1}{12} f(t_{i-1}, y_{i-1}) \right),
\text{ for } i = 1, 2, \ldots, N - 1.
\]

Local truncation error:
\[
\tau_{i+1}(h) = -\frac{1}{24} y^{(4)}(t_i^*) h^3, \text{ where } t_i^* \text{ is in } (t_{i-1}, t_{i+1}).
\]

How to derive the local truncation error?
\[
\tau_{i+1}(h) = \frac{1}{h} (y(t_{i+1}) - y(t_i)) - \left( \frac{5}{12} f(t_{i+1}, y(t_{i+1})) + \frac{8}{12} f(t_i, y(t_i)) - \frac{1}{12} f(t_{i-1}, y(t_{i-1})) \right)
= \frac{1}{h} (y(t_{i+1}) - y(t_i)) = y'(t_i) + \frac{y''(t_i)}{2} h + \frac{y'''(t_i)}{6} h^2 + \frac{y^{(4)}(t_i^*)}{24} h^3
\]
\[
= \frac{5}{12} y'(t_{i+1}) + \frac{8}{12} y'(t_i) - \frac{1}{12} y'(t_{i-1})
= \frac{5}{12} \left( y'(t_i) + \frac{y''(t_i)}{1!} h + \frac{y'''(t_i)}{2!} h^2 + \frac{y^{(4)}(t_i^*)}{3!} h^3 \right) + \frac{8}{12} y'(t_i)
= y'(t_i) + h \frac{y''(t_i)}{2} h + \frac{h^2}{6} y'''(t_i) + \frac{h^3}{2(3!)} y^{(4)}(t_i^*), \text{ where } t_i^* \text{ in } (t_{i-1}, t_{i+1})
\]
\[
= y'(t_i) + \frac{h^2}{6} y'''(t_i) + \frac{h^3}{2(3!)} y^{(4)}(t_i^*), \text{ where } t_i^* \text{ in } (t_{i-1}, t_{i+1})
\]
\[
\tau_{i+1}(h) = y'(t_i) + \frac{y''(t_i)}{2} h + \frac{y'''(t_i)}{6} h^2 + \frac{y^{(4)}(t_i^*)}{24} h^3
- \left( y'(t_i) + \frac{h^2}{6} y'''(t_i) + \frac{h^3}{2(3!)} y^{(4)}(t_i^*) \right)
= h^3 \left( \frac{1}{24} y^{(4)}(t_i^*) - \frac{1}{12} y^{(4)}(t_i^*) \right) = -\frac{h^3}{24} y^{(4)}(\hat{t}_i) \text{ where } \hat{t}_i \text{ in } (t_{i-1}, t_{i+1})
\]

c. 3-step Implicit Method:
\[
y_0 = a, \ y_1 = \alpha_1, \ y_2 = \alpha_2 \\
y_{i+1} = y_i + h \left( \frac{9}{24} f(t_{i+1}, y_{i+1}) + \frac{10}{24} f(t_i, y_i) - \frac{5}{24} f(t_{i-1}, y_{i-1}) + \frac{1}{24} f(t_{i-2}, y_{i-2}) \right),
\text{ for } i = 2, 3, \ldots, N - 1.
\]

Local truncation error:
Consider the initial-value problem: \( y'(t) = t^2 + 1, \quad 0 \leq t \leq 2, \quad y(0) = 0.5. \) Let \( h = 0.1. \)

a. Use the Midpoint Method to generate estimate: \( y_1; \)
b. generate $y_2^*$ using Adams-Bashforth 2-step method; and
c. generate $y_2$ using Adams-Moulton 1-step method.

What is the order of the local truncation error for $y_2$?

$$t_0 = 0, \ h = 0.1, \ y_0 = 0.5$$

$$y_1^* = y_0 + \frac{h}{2} f(t_0, y_0) = 0.5 + \frac{(0.1)}{2} (0.5 - 0^2 + 1) = 0.575$$

$$y_1 = y_0 + h f\left(t_0 + \frac{h}{2}, \ y_1^*\right) = 0.5 + (0.1) \left(0.575 - \left(\frac{0.1}{2}\right)^2 + 1\right) = 0.65725$$

$$y_2^* = y_1 + h \left(\frac{3}{2} f(t_1, y_1) - \frac{1}{2} f(t_0, y_0)\right)$$

$$= 0.65725 + (0.1) \left(\frac{3}{2} (0.65725 - (0.1)^2 + 1) - \frac{1}{2} (0.5 - 0^2 + 1)\right) = 0.829375$$

$$y_2 = y_1 + h \left(\frac{1}{2} f(t_2, y_2^*) + \frac{1}{2} f(t_1, y_1)\right)$$

$$= 0.65725 + (0.1) \left(\frac{1}{2} (0.829375 - (0.2)^2 + 1) + \frac{1}{2} (0.65725 - (0.1)^2 + 1)\right)$$

$$= 0.8290794$$

$$\text{true error} = \left| (1 + 0.2)^2 - 0.5e^{0.2} - 0.8290794 \right| = 2.192209 \times 10^{-4}$$

```matlab
>> clear; clf
>> fun=@(t,y) y-t.^2+1;
>> fun1=@(t,y) (y-t.^2+1)-2*t;
>> [tv,yvtaylor,n]=tayfun2(fun,fun1,0,2,1/2,0.1);
>> [tv,yvmidpt,n]=rk2mid(fun,0,2,1/2,0.1);
>> [tv,yvab,n]=adambash(fun,0,2,1/2,0.1,2);
>> [tv,yvam,n]=adammoul(fun,0,2,1/2,0.1,1);
>> ysol=(tv+1).^2-0.5*exp(tv);
>> plot(tv,yvtaylor,'r-o',tv,yvmidpt,'b-*',tv,yvab,'m-+',tv,yvam,'g-x',tv,ysol,'k-')
>> title('dy/dt=ty^2+1, y(0)=1/2, t in [0,2], h=0.1, y(t)=(t+1)^2-0.5e^t')
>> axis([0 2 0 5.5])
(Or MatLab program lect5_5_ex1.m)
4. Other Multistep Methods:
   a. Milne’s Explicit Method:
   
\[
y_{i+1} = y_{i-3} + h \left( \frac{8}{3} f(t_i, y_i) - \frac{4}{3} f(t_{i-1}, y_{i-1}) + \frac{8}{3} f(t_{i-2}, y_{i-2}) \right)
\]
which has local truncation error $\tau_{i+1}(h) = \frac{14}{45} h^4 y^{(5)}(t^*_i)$ where $t^*_i$ is in $(t_{i-3}, t_{i+1})$

b. Simpson’s Implicit Method:

$$ y_{i+1} = y_{i-1} + h \left( \frac{1}{3} f(t_{i-1}, y_{i-1}) + \frac{4}{3} f(t_i, y_i) + \frac{1}{3} f(t_{i+1}, y_{i+1}) \right) $$

which has local truncation error $\tau_{i+1}(h) = -\frac{1}{90} h^4 y^{(5)}(t^*_i)$ where $t^*_i$ is in $(t_{i-1}, t_{i+1})$

The derivation for Milne’s Method:

$$ y(t_{i+1}) - y(t_{i-3}) = \int_{t_{i-3}}^{t_{i+1}} f(t, y(t)) dt $$

$$ y_{i+1} = y_{i-3} + h \left( \frac{8}{3} f(t_{i-2}, y_{i-2}) - \frac{4}{3} f(t_{i-1}, y_{i-1}) + \frac{8}{3} f(t_{i}, y_{i}) \right) $$

The local truncation error:

$$ \tau_{i+1}(h) = \frac{1}{h} (y(t_{i+1}) - y(t_{i-3})) - \left( \frac{8}{3} f(t_{i-2}, y_{i-2}) - \frac{4}{3} f(t_{i-1}, y_{i-1}) + \frac{8}{3} f(t_{i}, y_{i}) \right) $$

$$ = \frac{1}{h} \left( y(t_i) + \frac{y'(t_i)}{1!} h + \frac{y''(t_i)}{2!} h^2 + \frac{y'''(t_i)}{3!} h^3 + \frac{y^{(4)}(t_i)}{4!} h^4 + \frac{y^{(5)}(t^*_i)}{5!} h^5 \right) $$

$$ - \frac{1}{h} \left( y(t_i) + \frac{y'(t_i)}{1!} (-3h) + \frac{y''(t_i)}{2!} (-3h)^2 + \frac{y'''(t_i)}{3!} (-3h)^3 + \frac{y^{(4)}(t_i)}{4!} (-3h)^4 + \frac{y^{(5)}(t^*_i)}{5!} (-3h)^5 \right) $$

$$ - \frac{8}{3} \left( y'(t_i) + \frac{y''(t_i)}{2!} (-2h) + \frac{y'''(t_i)}{3!} (-2h)^2 + \frac{y^{(4)}(t_i)}{4!} (-2h)^3 + \frac{y^{(5)}(t^*_i)}{5!} (-2h)^4 \right) $$

$$ + \frac{4}{3} \left( y'(t_i) + \frac{y''(t_i)}{2!} (-h) + \frac{y'''(t_i)}{3!} (-h)^2 + \frac{y^{(4)}(t_i)}{4!} (-h)^3 + \frac{y^{(5)}(t^*_i)}{5!} (-h)^4 \right) $$

$$ - \frac{8}{3} y'(t_i) $$

$$ = y'(t_i) \left( 1 - \frac{8}{3} + \frac{4}{3} - \frac{8}{3} \right) + y''(t_i) \left( \frac{1}{2} - \frac{9}{2} + \frac{16}{3} - \frac{4}{3} \right) h + y'''(t_i) \left( \frac{1}{6} + \frac{9}{2} - \frac{16}{3} + \frac{2}{3} \right) h^2 $$

$$ + y^{(4)}(t_i) \left( \frac{1}{24} - \frac{27}{8} + \frac{32}{9} - \frac{2}{9} \right) h^3 + y^{(5)}(t^*_i) h^4 $$

$$ = \frac{14}{45} y^{(5)}(t^*_i) h^4, \ t^*_i \text{ is in } (t_{i-3}, t_{i+1}). $$

5. Predictor-Corrector Schemes:

In an implicit method, the current step $y'_{i+1}$ is on the both sides of the formula for computing $y_{i+1}$. A predictor-Corrector Scheme computes $y'_{i+1}$ on the right side by an explicit method, called it $\bar{y}'_{i+1}$ and then computes $y_{i+1}$ on the left side using $\bar{y}_{i+1}$ in the formula. The obtained $y'_{i+1}$ is called a predictor and $y_{i+1}$ is called a corrector. Note that the approximation error of the chosen explicit method should be in the same order as it is for the implicit method. For example, in the 1-step Adams-Moulton method, $y'_{i+1}$ should be computed by the 2-step Adams-Bashforth method or a second-order Runge-Kutta method, or a second-order Taylor method.

Exercises:

1. Apply the 2-step Adams-Bashforth method to approximate the solutions of the following initial-value problems with $h = 0.2$. Compute $y_2$ without using the MatLab program. Use the Midpoint Method to find $y_1$.

\begin{enumerate}
\item $y' = ty^3 - y, \ 0 \leq t \leq 1, \ y(0) = 1$
\item $y' + \frac{4}{t} y = t^4, \ 1 \leq t \leq 3, \ y(1) = 1$
\item $y' + 2y^2 = t^2 - 1, \ 0 \leq t \leq 1, \ y(0) = 0$
\end{enumerate}
(4) \[ y' = \frac{(1 + y^2)}{t}, \ 1 \leq t \leq 4, \ y(1) = 0 \]

2. Derive the 3-step Adams-Bashforth method.

3. Derive the truncation error for the 1-step Adams-Moulton (trapezoidal) method.

4. Consider the population model

\[ y' = ry \left(1 - \frac{y}{k}\right) - \frac{y^2}{1 + y^2}. \]

The first term on the right side is known as the logistic growth term and the second one represents harvesting/predation of the species by some other species. The parameters \( r \) and \( k \) are called the natural growth rate of the population and the environmental carrying capacity, respectively. Let \( r = 0.4 \) and \( k = 20 \) and the initial population be 2.44.

1. Use one of the 2nd order Runge-Kutta Method to approximate \( y(t) \) with \( h = 0.25 \).
2. Use Adams-Bashforth 2-step method to approximation \( y(t) \) with \( h = 0.25 \).
3. Use Adams-Moulton 1-step method to approximation \( y(t) \) with \( h = 0.25 \).
4. Plot sets \( \{y_i\} \) obtained in (1), (2) and (3).
5. Determine the eventual population level (as \( t \to \infty \)) reached from the initial population.

5. A genetic switch is a biochemical mechanism that governs whether a particular protein product of a cell is synthesized or not. The following initial-value problem has been proposed as a model for a genetic switch:

\[ g'(t) = 0.4 - 1.41g + 3.03 \frac{g^2}{1 + g^2}, \ 0 \leq t \leq 12, \ g(0) = 0. \]

1. Use one of the 2nd order Runge-Kutta Method to approximate \( y(t) \) with \( h = 0.25 \).
2. Use Adams-Bashforth 2-step method to approximation \( y(t) \) with \( h = 0.25 \).
3. Use Adams-Moulton 1-step method to approximation \( y(t) \) with \( h = 0.25 \).
4. Plot sets \( \{y_i\} \) obtained in (1), (2) and (3).
5. Predict the limit of \( g(t) \) as \( t \to \infty \) (you may enlarge the interval of \( t \) to obtain more information).

6. Apply the 2-step Adams-Bashforth method with the second-order Taylor method (for computing \( y_1 \)) to approximate the solution of the following initial-value problems with given \( h \). Compute \( y_2 \) without using the MatLab program.

a. \[ y' = 1 - y + e^{2ty^2}, \ 0 \leq t \leq 0.9, \ y(0) = 0, \ h = 0.1 \]

b. \[ y' = y^2 - \frac{y}{t}, \ 1 \leq t \leq 5, \ y(1) = -\frac{1}{\ln(2)}, \ h = 0.2 \]

c. \[ y' = \frac{4y}{t} + t^4e^t, \ 1 \leq t \leq 2, \ y(1) = 0, \ h = 0.2. \]

7. Apply the 1-step Adams-Moulton method with the Modified Euler method (for computing \( y_1 \)) and with the 2-step Adams-Bashforth method (for computing \( y_2 \)) to approximate the solutions of the initial-value problems given in 1. with \( h = 0.2 \). Compute \( y_2 \) without using the MatLab program.