5.6 Multistep Methods

Motivation: Solve the IVP: $y' = f(t,y), \quad a \le t \le b, \ y(a) = \alpha$. To compute solution at t_{i+1}, approximate solutions at mesh points $t_0, t_1, t_2, \ldots\ t_i$ are already obtained. Since in general error $|y(t_{i+1}) - w_{i+1}|$ grows with respect to time t, it then makes sense to use more previously computed approximate solution $w_i,\ w_{i-1}, w_{i-2}, \ldots$ when computing w_{i+1}.

Adams-Bashforth two-step *explicit* method.

$$w_0 = \alpha, \quad \boxed{w_1 = \alpha_1}$$

$$w_{i+1} = w_i + \frac{h}{2}[3f(t_i, w_i) - f(t_{i-1}, w_{i-1})] \quad \text{where } i = 1, 2, \ldots N-1.$$

Adams-Moulton two-step *implicit* method.

$$w_0 = \alpha, \quad \boxed{w_1 = \alpha_1}$$

$$w_{i+1} = w_i + \frac{h}{12}[5f(t_{i+1}, w_{i+1}) + 8f(t_i, w_i) - f(t_{i-1}, w_{i-1})] \quad \text{where } i = 1, 2, \ldots N-1.$$

Example. Solve the IVP $y' = y - t^2 + 1, \quad 0 \le t \le 2, \ y(0) = 0.5$ by Adams-Bashforth two-step explicit method and Adams-Moulton two-step implicit method respectively. Use the exact values given by $y(t) = (t+1)^2 - 0.5e^t$ to get needed starting values for approximation and $h = 0.2$.

Solution:

$$w_0 = 0.5$$

$$w_1 = y(0.2) = (0.2+1)^2 - 0.5e^{0.2} = 0.8292986 \qquad \text{(by using } y(t)\text{)}$$

1) Adams-Bashforth two-step explicit method

$$w_{i+1} = w_i + \frac{h}{2}[3(w_i - {t_i}^2 + 1) - (w_{i-1} - {t_{i-1}}^2 + 1)]$$

$$w_2 = 0.8292986 + 0.1[3(0.8292986 - 0.2^2 + 1) - (0.5 + 1)] = 1.2160882$$

$$w_3 = 1.2160882 + 0.1[3(1.2160882 - 0.4^2 + 1) - (0.8292986 - 0.2^2 + 1)] = 1.6539848$$

…. and so on till to compute w_{10}.

2) Adams-Moulton two-step implicit method

$$w_{i+1} = w_i + \frac{h}{12}[5(w_{i+1} - {t_{i+1}}^2 + 1) + 8(w_i - {t_i}^2 + 1) - (w_{i-1} - {t_{i-1}}^2 + 1)]$$

$$w_2 = 0.8292986 + \frac{0.2}{12}[5(w_2 - 0.4^2 + 1) + 8(0.8292986 - 0.2^2 + 1) - (0.5 + 1)]$$

Solve for w_2:

$$w_2 = 1.21404191$$

$$w_3 = 1.21404191 + \frac{0.2}{12}[5(w_3 - 0.6^2 + 1) + 8(1.21404191 - 0.4^2 + 1) - (0.8292986 - 0.2^2 + 1)]$$

Solve for w_3: ...

… and so on till to compute w_{10}.

Definition

An m-step multistep method for solving the IVP

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

has a difference equation for approximate w_{i+1} at t_{i+1}:

$$\begin{aligned} w_{i+1} &= a_{m-1}w_i + a_{m-2}w_{i-1} + \cdots + a_0 w_{i+1-m} \\ &+ h[b_m f(t_{i+1}, w_{i+1}) + b_{m-1} f(t_i, w_i) + \cdots \\ &+ b_0 f(t_{i+1-m}, w_{i+1-m})], \end{aligned}$$

where $h = \frac{b-a}{N}$, and starting values are specified:

$$w_0 = \alpha, \ w_1 = \alpha_1, \dots, w_{m-1} = \alpha_{m-1}.$$

Explicit method if $b_m = 0$, ***implicit*** method if $b_m \neq 0$.

Example. Derive Adams-Bashforth two-step *explicit* method: Solve the IVP: $y' = f(t,y), \quad a \le t \le b, \; y(a) = \alpha$.
Integrate $y' = f(t,y)$ over $[y_i, y_{i+1}]$

$$y_{i+1} - y_i = \int_{t_i}^{t_{i+1}} y'(t)dt = \int_{t_i}^{t_{i+1}} f(t, y(t))dt$$

Use (t_i, y_i) and (t_{i-1}, y_{i-1}) to form interpolating polynomial $P_1(t)$ (by Newton backward difference (Page 129)) to approximate $f(t,y)$.

$$\int_{t_i}^{t_{i+1}} f(t,y)dt = \int_{t_i}^{t_{i+1}} \left(f(t_i,y_i) + \nabla f(t_i,y_i)\frac{(t-t_i)}{h} + \text{error}\right)dt$$

$$y_{i+1} - y_i = h\left[f(t_i,y_i) + \frac{1}{2}\big(f(t_i,y_i) - f(t_{i-1},y_{i-1})\big)\right] + \text{Error}$$

where $h = t_{i+1} - t_i$, and the backward difference $\nabla f(t_i,y_i) = hf[t_i, t_{i-1}] = (f(t_i,y_i) - f(t_{i-1},y_{i-1}))$.
Consequently, Adams-Bashforth two-step *explicit* method is:

$$w_0 = \alpha, \quad w_1 = \alpha_1$$

$$w_{i+1} = w_i + \frac{h}{2}[3f(t_i,w_i) - f(t_{i-1},w_{i-1})] \quad \text{where } i = 1, 2, \ldots N-1.$$

Local Truncation Error. If $y(t)$ solves the IVP $y' = f(t,y), \quad a \le t \le b, \; y(a) = \alpha$ and

$$\begin{gathered} w_{i+1} = a_{m-1}w_i + a_{m-2}w_{i-1} + \cdots + a_0 w_{i+1-m} \\ h[b_m f(t_{i+1}, w_{i+1}) + b_{m-1} f(t_i, w_i) + \cdots \\ + b_0 f(t_{i+1-m}, w_{i+1-m})], \end{gathered}$$

the local truncation error is

$$\tau_{i+1}(h) = \frac{y(t_{i+1}) - a_{m-1}y(t_i) - a_{m-2}y(t_{i-1}) - \cdots - a_0 y(t_{i+1-m})}{h} - [b_m f\big(t_{i+1}, y(t_{i+1})\big) + \cdots + b_0 f(t_{i+1-m}, y(t_i))]$$

NOTE: the local truncation error of a *m*-step *explicit* step is $O(h^m)$.

the local truncation error of a *m*-step *implicit* step is $O(h^{m+1})$.

m-step *explicit* step method *vs.* (m-1)-step *implicit* step method

a) both have the same order of local truncation error, $O(h^m)$.

b) Implicit method usually has greater stability and smaller round-off errors.
For example, local truncation error of Adams-Bashforth 3-step explicit method, $\tau_{i+1}(h) = \frac{3}{8} y^{(4)}(\mu_i)h^3$.
Local truncation error of Adams-Moulton 2-step implicit method, $\tau_{i+1}(h) = -\frac{1}{24} y^{(4)}(\xi_i)h^3$.

Predictor-Corrector Method

Motivation: **(1)** Solve the IVP $y' = e^y, \quad 0 \le t \le 0.25, \ y(0) = 1$ by the three-step Adams-Moulton method.
Solution: The three-step Adams-Moulton method is

$$w_{i+1} = w_i + \frac{h}{24}[9e^{w_{i+1}} + 19e^{w_i} - 5e^{w_{i-1}} + e^{w_{i-2}}] \qquad Eq.(1)$$

$Eq.(1)$ can be solved by Newton's method. However, this can be quite computationally expensive.

(2) combine explicit and implicit methods.

4th order Predictor-Corrector Method

(we will combine 4th order Runge-Kutta method + 4th order 4-step explicit Adams-Bashforth method + 4th order three-step Adams-Moulton implicit method)

Step 1: Use 4th order Runge-Kutta method to compute w_0, w_1, w_2 and w_3.

Step 2: For $i = 3, 5, \ldots N$

(a) Predictor sub-step. Use 4th order 4-step explicit Adams-Bashforth method to compute a predicated value

$$w_{i+1,p} = w_i + \frac{h}{24}[55f(t_i, w_i) - 59f(t_{i-1}, w_{i-1}) + 37f(t_{i-2}, w_{i-2}) - 9f(t_{i-3}, w_{i-3})]$$

(b) Correction sub-step. Use 4th order three-step Adams-Moulton implicit method to compute a correction (the approximation at $i + 1$ time step)

$$w_{i+1} = w_i + \frac{h}{24}[9f\left(t_{i+1}, w_{i+1,p}\right) + 19f(t_i, w_i) - 5f(t_{i-1}, w_{i-1}) + f(t_{i-2}, w_{i-2})]$$