Numerical Solution of ODEs

Exercise Class

7th November 2025

Multistep Methods

General m-step method,

$$\begin{aligned} &a_m u_{j+m} + a_{m-1} u_{j+m-1} + \cdots + a_0 u_j \\ &\qquad = \tau(b_m f(t_{j+m}, u_{j+m}) + b_{m-1} f(t_{j+m-1}, u_{j+m-1}) + \cdots + b_0 f(t_j, u_j)), \end{aligned}$$

for $j = 0, \ldots, N-m$, where $a_i, b_i \in \mathbb{R}$, $i \leq m$, $a_m = 1$. Note that,

$$\begin{aligned} b_m &= 0 &&\text{— \textit{explicit} method,} \\ b_m &\neq 0 &&\text{— \textit{implicit} method.} \end{aligned}$$

Adams Method

Set

$$a_m = 1, \qquad a_{m-1} = -1, \qquad a_{m-2} = \cdots = a_0 = 0.$$

We can define the recursive formulation for *Adams method* as

$$u_{j+m} - u_{j+m-1} = \tau(b_m f(t_{j+m}, u_{j+m}) + b_{m-1} f(t_{j+m-1}, u_{j+m-1}) + \cdots + b_0 f(t_j, u_j)).$$

By shifting the indices we can re-write this as

$$u_{j+1} - u_j = \tau(b_m f(t_{j+1}, u_{j+1}) + b_{m-1} f(t_j, u_j) + \cdots + b_0 f(t_{j-m+1}, u_{j-m+1})).$$

We deduce $b_m, \ldots, b_0$ such that the method is the *highest order* possible. Using Theorem 3.2 we obtain the criteria for the highest order method.

From these we can obtain the following requirements for the explicit two-step method ($m = 2$):

$$\sum_{i=0}^{m} a_i = 0, \qquad \sum_{i=1}^{m} (i^\ell a_i - \ell i^{\ell-1} b_i) = 0, \quad \ell = 1, \ldots, p,$$

where $p = 2$ is the order of the method. Using these conditions we get the linear system

$$\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \end{bmatrix} = \begin{bmatrix} a_1 + 2a_2 \\ {}^{a_1}/_2 + 2a_2 \end{bmatrix};$$

therefore, as the method is explicit $b_0 = -{}^1/_2$, $b_1 = {}^3/_2$, and $b_2 = 0$.

Adams-Bashfort

These are the explicit Adams methods for $m = 1, 2, 3, 4$, respectively:

$$u_{j+1} = u_j + \tau f(t_j, u_j), \tag{ab1}$$

$$u_{j+2} = u_{j+1} + \tau \left(\frac{3}{2} f(t_{j+1}, u_{j+1}) - \frac{1}{2} f(t_j, u_j) \right), \tag{ab2}$$

$$u_{j+3} = u_{j+2} + \tau \left(\frac{23}{12} f(t_{j+2}, u_{j+2}) - \frac{4}{3} f(t_{j+1}, u_{j+1}) + \frac{5}{12} f(t_j, u_j) \right), \tag{ab3}$$

$$u_{j+4} = u_{j+3} + \tau \left(\frac{55}{24} f(t_{j+3}, u_{j+3}) - \frac{59}{24} f(t_{j+2}, u_{j+2}) + \frac{37}{24} f(t_{j+1}, u_{j+1}) - \frac{3}{8} f(t_j, u_j) \right) \tag{ab4}$$

The m-step Adams-Bashfort is order $p = m$. Note that ab1 $\equiv$ Euler.

Adams-Moulton

These are the implicit Adams methods ($b_m \neq 0$) for $m = 1, 2, 3, 4$, respectively:

$$u_{j+1} = u_j + \frac{1}{2} \tau \left(f(t_{j+1}, u_{j+1}) + f(t_j, u_j) \right), \tag{am1}$$

$$u_{j+2} = u_{j+1} + \tau \left(\frac{5}{12} f(t_{j+2}, u_{j+2}) + \frac{2}{3} f(t_{j+1}, u_{j+1}) - \frac{1}{12} f(t_j, u_j) \right), \tag{am2}$$

$$\begin{aligned} u_{j+3} = u_{j+2} + \tau \bigg(&\frac{3}{8} f(t_{j+3}, u_{j+3}) + \frac{19}{24} f(t_{j+2}, u_{j+2}) \\ &- \frac{5}{24} f(t_{j+1}, u_{j+1}) + \frac{1}{24} f(t_j, u_j) \bigg), \end{aligned} \tag{am3}$$

$$\begin{aligned} u_{j+4} = u_{j+3} + \tau \bigg(&\frac{251}{720} f(t_{j+4}, u_{j+4}) + \frac{646}{720} f(t_{j+3}, u_{j+3}) - \frac{264}{720} f(t_{j+2}, u_{j+2}) \\ &+ \frac{106}{720} f(t_{j+1}, u_{j+1}) - \frac{19}{720} f(t_j, u_j) \bigg). \end{aligned} \tag{am4}$$

The m-step Adams-Moulton is order $p = m + 1$. Note that am1 $\equiv$ Crank-Nicholson.

Numerical Integration Definition

Consider the initial value problem

$$x' = f(t, x), \qquad x(t_0) = x_0.$$

We can define $u(t) = \psi(t, t_0, x_0)$, $t_0 \leq t \leq T$, using the integral formula

$$u(t) = u(t_0) + \int_{t_0}^{t} f(\tau, u(\tau)) \, \mathrm{d}\tau.$$

Using the equidistant partition $\{t_j\}_{j=0}^{N}$, $t_j = t_0 + \tau j$ we have that

$$u(t_{j+1}) = u(t_{j-k}) + \int_{t_{j-k}}^{t_{j+1}} f(s, u(s)) \, \mathrm{d}s, \qquad k = 0, 1, 2, \ldots.$$

Considering the Lagrange interpolation of $f(\cdot, u(\cdot))$ at the nodes t_i, $i = j - q, \ldots, j + \ell$, $q \in \mathbb{N}_0$, $\ell \in \{0, 1\}$, given by

$$f(s, u(s)) \approx \mathcal{L}_{j-q}(s) f_{j-q} + \cdots + \mathcal{L}_j(s) f_j + \cdots + \mathcal{L}_{j-\ell}(s) f_{j-\ell}$$

where

$$
\begin{aligned}
f_i &= f(t_i, u(t_i)) && i = j-q, \dots, j+\ell, \\
\mathcal{L}_{j-q+i}(s) &= \prod_{\substack{k=0 \\ k\neq i}}^{q+\ell} \frac{s - t_{j-q+k}}{t_{j-q+i} - t_{j-q+k}}, && t_{j-q} \leq s \leq t_{j+1}, i = 0, \dots, q+\ell.
\end{aligned}
$$

Then,

$$
u(t_{j+1}) - u(t_{j-k}) = \int_{t_{j-k}}^{t_{j+1}} f(s, u(s))\,\mathrm{d}s \approx \sum_{i=0}^{q+\ell} f_{j-q+i} \underbrace{\int_{t_{j-k}}^{t_{j+1}} \mathcal{L}_{j-q+i}(s)\,\mathrm{d}s}_{b}.
$$

We note that $\ell = 0$ defines an explicit method and $\ell = 1$ defines an implicit method. Let $q = 1$, $k = 0$, $\ell = 1$; then, we derive Adams-Moulton 2-step ($m = 2$).

Exercises

1. Derive the formula for Adams-Bashfort and Adams-Moulton for $m = 3$.
2. Modify *Adams-Bashfort 2-step* (`ab2.m`) and *Adams-Bashfort 2-step* (`ab3.m`) to use *Euler* rather than *Runge-Kutta* for the initialisation steps. Run `run_ab.m` with these modified *Adams-Bashfort* implementations. Are there any differences to when using *Runge-Kutta*? Can you find a reason for this behaviour?
3. Compare the two *Adams-Moulton 2-step* methods (`am2.m` and `am2_mod.m`) to the *Adams-Bashfort 2-step* and *3-step* methods for solving the following ODEs:
   (a) Logistic problem (`logistic.m`) on the time interval $t \in [0, 2]$, with $\tau = 0.1$:
   $$
   \begin{aligned}
   x' &= (1-x)x \\
   x(0) &= 2
   \end{aligned}
   $$
   (b) Linear oscillator (`oscillator.m`) on the time interval $t \in [0, 10]$, with $\tau = 0.1$:
   $$
   \begin{aligned}
   x_1' &= x_2 \\
   x_2' &= -9x + 10\cos(t), \\
   \boldsymbol{x}(0) &= \begin{pmatrix} 2 \\ 1 \end{pmatrix}
   \end{aligned}
   $$
   For comparisons plot t vs. x_1.
   (c) Stiff linear system (`linsystem.m`) on the time interval $t \in [0, 0.1]$ with $\tau = 0.001$:
   $$
   \begin{aligned}
   \boldsymbol{x}' &= \begin{pmatrix} 998 & 1998 \\ -999 & -1999 \end{pmatrix} \boldsymbol{x} \\
   \boldsymbol{x}(0) &= \begin{pmatrix} 2 \\ 1 \end{pmatrix}
   \end{aligned}
   $$
   Remark. *For comparisons plot t vs. x_1 you will need to restrict the y-axis (x_1) limits. The following MATLAB command will restrict the axis to a sensible limit, if executed after the plot is displayed:*
   ```
   ylim([-5 15]);
   ```

   Also run convergence analysis using `conv_analysis.m` to deduce the order of the *Adams-Moulton 2-step* methods. Additionally, have a look at the tolerance `tol` used in the fixed point iteration. Does modifying this tolerance have any effect on convergence?