Numerical analysis, week 9: ODEs, multi-step and implicit methods

Dr. Dmitry Batenkov

May 8, 2021

1. A single-step method is a scheme where y_{n+1} depends only on y_n . In contrast, a multi-step method expresses y_{n+1} as a function of $\{y_{n-p}, \ldots, y_n\}$ for $p \geq 1$.

2. Main idea: use $y'(x) = f(x, y(x))$ and apply an integration scheme for

$$y(x_{n+1}) - y(x_n) = \int_{x_n}^{x_{n+1}} y'(x)\, dx$$

$$\underbrace{y(x_{n+1})}_{\approx y_{n+1}} = \underbrace{y(x_n)}_{\approx y_n} + \int_{x_n}^{x_{n+1}} \underbrace{f(x, y(x))}_{g(x)}\, dx \tag{1}$$

3. **Adams-Bashforth** scheme:

   (a) use the interpolation nodes $x_n, x_{n-1} = x_n - h, x_{n-2} = x_n - 2h, x_{n-3} = x_n - 3h$:

$$y_{n+1} = y_n + A_0 g(x_n) + A_1 g(x_{n-1}) + A_2 g(x_{n-2}) + A_3 g(x_{n-3})$$

   (b) require exactness for polynomials of degree up to 3. Substitute $g(x) = 1, x - x_n, (x - x_n)^2, (x - x_n)^3$:

$$\begin{aligned} h &= \textstyle\int_{x_n}^{x_{n+1}} 1 dx = A_0 \cdot 1 + A_1 \cdot 1 + A_2 \cdot 1 + A_3 \cdot 1 \\ h^2/2 &= \textstyle\int_{x_n}^{x_{n+1}} (x - x_n)\, dx = A_0 \cdot 0 + A_1 \cdot (-h) + A_2 \cdot (-2h) + A_3 \cdot (-3h), \\ h^3/3 &= \textstyle\int_{x_n}^{x_{n+1}} (x - x_n)^2\, dx = A_0 \cdot 0 + A_1 \cdot (-h)^2 + A_2 \cdot (-2h)^2 + A_3 \cdot (-3h)^2 \\ h^4/4 &= \textstyle\int_{x_n}^{x_{n+1}} (x - x_n)^3\, dx = A_0 \cdot 0 + A_1 \cdot (-h)^3 + A_2 \cdot (-2h)^3 + A_3 \cdot (-3h)^3. \end{aligned}$$

   (c) Solving the system gives

$$A_0 = \frac{55}{24}h, A_1 = \frac{-59}{24}h, A_2 = \frac{37}{24}h, A_3 = \frac{-9}{24}h$$

   so the final scheme is

$$y_{n+1} = y_n + \frac{h}{24}\left[55 f(x_n, y_n) - 59 f(x_{n-1}, y_{n-1}) + 37 f(x_{n-2}, y_{n-2}) - 9 f(x_{n-3}, y_{n-3})\right]. \tag{2}$$

(d) Rate of convergence: use the error formula for the interpolation polynomial for $g(x)$ to compute the truncation error:

$$g(x) = P_3(x) + g[x_n, x_{n-1}, x_{n-2}, x_{n-3}, x](x - x_n)(x - x_{n-1})(x - x_{n-2})(x - x_{n-3})$$

$$E = \int_{x_n}^{x_{n+1}} g[x_n, x_{n-1}, x_{n-2}, x_{n-3}, x](x - x_n)(x - x_{n-1})(x - x_{n-2})(x - x_{n-3})\,dx$$

$$= g[x_n, x_{n-1}, x_{n-2}, x_{n-3}, \xi]\frac{251}{30}h^5 = \frac{251}{720}g^{(4)}(\xi)h^5 = \frac{251}{720}y^{(5)}(\xi)h^5$$

(e) Example: what is the truncation error for $y' = -y^2, y(0) = 1, h = 0.1$ at $x = 0$?
Solution: the exact solution to the IVP is $y = (1+x)^{-1}$ and so

$$|E| = h^5\frac{251}{720}\left|\underbrace{y^{(5)}(\xi)}_{-\frac{5!}{(1+x)^6}}\right| \leq 5! \cdot h^5\frac{251}{720} = \frac{251}{6}10^{-5} \approx 0.0004.$$

4. Implicit schemes. **Backward Euler**: use the rectangle rule approximation (or the terminal slope)

$$\int_{x_n}^{x_{n+1}} g(x)\,dx \approx hg(x_{n+1})$$

$$y_{n+1} = y_n + hf(x_{n+1}, y_{n+1})$$

(a) y_{n+1} is given implicitly as a function of y_n, in general need to solve a possibly nonlinear equation

(b) Local truncation error:

$$\int_{x_n}^{x_{n+1}} f(x, y(x))dx = \int_{x_n}^{x_{n+1}} g(x)dx = \underbrace{\int_{x_n}^{x_{n+1}} g(x_{n+1})\,dx}_{hg(x_{n+1})} + \underbrace{\int_{x_n}^{n_{n+1}} g[x_{n+1}, x](x - x_{n+1})\,dx}_{\tau_n = \text{ truncation error}}.$$

$$\tau_n = \int_{x_n}^{x_{n+1}} g[x_{n+1}, x]\underbrace{(x - x_{n+1})}_{\leq 0 \text{ in the interval}}\,dx = g[x_{n+1}, d]\int_{x_n}^{x_{n+1}} (x - x_{n+1})\,dx$$

$$= g[x_{n+1}, d]\left[\frac{(x - x_{n+1})^2}{2}\right]_{x_n}^{x_{n+1}} = -\frac{g'(c_n)}{2}h^2.$$

(c) Global error is of order 1

5. **Crank-Nicolson scheme (implicit Trapezoidal scheme).** Apply Trapezoidal rule to approximating the integral (1)

$$y_{n+1} = y_n + \frac{h}{2}[f(x_n, y_n) + f(x_{n+1}, y_{n+1})]$$

(a) Local truncation error:

$$\int_{x_n}^{x_{n+1}} f(x,y(x))dx = \int_{x_n}^{x_{n+1}} g(x)dx = \frac{h}{2}\left[g\left(x_n\right)+g\left(x_{n+1}\right)\right]$$
$$+\underbrace{\int_{x_n}^{x_{n+1}} g\left[x_n,x_{n+1},x\right]\left(x-x_n\right)\left(x-x_{n+1}\right)dx}_{\tau_n=\text{ truncation error}}.$$

$$\begin{aligned}\tau_n &= \sum_{x_n}^{x_{n+1}} g\left[x_n,x_{n+1},x\right]\underbrace{\left(x-x_n\right)}_{\geq 0\text{ in the interval}}\cdot\underbrace{\left(x-x_{n+1}\right)}_{\leq 0\text{ in the interval}}dx\\ &= g\left[x_n,x_{n+1},d\right]\int_{x_n}^{x_{n+1}}\left(x-x_n\right)\left(x-x_{n+1}\right)dx\\ &\overset{t=x-x_n}{=} g\left[x_n,x_{n+1},d\right]\int_0^h t(t-h)dx = \frac{g''\left(c_n\right)}{2!}\left[\frac{t^3}{3}-h\frac{t^2}{2}\right]_0^h\\ &= -\frac{g''\left(c_n\right)}{12}h^3.\end{aligned}$$

(b) Global error of order 2.

6. **Leap-Frog scheme.** The general rule (1) can be extended to integrate from x_{n-p} to x_{n+1}.

(a) For $p=1$ we get

$$y\left(x_{n+1}\right)-y\left(x_{n-1}\right)=\int_{x_{n-1}}^{x_{n+1}} f(x,y(x))dx$$

(b) Approximating the integral by the midpoint rule:

$$y_{n+1}=y_{n-1}+2hf\left(x_n,y_n\right)$$

(c) Local truncation error: use the symmetry $\int_{x_{n-1}}^{x_{n+1}}\left(x-x_n\right)dx=0$ and get

$$y\left(x_{n+1}\right)-y\left(x_{n-1}\right)=\int_{x_{n-1}}^{x_{n+1}} g(x)dx = 2hg\left(x_n\right)+\underbrace{\int_{x_{n-1}}^{x_{n+1}} g\left[x_n,x\right]\left(x-x_n\right)dx}_{\tau_n=\text{ truncation error}}.$$

$$\begin{aligned}\tau_n &= \int_{x_{n-1}}^{x_{n+1}} g\left[x_n,x\right]\left(x-x_n\right)dx = g\left[x_n,\tilde{x}_n,d\right]\int_{x_{n-1}}^{x_{n+1}}\left(x-x_n\right)\left(x-\tilde{x}_n\right)dx\\ &= g\left[x_n,x_n,d\right]\int_{x_{n-1}}^{x_{n+1}}\left(x-x_n\right)^2dx = \frac{g''(c)}{3}h^3.\end{aligned}$$

(d) Global error of order 2.

7. Discrete difference equations

   (a) Our schemes are usually of the form of difference equation with constant coefficients:

$$y_{n+N} + a_{N-1}y_{n+N-1} + a_{N-1}y_{n+N-2} + \ldots. + a_0 y_n = b_n$$

   for some constants $\{a_0, \ldots, a_{N-1}\}$ and initial conditions $\{y_0, \ldots, y_{N-1}\}$

   (b) Note the similarity to differential equation with constant coefficients

$$y^{(N)} + a_{N-1}y^{(N-1)} + a_{N-1}y^{(N-2)} + \ldots. + a_0 y_n = g(x).$$

   (c) For the homogeneous equation ($b_n = 0$) we will look for solutions of the form

$$y_n = \beta^n$$

   Substitution and division by β^n gives:

$$\beta^N + a_{N-1}\beta^{N-1} + a_{N-1}\beta^{N-2} + \ldots. a_1\beta + a_0 = 0$$

   This is an algebraic equation of order N. If it has N distinct real roots (we will deal only with such cases) $\{\beta_k\}_{k=1}^{N}$ then the general solution of the homogeneous equation is

$$y_n^{(H)} = C_1\beta_1^n + C_2\beta_2^n + C_3\beta_3^n + \ldots. + C_N\beta_N^n.$$

   (d) For the inhomogeneous part, we will deal only with the case when $y_n^{(P)} = C$ is a solution to the inhomogeneous equation, and therefore the solution is of the form

$$y_n = y_n^{(H)} + C$$

8. Convergence

   (a) Setup: a discrete scheme computing $\{y_n\}_{n=0}^{N}$ at the points $\{x_i\}_{i=0}^{N}$ with $x_i = x_0 + ih \in [a, b]$, with $N = N(h)$ and $Nh = x_N - x_0 \leq b - a$.

   (b) Definition: the scheme **converges** to the solution $y(x)$ of the IVP if

$$\lim_{h\to 0} \max_{0\leq n\leq N(h)} |y(x_n) - y_n| = 0.$$

   (c) We saw that Euler's scheme converges when the conditions of the theorem are satisfied, with the rate $O(h)$.

   (d) Example of **Backward Euler**:

$$\begin{aligned}
&y' = -2y, \quad f(x, y) = -2y \\
&y_{n+1} = y_n + hf(x_{n+1}, y_{n+1}), \quad n = 0, 1, \ldots \\
&y_0 = y(0) = A \\
&(1 + 2h)y_{n+1} = y_n, \quad n = 0, 1, \ldots \\
&y_{n+1} = \tfrac{y_n}{(1+2h)}, \quad n = 0, 1, \ldots \\
&y_n = \tfrac{y_0}{(1+2h)^n} = y_0 \cdot (1 + 2h)^{-n} \\
&\quad = y_0 \cdot (1 + 2h)^{-(x_n/h)} = y_0 \cdot \left((1 + 2h)^{\frac{1}{2h}}\right)^{-2x_n} \\
&\xrightarrow{h\to 0} y_0 \cdot e^{-2x_n}.
\end{aligned}$$

(e) Example of **Crank-Nicolson**:

$$
\begin{aligned}
&y' = -2y, \quad f(x,y) = -2y \\
&y_{n+1} = y_n + \tfrac{h}{2}\left[f(x_n, y_n) + f(x_{n+1}, y_{n+1})\right], \quad n = 0,1,\dots \\
&y_0 = y(0) = A \\
&y_{n+1} = y_n - h\left(y_n + y_{n+1}\right), \quad n = 0,1,\dots \\
&(1+h)y_{n+1} = (1-h)y_n, \quad n = 0,1,\dots. \\
&y_{n+1} = \tfrac{(1-h)}{(1+h)} y_n, \quad n = 0,1,\dots \\
&y_n = \left(\tfrac{1-h}{1+h}\right)^n y_0 \\
&\quad = y_0 \cdot \left(\tfrac{1+h-2h}{1+h}\right)^n = y_0\left(1 - 2\tfrac{h}{1+h}\right)^{(x_n/h)} \\
&\quad = y_0 \cdot \left(1 - 2\tfrac{h}{1+h}\right)^{\frac{1+h}{-2h}\cdot\frac{-2x_n}{1+h}} = y_0 \cdot \left(\left(1 - 2\tfrac{h}{1+h}\right)^{\frac{1+h}{-2h}}\right)^{\frac{-2x_n}{1+h}} \\
&\quad \xrightarrow{h\to 0} y_0 \cdot e^{-2x_n}
\end{aligned}
$$

(f) Example of **Leap-Frog**: we get a finite difference equation (homogeneous)

$$
\begin{aligned}
y' = -2y, \quad f(x,y) = -2y, \quad y(0) = y_0 = 3 \Rightarrow y(x) = 3e^{-2x} \\
y_{n+1} = y_{n-1} - 4hy_n \\
y_{n+1} + 4hy_n - y_{n-1} = 0 \\
\beta^2 + 4\beta h - 1 = 0 \to \beta = -2h \pm \sqrt{4h^2+1} \\
y_n = C_1\left(-2h + \sqrt{4h^2+1}\right)^n + C_2\left(-2h - \sqrt{4h^2+1}\right)^n
\end{aligned}
$$

The first summand converges to the solution (use Taylor approximation $\sqrt{1+x} \approx 1 + \frac{1}{2}x$:

$$
C_1\left(-2h + \sqrt{4h^2+1}\right)^n \approx C_1\left(-2h + 1 + \frac{4h^2}{2}\right)^n \approx C_1(-2h+1)^n = C_1(-2h+1)^{(x_n - x_0)/h} \approx
$$
$$
\approx C_1 e^{-2(x_n - x_0)} = K_1 e^{-2x_n}
$$

but the second summand does not:

$$
\begin{aligned}
C_2\left(-2h - \sqrt{4h^2+1}\right)^n \approx C_2\left(-2h - 1 - \tfrac{4h^2}{2}\right)^n \approx C_2(-2h-1)^n = \\
= C_2(-1)^n(1+2h)^{(x_n - x_0)/h} \approx K_2(-1)^n e^{2x_n}
\end{aligned}
$$

This is a "parasitic" term. So **the scheme does not necessarily converge**.

9. Stability of single-step methods

   (a) An iterative process is considered to be stable if small errors cannot "blow up" after many steps

   (b) Consider $y' = \lambda y$

(c) Write a one-step method as a homogeneous difference equation (see above)

$$y_{n+1} = g(\lambda h) y_n$$

(d) If we commit a small error in the initial condition $\tilde{y}_0 = y_0 + \delta_0$ then the sequence of iterates $\{\tilde{y}_n\}_{n=0}^{N}$ will have errors $\tilde{y}_n - y_n = \delta_n$ which also satisfy the difference equation

(e) Definition: the scheme is **stable** if $|g(\lambda h)| \leq 1$, and **strongly stable** if $|g(\lambda h)| < 1$.

(f) Example of **Forward Euler:** need $|1 + \lambda h| \leq 1$ for **stability**

   i. This is a problem for λ large and negative (need very small h). The Euler method "overshoots" the true slope

(g) Example of **Backward Euler:**

$$\begin{aligned} y_{n+1} &= y_n + h\lambda y_{n+1} \\ y_{n+1} &= \frac{y_n}{1 - h\lambda} \\ g\left(\lambda h\right) &= \frac{1}{1 - \lambda h} \end{aligned}$$

Stability when $\left|\frac{1}{1-\lambda h}\right| \leq 1 \Rightarrow |1 - \lambda h| \geq 1$. If $\lambda < 0$ the scheme is strongly stable for all h.

(h) Example of **Crank-Nicolson:**

$$\begin{aligned} y_{n+1} &= \frac{(1 + \lambda h)}{(1 - \lambda h)} y_n \\ |1 + \lambda h| &\leq |1 - \lambda h| \\ \Re\lambda \leq 0 &\iff \text{stability} \end{aligned}$$

10. Stability in general

(a) More generally, a discrete scheme (one-step/multi-step) is given by a difference equation with initial conditions $\{y_n\}_{n=0}^{p}$. The scheme is stable if a small change in the initial conditions will not cause an unbounded change in the solution.

(b) Definition: an approximation scheme is **stable** if for every $\varepsilon > 0$ there exist constants h_0, Z s.t. for every $h \leq h_0$ and every initial condition $\{\tilde{y}_n\}_{n=0}^{p}$ with

$$\max_{0 \leq n \leq p} |y_n - \tilde{y}_n| < \varepsilon \tag{3}$$

the solution $\{\tilde{y}_n\}_{n=0,1,\ldots}$ will satisfy

$$|y_n - \tilde{y}_n| \leq Z \qquad \forall n \geq 0.$$

We will prefer stable schemes, otherwise the numerical solution may diverge from the true one after a large number of steps.

11. Example of **Leap-Frog**: with $y' = \lambda y$ we can repeat the previous calculation to derive the general solution of the form

$$y_{n+1} - 2\lambda h y_n - y_{n-1} = 0 \tag{4}$$

$$y_n = C_1 \Bigg(\underbrace{\lambda h + \sqrt{\lambda^2 h^2 + 1}}_{\beta_1} \Bigg)^n + C_2 \Bigg(\underbrace{\lambda h - \sqrt{\lambda^2 h^2 + 1}}_{\beta_2} \Bigg)^n \tag{5}$$

$$\text{Initial conditions: } \begin{array}{l} y_0 = C_1 + C_2 \\ y_1 = C_1 \left(\lambda h + \sqrt{\lambda^2 h^2 + 1}\right) + C_2 \left(\lambda h - \sqrt{\lambda^2 h^2 + 1}\right) \end{array}$$

$$\text{Solving for } C_1, C_2 : C_1 = \frac{y_0 \beta_2 - y_1}{\beta_2 - \beta_1}, \quad C_2 = \frac{y_0 \beta_1 - y_1}{\beta_1 - \beta_2}$$

$$\text{For } h \ll 1 : |C_1| \leq \frac{1}{2\sqrt{\lambda^2 h^2 + 1}} \left(|\beta_2 y_0| + |y_1|\right) \leq \frac{2\,|y_0| + |y_1|}{2}$$

$$|C_2| \leq \frac{1}{2\sqrt{\lambda^2 h^2 + 1}} \left(|\beta_1 y_0| + |y_1|\right) \leq \frac{2\,|y_0| + |y_1|}{2}$$

(a) Since both sequences $\{y_n\}$ and $\{\tilde{y}_n\}$ satisfy the homogeneous equation (4), their difference does so as well.

(b) If (3) is satisfied, then we can find h_0 such that for every $h \leq h_0$ the constants C_1, C_2 are bounded.

(c) In order to have the general term (5) bounded for every n, we therefore need to ensure that $|\beta_1| \leq 1$ and $|\beta_2| \leq 1$.

(d) Clearly if $\lambda = 0$ then $\beta_1 = 1,\ \beta_2 = -1$ and so we have stability.

(e) If $\lambda > 0$ then $\beta_1 > 1$, and if $\lambda < 0$ then $\beta_2 < -1$

(f) Conclusion: Leap-Frog is stable only if $\lambda = 0$.