CALCULUS

Lecture 14: Numerical Methods for ODEs

Anshid Aboobacker

Table of Contents

When Exact Solutions Fail

So far, we have solved ODEs analytically (separation of variables, integrating factors, Laplace transforms). But most real-world DEs cannot be solved exactly.

Examples where closed-form solutions are unavailable:

- $y' = x^2 + y^2$ (Riccati equation — no general closed form)
- $y'' + \sin(y) = 0$ (nonlinear pendulum)
- Complex engineering or biological systems

Numerical methods approximate the solution by computing y at a sequence of discrete points $x_0, x_1, x_2, \ldots$ starting from the initial condition.

Running example throughout this lecture:

$$\frac{dy}{dx} = 2xy, \quad y(0) = 1, \quad \text{exact solution: } y = e^{x^2}$$

(We use a solvable example so we can measure accuracy.)

Slope Fields and Step-by-Step Idea

Key observation: At any point (x, y), the ODE tells us the slope:

$$y'(x) = f(x, y)$$

The general approach:

- Start at (x_0, y_0) — the initial condition
- Use the known slope $f(x_0, y_0)$ to take a small step of size h
- Arrive at (x_1, y_1); repeat from the new point
- Each step uses: $x_{n+1} = x_n + h$

The step size h: smaller h gives more accuracy but requires more computation. The trade-off between accuracy and cost is the central question in numerical methods.

Error Terminology

- **Local truncation error (LTE):** the error introduced in a *single step*, assuming the previous value is exact. Measures how well the formula approximates the Taylor series.
- **Global error:** the cumulative error at a fixed point x after many steps. Approximately $=$ (number of steps) $\times$ LTE per step.
- For step size h and $n = \lfloor (x - x_0)/h \rfloor$ steps: global error $\approx \frac{x-x_0}{h} \times$ LTE.

Order of a method: If LTE $= O(h^{p+1})$, global error $= O(h^p)$ and we say the method is **order** p.

- Euler: order 1 (global error $\sim h$)
- Improved Euler: order 2 (global error $\sim h^2$)
- RK4: order 4 (global error $\sim h^4$)

Table of Contents

Geometric Motivation

At (x_n, y_n), the tangent line has slope $f(x_n, y_n)$. Use it to step forward by h:

$$y_{n+1} \approx y_n + h \cdot y'(x_n) = y_n + h \cdot f(x_n, y_n)$$

This is the first two terms of the Taylor expansion of $y(x_n + h)$:

$$y(x_n + h) = y(x_n) + h\, y'(x_n) + \underbrace{\frac{h^2}{2} y''(x_n) + \cdots}_{\text{truncated}}$$

The truncated terms give the local error $O(h^2)$, so the global error is $O(h)$.

Euler's Method

$$y_{n+1} = y_n + h\, f(x_n,\, y_n), \qquad x_{n+1} = x_n + h$$

Euler's Method: Worked Example (Setup)

Problem: Solve $y' = 2xy$, $y(0) = 1$, using $h = 0.25$ from $x = 0$ to $x = 1$.

Here $f(x, y) = 2xy$. We compute y_1, y_2, y_3, y_4.

Step 1 ($x_0 = 0$, $y_0 = 1$):

$$f(0, 1) = 2(0)(1) = 0$$
$$y_1 = 1 + 0.25 \cdot 0 = 1.0000$$

Step 2 ($x_1 = 0.25$, $y_1 = 1.0000$):

$$f(0.25,\ 1.0000) = 2(0.25)(1.0000) = 0.5000$$
$$y_2 = 1.0000 + 0.25(0.5000) = 1.1250$$

Step 3 ($x_2 = 0.5$, $y_2 = 1.1250$):

$$f(0.5,\ 1.1250) = 2(0.5)(1.1250) = 1.1250$$
$$y_3 = 1.1250 + 0.25(1.1250) = 1.4063$$

Euler's Method: Full Table and Comparison

Step 4 ($x_3 = 0.75$, $y_3 = 1.4063$): $f(0.75, 1.4063) = 2.1094$, $y_4 = 1.9336$

Comparison with exact solution $y = e^{x^2}$:

x_n	y_n (Euler)	y exact	Error
0.00	1.0000	1.0000	0.0000
0.25	1.0000	1.0645	0.0645
0.50	1.1250	1.2840	0.1590
0.75	1.4063	1.7551	0.3488
1.00	1.9336	2.7183	0.7847

Error at $x = 1$ is 0.78 — nearly 29%. Euler's method needs a much smaller h for acceptable accuracy.

Effect of Step Size and Error Analysis

Halving h: With $h = 0.125$, the error at $x = 1$ drops to approximately 0.39 — roughly half.

This is consistent with the **first-order** ($O(h)$) global error: halving h halves the error.

Euler's Method Error

- Local truncation error: $O(h^2)$
- Global error: $O(h)$ — first-order method
- Simple to implement but requires very small h for accuracy

To achieve error < 0.01 at $x = 1$ for this example, we would need $h \approx 0.003$ — over 300 steps. Better methods are needed.

Table of Contents

Idea: Average the Slope

Euler's weakness: uses only the slope at the *start* of the interval.

Better idea: average the slopes at the start and end of the interval.

- **Predictor** (Adams-Bashforth step): use Euler to get a rough estimate y^*_{n+1}
- Evaluate f at the predicted point to get the slope at the end
- **Corrector**: average the two slopes and re-step

Improved Euler / Predictor-Corrector Formulas

$$y^*_{n+1} = y_n + h\,f(x_n,\, y_n) \qquad \text{(Predictor)}$$

$$y_{n+1} = y_n + \frac{h}{2}\left[f(x_n,\, y_n) + f(x_{n+1},\, y^*_{n+1})\right] \qquad \text{(Corrector)}$$

Global error: $O(h^2)$ — **second-order** method. Requires 2 function evaluations per step.

Improved Euler: Steps 1 and 2

Same problem: $y' = 2xy$, $y(0) = 1$, $h = 0.25$.
Step 1 ($x_0 = 0$, $y_0 = 1$):

$$y_1^* = 1, \quad f(0.25, 1) = 0.5, \quad y_1 = 1 + 0.125[0 + 0.5] = 1.0625$$

Exact: 1.0645. Error: 0.0020 (vs Euler's 0.0645).

Step 2 ($x_1 = 0.25$, $y_1 = 1.0625$):

$$\begin{aligned} f(0.25,\ 1.0625) &= 0.5313, \quad y_2^* = 1.1953 \\ f(0.5,\ 1.1953) &= 1.1953 \\ y_2 &= 1.0625 + 0.125[0.5313 + 1.1953] = 1.2783 \end{aligned}$$

Exact: 1.2840. Error: 0.0057 (vs Euler's 0.1590).

Improved Euler: Error Comparison

x_n	Euler	Imp. Euler	Exact
0.00	1.0000	1.0000	1.0000
0.25	1.0000	1.0625	1.0645
0.50	1.1250	1.2783	1.2840
0.75	1.4063	1.7279	1.7551
1.00	1.9336	2.6411	2.7183

Errors at $x = 1$: Euler $=$ 0.785 (29%), Imp. Euler $=$ 0.077 (2.8%).
10$\times$ accuracy improvement for only 2$\times$ the computation. The $O(h^2)$ order means halving h quarters the error.

Practice: Improved Euler

Problem: Use Improved Euler with $h = 0.5$ to approximate $y(1)$ for $y' = y$, $y(0) = 1$. (Exact: e.)

Solution: $f(x, y) = y$.
Step 1 ($x_0 = 0$, $y_0 = 1$):

$$\begin{aligned} y_1^* &= 1 + 0.5(1) = 1.5 \\ y_1 &= 1 + \tfrac{0.5}{2}\left[f(0,1) + f(0.5,1.5)\right] \\ &= 1 + 0.25[1 + 1.5] = 1 + 0.625 = 1.625 \end{aligned}$$

Step 2 ($x_1 = 0.5$, $y_1 = 1.625$):

$$y_2^* = 1.625 + 0.5(1.625) = 2.4375$$
$$y_2 = 1.625 + 0.25[1.625 + 2.4375] = 1.625 + 1.0156 = 2.6406$$

Exact $e \approx 2.7183$. Error: 0.0777. (Euler same h: $y(1) = 2.25$, error $= 0.47$)

Table of Contents

Motivation: Sampling More Slopes

Idea: sample the slope at *four* strategically chosen points within the interval and form a weighted average.

The four samples approximate the slope at:

- k_1: the start of the interval
- k_2: the midpoint, using k_1 for prediction
- k_3: the midpoint again, using k_2 for a better prediction
- k_4: the end of the interval, using k_3

The final update is a **weighted average**: $y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$.

RK4 Formulas

$$
\begin{aligned}
k_1 &= h\,f(x_n,\ y_n) \\
k_2 &= h\,f\left(x_n + \tfrac{h}{2},\ y_n + \tfrac{k_1}{2}\right) \\
k_3 &= h\,f\left(x_n + \tfrac{h}{2},\ y_n + \tfrac{k_2}{2}\right) \\
k_4 &= h\,f(x_n + h,\ y_n + k_3) \\
y_{n+1} &= y_n + \tfrac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)
\end{aligned}
$$

Global error: $O(h^4)$ — **fourth-order** method. Requires 4 function evaluations per step. Self-starting.

RK4: First Step Computation

Problem: $y' = 2xy$, $y(0) = 1$, $h = 0.25$. Compute y_1.

$f(x, y) = 2xy$. At $x_0 = 0$, $y_0 = 1$:

$$k_1 = 0.25 \cdot f(0,\ 1) = 0.25 \cdot 0 = 0$$

$$k_2 = 0.25 \cdot f(0.125,\ 1) = 0.25 \cdot 2(0.125)(1) = 0.0625$$

$$\begin{aligned} k_3 &= 0.25 \cdot f(0.125,\ 1 + 0.0625/2) = 0.25 \cdot f(0.125,\ 1.03125) \\ &= 0.25 \cdot 2(0.125)(1.03125) = 0.25 \cdot 0.25781 = 0.06445 \end{aligned}$$

$$\begin{aligned} k_4 &= 0.25 \cdot f(0.25,\ 1 + 0.06445) = 0.25 \cdot f(0.25,\ 1.06445) \\ &= 0.25 \cdot 2(0.25)(1.06445) = 0.25 \cdot 0.53223 = 0.13306 \end{aligned}$$

$$y_1 = 1 + \tfrac{1}{6}(0 + 2(0.0625) + 2(0.06445) + 0.13306) = 1.06449$$

Exact: $e^{0.0625} = 1.06449$. Error: < 0.00001

RK4: Full Table and Accuracy

x_n	y_n (RK4)	y exact	Error
0.00	1.00000	1.00000	< 0.00001
0.25	1.06449	1.06449	< 0.00001
0.50	1.28403	1.28403	< 0.00001
0.75	1.75505	1.75505	< 0.00001
1.00	2.71822	2.71828	0.00006

RK4 vs Euler (same $h = 0.25$):

- Euler error at $x = 1$: 0.785
- RK4 error at $x = 1$: 0.00006

RK4 is a **fourth-order** method: global error $O(h^4)$. Halving h reduces error by a factor of 16. This is why RK4 is the standard workhorse of numerical ODEs.

Practice: RK4

Problem: Use RK4 with $h = 0.5$ to find $y(0.5)$ for $y' = y$, $y(0) = 1$. (Exact: $e^{0.5} \approx 1.6487$.)

Solution: $f(x, y) = y$. At $x_0 = 0$, $y_0 = 1$:

$$k_1 = 0.5 \cdot f(0,\ 1) = 0.5$$
$$k_2 = 0.5 \cdot f(0.25,\ 1 + 0.25) = 0.5 \cdot 1.25 = 0.625$$
$$k_3 = 0.5 \cdot f(0.25,\ 1 + 0.3125) = 0.5 \cdot 1.3125 = 0.65625$$
$$k_4 = 0.5 \cdot f(0.5,\ 1 + 0.65625) = 0.5 \cdot 1.65625 = 0.82813$$

$$\begin{aligned} y_1 &= 1 + \tfrac{1}{6}(0.5 + 1.25 + 1.3125 + 0.82813) \\ &= 1 + \tfrac{1}{6}(3.89063) = 1.64844 \end{aligned}$$

Error: $|1.64844 - 1.64872| = 0.00028$. Compare: Euler gives 1.5 (error 0.15).

Table of Contents

Multi-Step Methods: Idea

Single-step methods (Euler, RK4) use only y_n to get y_{n+1}.

Multi-step methods reuse previously computed values $y_n, y_{n-1}, \ldots$, which costs almost nothing extra.

ABM (Adams-Bashforth-Moulton): 4-step predictor-corrector:

1. **Startup:** Use RK4 to generate y_0, y_1, y_2, y_3
2. **Predictor:** Adams-Bashforth 4-step explicit formula
3. **Corrector:** Adams-Moulton 4-step implicit formula

ABM achieves $O(h^4)$ accuracy with **2 evaluations/step** (vs. RK4's 4).

Adams-Bashforth Predictor

Let $y_n' = f(x_n, y_n)$. The **Adams-Bashforth 4-step predictor** is:

$$y_{n+1}^* = y_n + \frac{h}{24}\left(55y_n' - 59y_{n-1}' + 37y_{n-2}' - 9y_{n-3}'\right)$$

Derivation: Fit a polynomial through $(x_{n-3}, y_{n-3}'), \ldots, (x_n, y_n')$ and integrate over $[x_n, x_{n+1}]$.

For our example $y' = 2xy$, $y(0) = 1$, $h = 0.25$, using RK4 startup values:

$$\begin{aligned}
y_0 &= 1.00000, & y_0' &= 0 \\
y_1 &= 1.06449, & y_1' &= 2(0.25)(1.06449) = 0.53225 \\
y_2 &= 1.28403, & y_2' &= 2(0.50)(1.28403) = 1.28403 \\
y_3 &= 1.75505, & y_3' &= 2(0.75)(1.75505) = 2.63258
\end{aligned}$$

ADAMS-MOULTON CORRECTOR AND PRACTICE

ABM Predictor ($n = 3$, finding y_4 at $x = 1$):

$$\begin{aligned} y_4^* &= 1.75505 + \tfrac{0.25}{24}\big(55(2.63258) - 59(1.28403) \\ &\qquad + 37(0.53225) - 9(0)\big) \\ &= 1.75505 + 0.01042(88.73) = 2.6793 \end{aligned}$$

$y_4'^* = 2(1)(2.6793) = 5.3586.$ **Adams-Moulton Corrector:**

$$y_{n+1} = y_n + \tfrac{h}{24}(9y'_{n+1} + 19y'_n - 5y'_{n-1} + y'_{n-2})$$

Corrected y_4: $1.75505 + \frac{0.25}{24}(92.34) = 2.7169.$ Exact: 2.7183, error: 0.0014

Practice: ABM Method

Problem: Given $y' = y$, $y(0) = 1$, $h = 0.5$, startup values from RK4:

$$y_0 = 1.00000,\; y_0' = 1.00000 \quad y_1 = 1.64872,\; y_1' = 1.64872$$
$$y_2 = 2.71828,\; y_2' = 2.71828 \quad y_3 = 4.48169,\; y_3' = 4.48169$$

Use ABM to predict and correct y_4 at $x = 2.0$.
Solution (Predictor):

$$y_4^* = 4.48169 + \tfrac{0.5}{24}(138.11) = 4.48169 + 3.69 = 7.359$$

(Exact $e^2 = 7.389$; corrector closes the gap further.)

Table of Contents

Comparison of Numerical Methods

Method	Order	Evals/step	Notes
Euler	$O(h)$	1	Simple; large errors
Improved Euler	$O(h^2)$	2	Predictor-corrector
RK4	$O(h^4)$	4	Standard workhorse
ABM (4-step)	$O(h^4)$	2 (after startup)	Efficient for long runs

Practical guidelines:

- **Euler:** useful for teaching concepts only
- **RK4:** default choice for most problems; self-starting; robust
- **ABM:** preferred for long integrations; needs RK4 startup; not self-starting
- **Adaptive step size:** adjusts h to control error (not covered here)

Practice Problem 1: Euler's Method

Problem: Approximate $y(0.4)$ using Euler's method with $h = 0.2$ for

$$y' = x + y, \quad y(0) = 1$$

Solution: $f(x, y) = x + y$.
Step 1 ($x_0 = 0$, $y_0 = 1$):

$$y_1 = 1 + 0.2(0 + 1) = 1.2$$

Step 2 ($x_1 = 0.2$, $y_1 = 1.2$):

$$y_2 = 1.2 + 0.2(0.2 + 1.2) = 1.2 + 0.28 = 1.48$$

Exact solution: $y = 2e^x - x - 1$. $\quad y(0.4) = 2e^{0.4} - 1.4 \approx 1.584$.
Error: $|1.48 - 1.584| = 0.104$.

Practice Problem 2: RK4

Problem: Compute y_1 using RK4 with $h = 0.2$ for $y' = x^2 + y$, $y(0) = 1$.

Solution: $f(x, y) = x^2 + y$.

$$k_1 = 0.2 \cdot f(0,\ 1) = 0.2(0 + 1) = 0.2$$

$$k_2 = 0.2 \cdot f(0.1,\ 1.1) = 0.2(1.11) = 0.222$$

$$k_3 = 0.2 \cdot f(0.1,\ 1.111) = 0.2(1.121) = 0.2242$$

$$k_4 = 0.2 \cdot f(0.2,\ 1.2242) = 0.2(1.2642) = 0.2528$$

$$\begin{aligned} y_1 &= 1 + \tfrac{1}{6}(0.2 + 2(0.222) + 2(0.2242) + 0.2528) \\ &= 1 + \tfrac{1}{6}(1.3452) = 1.2242 \end{aligned}$$

Thank You