05b_numerical_methods_IVPs

December 10, 2025

1 One-step Methods for Initial Value Problems

1.0.1 *Lecture 2*

1.1 Overview

In today's lecture we will introduce **forward** and **backward Euler methods** and discuss some of their main properties.
These notes are based largely on Chapters 5–7 from the book by Randall LeVeque *Finite Difference Methods for Ordinary and Partial Differential Equations* (SIAM, Philadelphia, PA, 2007), and Chapter 5 of *Numerical Analysis* (Brooks/Cole, 2010) by R. L. Burden and J. D. Faires.

1.2 Some Basic Numerical Methods: Forward and Backward Euler

To solve initial value problems numerically, we start at t_0 and seek an approximate solution U^n at successive times
$t_1, t_2, t_3, \ldots$ with time step $k = t_n - t_{n-1}$ so that $t_n = nk$.

We set $U^0 = \eta$ since that is the given initial condition, and we aim to determine
$U^1, U^2, \ldots$ with $U^n \approx u(t_n)$.

1.2.1 Forward Euler Method

Replacing $u'(t_n)$ in $u' = f(u)$ by the forward finite difference approximation gives

$$\frac{U^{n+1} - U^n}{k} = f(U^n) \qquad \Longrightarrow \qquad U^{n+1} = U^n + kf(U^n).$$

1.2.2 Backward Euler Method

Replacing $u'(t_n)$ by the backward finite difference approximation gives

$$\frac{U^{n+1} - U^n}{k} = f(U^{n+1}) \qquad \Longrightarrow \qquad U^{n+1} = U^n + kf(U^{n+1}).$$

The forward Euler method gives an **explicit** update (easy to compute).
The backward Euler method gives an **implicit** update (requires solving for U^{n+1}).

1.2.3 Trapezoidal Method

Another implicit method is the **trapezoidal method**, obtained by averaging the two Euler methods:

$$\frac{U^{n+1} - U^n}{k} = \frac{1}{2}\big(f(U^n) + f(U^{n+1})\big).$$

This symmetric approximation is **second-order accurate**, whereas the Euler methods are **first-order**.

1.2.4 One-step Methods

The above methods are called **one-step methods**, meaning U^{n+1} depends only on U^n (not on earlier values).

1.3 Local Truncation Error

1.3.1 Definition

For the forward Euler method, the **local truncation error (LTE)** at step n is defined as

$$\tau^n = \frac{u(t_{n+1}) - u(t_n)}{k} - f(u(t_n)),$$

where $u(t_n)$ is the exact solution.

Using the Taylor expansion of $u(t_{n+1})$ about t_n, we get

$$\tau^n = u'(t_n) + \frac{k}{2}u''(t_n) + \frac{k^2}{6}u'''(t_n) - f(u(t_n)) + O(k^3) = \frac{k}{2}u''(t_n) + O(k^2).$$

Thus, the local truncation error of the forward Euler method is $O(k)$.

1.3.2 Definition (Consistency)

A numerical method for an IVP is **consistent** if $\tau^n \to 0$ as $k \to 0$.

Hence, the forward Euler method is consistent.

1.4 One-step Error

We also define a **one-step error** $\mathcal{L}^n$ based on the update form of a method.

1.4.1 Definition

For forward Euler,

$$\mathcal{L}^n = u(t_{n+1}) - u(t_n) - kf(u(t_n)).$$

Expanding $u(t_{n+1})$ in Taylor series gives

$$\mathcal{L}^n = ku'(t_n) + \frac{k^2}{2}u''(t_n) - kf(u(t_n)) + O(k^3) = \frac{k^2}{2}u''(t_n) + O(k^3).$$

Notice that $\mathcal{L}^n = k\tau^n$, so $\mathcal{L}^n = O(k^2)$.
It measures the error after one step assuming past values are exact.

1.5 Convergence

A numerical method **converges** if, for a fixed final time $T > 0$, the approximate solution U^N satisfies

$$\lim_{\substack{k\to 0\\ Nk=T}} U^N = u(T).$$

That is, as $k \to 0$, the approximation approaches the true solution.

1.5.1 Definition (Convergence)

A one-step numerical method is **convergent** if, for any Lipschitz continuous $f(u,t)$ and $u(t_0) = U^0 = \eta$,

$$\lim_{\substack{k\to 0\\ Nk=T}} U^N = u(T)$$

for every fixed $T > 0$ where the ODE has a unique solution.

1.5.2 Definition (Convergence of Order p)

A method is **convergent of order** p if the global error satisfies

$$\|E^n\|_\infty = \|U^n - u(t_n)\|_\infty = O(k^p), \quad n = 1, 2, \dots, N.$$

1.6 Convergence of Euler's Forward Method

1.6.1 Linear Problems

Consider the linear test problem

$$
\begin{aligned}
u'(t) &= \lambda u(t) + g(t), \\
u(t_0) &= \eta,
\end{aligned}
$$

with exact solution (Duhamel's principle):

$$u(t) = e^{\lambda(t-t_0)}\eta + \int_{t_0}^{t} e^{\lambda(t-\tau)} g(\tau)\, d\tau.$$

For forward Euler:

$$U^{n+1} = (1 + k\lambda)U^n + kg(t_n). \tag{1}$$

For the exact solution,

$$u(t_{n+1}) = (1 + k\lambda)u(t_n) + kg(t_n) + k\tau^n, \tag{2}$$

where $\tau^n = \frac{k}{2}u''(t_n) + O(k^2)$.

Subtracting (1) and (2) gives

$$E^{n+1} = (1 + k\lambda)E^n - k\tau^n.$$

Solving recursively,

$$E^n = (1 + k\lambda)^n E^0 - k \sum_{m=1}^{n} (1 + k\lambda)^{n-m} \tau^{m-1}.$$

Bounding this using $|(1 + k\lambda)^{n-m}| \leq e^{|\lambda|T}$ gives

$$|E^n| \leq e^{|\lambda|T}(|E^0| + T\|\tau\|_\infty).$$

Since $E^0 = 0$, we find

$$|E^n| \leq e^{|\lambda|T} T\|\tau\|_\infty = O(k).$$

Hence, **forward Euler is first-order convergent**.

1.6.2 Nonlinear Problems

For a nonlinear problem $u' = f(u)$ with Lipschitz continuous f, forward Euler gives

$$U^{n+1} = U^n + kf(U^n).$$

Then

$$E^{n+1} = E^n + k[f(U^n) - f(u(t_n))] - k\tau^n.$$

Using Lipschitz continuity:

$$|f(U^n) - f(u(t_n))| \leq L|E^n|,$$

so

$$|E^{n+1}| \leq (1 + kL)|E^n| + k|\tau^n|.$$

By induction,

$$|E^n| \leq (1 + kL)^n |E^0| + k \sum_{m=1}^{n} (1 + kL)^{n-m} |\tau^{m-1}|.$$

Thus,

$$|E^n| \leq e^{LT} T \|\tau\|_\infty = O(k), \quad \text{as } k \to 0.$$

1.7 Exercises

1.7.1 Exercise 1

Given the IVP

$$\begin{cases} u' = cu, & t \in [0, 1], \\ u(0) = \eta, \end{cases}$$

(a) Derive Euler's method formula.

(b) Find the exact solution.

(c) Show that Euler's method converges as $n \to \infty$.

1.7.2 Exercise 2

Apply Euler's method with step size $k = 1/4$ to

$$\begin{cases} u' = 2(t+1)u, \quad t \in [0,1], \\ u(0) = 1. \end{cases}$$

Compute U^n for $n = 0, \dots, 4$, and compare with the exact solution at $t = 1$.

1.7.3 Exercise 3

Show that applying Euler's method to

$$u' = f(t), \quad u(a) = 0,$$

over $[a, b]$ gives

$$u(b) \approx \sum_{i=0}^{N-1} f(t_i)k,$$

which is a **Riemann sum** for the integral $\int_a^b f(t)\,dt$.

1.7.4 Exercise 4

Show that Euler's method fails to approximate

$$u' = 1.5u^{1/3}, \quad u(0) = 0,$$

whose exact solution is $u(t) = t^{3/2}$.
Explain the difficulty encountered.

1.7.5 Computational Exercise

Consider the IVP

$$u' = \frac{u^2 + u}{t}, \quad 1 \le t \le 5, \quad u(1) = -2,$$

whose exact solution is $u(t) = \frac{2t}{1-2t}$.

Implement the following in Python:

1. **Forward Euler**:

$$U^{n+1} = U^n + kf(U^n, t_n)$$

2. **Backward Euler**:

$$U^{n+1} = U^n + kf(U^{n+1}, t_{n+1})$$

Tasks:

(a) Compute with $k = 0.1$ and $k = 0.05$, plot both results.

(b) Compare with exact solution and plot errors.

(c) Check if global error halves when k is halved.

(d) Compute convergence rate using

$$p = \log_2 \left(\frac{\|E(k)\|}{\|E(k/2)\|} \right)$$

and verify that

$$\log \|E(k)\| = p \log(k) + \log A.$$

[ ]:

```

```