Lecture on
Initial Value Problems

Eskil Hansen

Approx. chap. 6.1–6.6 in Sauer
Initial value problems (IVP)

Consider the nonlinear IVP

\[ y'(t) = f(t, y(t)), \quad t > a, \]

with the initial value \( y(a) = y_0 \).

**Simplest possible example:** Radioactive decay

The unknown \( y(t) \) models the amount of radioactive substance at time \( t \) and

\[ y'(t) = -\lambda y(t), \quad t > 0, \]

with \( \lambda > 0 \) and the initial amount \( y(0) = y_0 \).

**Solution:** \( y(t) = y_0 e^{-\lambda t} \), i.e., \( y(t) \to 0 \) as \( t \to \infty \).
Transformation to standard form

Example:

\[ v''(t) = g(t, v(t), v'(t)), \quad v(0) = v_0, \text{ and } v'(0) = v'_0 \]

**Standard substitution:** Introduce

\[ y_1 = v \quad \text{and} \quad y_2 = v', \]

and we then obtain

\[
y' = \begin{bmatrix} y'_1 \\ y'_2 \end{bmatrix} = \begin{bmatrix} y_2 \\ g(t, y_1, y_2) \end{bmatrix} = f(t, y)
\]

with \( y(0) = [y_1(0), y_2(0)]^T = [v_0, v'_0]^T. \)
Discretization: Explicit Euler (1768)

Discretize: Continuous $\rightarrow$ Discrete

$t \geq 0 \rightarrow t_n = n\Delta t, \ n = 0, 1, 2, \ldots$

$y(t) \rightarrow u_n$

$\frac{d}{dt}y = f(t, y) \rightarrow \Delta/\Delta t \ u_n = f(t_n, u_n)$

This gives us the discrete problem

$$u_{n+1} = u_n + \Delta t \ f(t_n, u_n), \ u_0 = y_0,$$

where $u_n \approx y(t_n)$.

This discretization is referred to as the **Explicit Euler** scheme ($u_{n+1}$ is explicitly given by $u_n$).
Explicit Euler: alternative derivation (one of many)

We can also derive the EE scheme via a Taylor expansion:

\[ y(t_n + \Delta t) = y(t_n) + \Delta t \, y'(t_n) + O(\Delta t^2) \]

\[ \approx y(t_n) + \Delta t \, f(t_n, y(t_n)) \]

\[ \Rightarrow \]

\[ u_{n+1} = u_n + \Delta t \, f(t_n, u_n) \]

“Take a step of size \( \Delta t \) in the direction of the tangent.”
Explicit Euler...

“Take a step of size $\Delta t$ in the direction of the tangent.”

Each step introduces an error and ends up on a different solution trajectory (dashed curves).
Implicit Euler

Another approach is to use the expansion

\[ y(t_n - \Delta t) = y(t_n) - \Delta t \, y'(t_n) + \mathcal{O}(\Delta t^2) \]

\[ \approx y(t_n) - \Delta t \, f(t_n, y(t_n)) \]

\[ \Rightarrow \]

\[ y(t_n) \approx y(t_n - \Delta t) + \Delta t \, f(t_n, y(t_n)) \]

which gives us the **Implicit Euler** scheme

\[ u_{n+1} = u_n + \Delta t \, f(t_{n+1}, u_{n+1}), \quad u_0 = y_0, \]

where \( u_n \approx y(t_n) \).
Implicit Euler...

\[ u_{n+1} = u_n + \Delta t f(t_{n+1}, u_{n+1}), \quad u_0 = y_0, \]

Note:

- The scheme is referred to as **implicit** as \( u_{n+1} \) is not directly given.
- **Accuracy:** Explicit and implicit Euler have comparable accuracy after a single step.
- **Computational expense:** Try to trade equation solving costs for much larger time steps.
- **Stability:** Implicit Euler has better stability (more on this later).
Other simple methods

Trapezoidal Rule

\[ u_{n+1} = u_n + \Delta t \left( \frac{f(t_n, u_n)}{2} + \frac{f(t_{n+1}, u_{n+1})}{2} \right) \]

Implicit Midpoint Rule

\[ u_{n+1} = u_n + \Delta t f \left( t_n + \frac{1}{2} \Delta t, \frac{u_n + u_{n+1}}{2} \right) \]

Heun’s method

\[ k_1 = f(t_n, u_n), \quad k_2 = f(t_n + \Delta t, u_n + \Delta t k_1) \quad \text{and} \]

\[ u_{n+1} = u_n + \frac{1}{2} \Delta t (k_1 + k_2) \]
The Lotka–Volterra model

\[ r(t) = \text{number of rabbits (prey)} \]
\[ f(t) = \text{number of foxes (predator)} \]

The populations are modeled as

\[
\begin{align*}
    r'(t) &= 2r(t) - 0.01 r(t)f(t) \\
    f'(t) &= -f(t) + 0.01 r(t)f(t)
\end{align*}
\]

with \( r(0) = 300 \) and \( f(0) = 150 \).

Question: How do the populations vary with time?
Local and global errors

Global error: \[ g_n = y(t_n) - u_n \]

Local error: \[ \ell_{n+1} = \hat{y}(t_{n+1}) - u_{n+1} \]

where \( \hat{y}(t) \) is the exact solution with \( \hat{y}(t_n) = u_n \).
Error propagation for explicit Euler

Local error:

Let \( \hat{y}(t) \) be the solution with \( \hat{y}(t_n) = u_n \), then

\[
\hat{y}(t_{n+1}) = \hat{y}(t_n + \Delta t) = \hat{y}(t_n) + \Delta t \hat{y}'(t_n) + \mathcal{O}(\Delta t^2)
\]

\[
= u_n + \Delta t f(t_n, u_n) + \mathcal{O}(\Delta t^2)
\]

Hence, \( \ell_{n+1} = \hat{y}(t_{n+1}) - u_{n+1} = \mathcal{O}(\Delta t^2) \).

Global error: "Best case scenario"

\[
\|g_n\| = C \sum_{j=1}^{n} \|\ell_j\| = n \mathcal{O}(\Delta t^2) = \mathcal{O}(\Delta t)
\]
Convergence order

If the global error satisfies

$$\|g_n\| = O(\Delta t^p)$$

then the method’s order of convergence is $p$.

Note: In the well behaved case we have

$$\|\ell_n\| = O(\Delta t^{p+1}) \Rightarrow \|g_n\| = O(\Delta t^p)$$

Examples:

- Explicit Euler $p = 1$
- Implicit Euler $p = 1$
- Trapezoidal Rule $p = 2$
- Implicit Midpoint Rule $p = 2$
- Heun’s method $p = 2$
Classical Runge–Kutta method

Higher order methods?

**RK4** (1895) Order $p = 4$.

\[
\begin{align*}
  k_1 &= f(t_n, u_n) \\
  k_2 &= f(t_n + \frac{1}{2} \Delta t, u_n + \frac{1}{2} \Delta t k_1) \\
  k_3 &= f(t_n + \frac{1}{2} \Delta t, u_n + \frac{1}{2} \Delta t k_2) \\
  k_4 &= f(t_n + \Delta t, u_n + \Delta t k_3) \\
  u_{n+1} &= u_n + \frac{1}{6} \Delta t \left( k_1 + 2k_2 + 2k_3 + k_4 \right)
\end{align*}
\]
Numerical instability

Example: Apply explicit Euler method to

\[ y'(t) = -100 \, y(t), \quad \text{i.e.,} \quad y(t) = e^{-100t} y_0 \]

Case 1 (Numerical instability): Step size \( \Delta t = 0.1 \) yields

\[ u_{n+1} = (1 - 0.1 \cdot 100) \cdot u_n = -9u_n \quad \Rightarrow \quad u_n = (-9)^n u_0 \]

Exponentially growing oscillations!

Case 2: Step size \( \Delta t = 0.001 \) yields

\[ u_{n+1} = (1 - 0.001 \cdot 100) \cdot u_n = 0.9u_n \quad \Rightarrow \quad u_n = 0.9^n u_0 \]

Smooth, decaying solution.
Stability analysis

“Test equation”: \( y'(t) = \lambda y(t), \) solution \( y(t) = e^{t\lambda}y_0. \)

Mathematical stability condition:

\[
\text{Bounded solution } \iff \text{Re } \lambda \leq 0
\]

Explicit Euler: \( u_{n+1} = (1 + \Delta t \lambda)u_n \)

Numerical stability condition:

\[
\text{Bounded solution } \iff |1 + \Delta t \lambda| \leq 1
\]

Stability region:

\[
S = \{ \Delta t \lambda \in \mathbb{C} : |1 + \Delta t \lambda| \leq 1 \}
\]
Stability analysis...

Apply the implicit Euler method to the test equation:

\[ u_{n+1} = u_n + \Delta t \lambda u_{n+1} \implies u_{n+1} = (1 - \Delta t \lambda)^{-1} u_n \]

**Numerical stability condition:**

Bounded solution \( \iff |1 - \Delta t \lambda| \geq 1 \)

**Stability region:**

\[ S = \{ \Delta t \lambda \in \mathbb{C} : |1 - \Delta t \lambda| \geq 1 \} \]
Stiffness

Solve: \( y' = \lambda y - \lambda \sin t + \cos t \) with \( \lambda = -50 \).

Solutions: Particular: \( y_P(t) = \sin t \)
Homogeneous: \( y_H(t) = e^{\lambda t} \)
General: \( y(t) = e^{\lambda(t-a)}(y_0 - \sin a) + \sin t \)
Stiffness...

“Stiff” IVP are characterized by homogeneous solutions being strongly damped.

**Example:** \( y' = \lambda y - \lambda \sin t + \cos t \) with \( \lambda \ll -1 \).

Stability regions of explicit methods are bounded. \( \Delta t \lambda \in S \) puts strong restriction on \( \Delta t \).

**Example:** Explicit Euler applied to the problem above requires

\[
\Delta t \leq -2/\lambda \ll 1
\]

Implicit methods with unbounded stability regions put no restrictions on \( \Delta t \) (only restricted by accuracy).