18-660: Numerical Methods for Engineering Design and Optimization

Xin Li
Department of ECE
Carnegie Mellon University
Pittsburgh, PA 15213
Overview

- Ordinary Differential Equation (ODE)
  - Numerical integration
  - Stability
Ordinary Differential Equation (ODE)

- Transient analysis for electrical circuit

\[
\dot{x}(t) = u(t) - x(t) \quad \rightarrow \quad \text{Ordinary differential equation}
\]

\[
x(0) = 0 \quad \rightarrow \quad \text{Initial condition}
\]

\[
u(t) = 1 \quad (t \geq 0)
\]
**Ordinary Differential Equation (ODE)**

**General mathematical form**

\[
F[\dot{x}(t), x(t), u(t)] = 0 \quad x(0) = X
\]

- \(x(t)\): N-dimensional vector of unknown variables
- \(u(t)\): Vector of input sources
- \(F\): Nonlinear operator
- \(X\): Initial condition
Numerical Integration

- In general, closed-form solution does not exist
  - Even if ODE is linear, we cannot find analytical solutions in many practical cases

- Numerical methods must be applied to approximate the solution – numerical solution
  - Numerical integration for differential operator

\[ \dot{x}(t) \approx ??? \]

\[ \text{Algebraic equation} \]
Several different formulas exist for numerical integration

One-step numerical integration approximates differential operator from two successive time points

\[
\dot{x} = f(x)
\]

\[
\frac{x(t_{n+1}) - x(t_n)}{\Delta t} \approx f(x)
\]

Forward Euler (FE)

\[
\frac{x(t_{n+1}) - x(t_n)}{\Delta t} \approx f(t_n) = f(t_n)
\]

Backward Euler (BE)

\[
\frac{x(t_{n+1}) - x(t_n)}{\Delta t} \approx f(t_{n+1})
\]

Trapezoidal (TR)

\[
\frac{x(t_{n+1}) - x(t_n)}{\Delta t} \approx \frac{f(t_n) + f(t_{n+1})}{2}
\]
Trapezoidal Approximation

- Trapezoidal is often more accurate but also more expensive than BE and FE

\[
\frac{x(t_{n+1}) - x(t_n)}{\Delta t} \approx f[x(t_n)] + f[x(t_{n+1})] \quad \frac{1}{2}
\]

\[
x(t_{n+1}) - x(t_n) = \int_{t_n}^{t_{n+1}} f[x(t)] \cdot dt
\]

\[
\approx \Delta t \cdot \frac{f[x(t_n)] + f[x(t_{n+1})]}{2}
\]

\[
\dot{x} = f(x)
\]
Backward Euler Approximation

- Similar to TR but is less accurate and expensive
- Widely used for practical applications

\[ \dot{x} = f(x) \]

\[ \frac{x(t_{n+1}) - x(t_n)}{\Delta t} \approx f[x(t_{n+1})] \]

\[ x(t_{n+1}) - x(t_n) = \int_{t_n}^{t_{n+1}} f[x(t)] \cdot dt \]

\[ \approx \Delta t \cdot f[x(t_{n+1})] \]
Backward Euler Example

First-order system example

\[
\dot{x}(t) = u(t) - x(t) \\
x(0) = 0 \\
u(t) = 1 \quad (t \geq 0)
\]

\[
x(t_{n+1}) - x(t_n) \approx f(t_{n+1})
\]

\[
\frac{x(t_{n+1}) - x(t_n)}{\Delta t} = 1 - x(t_{n+1})
\]

\[
x(t_0) = 0
\]
**Backward Euler Example**

- **First-order system example**

\[
\frac{x(t_{1}) - x(t_{0})}{\Delta t} = 1 - x(t_{1})
\]

\[
x(t_{0}) = 0
\]

\[
x(t_{1}) - x(t_{0}) = \Delta t - \Delta t \cdot x(t_{1})
\]

\[
x(t_{1}) = \frac{\Delta t + x(t_{0})}{1 + \Delta t} = \frac{\Delta t}{1 + \Delta t}
\]
Backward Euler Example

First-order system example

\[
\frac{x(t_2) - x(t_1)}{\Delta t} = 1 - x(t_2)
\]

\[x(t_2) - x(t_1) = \Delta t - \Delta t \cdot x(t_2)\]

\[x(t_2) = \frac{\Delta t + x(t_1)}{1 + \Delta t}\]

\[
\frac{x(t_{n+1}) - x(t_n)}{\Delta t} = 1 - x(t_{n+1})
\]

\[x(t_0) = 0\]

Continue iteration to determine \(x(t)\)
Forward Euler Approximation

- Least accurate compared to TR and BE
- Difficult to guarantee stability

\[ \dot{x} = f(x) \]

\[ \frac{x(t_{n+1}) - x(t_n)}{\Delta t} \approx f[x(t_n)] \]

\[ x(t_{n+1}) - x(t_n) = \int_{t_n}^{t_{n+1}} f[x(t)] \cdot dt \]

\[ \approx \Delta t \cdot f[x(t_n)] \]
Forward Euler Example

- First-order system example

\[ \dot{x}(t) = u(t) - x(t) \]
\[ x(0) = 1 \quad u(t) = 0 \]

\[ \frac{x(t_{n+1}) - x(t_n)}{\Delta t} = -x(t_n) \]
\[ x(t_0) = 1 \]
First-order system example

\[
\frac{x(t_1) - x(t_0)}{\Delta t} = -x(t_0)
\]

\[
x(t_0) = 1
\]

\[
x(t_1) = -\Delta t + 1
\]

\[
x(t_1) = 1 - \Delta t
\]
First-order system example

\[
\frac{x(t_2) - x(t_1)}{\Delta t} = -x(t_1)
\]

\[
x(t_1) = 1 - \Delta t
\]

\[
x(t_2) = -\Delta t \cdot x(t_1) + x(t_1)
\]

\[
x(t_2) = (1 - \Delta t) \cdot x(t_1)
\]

\[
= (1 - \Delta t)^2
\]
Forward Euler Example

First-order system example

\[ x(t_3) = (1 - \Delta t)^3 \]

\[ x(t_4) = (1 - \Delta t)^4 \]

\[ x(t_n) = (1 - \Delta t)^n \]
Forward Euler Example

- First-order system example
  \[ x(t_n) = (1 - \Delta t)^n \]

\[ \Delta t = 0.1 \text{ (correct answer)} \]

\[ \Delta t = 3 \text{ (fail to converge)} \]

Forward Euler fails to converge if \( \Delta t \) is too large.
Numerical Integration Stability

- $\Delta t$ must be sufficiently small for FE to guarantee stability
  - In practice, it is not easy to determine the appropriate $\Delta t$

- BE and TR do not suffer from stability issue
  - Stability is guaranteed for any $\Delta t > 0$

- First-order system example

\[
\dot{x}(t) = u(t) - x(t) \\
x(0) = 1 \quad u(t) = 0
\]

BE

\[
x(t_n) = \frac{1}{(1 + \Delta t)^n}
\]

Always stable for $\Delta t > 0$
Nth-Order Linear ODE

- Our simple example solves a first-order linear ODE
  \[ \dot{x}(t) = u(t) - x(t) \]

- In general, an Nth-order linear time-invariant dynamic system is described by the following ODE:
  \[ \dot{x}(t) = A \cdot x(t) + B \cdot u(t) \]

  \[ x(0) = 0 \]

  → Ordinary differential equation

  → Initial condition

- \( x(t) \): N-dimensional vector of unknown variables
- \( u(t) \): Vector of input sources
- \( A, B \): Matrices
Nth-Order Linear ODE

- **Backward Euler example**

\[ \dot{x}(t) = A \cdot x(t) + B \cdot u(t) \quad x(0) = 0 \]

\[ \frac{x(t_{n+1}) - x(t_n)}{\Delta t} = A \cdot x(t_{n+1}) + B \cdot u(t_{n+1}) \quad x(t_0) = 0 \]

\[ x(t_{n+1}) - x(t_n) = \Delta t \cdot A \cdot x(t_{n+1}) + \Delta t \cdot B \cdot u(t_{n+1}) \quad x(t_0) = 0 \]

\[ x(t_{n+1}) = (I - \Delta t \cdot A)^{-1} \cdot \left[ x(t_n) + \Delta t \cdot B \cdot u(t_{n+1}) \right] \quad x(t_0) = 0 \]

Solve linear algebraic equation to find \( x(t_{n+1}) \)
Many physical systems are both high-order and nonlinear

\[ F[\dot{x}(t), x(t), u(t)] = 0 \quad x(0) = 0 \]

\( x(t) \): N-dimensional vector of unknown variables

\( u(t) \): Vector of input sources

\( F \): Nonlinear operator
Nth-Order Nonlinear ODE

- Backward Euler example

\[ F[\dot{x}(t), x(t), u(t)] = 0 \quad x(0) = 0 \]

\[ F\left[ \frac{x(t_{n+1}) - x(t_n)}{\Delta t}, x(t_{n+1}), u(t_{n+1}) \right] = 0 \quad x(t_0) = 0 \]

Solve nonlinear algebraic equation to find \( x(t_{n+1}) \)

- Solving nonlinear algebraic equation requires iterative algorithm
  - More details in future lectures...
Advanced Topics for ODE Solver

- Local truncation error estimation
  - Estimate approximation error for numerical integration

- Adaptive time step control
  - Dynamically determine $\Delta t$

- High-order integration formula
  - Apply multi-step numerical integration

- Some of these advanced topics are covered by 18-762 that particularly focuses on ODE solver for circuit simulation
Ordinary differential equation (ODE)

- Numerical integration
- Stability