ODE Solutions

J. B. Snively (EP501)

Classifications…

- **Initial Value Problems (IVP)**
  - Open Domain, evolves from initial y_o at t_o.
  - Progress Forward in Time (“time-like” coordinate).
- **Boundary Value Problems (BVP)**
  - Closed Domain, constrained by values y_1, y_2 at two points x_1, x_2.
  - May satisfy an equilibrium solution in space.

Methods

- **First Order**
  - Forward Euler (Explicit)
  - Backward Euler (Implicit)
- **Second (and Higher) Order**
  - Modified Midpoint
  - Modified Euler
  - Runge-Kutta

Recall: Forward Difference

Define: $\Delta x = x - x_0 \longrightarrow x = x_0 + \Delta x$

Use those expressions to solve for derivative term…

$$f'(x_0) = \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x} + \mathcal{O}(\Delta x)$$

Result is first-order accurate.

Forward Euler Method

Solve: $y'_n = \frac{y_{n+1} - y_n}{\Delta t} = f(t_n, y_n)$

Construct an explicit method:

$$y_{n+1} = y_n + \Delta t f_n$$

Result is first-order accurate, conditionally stable, and easy to implement.

Recall: Backward Difference

Define: $\Delta x = x - x_0 \longrightarrow x = x_0 + \Delta x$

Use those expressions to solve for derivative term…

$$f'(x_0) = \frac{f(x_0) - f(x_0 - \Delta x)}{\Delta x} + \mathcal{O}(\Delta x)$$

Result is first-order accurate.

Backward Euler Method

Solve: $y'_{n+1} = \frac{y_{n+1} - y_n}{\Delta t} = f(t_{n+1}, y_{n+1})$

Construct an implicit method:

$$y_{n+1} = y_n + \Delta t f_{n+1}$$

Result is first-order accurate, unconditionally stable, and requires additional numerical solution.

Recall: Centered Difference

Define: $\Delta x = x - x_0 \longrightarrow x = x_0 + \Delta x$

Subtract expressions to solve for derivative term…

$$\boxed{f'(x_0) = \frac{f(x_0 + \Delta x) - f(x_0 - \Delta x)}{2\Delta x} + \mathcal{O}(\Delta x^2)}$$

Result is second-order accurate, and “centered”.

Midpoint Method

Solve: $y'_{n+1/2} = \frac{y_{n+1} - y_n}{\Delta t} = f(t_{n+1/2}, y_{n+1/2})$

Construct an implicit method:

$$\boxed{y_{n+1/2} = y_n + \Delta t f_{n+1/2}}$$

Practically, apply predictor-corrector approach to obtain an explicit method, where $f_{n+1/2}$ is predicted.

$$\begin{aligned} y^p_{n+1/2} &= y_n + \frac{\Delta t}{2} f_n \\ y^c_{n+1} &= y_n + \Delta t f^p_{n+1/2} \end{aligned}$$

Result is second-order accurate.

Midpoint Method

Alternatively, apply predictor-corrector approach to obtain an explicit method, where $f_{n+1/2}$ is approximated via a Taylor series expansion:

$$
\begin{aligned}
y^p_{n+1} &= y_n + \Delta t f_n \\
y^c_{n+1} &= y_n + \frac{\Delta t}{2}(f_n + f^p_{n+1})
\end{aligned}
$$

Also called "modified Euler method", or "modified trapezoid method".

Again, result is second-order accurate.

Runge-Kutta Methods

Based on weighted sums of increments to solution Δy_i …

$$\Delta y = C1\Delta y_1 + C2\Delta y_2 + C3\Delta y_3 + C4\Delta y_4 + ...$$

$$y_{n+1} = y_n + C1\Delta y_1 + C2\Delta y_2 + C3\Delta y_3 + C4\Delta y_4 + ...$$

For second order, two coefficients and two Δy_i are needed. For fourth order, four of each are needed.

Perhaps the most familiar and popular RK method is the 4th order formulation…

Runge-Kutta (RK4) Fourth-Order Method

$$y_{n+1} = y_n + \frac{1}{6}(\Delta y_1 + 2\Delta y_2 + 2\Delta y_3 + \Delta y_4)$$

$$\begin{aligned}
\Delta y_1 &= \Delta t f(t_n, y_n) \\
\Delta y_2 &= \Delta t f(t_n + \frac{\Delta t}{2}, y_n + \frac{\Delta y_1}{2}) \\
\Delta y_3 &= \Delta t f(t_n + \frac{\Delta t}{2}, y_n + \frac{\Delta y_2}{2}) \\
\Delta y_4 &= \Delta t f(t_n + \Delta t, y_n + \Delta y_3)
\end{aligned}$$

Classifications…

- **Initial Value Problems (IVP)**
  - Open Domain, evolves from initial y_o at t_o.
  - Progress Forward in Time (“time-like” coordinate).
- **Boundary Value Problems (BVP)**
  - Closed Domain, constrained by values y_1, y_2 at two points x_1, x_2.
  - May satisfy an equilibrium solution in space.

Two Approaches

- **"Shooting" methods**
  - Evolve from initial boundary x_1 with value y_1.
  - Iterate parameters (e.g., dy/dx @ 1) until second value y_2 at next boundary x_2 is satisfied.
- **"Equilibrium" methods**
  - Construct finite difference solution to problem subject to boundary conditions.
  - Solve simultaneously as a system.