MATLAB Applications to
ODE (Ordinary Differential Equation) Solver

Prof. Chang-Joo Kim

Problem Statement of ODEs

First order System of ODEs (Ordinary Differential Equations)

Problem Statement

Solve $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t)$ over $t \in [t_0, t_f]$ With the initial conditions $\mathbf{x}(t_0) = \mathbf{x}_0$

How to transform the higher-order ODE into 1st order System of the ODEs

Example #1 $\ddot{x} + 2\dot{x} + 16x = 0.1\cos t, \quad t \in [0, 2]$ with initial conditions of $x(0) = 0, \dot{x}(0) = 1$

$$
\begin{aligned}
x_1 &= x \\
x_2 &= \dot{x}
\end{aligned}
\quad \rightarrow \quad
\begin{aligned}
\dot{x}_1 &= \dot{x} = x_2 \\
\dot{x}_2 &= \ddot{x} = -2\dot{x} - 16x + 0.1\cos t \\
&= -2x_2 - 16x_1 + 0.1\cos t
\end{aligned}
\quad \rightarrow \quad
\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} x_2 \\ -2x_2 - 16x_1 + 0.1\cos t \end{pmatrix}
$$

$$
\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \qquad \mathbf{f}(\mathbf{x}, t) = \begin{pmatrix} x_2 \\ -2x_2 - 16x_1 + 0.1\cos t \end{pmatrix} \qquad \mathbf{x}_0 = \begin{pmatrix} x(0)_0 \\ \dot{x}(0) \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}
$$

Problem Statement of ODEs

How to transform the higher-order ODE into 1st order System of the ODEs

Example #2

$$m(\ddot{r} - r\dot{\theta}^2) = -\frac{Gm_E m}{r^2} \quad \text{over} \quad t \in [t_0, t_f]$$

$$m(2\dot{r}\dot{\theta} + r\ddot{\theta}) = 0$$

With the initial conditions

$$r(t_0) = r_0 \qquad \dot{r}(t_0) = 0$$

$$\theta(t_0) = 0 \qquad \dot{\theta}(t_0) = \dot{\theta}_0$$

$$\begin{aligned} x_1 &= r \\ x_2 &= \theta \\ x_3 &= \dot{r} \\ x_4 &= \dot{\theta} \end{aligned} \quad \rightarrow \quad \begin{aligned} \dot{x}_1 &= \dot{r} = x_3 \\ \dot{x}_2 &= \dot{\theta} = x_4 \\ \dot{x}_3 &= \ddot{r} = x_1 x_4^2 - Gm_E / x_1^2 \\ \dot{x}_4 &= \ddot{\theta} = -2x_3 x_4 / x_1 \end{aligned} \quad \rightarrow \quad \begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{pmatrix} = \begin{pmatrix} x_3 \\ x_4 \\ x_1 x_4^2 - Gm_E / x_1^2 \\ -2x_3 x_4 / x_1 \end{pmatrix}$$

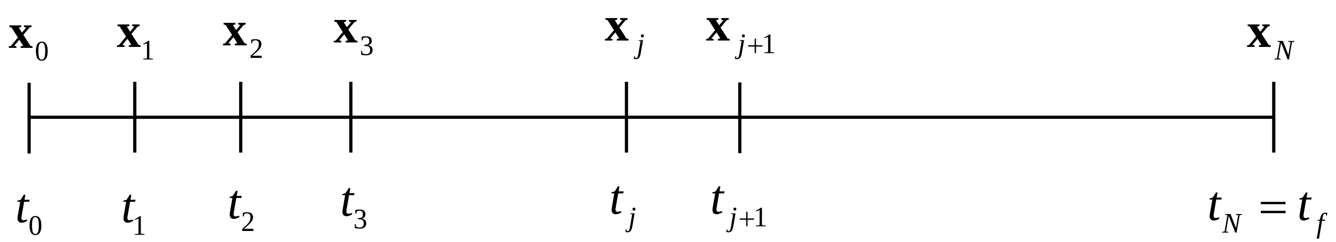
$$\boxed{\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} \qquad \mathbf{f}(\mathbf{x}, t) = \begin{pmatrix} x_3 \\ x_4 \\ x_1 x_4^2 - Gm_E / x_1^2 \\ -2x_3 x_4 / x_1 \end{pmatrix} \qquad \mathbf{x}_0 = \begin{pmatrix} r(t_0) \\ \theta(t_0) \\ \dot{r}(t_0) \\ \dot{\theta}(t_0) \end{pmatrix} = \begin{pmatrix} r_0 \\ 0 \\ 0 \\ \dot{\theta}_0 \end{pmatrix}}$$

Time Integrator (1): Euler Method

Problem Statement

Solve $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t)$ over $t \in [t_0, t_f]$ With the initial conditions $\mathbf{x}(t_0) = \mathbf{x}_0$

Discrete Solution Approach using Time Nodes with Equal Spacing $\{t_j\}_{j=0}^{j=N}$



Time-Node Generation

$$h = \Delta t = t_{j+1} - t_j = \frac{t_f - t_0}{N}$$

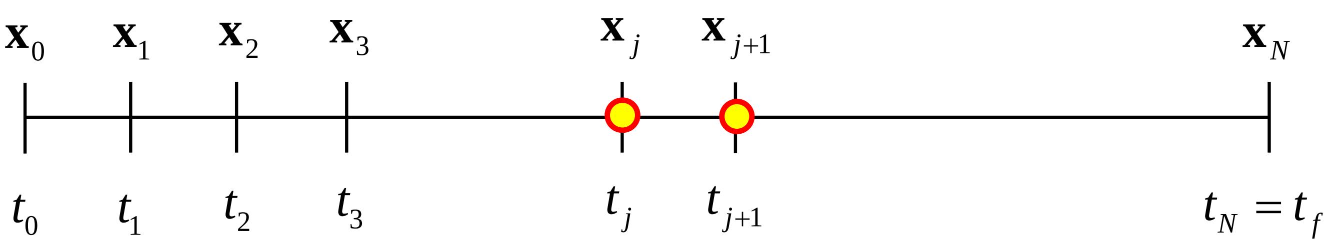
$$t_j = t_0 + j\Delta t \quad (j = 0, 1, 2, \cdots, N)$$

Time Integrator (1): Euler Method

Problem Statements

Solve $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t)$ over $t \in [t_0, t_f]$ With the initial conditions $\mathbf{x}(t_0) = \mathbf{x}_0$

Discrete Solution Approach using Time Nodes with Equal Spacing $\{t_j\}_{j=0}^{j=N}$



$$\boxed{\begin{aligned} h = \Delta t &= t_{j+1} - t_j \\ &= \frac{t_f - t_0}{N} \end{aligned}}$$

When the solution $\mathbf{x}(t_j) = \mathbf{x}_j$ at $t = t_j$ is known

$$\begin{aligned} \mathbf{x}(t_j + h) &= \mathbf{x}(t_j) + \dot{\mathbf{x}}(t_j)\Delta t + \frac{1}{2!}\ddot{\mathbf{x}}(t_j)\Delta t^2 + \frac{1}{3!}\dddot{\mathbf{x}}(t_j)\Delta t^3 + \cdots \\ &= \mathbf{x}(t_j) + \sum_{k=1}^{k=\infty} \frac{1}{k!}\mathbf{x}^{(k)}(t_j)\Delta t^k \leftarrow \mathbf{x}^{(k)} = \frac{d^k \mathbf{x}}{dt^k} \\ &\approx \mathbf{x}(t_j) + \dot{\mathbf{x}}(t_j)\Delta t = \mathbf{x}(t_j) + \mathbf{f}(\mathbf{x}_j, t_j)\Delta t \end{aligned}$$

$$\boxed{\mathbf{x}_{j+1} = \mathbf{x}(t_j + h) \approx \mathbf{x}_j + \mathbf{f}(\mathbf{x}_j, t_j)\Delta t, \quad j = 0, 1, 2, \cdots, N}$$

Time Integrator (2): Heun's Predictor-Corrector Method

Problem Statement

Solve $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t)$ over $t \in [t_0, t_f]$ With the initial conditions $\mathbf{x}(t_0) = \mathbf{x}_0$

Another Form of ODEs: Integral Equation Form

$$\mathbf{x}(t) = \mathbf{x}_0 + \int_{t_0}^{t} \mathbf{f}(\mathbf{x}, t) dt \qquad \boxed{\mathbf{x}(t_{j+1}) = \mathbf{x}(t_j) + \int_{t_j}^{t_{j+1}} \mathbf{f}(\mathbf{x}, t) dt}$$

How to Compute the Integral over $t \in [t_j, t_{j+1}]$: using Trapezoidal Rule (사다리꼴 공식)

$$\boxed{\int_{t_j}^{t_{j+1}} \mathbf{f}(\mathbf{x}, t) dt = \frac{1}{2}\left\{\mathbf{f}(\mathbf{x}_j, t_j) + \mathbf{f}(\mathbf{x}_{j+1}, t_{j+1})\right\}\Delta t}$$

We don't know

Predict $\mathbf{x}_{j+1}$ using the Euler Method

$$\mathbf{x}_{j+1} = \mathbf{x}(t_j + h) \approx \mathbf{x}_j + \mathbf{f}(\mathbf{x}_j, t_j)\Delta t$$

Heun's Predictor-Corrector Update Algorithm

Predictor Step:

$$\boxed{\mathbf{x}_{j+1}^{p} \approx \mathbf{x}_j + \mathbf{f}(\mathbf{x}_j, t_j)\Delta t}$$

Corrector Step:

$$\boxed{\mathbf{x}(t_{j+1}) = \mathbf{x}(t_j) + \frac{1}{2}\left\{\mathbf{f}(\mathbf{x}_j, t_j) + \mathbf{f}(\mathbf{x}_{j+1}^{p}, t_{j+1})\right\}\Delta t}$$

Time Integrator (3): Runge-Kutta Method

Problem Statement

Solve $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t)$ over $t \in [t_0, t_f]$ With the initial conditions $\mathbf{x}(t_0) = \mathbf{x}_0$

4-stage Runge-Kutta Integrator:

(You will learn the detailed derivation Process in Numerical Analysis Lecture)

$$\mathbf{x}_{j+1} = \mathbf{x}_j + \frac{1}{6} h \left(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4 \right)$$

$$\mathbf{k}_1 = \mathbf{f}(t_j, \mathbf{x}_j)$$

$$\mathbf{k}_2 = \mathbf{f}(t_j + \frac{1}{2} h, \mathbf{x}_j + \frac{1}{2} \mathbf{k}_1 h)$$

$$\mathbf{k}_3 = \mathbf{f}(t_j + \frac{1}{2} h, \mathbf{x}_j + \frac{1}{2} \mathbf{k}_2 h)$$

$$\mathbf{k}_4 = \mathbf{f}(t_j + h, \mathbf{x}_j + \mathbf{k}_3 h)$$

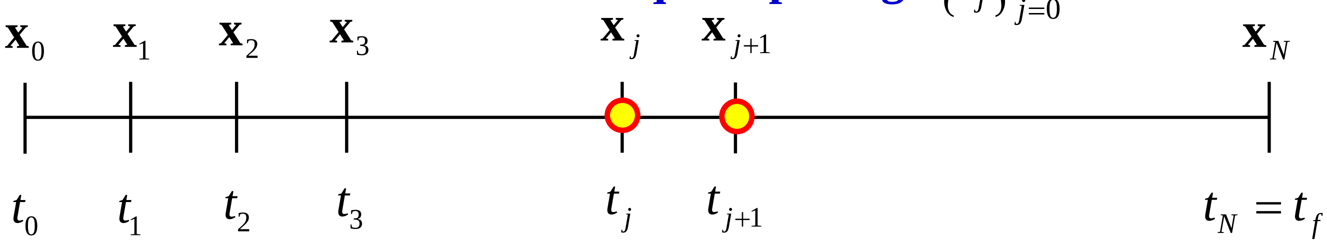
$$h = \Delta t_j = t_{j+1} - t_j$$

Time Integrator (4): Summary

Problem Statement

Solve $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t)$ over $t \in [t_0, t_f]$ With the initial conditions $\mathbf{x}(t_0) = \mathbf{x}_0$

Generation of Time-Nodes with Equal Spacing $\{t_j\}_{j=0}^{j=N}$



$$h = \Delta t = t_{j+1} - t_j = \frac{t_f - t_0}{N}$$

Euler Method

$$\mathbf{x}_{j+1} = \mathbf{x}(t_j + h) \approx \mathbf{x}_j + \mathbf{f}(\mathbf{x}_j, t_j)\Delta t, \quad j = 0, 1, 2, \cdots, N$$

Heun's Predictor-Corrector Method

$$\mathbf{x}(t_{j+1}) = \mathbf{x}(t_j) + \frac{1}{2}\left\{\mathbf{f}(\mathbf{x}_j, t_j) + \mathbf{f}(\mathbf{x}_{j+1}^p, t_{j+1})\right\}\Delta t$$

$$\mathbf{x}_{j+1}^p \approx \mathbf{x}_j + \mathbf{f}(\mathbf{x}_j, t_j)\Delta t$$

4-stage Runge-Kutta Integrator

$$\mathbf{x}_{j+1} = \mathbf{x}_j + \frac{1}{6}h\left(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4\right)$$

$$\mathbf{k}_1 = \mathbf{f}(t_j, \mathbf{x}_j)$$

$$\mathbf{k}_2 = \mathbf{f}(t_j + \frac{1}{2}h, \mathbf{x}_j + \frac{1}{2}\mathbf{k}_1 h)$$

$$\mathbf{k}_3 = \mathbf{f}(t_j + \frac{1}{2}h, \mathbf{x}_j + \frac{1}{2}\mathbf{k}_2 h)$$

$$\mathbf{k}_4 = \mathbf{f}(t_j + h, \mathbf{x}_j + \mathbf{k}_3 h)$$

How to Program in MATLAB : Example for Euler Method

Input $N, \quad t_0, \quad t_f, \quad \mathbf{x}_0$

Program Structure for Euler Method

(1) Node Generation

$$\boxed{\begin{aligned} h &= \frac{t_f - t_0}{N} \\ \Delta t &= h \end{aligned}} \qquad \boxed{t_j = t_0 + jh}$$

(2) for-loop

$\mathbf{x}=\mathbf{x}_0$

for j=1: N

$$\boxed{t_j = t_0 + jh}$$

$$\boxed{\begin{aligned} \mathbf{x} &= \mathbf{x} + \mathbf{f}(\mathbf{x}, t_j)h \\ \mathbf{x}_j &= \mathbf{x} \end{aligned}}$$

We need a function to compute $\boxed{\mathbf{f}(\mathbf{x}, t_j)}$

end

End of Lecture