筑波大学
University of Tsukuba

IMAGINE THE FUTURE.

力学的システムの動的シミュレーション

Dynamic Simulation for Mechanical System

車両-橋梁相互作用 01

Vehicle-Bridge Interaction

システム情報系 助教

山本亨輔 Kyosuke YAMAMOTO

yamamoto_k@kz.tsukuba.ac.jp

作成：2018年6月5日

Today's Goal

Numerical Solutions of Equation of Motion

（運動方程式の数値解）

- Equation of Motion: $\{\boldsymbol{F}\} = [\mathbf{M}]\{\ddot{\boldsymbol{x}}\}$

  - $\{\boldsymbol{F}\}$ External Force Vector ($\boldsymbol{F} \in R^n$): **input**, usually given
  - $[\mathbf{M}]$ Mass Matrix ($\mathbf{M} \in R^{n \times n}$): **system parameter**
  - $\{\boldsymbol{x}\}$ Displacement Vector ($\boldsymbol{x} \in R^n$): **output**, or solution

- Considering a mechanical "**system**", force and displacement fields are input and output of the system, respectively
- A matrix acting on the output is system parameters: e.g.) Mass Matrix $\mathbf{M}$ and Stiffness Matrix $\mathbf{K}$ ($\boldsymbol{f} = \mathbf{K}\boldsymbol{x}$)
- n is DOF (degree of freedom, 自由度): = "**point**" × "**direction**"
- The displacement field $\{\boldsymbol{x}\}$ is the "**solution**" of the equation of motion as "**differential equation**"
- So-called "**Simulation**" is a method to calculate solutions numerically (**numerical solution**)

Skill you can learn

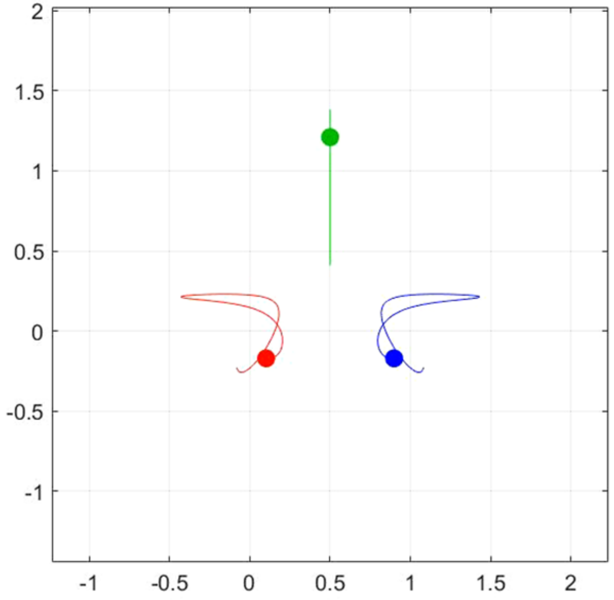
Model, Solve and Visualize It!

（モデル化し、解いて、視覚化する）

- To simulate the motions of the mass-points

(Basic Elements of System Parameters)

- m_i : mass
- c_i : damping
- k_i : spring stiffness



SDOF: Single Degree of Freedom

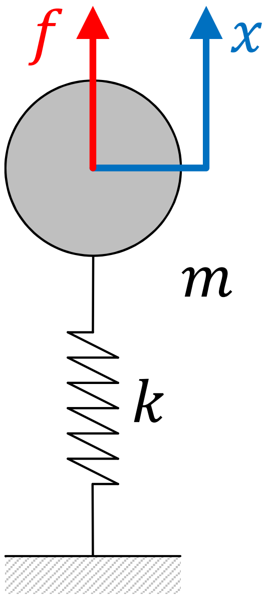
- Equation of Motion: $F = m\ddot{x}$

$$\underbrace{\boxed{f - kx}}_{\text{sum of external force}} = \underbrace{m\ddot{x}}_{\text{inertia term}}$$

- **Force** is an agent that changes the **shape** and **motion** of an object
  - **motion** is **velocity**, and the change of motion is **acceleration**

$$m\ddot{x} + kx = f$$

- **Load**: f becomes
  - **acceleration** = the change of motion: $m\ddot{x}$
  - **displacement** = the change of shape: kx



Special Conditions of Equation of Motion: $f - kx = m\ddot{x}$

- **Equilibrium: $\ddot{x} = 0$**
  - The sum of external force equals to zero: $F = 0$
  - The acceleration (the change of motion) becomes zero from $F = m\ddot{x}$
  - **Inertia**: "object keeps its state of motion" = "no force, same velocity"

$$f = kx$$

- **Free Vibration: $f = 0$**
  - If there is no Load: f, the system is in the state of free vibration
  - The equation of motion becomes LHDE: Linear homogeneous Differential Equation

$$m\ddot{x} + kx = 0$$

Analysis

- **Analysis** is "Mathematics to calculate exact solution"
- Free Vibration: $m\ddot{x} + kx = 0$

Assuming that $x(t) = A\sin(\omega t)$

$$\dot{x}(t) = A\omega\cos(\omega t)$$

$$\ddot{x}(t) = -A\omega^2\sin(\omega t) = -\omega^2 x(t)$$

Substitution of $\ddot{x}(t)$ and $x(t)$ into the given equation:

$$\overset{-\omega^2 x}{\downarrow}$$

$$m\ddot{x} + kx = 0$$

$$\downarrow$$

$$m(-\omega^2 x) + kx = 0 \Rightarrow (-m\omega^2 + k)x(t) = 0$$

$$\begin{cases} -m\omega^2 + k = 0 \Rightarrow \text{Angular Frequency } \boxed{\omega = \sqrt{\dfrac{k}{m}}} \Rightarrow \boxed{x(t) = A\sin\left(\sqrt{\dfrac{k}{m}}t\right)} \\ x(t) = 0 \quad \text{Not Requested} \end{cases}$$

Solution

Taylor Expansion for approximate formula

（テイラー展開による近似式の導入）

Discretization of time Differential

$$\boxed{\dot{x}(t) = \frac{x(t+\Delta t) - x(t-\Delta t)}{2 \cdot \Delta t}}$$ Central Differential

Taylor Expansion

$$x(t+\Delta t) = x(t) + \dot{x}(t)\Delta t + \frac{1}{2}\ddot{x}(t)\Delta t^2 + \frac{1}{6}\dddot{x}(t)\Delta t^3 + \cdots \Rightarrow \dot{x}(t) = \frac{x(t+\Delta t) - x(t)}{\Delta t}$$ Forward Scheme

$$-)\quad x(t-\Delta t) = x(t) - \dot{x}(t)\Delta t + \frac{1}{2}\ddot{x}(t)\Delta t^2 - \frac{1}{6}\dddot{x}(t)\Delta t^3 + \cdots \Rightarrow \dot{x}(t) = \frac{x(t) - x(t-\Delta t)}{\Delta t}$$ Backstep Scheme

$$\boxed{x(t+\Delta t) - x(t-\Delta t) = 2\dot{x}(t)\Delta t} \quad + \frac{2}{6}\dddot{x}(t)\Delta t^3 + \cdots$$

Example 01:

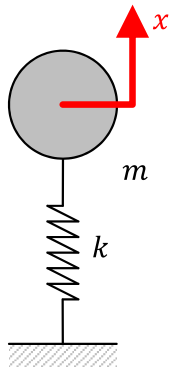
Numerical Simulation of Free Vibration for SDOF by Forward Differential

- Governing Equation: $\boxed{m\ddot{x}(t) = -kx(t)}$
- Discretization:

$$\ddot{x}(t) = \frac{1}{\Delta t}\big(\dot{x}(t+\Delta t) - \dot{x}(t)\big)$$
$$= \frac{1}{\Delta t^2}\big(x(t+2\Delta t) - 2\,x(t+\Delta t) + x(t)\big)$$

- Substitution:

$$\boxed{m\frac{1}{\Delta t^2}\big(x(t+2\Delta t) - 2\,x(t+\Delta t) + x(t)\big) = -kx(t)}$$



- Discretized Equation:

$$x(t+2\Delta t) = 2 \times \underbrace{x(t+\Delta t)}_{\text{known}} - \left(\frac{k}{m}\Delta t^2 + 1\right) \underbrace{x(t)}_{\text{known}}$$

- Matlab Coding

```
x(tt,:) = 2*x(tt-1,:)-(k/m*dt^2+1)*x(tt-2,:);
```

Example 01:
Numerical Simulation of Free Vibration for SDOF by Forward Differential

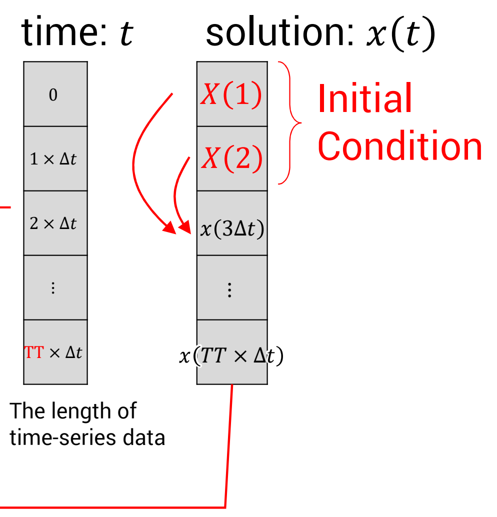
Function File:

```
function [ t, x ] = ex_01_sdof( m, k, X, dt ,T )
%-- preparation
t = (0:dt:T)'; %-- time
TT = length(t); %-- length of solution

%-- initialization
x = zeros(TT,size(X,2));
x([1 2],:) = X([1 2],:);

%-- upwind scheme
for tt = 3:TT
    x(tt,:) = 2*x(tt-1,:)-(k/m*dt^2+1)*x(tt-2,:);
end

end
```



Example 01:
Numerical Simulation of Free Vibration for SDOF by Forward Differential

Launcher Script:

m	k	Δt	$v(0)$
1.0	1.0	0.0001	1

m	k	Δt	$v(0)$
1.0	1.0	0.01	1

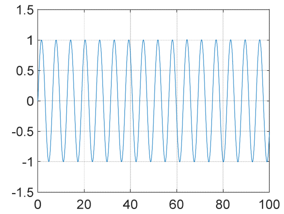
```
[ t, x ] = ex_01_sdof( m, k,           X,      dt ,  T )
[ t, x ] = ex_01_sdof( 1, 1, [0;0.0001], 0.0001 ,100 );
figure(1); plot(t,x)
set(gca,'XLim', [0 100], 'XGrid', 'on')
set(gca,'YLim', [-1.5 1.5], 'YGrid', 'on')
set(gca,'FontSize',18)
print('ex_01_sdof_01','-dmeta','-r1200')
```

print out

```
[ t, x ] = ex_01_sdof( 1, 1, [0;0.01], 0.01 ,100 );
figure(2); plot(t,x)
set(gca,'XLim', [0 100], 'XGrid', 'on')
set(gca,'YLim', [-1.5 1.5], 'YGrid', 'on')
set(gca,'FontSize',18)
print('ex_01_sdof_02','-dmeta','-r1200')
```

$$v(0) = \frac{X(1) - X(0)}{\Delta t}$$

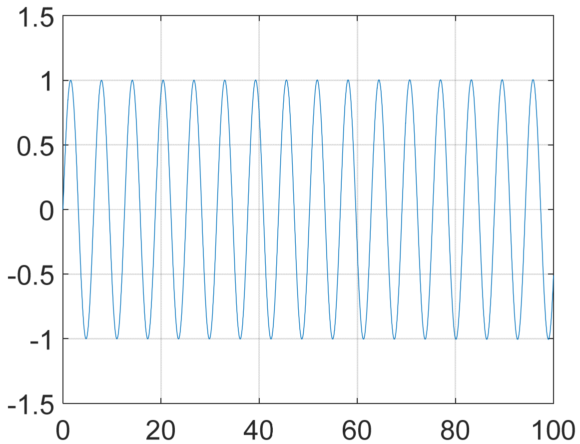
visualization



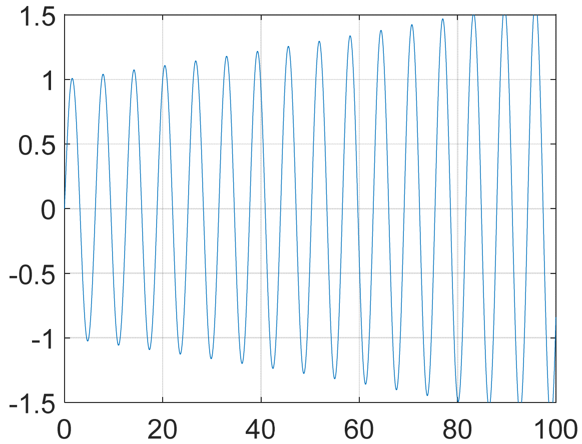
Example 01:
Numerical Simulation of Free Vibration for SDOF by Forward Differential

Results and Discussion

m	k	Δt	$v(0)$
1.0	1.0	0.0001	1



m	k	Δt	$v(0)$
1.0	1.0	0.01	1



- Larger Δt is Low Computation Cost and Low Accuracy
- For not enough small Δt, the explosion of solution occurs

Newmark-β scheme

（ニューマークβ法による高精度な計算）

☐ Discretization

1) The unknowns $\begin{Bmatrix} x(t) \\ \dot{x}(t) \end{Bmatrix}$ are replaced with $\begin{Bmatrix} x(t-\Delta t) \\ \dot{x}(t-\Delta t) \\ \ddot{x}(t-\Delta t) \\ \ddot{x}(t) \end{Bmatrix}$ in the governing equation

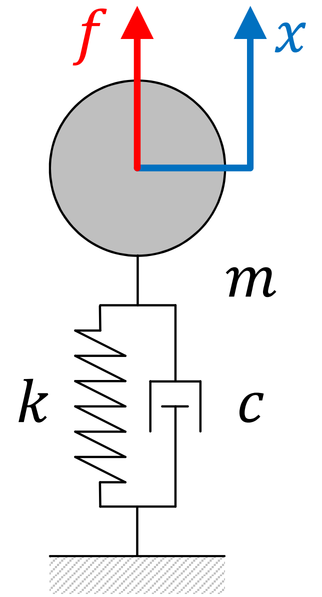
2) All factors consisting the obtained equation are known, except $\ddot{x}(t)$

$$\begin{cases} \dot{x}(t) = \dot{x}(t-\Delta t) + \Delta t\big((1-\gamma)\ddot{x}(t-\Delta t) + \gamma\ddot{x}(t)\big) \\ x(t) = x(t-\Delta t) + \Delta t \cdot \dot{x}(t-\Delta t) + \Delta t^2\left(\left(\dfrac{1}{2} - \beta\right)\ddot{x}(t-\Delta t) + \beta\ddot{x}(t)\right) \end{cases}$$

SDOF: Single Degree of Freedom

- Equation of Motion: $F = m\ddot{x}$

$$\underbrace{\boxed{f - kx - c\dot{x}}}_{\text{sum of external force}} = \underbrace{m\ddot{x}}_{\text{inertia term}}$$



- The Governing Equation:

$$m\ddot{x} + c\dot{x} + kx = f$$

  - MCK-system

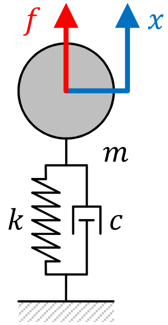
Example 02:

Numerical Simulation of SDOF by Newmark-β method

- **Governing Equation:** $m\ddot{x}(t) + c\dot{x}(t) + kx(t) = f(t)$
- Discretization:

$$\dot{x}(t) = \dot{x}(t-\Delta t) + \Delta t\big((1-\gamma)\ddot{x}(t-\Delta t) + \gamma\ddot{x}(t)\big)$$

$$x(t) = x(t-\Delta t) + \Delta t \cdot \dot{x}(t-\Delta t) + \Delta t^2\left(\left(\frac{1}{2}-\beta\right)\ddot{x}(t-\Delta t) + \beta\ddot{x}(t)\right)$$



- Substitution:

$$(m + \Delta t\gamma c + \Delta t^2\beta k)\ddot{x}(t) = f(t) - \left\{\Delta t(1-\gamma)c + \Delta t^2\left(\frac{1}{2}-\beta\right)k\right\}\ddot{x}(t-\Delta t) - \{c + \Delta t\, k\}\dot{x}(t-\Delta t) - kx(t-\Delta t)$$

- **Discretized Equation:**

$$\ddot{x}(t) = \boxed{\frac{f(t) - \left\{\Delta t(1-\gamma)c + \Delta t^2\left(\frac{1}{2}-\beta\right)k\right\}\ddot{x}(t-\Delta t) - \{c + \Delta t\, k\}\dot{x}(t-\Delta t) - kx(t-\Delta t)}{(m + \Delta t\gamma c + \Delta t^2\beta k)}}$$

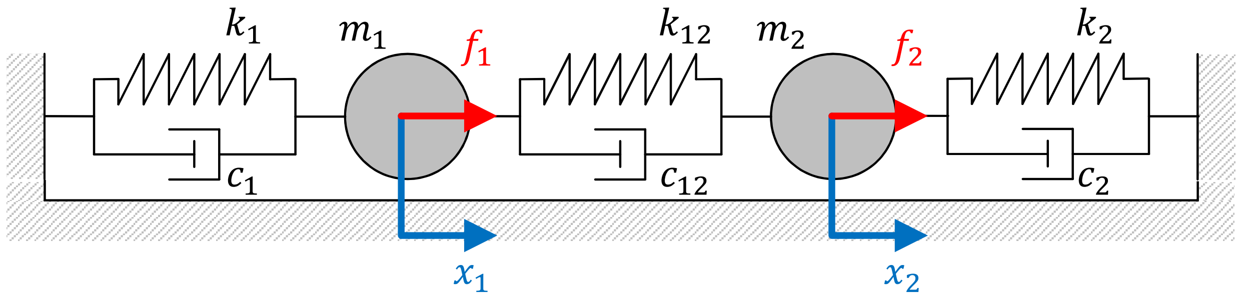
known

MDOF: Multi-Degree of Freedom

- **Equation of Motion:** $\boldsymbol{F} = \mathbf{M}\ddot{\boldsymbol{x}}$

$$m_1\ddot{x}_1 = f_1 - k_1x_1 - k_{12}(x_1 - x_2) - c_1\dot{x}_1 - c_{12}(\dot{x}_1 - \dot{x}_2)$$

$$m_2\ddot{x}_2 = f_2 - k_2x_2 - k_{12}(x_2 - x_1) - c_2\dot{x}_2 - c_{12}(\dot{x}_2 - \dot{x}_1)$$



$$\begin{bmatrix} m_1 & \\ & m_2 \end{bmatrix}\begin{Bmatrix} \ddot{x}_1(t) \\ \ddot{x}_2(t) \end{Bmatrix} + \begin{bmatrix} c_1 + c_{12} & -c_{12} \\ -c_{12} & c_2 + c_{12} \end{bmatrix}\begin{Bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{Bmatrix} + \begin{bmatrix} k_1 + k_{12} & -k_{12} \\ -k_{12} & k_2 + k_{12} \end{bmatrix}\begin{Bmatrix} x_1(t) \\ x_2(t) \end{Bmatrix} = \begin{Bmatrix} f_1(t) \\ f_2(t) \end{Bmatrix}$$

$$\boxed{[\mathbf{M}]\{\ddot{\boldsymbol{x}}\} + [\mathbf{C}]\{\dot{\boldsymbol{x}}\} + [\mathbf{K}]\{\boldsymbol{x}\} = \{\boldsymbol{f}\}}$$

MDOF: Multi-Degree of Freedom

- **Governing Equation:** $\boxed{\mathbf{M}\ddot{\boldsymbol{x}}(t) + \mathbf{C}\dot{\boldsymbol{x}}(t) + \mathbf{K}\boldsymbol{x}(t) = \boldsymbol{f}(t)}$
- Discretization:

$$\dot{\boldsymbol{x}}(t) = \dot{\boldsymbol{x}}(t-\Delta t) + \Delta t\big((1-\gamma)\ddot{\boldsymbol{x}}(t-\Delta t) + \gamma\ddot{\boldsymbol{x}}(t)\big)$$

$$\boldsymbol{x}(t) = \boldsymbol{x}(t-\Delta t) + \Delta t \cdot \dot{\boldsymbol{x}}(t-\Delta t) + \Delta t^2\left(\left(\frac{1}{2}-\beta\right)\ddot{\boldsymbol{x}}(t-\Delta t) + \beta\ddot{\boldsymbol{x}}(t)\right)$$

- Substitution:

$$(\mathbf{M} + \Delta t\gamma\mathbf{C} + \Delta t^2\beta\mathbf{K})\ddot{\boldsymbol{x}}(t) = \boldsymbol{f}(t) - \left\{\Delta t(1-\gamma)\mathbf{C} + \Delta t^2\left(\frac{1}{2}-\beta\right)\mathbf{K}\right\}\ddot{\boldsymbol{x}}(t-\Delta t) - \{\mathbf{C} + \Delta t\,\mathbf{K}\}\dot{\boldsymbol{x}}(t-\Delta t) - \mathbf{K}\boldsymbol{x}(t-\Delta t)$$

- **Discretized Equation:**

$$\ddot{\boldsymbol{x}}(t) = [\mathbf{M} + \Delta t\gamma\mathbf{C} + \Delta t^2\beta\mathbf{K}]^{-1}\left\{\boldsymbol{f}(t) - \left\{\Delta t(1-\gamma)\mathbf{C} + \Delta t^2\left(\frac{1}{2}-\beta\right)\mathbf{K}\right\}\ddot{\boldsymbol{x}}(t-\Delta t) - \{\mathbf{C} + \Delta t\,\mathbf{K}\}\dot{\boldsymbol{x}}(t-\Delta t) - \mathbf{K}\boldsymbol{x}(t-\Delta t)\right\}$$

$$\boxed{\mathbf{A}\ddot{\boldsymbol{x}}(t) = \boldsymbol{b}(t)} \Rightarrow \boxed{\ddot{\boldsymbol{x}}(t) = \mathbf{A}^{-1}\boldsymbol{b}(t)}$$

Global Matrix

Solution

RHS
Right-Hand Side

The problem of Inverse Matrix

Newmark-β method

Function File

$[\boldsymbol{x}(0)\ \dot{\boldsymbol{x}}(0)\ \ddot{\boldsymbol{x}}(0)]$

$\boldsymbol{f}(t)$ Δt γ β the inputs

```
function [ t, x, dx, ddx ] = newmark_beta( m,c,k,f,dt,X, G, B )
%% initialization
x=zeros(size(f));
dx=zeros(size(f));
ddx=zeros(size(f));
%-- input initial values
x(:,1)=X(:, 1);
dx(:,1)=X(:, 2);
ddx(:,1)=X(:, 3);
%-- time vector
T=length(f);
t=(0:T-1)*dt;
%% solving
A = m + dt*G*c + dt^2*B*k; %-- global matrix
M = inv(A); %-- inverse matrix of A
for tt=2:T
    %-- solving preparation
    b1 = -c * ( dx(:,tt-1) + (1-G)*dt*ddx(:, tt-1) );
    b2 = -k * ( x(:,tt-1) + dt*dx(:,tt-1) + (1/2-B)*dt^2*ddx(:,tt-1) );
    b = f(:,tt) + b1 + b2; %-- right hand side
    %-- calculation of acceleration at t=t+1
    ddx(:,tt) = M*b;
    %-- calculation of velocity and displacement responses
    dx(:,tt) = dx(:,tt-1) + dt*(1-G)*ddx(:,tt-1) + dt*G*ddx(:,tt);
    x(:,tt) = x(:,tt-1) + dt*dx(:,tt-1) + dt^2*(1/2-B)*ddx(:,tt-1) + dt^2*B*ddx(:,tt);
end
end
```

$$\boldsymbol{b}_1(t) = -\mathbf{C}\{\dot{\boldsymbol{x}}(t-\Delta t) + \Delta t(1-\gamma)\ddot{\boldsymbol{x}}(t-\Delta t)\}$$

$$\boldsymbol{b}_2(t) = -\mathbf{K}\left\{\boldsymbol{x}(t-\Delta t) + \Delta t\dot{\boldsymbol{x}}(t-\Delta t) + \Delta t^2\left(\frac{1}{2}-\beta\right)\ddot{\boldsymbol{x}}(t-\Delta t)\right\}$$

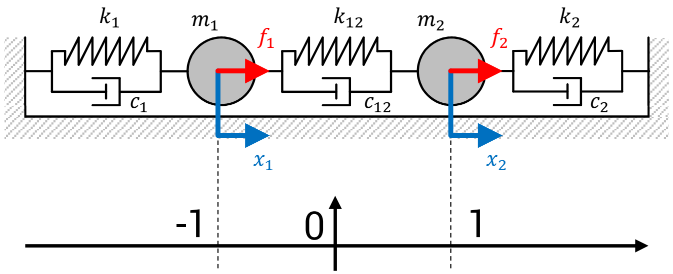
$$\boldsymbol{b}(t) = \boldsymbol{f}(t) + \boldsymbol{b}_1(t) + \boldsymbol{b}_2(t)$$

$$\ddot{\boldsymbol{x}}(t) = \mathbf{A}^{-1}\boldsymbol{b}(t)$$

Example 03:
Numerical Simulation of MDOF by Newmark-β method

- Governing Equation:

$$\begin{bmatrix} m_1 & \\ & m_2 \end{bmatrix}\begin{Bmatrix} \ddot{x}_1(t) \\ \ddot{x}_2(t) \end{Bmatrix} + \begin{bmatrix} c_1 + c_{12} & -c_{12} \\ -c_{12} & c_2 + c_{12} \end{bmatrix}\begin{Bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{Bmatrix} + \begin{bmatrix} k_1 + k_{12} & -k_{12} \\ -k_{12} & k_2 + k_{12} \end{bmatrix}\begin{Bmatrix} x_1(t) \\ x_2(t) \end{Bmatrix} = \begin{Bmatrix} f_1(t) \\ f_2(t) \end{Bmatrix}$$



- System Parameters:

m_1	m_2	c_1	c_2	c_{12}	k_1	k_2	k_{12}
1.0	2.0	0.05	0.01	0.01	1.0	4.0	8.0

- Input and Initial Condition:

$f_1(t)$	$f_2(t)$	$x_1(t)$	$x_2(t)$
0	0	-0.5	0.5

Example 03:

Numerical Simulation of MDOF by Newmark-β method

☐ Launcher Script:

```
%-- System Parameters
m1 = 1.00; m2 = 2.00;
c1 = 0.05; c2 = 0.01; c12= 0.01;
k1 = 1.00; k2 = 4.00; k12= 8.00;
%-- Simulation Parameters
G  = 1/2; B  = 1/6;
dt = 0.01; T  = 100;
t  = 0:dt:T; TT = length(t);
%-- Input
f  = zeros(2,TT);
%-- Initial Condition
X  = [-0.5  0  0;
       0.5  0  0];
%-- Making Matrices
M = [ m1     0      ;
      0      m2     ];
C = [ c1+c12 -c12   ;
      -c12   c1+c12];
K = [ k1+k12 -k12   ;
      -k12   k2+k12];
%-- Simulation
[ t, x, dx, ddx ] = newmark_beta( M,C,K,f,dt,X, G, B );
```

m_1	m_2
1.0	2.0

c_1	c_2	c_{12}
0.05	0.01	0.01

k_1	k_2	k_{12}
1.0	4.0	8.0

γ	β
1/2	1/6

Δt	T
0.01	100

$$\begin{bmatrix} x_1(0) & \dot{x}_1(0) & \ddot{x}_1(0) \\ x_2(0) & \dot{x}_2(0) & \ddot{x}_2(0) \end{bmatrix} = \begin{bmatrix} -0.5 & 0 & 0 \\ 0.5 & 0 & 0 \end{bmatrix}$$

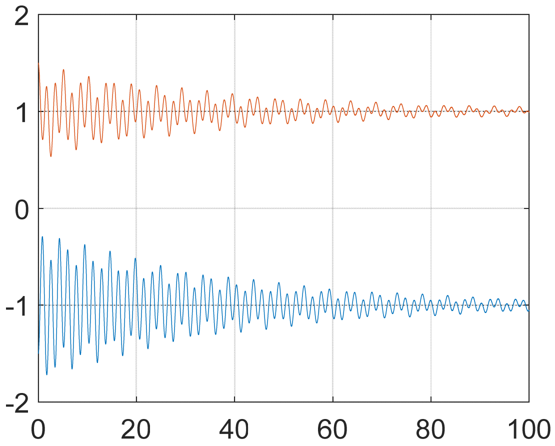
$$\begin{bmatrix} m_1 & \\ & m_2 \end{bmatrix} \begin{Bmatrix} \ddot{x}_1(t) \\ \ddot{x}_2(t) \end{Bmatrix} + \begin{bmatrix} c_1 + c_{12} & -c_{12} \\ -c_{12} & c_2 + c_{12} \end{bmatrix} \begin{Bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{Bmatrix} + \begin{bmatrix} k_1 + k_{12} & -k_{12} \\ -k_{12} & k_2 + k_{12} \end{bmatrix} \begin{Bmatrix} x_1(t) \\ x_2(t) \end{Bmatrix} = \begin{Bmatrix} f_1(t) \\ f_2(t) \end{Bmatrix}$$

Example 03:

Numerical Simulation of MDOF by Newmark-β method

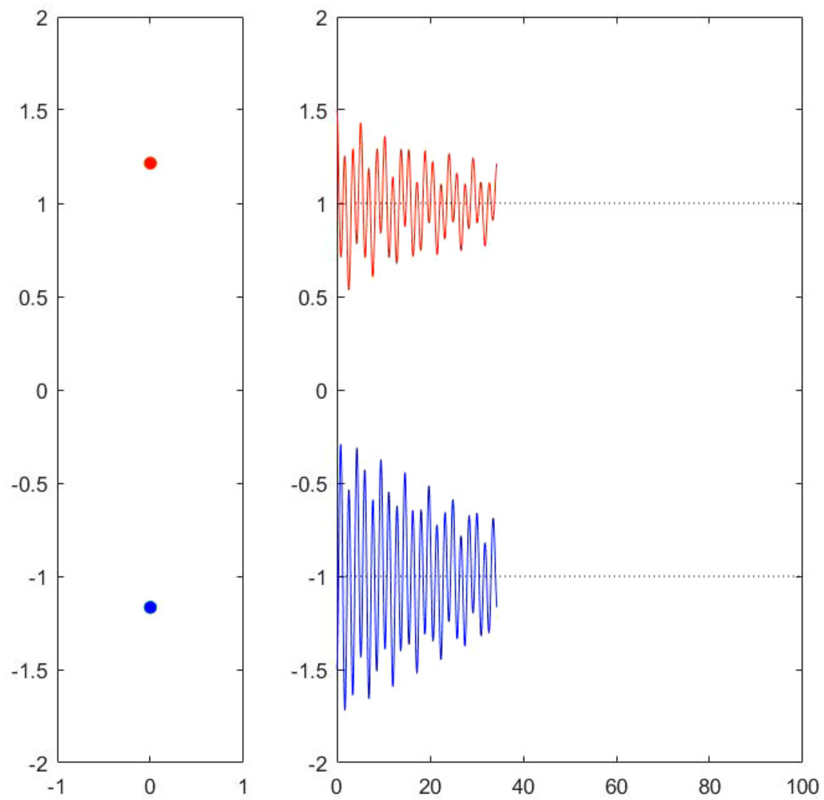
☐ Visualization

```
%-- Visualization
figure(1);
plot(t, [1;-1]*ones(1,TT),'k:')
hold on;
plot(t,[-1+x(1,:); 1+x(2,:)]);
hold off
set(gca,'XLim', [0 100], 'XGrid', 'on')
set(gca,'YLim', [-2 2], 'YGrid', 'on')
set(gca,'FontSize',18)
print('ex_03_nmb_01','-dmeta','-r1200')
```




```
%-- Animation
figure(2)
writerObj = VideoWriter('ex_03_nmb_sample.avi','Motion JPEG AVI');
writerObj.FrameRate = 20;
open(writerObj);
set(gcf, 'Unit', 'pixel', 'Position',[240, 120, 600, 600]);
fs=100; %-- flame size
for tt=1:TT
    figure(2);
    subplot('Position',[0.1 0.1 0.2 0.8])
    BALL_PLOT=plot(0,[-1+x(1,tt); 1+x(2,tt)],'o');
    set(BALL_PLOT(1), 'FaceColor', [0,0,1])
    set(BALL_PLOT(2), 'FaceColor', [1,0,0])
    ylim([-2 2])
    subplot('Position',[0.4 0.1 0.5 0.8])
    GRID_PLOT=plot(t, [-1;1]*ones(1,TT),'k:');hold on
    CURVE_PLOT=plot(t(1:tt),[-1+x(1,tt); 1+x(2,tt)],'-');
    hold off
    set(CURVE_PLOT(1), 'Color', [0,0,1])
    set(CURVE_PLOT(2), 'Color', [1,0,0])
    xlim([0 T])
    frame = getframe(figure(2));
    writeVideo(writerObj,frame);
end
close(writerObj); %-- end of animation
```

Example 03:
Numerical Simulation of MDOF by Newmark-β method



筑波大学
University of Tsukuba

IMAGINE
THE
FUTURE.