Homework 1: Equations of motion and eigenvalue problem

Name: Abdelrahman Fathy Abdelhaleem Aly Abdelrahman

Matr.-Nr.: 108023251500

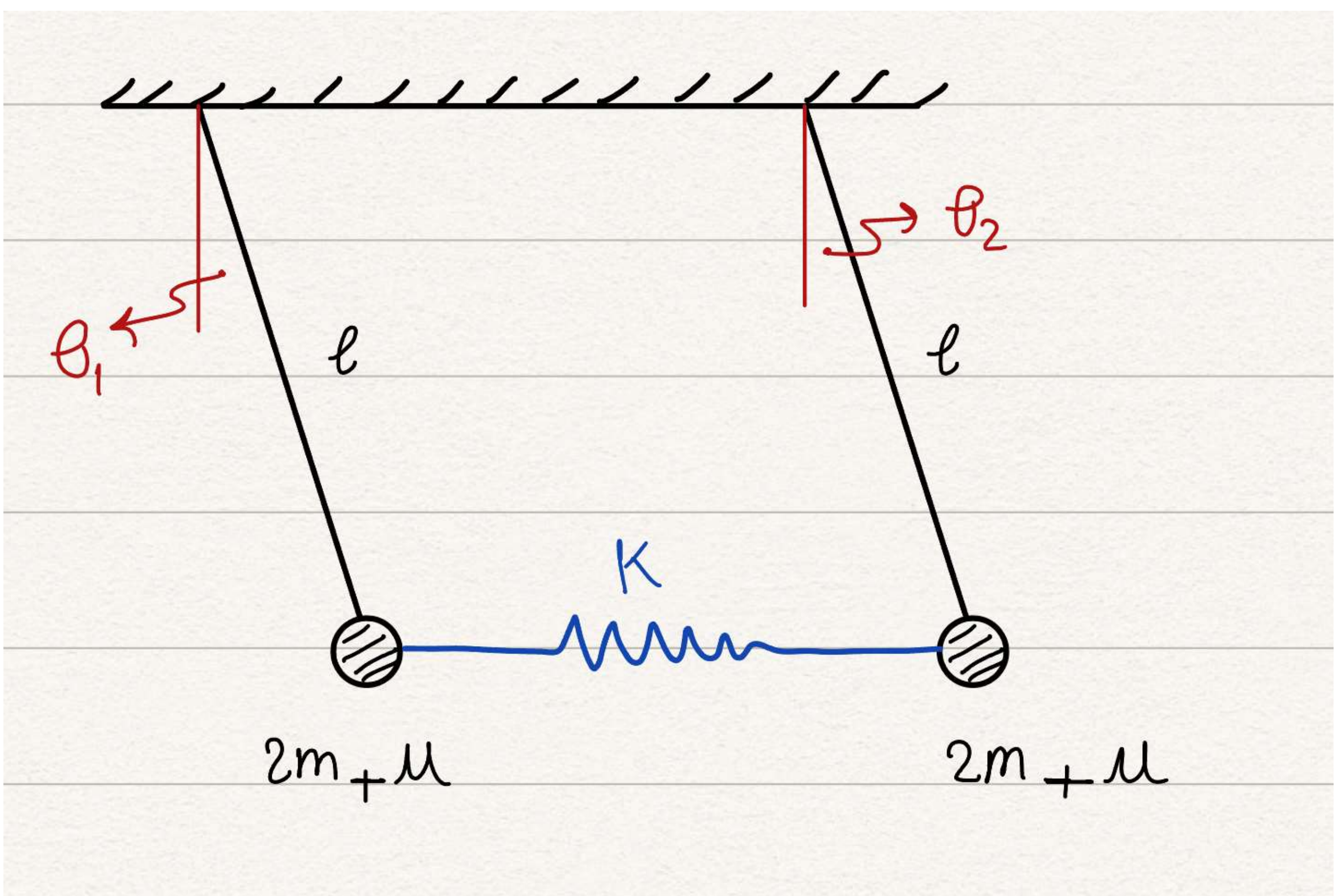
```julia
using Images, FileIO,PlutoUI
```

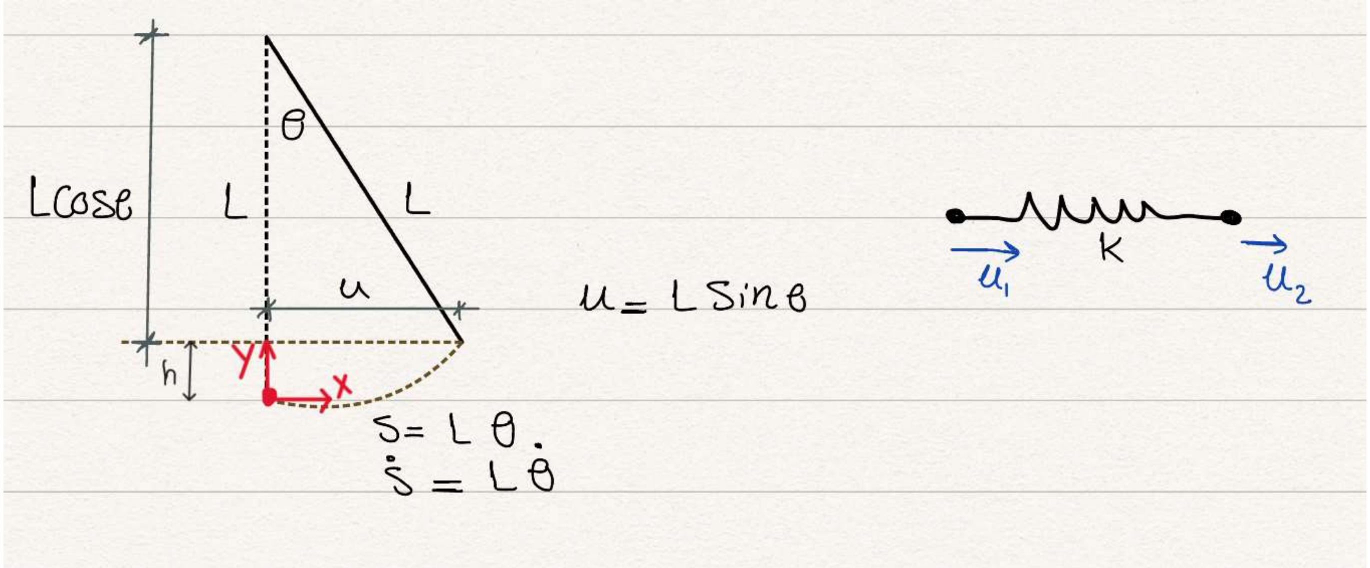
Task 1: Independent degrees of freedom:

The system can be reduced to a 2 DOFs system as shown in the following figure.

1.1. DOFs:

$$\theta_1, \theta_2$$





Task 2: Kinetic energy, Potential energy, Lagrangian:

Kinetic Energy (T):

$$T = \frac{1}{2} m' (\dot{s}_1^2 + \dot{s}_2^2) = \frac{1}{2} (2m + M) L^2 (\dot{\theta}_1^2 + \dot{\theta}_2^2)$$

Potential Energy (V):

> Note
>
> The potential energy will be due to two things; 1) height of the mass from the origin point, 2) deformation of spring (i.e. elastic energy).

$$V = m' g L (1 - \cos\theta_1) + m' g L (1 - \cos\theta_2) + \underbrace{\frac{1}{2} k \Delta u^2}_{\text{PE due to spring}}$$

$$V = (2m + M) g L (2 - \cos\theta_1 - \cos\theta_2) + \frac{1}{2} k L^2 (\sin\theta_2 - \sin\theta_1)^2$$

Lagrangian (L):

$$L = T - V$$

$$L = \frac{1}{2}(2m + M) L^2 (\dot{\theta}_1^2 + \dot{\theta}_2^2) - (2m + M) g L (2 - \cos\theta_1 - \cos\theta_2) - \frac{1}{2} k L^2 (\sin\theta_2 - \sin\theta_1)^2$$

Task 3: Euler-Lagrange differential equation:

$$\frac{\partial L}{\partial \theta_i} - \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_i}\right) = 0$$

For θ_1:

$$\frac{\partial L}{\partial \theta_1} = -(2m+M)gL(\sin\theta_1) - kL^2(\sin\theta_2 - \sin\theta_1)(-\cos\theta_1)$$

$$\frac{\partial L}{\partial \dot{\theta}_1} = (2m+M)L^2\dot{\theta}_1$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_1}\right) = (2m+M)L^2\ddot{\theta}_1$$

$$\therefore -(2m+M)gL(\sin\theta_1) - kL^2(\sin\theta_2 - \sin\theta_1)(-\cos\theta_1) - (2m+M)L^2\ddot{\theta}_1 = 0$$

For θ_2:

$$\frac{\partial L}{\partial \theta_2} = -(2m+M)gL(\sin\theta_2) - kL^2(\sin\theta_2 - \sin\theta_1)(\cos\theta_2)$$

$$\frac{\partial L}{\partial \dot{\theta}_2} = (2m+M)L^2\dot{\theta}_2$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}_2}\right) = (2m+M)L^2\ddot{\theta}_2$$

$$\therefore -(2m+M)gL(\sin\theta_2) - kL^2(\sin\theta_2 - \sin\theta_1)(\cos\theta_2) - (2m+M)L^2\ddot{\theta}_2 = 0$$

Putting all together:

> Note
>
> For small amplitude vibrations $\theta \approx 0$:
>
> $\sin\theta \approx \theta$
>
> $\cos\theta \approx 1 \rightarrow$ (e.g. $\cos 0 = 1$)

$$-(2m+M)gL\underbrace{(\sin\theta_1)}_{\approx\theta_1} + kL^2\underbrace{\cos\theta_1}_{\approx 1}\underbrace{(\sin\theta_2 - \sin\theta_1)}_{\approx(\theta_2-\theta_1)} - (2m+M)L^2\ddot{\theta}_1 = 0$$

$$-(2m+M)gL\underbrace{(\sin\theta_2)}_{\approx\theta_2} - kL^2\underbrace{\cos\theta_2}_{\approx 1}\underbrace{(\sin\theta_2 - \sin\theta_1)}_{\approx(\theta_2-\theta_1)} - (2m+M)L^2\ddot{\theta}_2 = 0$$

By dividing both equation by L^2:

$$(2m+M)\ddot{\theta}_1 + \frac{(2m+M)g}{L}\theta_1 - k(\theta_2 - \theta_1) = 0$$

$$(2m+M)\ddot{\theta}_2 + \frac{(2m+M)g}{L}\theta_2 + k(\theta_2 - \theta_1) = 0$$

Putting both equations in matrix form will yield:

$$\underbrace{\begin{bmatrix} 2m+M & 0 \\ 0 & 2m+M \end{bmatrix}}_{\boldsymbol{M}} \cdot \underbrace{\begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix}}_{\ddot{\boldsymbol{u}}} + \underbrace{\begin{bmatrix} \frac{(2m+M)g}{L} + k & -k \\ -k & \frac{(2m+M)g}{L} + k \end{bmatrix}}_{\boldsymbol{K}} \cdot \underbrace{\begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}}_{\boldsymbol{u}} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$
$$\therefore \boldsymbol{M} \cdot \ddot{\boldsymbol{u}} + \boldsymbol{K} \cdot \boldsymbol{u} = \boldsymbol{0}$$

Data given for the upcoming tasks:

$$m = 25\ Kg$$
$$M = 700\ Kg$$
$$k = 2 \times 10^2 N/m$$
$$l = 1\ m$$
$$g = 9.8\ m/s^2$$

```
10350.0
```

```
((2m+M_)*g)/L + k
```

Task 4: Eigenvalues and eigenvectors using characteristic polynomial of the system:

$$\boldsymbol{M} \cdot \ddot{\boldsymbol{u}} + \boldsymbol{K} \cdot \boldsymbol{u} = \boldsymbol{0} \rightarrow \text{(eq. 1)}$$
$$\boldsymbol{u} = \boldsymbol{\phi} f(t) \rightarrow \text{(eq. 2)}$$
$$\ddot{\boldsymbol{u}} = -\omega^2 \boldsymbol{\phi} f(t) \rightarrow \text{(eq. 3)}$$

where ω(angular frequency)$= \sqrt{\lambda}$, T(time period)$= \frac{2\pi}{\omega}$.
By inserting eq. 2 & eq. 3 in eq. 1 will yield:

$$\boldsymbol{K} \cdot \boldsymbol{\phi} = \underbrace{\omega^2}_{\lambda} \boldsymbol{M} \cdot \boldsymbol{\phi} \rightarrow (\boldsymbol{K} - \lambda \boldsymbol{M}) \cdot \boldsymbol{\phi} = \boldsymbol{0} \rightarrow \text{(eq. 4)}$$

for eq. 4 to be valid for nonzero $\boldsymbol{\phi}$, then $(\boldsymbol{K} - \lambda \boldsymbol{M})$ has to be singular matrix
$\iff \det(\boldsymbol{K} - \lambda \boldsymbol{M}) \overset{!}{=} 0.$

$$\Rightarrow \boldsymbol{K} - \lambda \boldsymbol{M} = \begin{bmatrix} 10350 & -3000 \\ -3000 & 10350 \end{bmatrix} - \lambda \begin{bmatrix} 750 & 0 \\ 0 & 750 \end{bmatrix} = \begin{bmatrix} 10350 - 750\lambda & -3000 \\ -3000 & 10350 - 750\lambda \end{bmatrix}$$
$$\Rightarrow \det\left(\begin{bmatrix} 10350 - 750\lambda & -3000 \\ -3000 & 10350 - 750\lambda \end{bmatrix} \right) = 0$$
$$\Rightarrow (10350 - 750\lambda)^2 - 9 \times 10^6 = 0$$
$$\Rightarrow 562500\lambda^2 - 15525000\lambda + 98122500 = 0$$
$$\Rightarrow \lambda^2 - 27.6\lambda + 174.44 = 0$$

solving the previous polynomial will yield the following eigenvalues:

$$\lambda_1 = 17.8$$
$$\lambda_2 = 9.8$$

Eigenvector for λ_1:

$$\begin{bmatrix} 10350 - 750\lambda_1 & -3000 \\ -3000 & 10350 - 750\lambda_1 \end{bmatrix} \cdot \begin{bmatrix} \phi_{11} \\ \phi_{12} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$
$$\Rightarrow \begin{bmatrix} -3000 & -3000 \\ -3000 & -3000 \end{bmatrix} \cdot \begin{bmatrix} \phi_{11} \\ \phi_{12} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

assume $\phi_{11} = 1$, then $\phi_{12} = -1$.

$$\bar{\boldsymbol{\phi}}_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \Rightarrow \boldsymbol{\phi}_1 = \frac{\bar{\boldsymbol{\phi}}_1}{||\bar{\boldsymbol{\phi}}_1||} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

Eigenvector for λ_2:

$$\begin{bmatrix} 10350 - 750\lambda_2 & -3000 \\ -3000 & 10350 - 750\lambda_2 \end{bmatrix} \cdot \begin{bmatrix} \phi_{21} \\ \phi_{22} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$
$$\Rightarrow \begin{bmatrix} 3000 & -3000 \\ -3000 & 3000 \end{bmatrix} \cdot \begin{bmatrix} \phi_{21} \\ \phi_{22} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

assume $\phi_{21} = 1$, then $\phi_{22} = 1$.

$$\bar{\boldsymbol{\phi}}_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \Rightarrow \boldsymbol{\phi}_2 = \frac{\bar{\boldsymbol{\phi}}_2}{||\bar{\boldsymbol{\phi}}_2||} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Task 5: Orthogonality of the eigenvectors:

since $\boldsymbol{\phi}_1 \cdot \boldsymbol{\phi}_2 = 0$, then the eigenvectors are orthogonal and the reason is that $\boldsymbol{K}$ is symmetric matrix and from linear algebra we know that for any symmetric matrix of *dim = n* (i.e. $\mathbb{R}^{nxn}$), it has *n* real eigenvalues and its eigenvectors are orthogonal.

Proof of the argument:

suppose $\boldsymbol{A}$ is a symmetric matrix (i.e. $\boldsymbol{A} = \boldsymbol{A}^T$) with two eigenvectors and eigenvalues.
i.e.,

$$\boldsymbol{A} \cdot \boldsymbol{\phi_1} = \lambda_1 \boldsymbol{\phi_1}$$
$$\boldsymbol{A} \cdot \boldsymbol{\phi_2} = \lambda_2 \boldsymbol{\phi_2}$$

Therefore,

$$\boldsymbol{\phi_2^T} \cdot \boldsymbol{A} \cdot \boldsymbol{\phi_1} = \lambda_1 \boldsymbol{\phi_2^T} \cdot \boldsymbol{\phi_1} \rightarrow \text{(eq. 5)}$$
$$(\boldsymbol{A} \cdot \boldsymbol{\phi_2})^T \cdot \boldsymbol{\phi_1} = \lambda_2 \boldsymbol{\phi_2^T} \cdot \boldsymbol{\phi_1} \rightarrow \text{(eq. 6)}$$

Both Equation *5* & *6* are equal and for equality to be valid for non zero λ then ϕ_1 and ϕ_2 has to be orthogonal.

Task 6: Forward iteration and inverse iteration implementation:

6.1. Forward iteration:

```
(m = 25.0, M_ = 700.0, k = 3000.0, L = 1.0, g = 9.8)
```

```
begin
    m =25.0
    M_ = 700.0
    k = 3e3
    L = 1.0
    g = 9.8
    (;m,M_,k,L,g)
end
```

```
struct Eigen{Eigenvalue<:Real,Eigenvector<:Vector}
    λ::Eigenvalue
    ϕ::Eigenvector
end
```

```
forward_iter (generic function with 1 method)
```

```julia
function forward_iter(K,M,x,tol)
    # K → stiffness matrix
    # M → mass matrix
    # x → initial guess
    # tol → tolerence

    y = K*x

    err = tol * 2 # any number above the tolerence
    ρₙ₁ = 0.0
    ρₙ = 0.0
    n = 0 # counter to count number of loops before convergence

    while err ≥ tol
        n +=1
        xbar = inv(M) * y
        ybar = K*xbar
        ρₙ = ρₙ₁
        ρₙ₁ = (xbar'*ybar)/(xbar'*y)
        err = abs(ρₙ₁ - ρₙ )/ ρₙ₁
        y = ybar/sqrt(xbar'*y)
     end

    λ = ρₙ₁
    ϕ = inv(K) * y
    ϕ = ϕ/norm(ϕ)

    eigen = Eigen(λ,ϕ)
    num_iter = n
    (;eigen,num_iter)
end
```

6.2. Inverse iteration:

```
inverse_iter (generic function with 1 method)
```

```julia
function inverse_iter(K,M,x,tol)
    # K → stiffness matrix
    # M → mass matrix
    # x → initial guess
    # tol → tolerence

    y = M*x

    err = tol * 2 # any number above the tolerence
    ρₙ₁ = 0.0
    ρₙ = 0.0
    n = 0 # counter to count number of loops before convergence

    while err ≥ tol
        n +=1
        xbar = inv(K) * y
        ybar = M*xbar
        ρₙ = ρₙ₁
        ρₙ₁ = (xbar'*y)/(xbar'*ybar)
        err = abs(ρₙ₁ - ρₙ )/ ρₙ₁
        y = ybar/sqrt(xbar'*ybar)
     end

    λ = ρₙ₁
    ϕ = inv(M) * y
    ϕ = ϕ/norm(ϕ)

    eigen = Eigen(λ,ϕ)
    num_iter = n
    (;eigen,num_iter)

end
```

```
(
   K = 2×2 Matrix{Float64}:
        10350.0  -3000.0
        -3000.0  10350.0
   M = 2×2 Matrix{Float64}:
        750.0    0.0
          0.0  750.0
   TOL = 1.0e-6
)
```

```julia
begin
    K = [((2m + M_)*g)/L + k -k; -k ((2m + M_)*g)/L + k]
    M = [2m+M_ 0;0 2m+M_]
    TOL = 1e-6
    (;K,M,TOL)
end
```

Task 7:

$$\text{initial guess: } \boldsymbol{x}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

```
x₁ =  [1, 0]
```

```
x₁ = [1;0]
```

7.1. Solve using forward itertion:

```
(eigen = Eigen(17.8, [0.707655, -0.706558]), num_iter = 12)
```

```
forward_iter(K,M,x₁,TOL)
```

> Note
>
> The returned eigenvalue from the **forward iteration** method is *17.8* which is the **largest**. Moreover, the algorithim **converged after 12 iterations**.

7.2. Solve using inverse iteration:

```
(eigen = Eigen(9.8, [0.707409, 0.706805]), num_iter = 13)
```

```
inverse_iter(K,M,x₁,TOL)
```

> Note
>
> The returned eigenvalue from the **inverse iteration** method is *9.8* which is the **smallest**. Moreover, the algorithim **converged after 13 iterations**.

Task 8: Check forward iteration convergence:

$$\text{initial guess: } \boldsymbol{x}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

```
x₂ =  [1, 1]
```

```
x₂ = [1;1]
```

```
(eigen = Eigen(9.8, [0.707107, 0.707107]), num_iter = 2)
```

```
forward_iter(K,M,x₂,TOL)
```

8.1. Explanation:

As observed from the previous result that *forward iteration* **converged to the smallest eigenvlue** rather than the largest and the reason is; the **initial guess** $[1 \;\; 1]^T$ is **orthogonal** to the eigenvector associated with the largest eigenvalue, accordingly, *forward iteration* will never converge to the largest eigenvalue anymore.