

Homework 1: Equations of motion and eigenvalue problem

Name: Abdelrahman Fathy Abdelhaleem Aly Abdelrahman

Matr.-Nr.: 108023251500

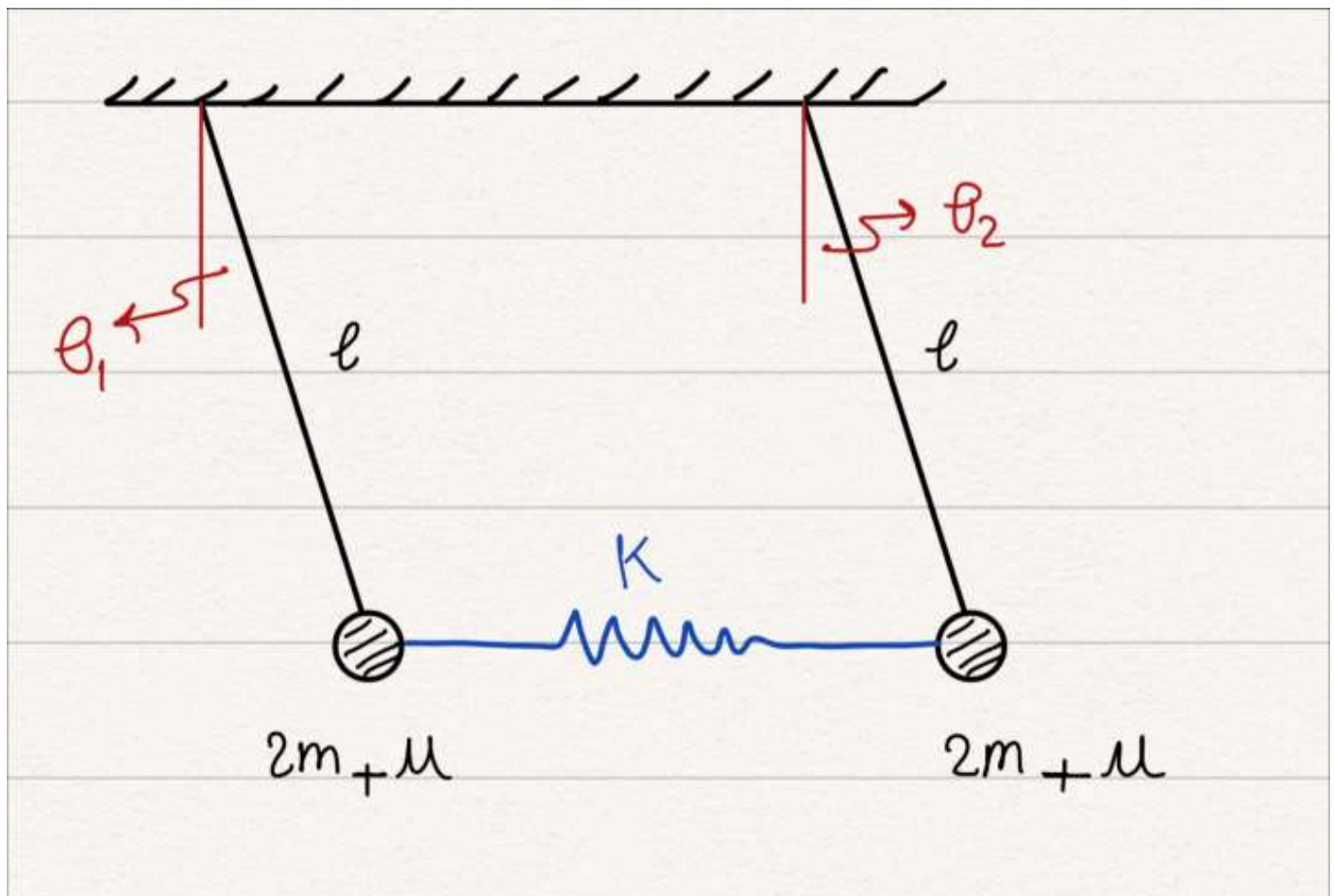
1 using Images, FileIO

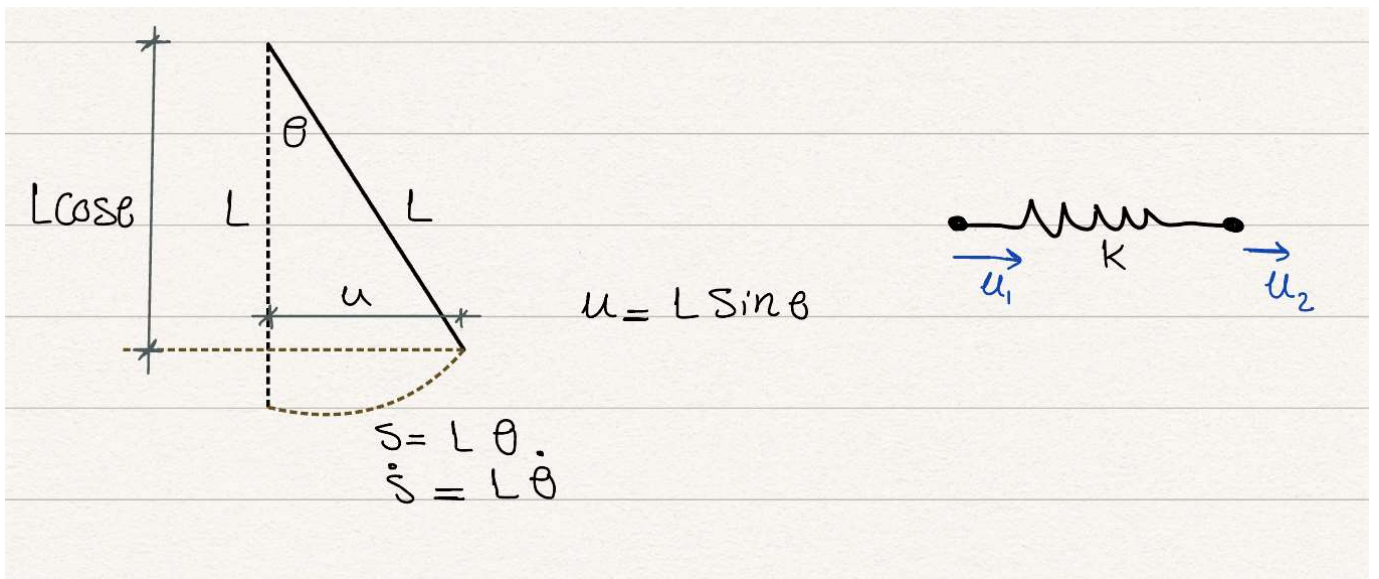
Task 1: Independent degrees of freedom:

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):

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

Potential Energy (V):

Note

The potential energy will be due to two things; 1) position of the mass, 2) deformation of spring (i.e. elastic energy).

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

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

Lagrangian (L):

$$L = T - V$$

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

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 :

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

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

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

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

For θ_2 :

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

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

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

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

Putting all together:

Note

For small amplitude vibrations $\theta \approx 0$:

$$\sin \theta \approx \theta$$

$$\cos \theta \approx 1 \rightarrow (\text{e.g. } \cos 0 = 1)$$

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

$$-(2m + M)L(\underbrace{\sin \theta_2}_{\theta_2}) - kL^2 \underbrace{\cos \theta_2}_1 (\underbrace{\sin \theta_2 - \sin \theta_1}_{(\theta_2 - \theta_1)}) - (2m + M)L^2\ddot{\theta}_2 = 0$$

By dividing both equation by L^2 :

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

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

Putting both equations in matrix form will yield:

- - -

-

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

$$\therefore \mathbf{M} \cdot \ddot{\mathbf{u}} + \mathbf{K} \cdot \mathbf{u} = \mathbf{0}$$

Data given for the upcoming tasks:

$$\begin{aligned} m &= 25 \text{ Kg} \\ M &= 700 \text{ Kg} \\ k &= 2 \times 10^2 \text{ N/m} \\ l &= 1 \text{ m} \\ g &= 9.8 \text{ m/s}^2 \end{aligned}$$

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

$$\mathbf{M} \cdot \ddot{\mathbf{u}} + \mathbf{K} \cdot \mathbf{u} = \mathbf{0} \rightarrow (\text{eq. 1})$$

$$\mathbf{u} = \phi f(t) \rightarrow (\text{eq. 2})$$

$$\ddot{\mathbf{u}} = -\omega^2 \phi f(t) \rightarrow (\text{eq. 3})$$

where ω (angular frequency) = $\sqrt{\lambda}$, T (time period) = $\frac{2\pi}{\omega}$.

By inserting eq. 2 & eq. 3 in eq. 1 will yield:

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

for eq. 4 to be valid for nonzero ϕ , then $(\mathbf{K} - \lambda \mathbf{M})$ has to be singular matrix

$$\iff \det(\mathbf{K} - \lambda \mathbf{M}) \stackrel{!}{=} 0.$$

$$\begin{aligned} \mathbf{K} - \lambda \mathbf{M} &= \begin{bmatrix} 3750 & -3000 \\ -3000 & 3750 \end{bmatrix} - \lambda \begin{bmatrix} 750 & 0 \\ 0 & 750 \end{bmatrix} = \begin{bmatrix} 3750 - 750\lambda & -3000 \\ -3000 & 3750 - 750\lambda \end{bmatrix} \\ \Rightarrow \det \left(\begin{bmatrix} 3750 - 750\lambda & -3000 \\ -3000 & 3750 - 750\lambda \end{bmatrix} \right) &= 0 \\ \Rightarrow (3750 - 750\lambda)^2 - 9 \times 10^6 &= 0 \\ \Rightarrow 562500\lambda^2 - 5625000\lambda + 5062500 &= 0 \\ \Rightarrow \lambda^2 - 10\lambda + 9 &= 0 \end{aligned}$$

solving the previous polynomial will yield the following eigenvalues:

$$\begin{aligned} \lambda_1 &= 9 \\ \lambda_2 &= 1 \end{aligned}$$

Eigenvector for λ_1 :

$$\begin{bmatrix} 3750 - 750\lambda_1 & -3000 \\ -3000 & 3750 - 750\lambda_1 \end{bmatrix} \cdot \begin{bmatrix} \phi_{11} \\ \phi_{12} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ \Rightarrow \begin{bmatrix} -3000 & -3000 \\ -3000 & -3000 \end{bmatrix} \cdot \begin{bmatrix} \phi_{11} \\ \phi_{12} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

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

$$\bar{\phi}_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \Rightarrow \phi_1 = \frac{\bar{\phi}_1}{\|\bar{\phi}_1\|} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

Eigenvector for λ_2 :

$$\begin{bmatrix} 3750 - 750\lambda_2 & -3000 \\ -3000 & 3750 - 750\lambda_2 \end{bmatrix} \cdot \begin{bmatrix} \phi_{21} \\ \phi_{22} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ \Rightarrow \begin{bmatrix} 3000 & -3000 \\ -3000 & 3000 \end{bmatrix} \cdot \begin{bmatrix} \phi_{21} \\ \phi_{22} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

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

$$\bar{\phi}_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \Rightarrow \phi_2 = \frac{\bar{\phi}_2}{\|\bar{\phi}_2\|} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Task 5:

since $\phi_1 \cdot \phi_2 = 0$, then the eigenvectors are orthogonal and the reason behind that is that \mathbf{K} is symmetric matrix and from linear algebra we know that for any symmetric matrix of $\dim = n$, it has n real eigenvalues and its eigenvectors are 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)

```
1 begin
2     m = 25.0
3     M_ = 700.0
4     k = 3e3
5     L = 1.0
6     g = 9.8
7     (;m,M_,k,L,g)
8 end
```

```

1 struct Eigen{Eigenvalue<:Real,Eigenvector<:Vector}
2     λ::Eigenvalue
3     φ::Eigenvector
4 end

```

forward_iter (generic function with 1 method)

```

1 function forward_iter(K,M,x,tol)
2     # K → stiffness matrix
3     # M → mass matrix
4     # x → initial guess
5     # tol → tolerance
6
7     y = K*x
8
9     err = tol * 2 # any number above the tolerance
10    ρn1 = 0.0
11    ρn = 0.0
12    n = 0 # counter to count number of loops before convergence
13
14    while err ≥ tol
15        n +=1
16        xbar = inv(M) * y
17        ybar = K*xbar
18        ρn = ρn1
19        ρn1 = (xbar'*ybar)/(xbar'*y)
20        err = abs(ρn1 - ρn )/ ρn1
21        y = ybar/sqrt(xbar'*y)
22    end
23
24    λ = ρn1
25    φ = inv(K) * y
26    φ = φ/norm(φ)
27
28    eigen = Eigen(λ,φ)
29    num_iter = n
30    (;eigen,num_iter)
31 end

```

6.2. Inverse iteration:

inverse_iter (generic function with 1 method)

```
1 function inverse_iter(K,M,x,tol)
2     # K → stiffness matrix
3     # M → mass matrix
4     # x → initial guess
5     # tol → tolerance
6
7     y = M*x
8
9     err = tol * 2 # any number above the tolerance
10    ρn1 = 0.0
11    ρn = 0.0
12    n = 0 # counter to count number of loops before convergence
13
14    while err ≥ tol
15        n +=1
16        xbar = inv(K) * y
17        ybar = M*xbar
18        ρn = ρn1
19        ρn1 = (xbar'*y)/(xbar'*ybar)
20        err = abs(ρn1 - ρn)/ ρn1
21        y = ybar/sqrt(xbar'*ybar)
22    end
23
24    λ = ρn1
25    φ = inv(M) * y
26    φ = φ/norm(φ)
27
28    eigen = Eigen(λ,φ)
29    num_iter = n
30    (;eigen,num_iter)
31
32 end
```

```
(K = 2x2 Matrix{Float64}[:, M = 2x2 Matrix{Float64}[:, TOL = 1.0e-6)
      3750.0  -3000.0      750.0    0.0
```

```
1 begin
2     K = [(2m + M-)/L + k -k; -k (2m + M-)/L + k]
3     M = [2m+M- 0;0 2m+M-]
4     TOL = 1e-6
5     (;K,M,TOL)
6 end
```

Task 7:

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

$\mathbf{x}_1 = [1, 0]$

```
1 x1 = [1;0]
```

7.1. Solve using forward iteration:

```
(eigen = Eigen(9.0, [0.707119, -0.707095]), num_iter = 5)
```

```
1 forward_iter(K,M,x1,TOL)
```

Note

The returned eigenvalue from the **forward iteration** method is 9 which is the **largest**. Moreover, the algorithm **converged after 5 iterations**.

7.2. Solve using inverse iteration:

```
(eigen = Eigen(1.0, [0.707119, 0.707095]), num_iter = 5)
```

```
1 inverse_iter(K,M,x1,TOL)
```

Note

The returned eigenvalue from the **inverse iteration** method is 1 which is the **smallest**. Moreover, the algorithm **converged after 5 iterations**.

Task 8: Check forward iteration convergence:

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

```
x2 = [1, 1]
```

```
1 x2 = [1;1]
```

```
(eigen = Eigen(1.0, [0.707107, 0.707107]), num_iter = 2)
```

```
1 forward_iter(K,M,x2,TOL)
```

8.1. Explanation:

As observed from the previous result that *forward iteration* **converged to the smallest eigenvalue** rather than the largest and the reason is the **initial guess** $\begin{bmatrix} 1 & 1 \end{bmatrix}^T$ which is **orthogonal** to the eigenvector associated with the largest eigenvalue, accordingly, *forward iteration* will never converge to the largest eigenvalue anymore.

