Proposal Presentations

- Wednesday 12 – 2 pm, November 4, 2024
- 5-minute presentation (~3-4 slides), 2-minute Q/A
- You are required to use one of the following two templates:
  - LaTeX template (highly recommended)
    http://ras.papercept.net/conferences/support/tex.php
  - MS Word template
    http://ras.papercept.net/conferences/support/word.php

  It is acceptable to modify the template (e.g., one column instead of two)
- At a minimum, your proposal and progress report should include **5 references.**

Resources

Resources

- Python & MATLAB codes made available on BruinLearn
- Comprehensive C++ Repositores
  - DisMech https://github.com/StructuresComp/dismech-rods
  - Rod-Contact Sim https://github.com/StructuresComp/rod-contact-sim

Prior Works that can Motivate Project Topics

- Simulation of ribbons
  - https://youtu.be/rrlAMWtce-E?si=q8RFNx3IinWMYt5A
  - https://doi.org/10.1016/j.jmps.2020.104168
- Simulation of knots
  - https://youtu.be/yq4-m0G0D4g?si=scSaKI-zg6eHw29G
  - https://doi.org/10.1016/j.eml.2022.101924
- ML-assisted reduced order models
  - https://youtu.be/mHCFa8U9Xpw?si=ZaIrU1qJowxXBuNI
  - https://doi.org/10.1016/j.eml.2022.101925
- Simulation of soft rolling robots
  - https://youtu.be/rCVr2qRQ8rw?si=pEry_IJS0UKodY_T
  - https://doi.org/10.1038/s41467-020-15651-9
- Simulation of ballooning of spiders
  - https://youtu.be/RKnIp5ERVbE?si=x_p1B0WZVyrFT86U
  - https://doi.org/10.1103/PhysRevE.105.034401
- Simulation & control of bacteria-inspired robots
  - https://youtu.be/WbI9zL-ogM8?si=MdIgW6J1-0OdGgEM
  - https://doi.org/10.1039/C9SM01843C AND https://doi.org/10.1115/1.4049548

STRUCTURES-COMPUTER
INTERACTION LABORATORY

Module 23

Discrete Elastic Plates (DEP) Algorithm

Goal

- Equations of motion of an elastic plate
- Programming implementation of the Discrete Elastic Plates (DEP) algorithm

Resources

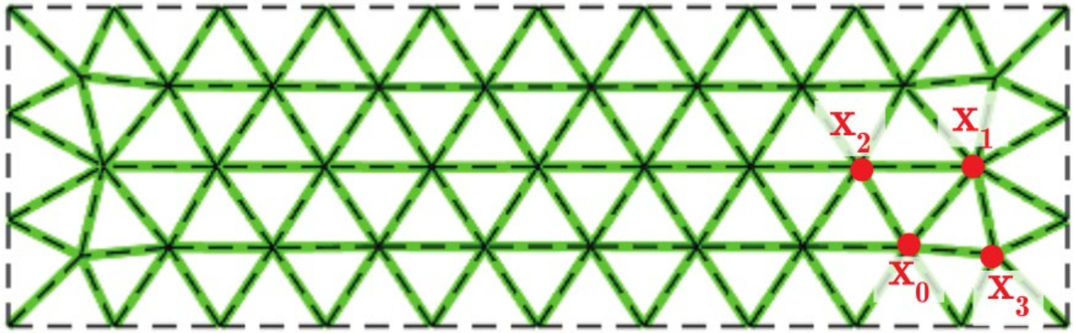
- Read *“Chapter 8: Discrete Elastic Plates and Shells”*
- Review the following papers:
  - T. Savin et al., “On the growth and form of the gut,” *Nature* (2011) 476(7358) 57–62
  - Baraff and Witkin, “Large steps in cloth simulation.” *Proceedings of the 25th annual conference on Computer graphics and interactive techniques* (1998)
  - R. Tamstorf and E. Grinspun, “Discrete bending forces and their jacobians,” *Graphical Models* (2013) 75(6) 362–370
  - Grinspun et al., “Discrete shells.” *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2003)

Degrees of Freedom (DOF)

$3N$ DOF for a plate with N nodes

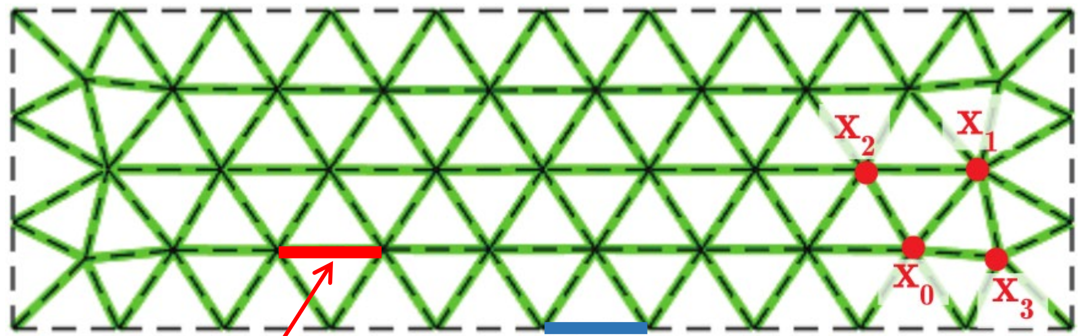
Degrees of Freedom (DOF) vector,

$$\mathbf{q} = \begin{bmatrix} \boxed{\begin{matrix} x_1 \\ y_1 \\ z_1 \end{matrix}} \\ \boxed{\begin{matrix} x_2 \\ y_2 \\ z_2 \end{matrix}} \\ \cdots \\ \cdots \\ \cdots \\ x_N \\ y_N \\ z_N \end{bmatrix} = \begin{bmatrix} \boxed{\mathbf{x}_1} \\ \boxed{\mathbf{x}_2} \\ \cdots \\ \cdots \\ \cdots \\ \mathbf{x}_N \end{bmatrix}$$



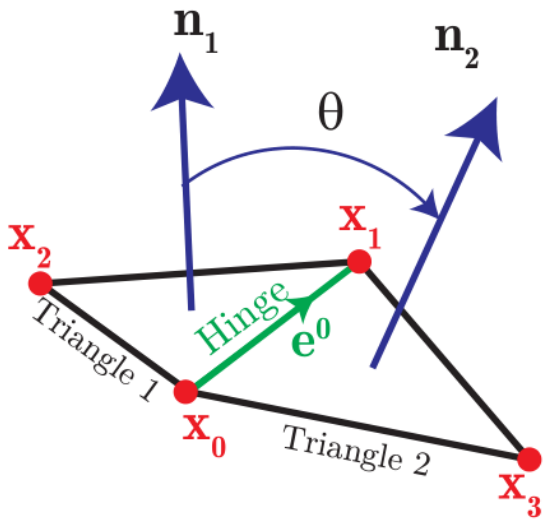
k-th node corresponds to DOF # $[3k-2, 3k-1, 3k]$
i.e. $\mathbf{x}_k = \mathbf{q}\left([3k-2, 3k-1, 3k]\right)$

Edge and Hinges



Edge AND Hinge

Edge NOT Hinge



An edge is a vector that connects on node to the next node.

A hinge is an edge is shared by two adjacent triangles.

Discrete Equations of Motion

$$\boxed{\mathbf{M}\,\ddot{\mathbf{q}} + \frac{\partial E_{\text{elastic}}}{\partial \mathbf{q}} - \mathbf{f}^{\text{ext}} = \mathbf{0}}$$

$$f_i \equiv \frac{m_i}{\Delta t}\left[\frac{q_i(t_{k+1}) - q_i(t_k)}{\Delta t} - \dot{q}_i(t_k)\right] + \boxed{\frac{\partial E_{\text{elastic}}}{\partial q_i}} - f_i^{\text{ext}} = 0$$

$$\mathbb{J}_{ij} = \frac{\partial f_i}{\partial q_j} = \mathbb{J}_{ij}^{\text{inertia}} + \boxed{\mathbb{J}_{ij}^{\text{elastic}}} + \mathbb{J}_{ij}^{\text{external}},$$

where

$$\begin{aligned}
\mathbb{J}_{ij}^{\text{inertia}} &= \frac{m_i}{\Delta t^2}\delta_{ij}, \\
\mathbb{J}_{ij}^{\text{elastic}} &= \frac{\partial^2 E_{\text{elastic}}}{\partial q_i \partial q_j}, \\
\mathbb{J}_{ij}^{\text{external}} &= -\frac{\partial f_i^{\text{ext}}}{\partial q_j}.
\end{aligned}$$

Discrete Elastic Energies

$$f_i \equiv \frac{m_i}{\Delta t}\left[\frac{q_i(t_{k+1}) - q_i(t_k)}{\Delta t} - \dot{q}_i(t_k)\right] + \boxed{\frac{\partial E_{\text{elastic}}}{\partial q_i}} - F_{i,\text{external}} = 0$$

$$\mathbb{J}_{ij} = \frac{\partial f_i}{\partial q_j} = \mathbb{J}_{ij}^{\text{inertia}} + \boxed{\mathbb{J}_{ij}^{\text{elastic}}} + \mathbb{J}_{ij}^{\text{external}},$$

We are ignoring it

$$E_{\text{elastic}} = \underbrace{\sum_{k=1}^{N_{\text{edge}}} E_{\text{s}}^k}_{\text{stretching energy}} + \underbrace{\sum_{k=1}^{N_{\text{hinge}}} E_{\text{b}}^k}_{\text{bending energy}} + \underbrace{\sum_{k=1}^{N_{\text{triangles}}} E_{\text{sh}}^k}_{\text{shearing energy}}$$

$$\mathbb{J}_{ij}^{\text{inertia}} = \frac{m_i}{\Delta t^2}\delta_{ij},$$

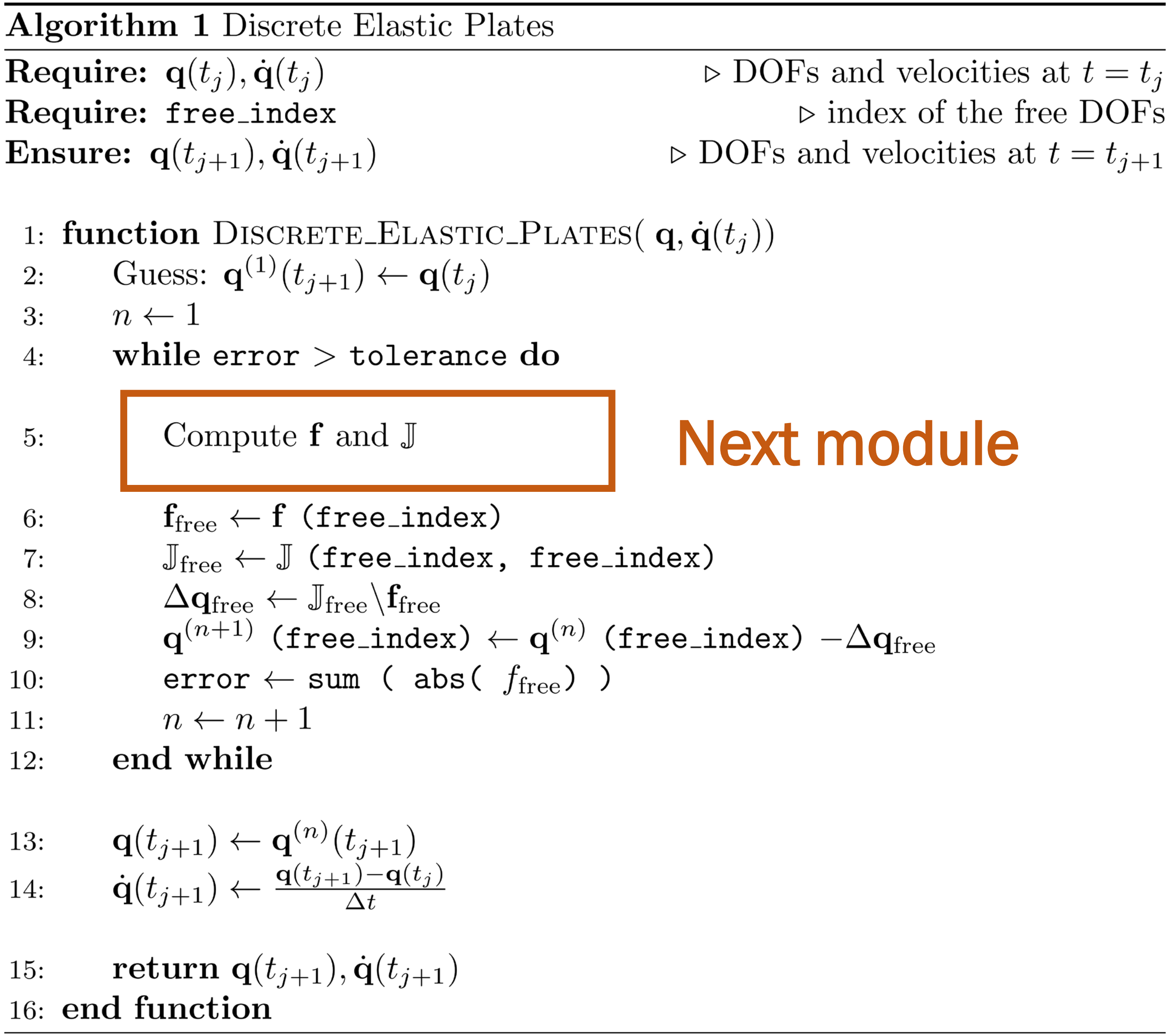
$$\mathbb{J}_{ij}^{\text{elastic}} = \frac{\partial^2 E_{\text{elastic}}}{\partial q_i \partial q_j},$$

$$\mathbb{J}_{ij}^{\text{external}} = -\frac{\partial f_i^{\text{ext}}}{\partial q_j}.$$

$$\boxed{\frac{\partial}{\partial q_i} E_{\text{elastic}} = \sum_{k=1}^{N_{\text{edge}}} \frac{\partial}{\partial q_i} E_s^k + \sum_{k=1}^{N_{\text{hinge}}} \frac{\partial}{\partial q_i} E_b^k}$$

$$\boxed{\frac{\partial^2}{\partial q_i \partial q_j} E_{\text{elastic}} = \sum_{k=1}^{N_{\text{edge}}} \frac{\partial^2}{\partial q_i \partial q_j} E_s^k + \sum_{k=1}^{N_{\text{hinge}}} \frac{\partial^2}{\partial q_i \partial q_j} E_b^k}$$

Discrete Elastic Plates (DEP)

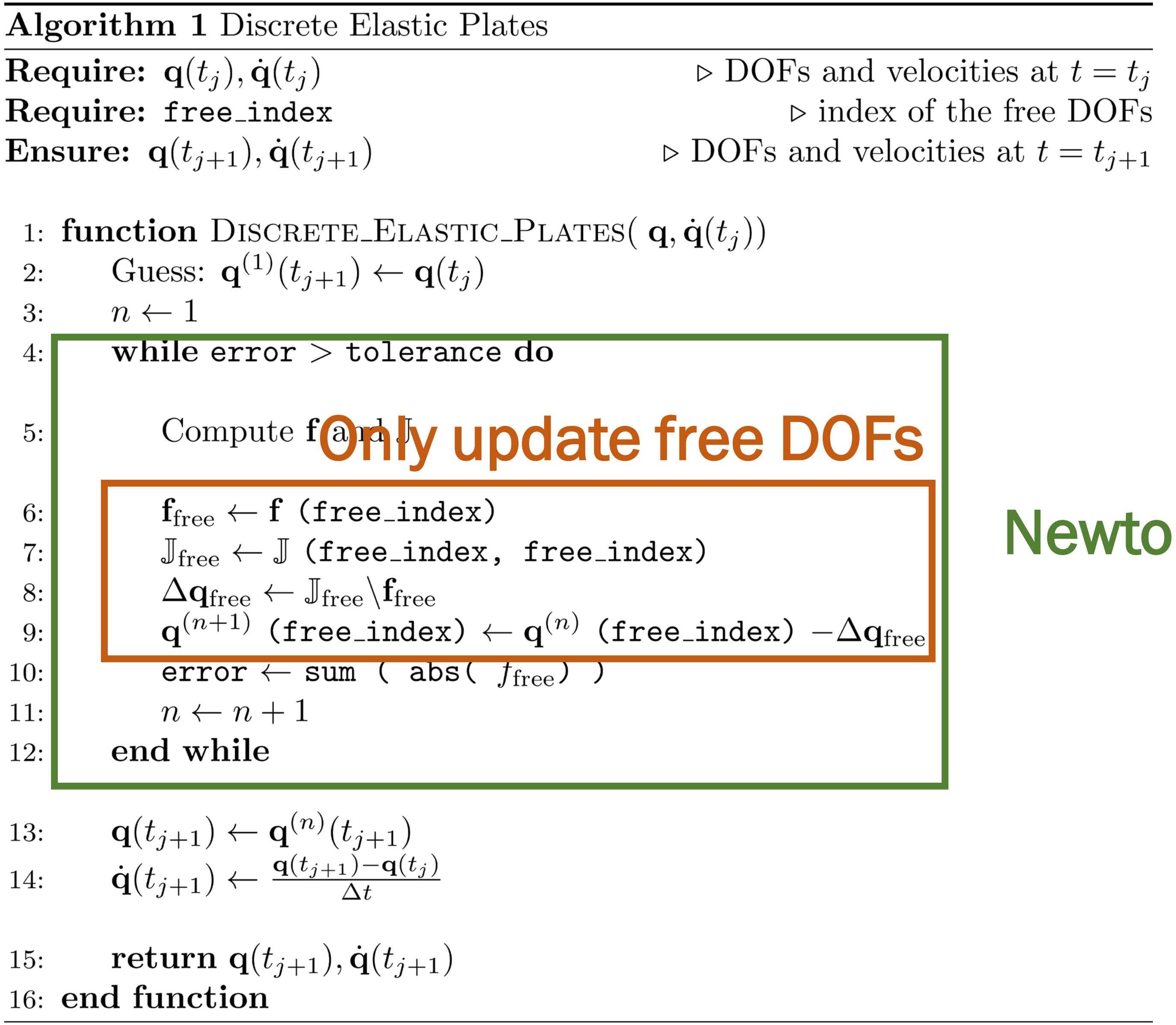


Algorithm 1 Discrete Elastic Plates

```
Require: $\mathbf{q}(t_j), \dot{\mathbf{q}}(t_j)$        ▷ DOFs and velocities at $t = t_j$
Require: free_index                                       ▷ index of the free DOFs
Ensure: $\mathbf{q}(t_{j+1}), \dot{\mathbf{q}}(t_{j+1})$  ▷ DOFs and velocities at $t = t_{j+1}$

function DISCRETE_ELASTIC_PLATES( $\mathbf{q}, \dot{\mathbf{q}}(t_j)$)
    Guess: $\mathbf{q}^{(1)}(t_{j+1}) \leftarrow \mathbf{q}(t_j)$
    $n \leftarrow 1$
    while error > tolerance do
        $\boxed{\text{Compute } \mathbf{f} \text{ and } \mathbb{J}}$
        $\mathbf{f}_{\text{free}} \leftarrow \mathbf{f}$ (free_index)
        $\mathbb{J}_{\text{free}} \leftarrow \mathbb{J}$ (free_index, free_index)
        $\Delta\mathbf{q}_{\text{free}} \leftarrow \mathbb{J}_{\text{free}} \backslash \mathbf{f}_{\text{free}}$
        $\mathbf{q}^{(n+1)}$ (free_index) $\leftarrow \mathbf{q}^{(n)}$ (free_index) $-\Delta\mathbf{q}_{\text{free}}$
        error $\leftarrow$ sum ( abs( $f_{\text{free}}$) )
        $n \leftarrow n + 1$
    end while

    $\mathbf{q}(t_{j+1}) \leftarrow \mathbf{q}^{(n)}(t_{j+1})$
    $\dot{\mathbf{q}}(t_{j+1}) \leftarrow \frac{\mathbf{q}(t_{j+1}) - \mathbf{q}(t_j)}{\Delta t}$

    return $\mathbf{q}(t_{j+1}), \dot{\mathbf{q}}(t_{j+1})$
end function
```

Next module

Discrete Elastic Plates (DEP)



Algorithm 1 Discrete Elastic Plates

```
Require: $\mathbf{q}(t_j), \dot{\mathbf{q}}(t_j)$                ▷ DOFs and velocities at $t = t_j$
Require: free_index                                ▷ index of the free DOFs
Ensure: $\mathbf{q}(t_{j+1}), \dot{\mathbf{q}}(t_{j+1})$          ▷ DOFs and velocities at $t = t_{j+1}$

function DISCRETE_ELASTIC_PLATES( $\mathbf{q}, \dot{\mathbf{q}}(t_j)$)
    Guess: $\mathbf{q}^{(1)}(t_{j+1}) \leftarrow \mathbf{q}(t_j)$
    $n \leftarrow 1$
    while error > tolerance do
        Compute $\mathbf{f}$ and $\mathbb{J}$
        $\mathbf{f}_{\text{free}} \leftarrow \mathbf{f}$ (free_index)
        $\mathbb{J}_{\text{free}} \leftarrow \mathbb{J}$ (free_index, free_index)
        $\Delta\mathbf{q}_{\text{free}} \leftarrow \mathbb{J}_{\text{free}} \backslash \mathbf{f}_{\text{free}}$
        $\mathbf{q}^{(n+1)}$ (free_index) $\leftarrow \mathbf{q}^{(n)}$ (free_index) $-\Delta\mathbf{q}_{\text{free}}$
        error ← sum ( abs( $f_{\text{free}}$) )
        $n \leftarrow n + 1$
    end while

    $\mathbf{q}(t_{j+1}) \leftarrow \mathbf{q}^{(n)}(t_{j+1})$
    $\dot{\mathbf{q}}(t_{j+1}) \leftarrow \frac{\mathbf{q}(t_{j+1}) - \mathbf{q}(t_j)}{\Delta t}$

    return $\mathbf{q}(t_{j+1}), \dot{\mathbf{q}}(t_{j+1})$
end function
```

Only update free DOFs

Newton-Raphson

Next Step

- Programming implementation of gradient and Hessian of elastic energies

$$\boxed{\frac{\partial}{\partial q_i} E_{\text{elastic}} = \sum_{k=1}^{N_{\text{edge}}} \frac{\partial}{\partial q_i} E_s^k + \sum_{k=1}^{N_{\text{hinge}}} \frac{\partial}{\partial q_i} E_b^k}$$

$$\boxed{\frac{\partial^2}{\partial q_i \partial q_j} E_{\text{elastic}} = \sum_{k=1}^{N_{\text{edge}}} \frac{\partial^2}{\partial q_i \partial q_j} E_s^k + \sum_{k=1}^{N_{\text{hinge}}} \frac{\partial^2}{\partial q_i \partial q_j} E_b^k}$$

STRUCTURES-COMPUTER
INTERACTION LABORATORY

Module 24

Gradient and Hessian of Elastic Energies in Discrete Plate

- Programming Implementation

Goal

- Understand the computation of gradient and Hessian of elastic energies in Discrete Elastic Plates (DEP)

$$\boxed{\frac{\partial}{\partial q_i} E_{\text{elastic}} = \sum_{k=1}^{N_{\text{edge}}} \frac{\partial}{\partial q_i} E_s^k + \sum_{k=1}^{N_{\text{hinge}}} \frac{\partial}{\partial q_i} E_b^k}$$

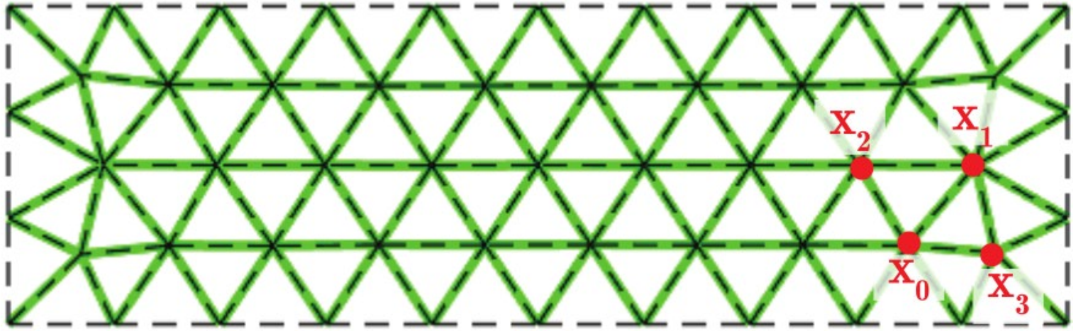
$$\boxed{\frac{\partial^2}{\partial q_i \partial q_j} E_{\text{elastic}} = \sum_{k=1}^{N_{\text{edge}}} \frac{\partial^2}{\partial q_i \partial q_j} E_s^k + \sum_{k=1}^{N_{\text{hinge}}} \frac{\partial^2}{\partial q_i \partial q_j} E_b^k}$$

Resources

- Read *“Chapter 8: Discrete Elastic Plates and Shells”*
- Review the following paper:
  - R. Tamstorf and E. Grinspun, “Discrete bending forces and their jacobians,” *Graphical Models* (2013) 75(6) 362–370

Degrees of Freedom (DOF)

$3N$ DOF for a plate with N nodes



Degrees of Freedom (DOF) vector,

$$\mathbf{q} = \begin{bmatrix} \boxed{\begin{matrix} x_1 \\ y_1 \\ z_1 \end{matrix}} \\ \boxed{\begin{matrix} x_2 \\ y_2 \\ z_2 \end{matrix}} \\ \dots \\ \dots \\ \dots \\ x_N \\ y_N \\ z_N \end{bmatrix} = \begin{bmatrix} \boxed{\mathbf{x}_1} \\ \boxed{\mathbf{x}_2} \\ \dots \\ \dots \\ \dots \\ \mathbf{x}_N \end{bmatrix}$$

Index of the k-th node, $\mathbf{x}_k$, in DOF vector:
$3k - 2, 3k - 1, 3k$, where $1 \leq k \leq N$

We will need this during programming implementation.

Discrete Elastic Energies

$$f_i \equiv \frac{m_i}{\Delta t}\left[\frac{q_i(t_{k+1}) - q_i(t_k)}{\Delta t} - \dot{q}_i(t_k)\right] + \boxed{\frac{\partial E_{\text{elastic}}}{\partial q_i}} - F_{i,\text{external}} = 0$$

$$\mathbb{J}_{ij} = \frac{\partial f_i}{\partial q_j} = \mathbb{J}^{\text{inertia}}_{ij} + \boxed{\mathbb{J}^{\text{elastic}}_{ij}} + \mathbb{J}^{\text{external}}_{ij},$$

$$\mathbb{J}^{\text{inertia}}_{ij} = \frac{m_i}{\Delta t^2}\delta_{ij},$$

$$\mathbb{J}^{\text{elastic}}_{ij} = \frac{\partial^2 E_{\text{elastic}}}{\partial q_i \partial q_j},$$

$$\mathbb{J}^{\text{external}}_{ij} = -\frac{\partial f^{\text{ext}}_i}{\partial q_j}.$$

$$E_{\text{elastic}} = \underbrace{\sum_{k=1}^{N_{\text{edge}}} E^k_{\text{s}}}_{\text{stretching energy}} + \underbrace{\sum_{k=1}^{N_{\text{hinge}}} E^k_{\text{b}}}_{\text{bending energy}}$$

$$\boxed{\frac{\partial}{\partial q_i} E_{\text{elastic}} = \sum_{k=1}^{N_{\text{edge}}} \frac{\partial}{\partial q_i} E^k_s + \sum_{k=1}^{N_{\text{hinge}}} \frac{\partial}{\partial q_i} E^k_b}$$

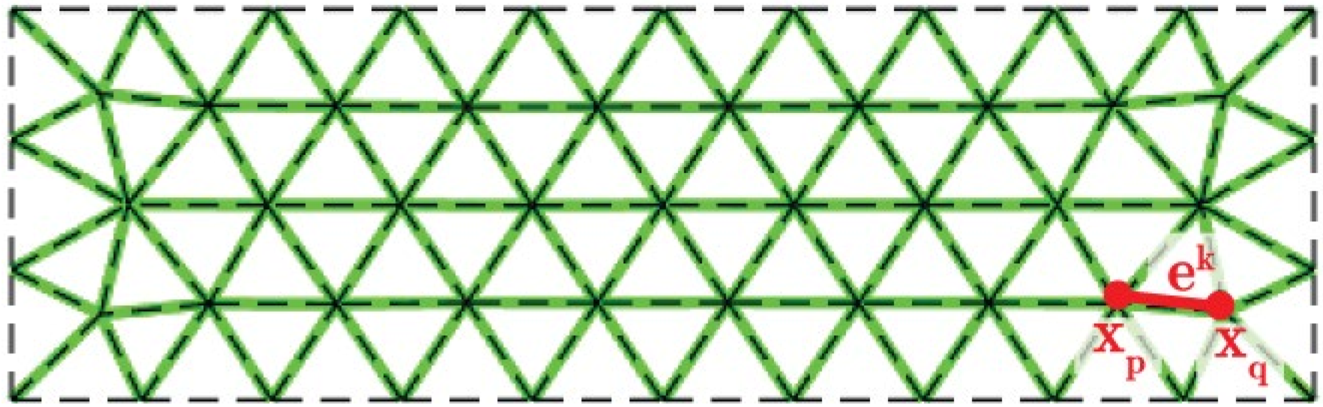
$$\boxed{\frac{\partial^2}{\partial q_i \partial q_j} E_{\text{elastic}} = \sum_{k=1}^{N_{\text{edge}}} \frac{\partial^2}{\partial q_i \partial q_j} E^k_s + \sum_{k=1}^{N_{\text{hinge}}} \frac{\partial^2}{\partial q_i \partial q_j} E^k_b}$$

Gradient of Elastic Energy Computation

$$\frac{\partial}{\partial q_i} E_{\text{elastic}} = \sum_{k=1}^{N_{\text{edge}}} \boxed{\frac{\partial}{\partial q_i} E_s^k} + \sum_{k=1}^{N_{\text{hinge}}} \frac{\partial}{\partial q_i} E_b^k$$

$$\mathbf{F} \equiv \frac{\partial}{\partial q_i} \rightarrow 3N \text{ sized vector}$$

“Big gradient vector” F



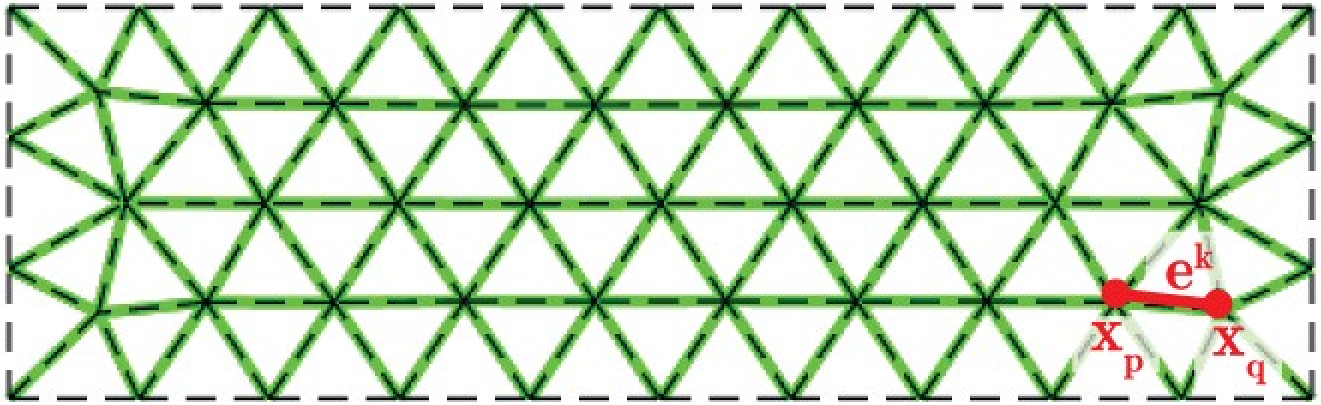
$$E_{\text{s}}^k = \frac{1}{2} k_s \left(\frac{\|\mathbf{e}^k\|}{l_k} - 1 \right)^2$$

Stretching stiffness, $k_s = \frac{\sqrt{3}}{2} Y h l_k^2$
Edge length, $\|\mathbf{e}^k\| = \|\mathbf{x}_q - \mathbf{x}_p\|$
l_k is the edge length in undeformed state
Y is Young’s modulus
h is the plate thickness

Gradient of Elastic Energy Computation

$$\frac{\partial}{\partial q_i} E_{\text{elastic}} = \sum_{k=1}^{N_{\text{edge}}} \boxed{\frac{\partial}{\partial q_i} E_s^k} + \sum_{k=1}^{N_{\text{hinge}}} \frac{\partial}{\partial q_i} E_b^k$$

$$\mathtt{F} \equiv \frac{\partial}{\partial q_i} \rightarrow 3N \text{ sized vector}$$

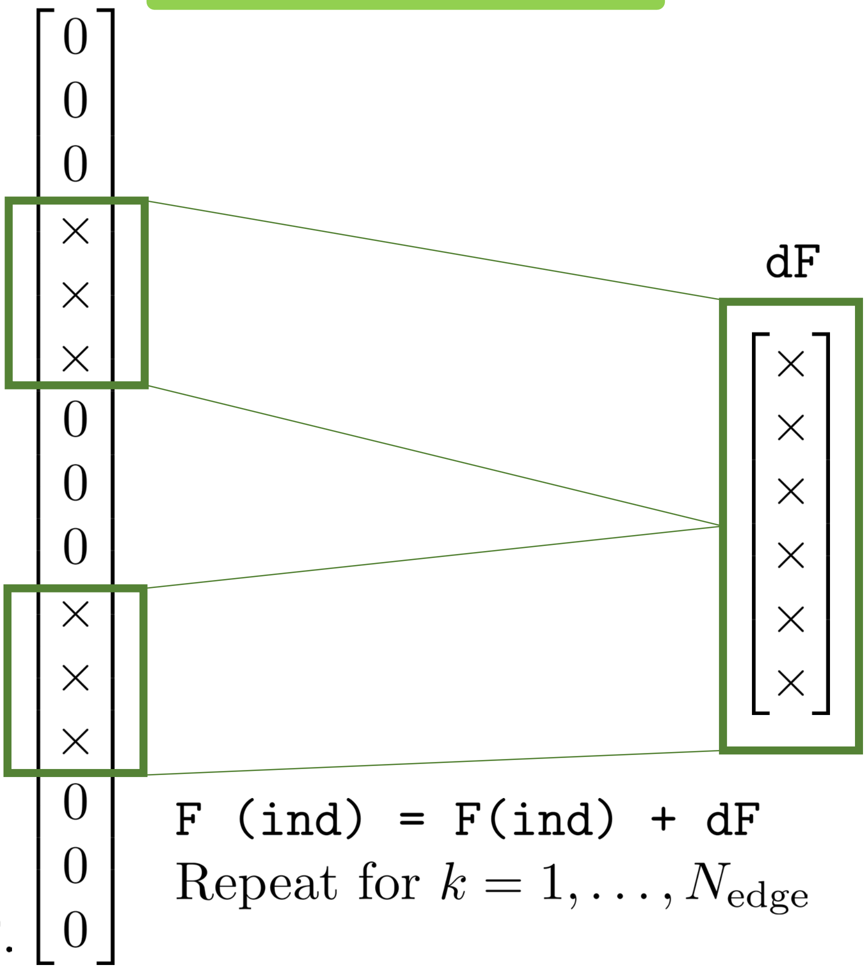


Stretching energy, E_k^s, only depends on $\mathbf{x}_p$ and $\mathbf{x}_q$.

The corresponding indices in the DOF vector are:
$\mathtt{ind} = [3p-2, 3p-1, 3p, 3q-2, 3q-1, 3q]$

$\frac{\partial}{\partial q_i} E_s^k$ has 6 non-zero elements.
See Appendix for codes to compute non-zero “small” gradient, $\mathtt{dF}$.

"Big gradient vector" F



F (ind) = F(ind) + dF

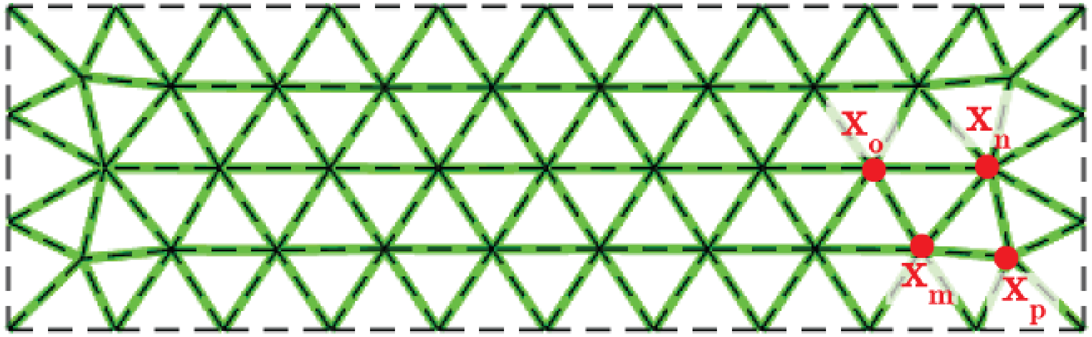
Repeat for $k = 1, \ldots, N_{\text{edge}}$

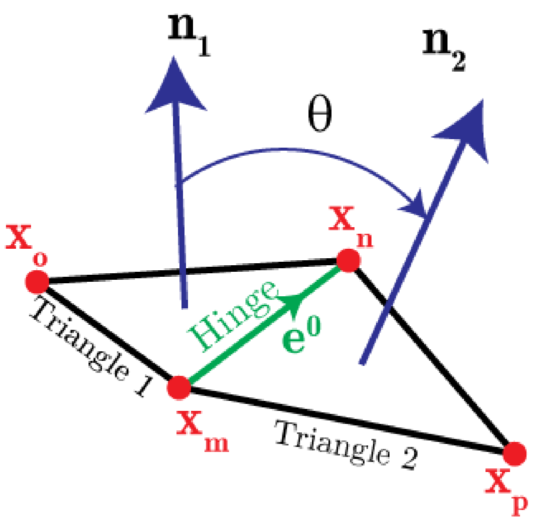
Gradient of Elastic Energy Computation

$$\frac{\partial}{\partial q_i} E_{\text{elastic}} = \sum_{k=1}^{N_{\text{edge}}} \frac{\partial}{\partial q_i} E_s^k + \sum_{k=1}^{N_{\text{hinge}}} \boxed{\frac{\partial}{\partial q_i} E_b^k}$$

$$\mathbf{F} \equiv \frac{\partial}{\partial q_i} \rightarrow 3N \text{ sized vector}$$

“Big gradient vector” F





$$E_{\text{b}}^k = \tfrac{1}{2} k_b \theta^2$$

Bending stiffness, $k_b = \frac{2}{\sqrt{3}} \frac{Y h^3}{12}$

Y is Young’s modulus

h is the plate thickness

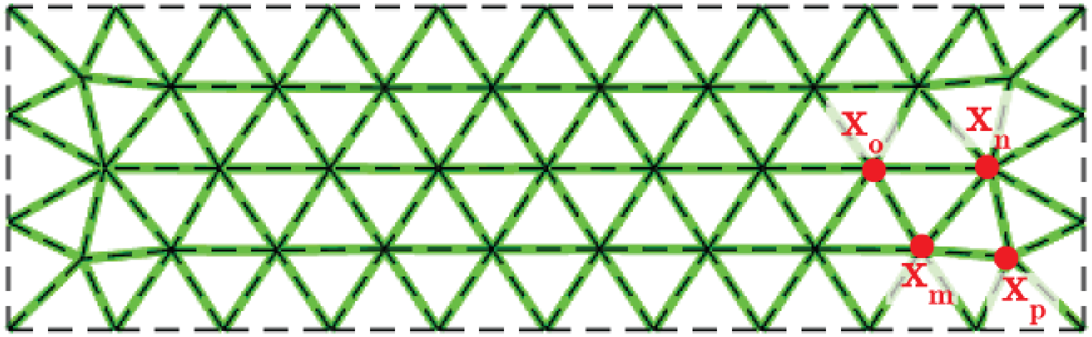
θ is a function of $\mathbf{x}_m, \mathbf{x}_n, \mathbf{x}_o$, and $\mathbf{x}_p$

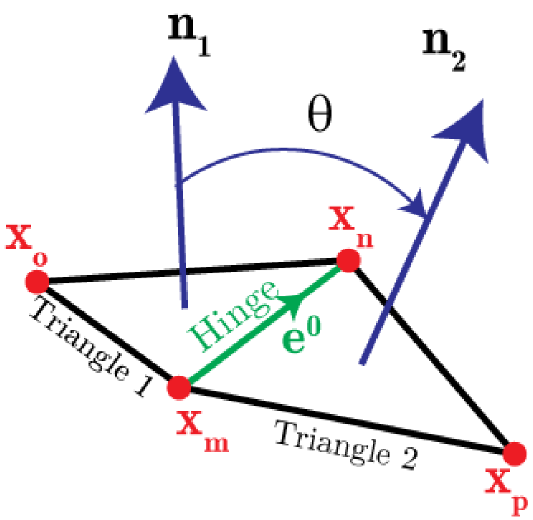
Gradient of Elastic Energy Computation

$$\frac{\partial}{\partial q_i} E_{\text{elastic}} = \sum_{k=1}^{N_{\text{edge}}} \frac{\partial}{\partial q_i} E_s^k + \sum_{k=1}^{N_{\text{hinge}}} \boxed{\frac{\partial}{\partial q_i} E_b^k}$$

$$\mathtt{F} \equiv \frac{\partial}{\partial q_i} \rightarrow 3N \text{ sized vector}$$

“Big gradient vector” F





Bending energy, E_k^b, only depends on $\mathbf{x}_m, \mathbf{x}_n, \mathbf{x}_o$, and $\mathbf{x}_p$.

The corresponding indices in the DOF vector are:

$\mathtt{ind} = [3m-2, 3m-1, 3m, 3n-2, 3n-1, 3n,$
$3o-2, 3o-1, 3o, 3p-2, 3p-1, 3p]$.

$\frac{\partial}{\partial q_i} E_b^k$ has 12 non-zero elements.

See Appendix for codes to compute non-zero “small” gradient, `dF`.

```
F (ind) = F(ind) + dF
```

Repeat for $k = 1, \ldots, N_{\text{hinge}}$

Hessian of Elastic Energy Computation

$$\frac{\partial^2}{\partial q_i \partial q_j} E_{\text{elastic}} = \sum_{k=1}^{N_{\text{edge}}} \frac{\partial^2}{\partial q_i \partial q_j} E_s^k + \sum_{k=1}^{N_{\text{hinge}}} \frac{\partial^2}{\partial q_i \partial q_j} E_b^k$$

$$\mathsf{J} \equiv \frac{\partial}{\partial q_i \partial q_j} E_{\text{elastic}} \rightarrow 3N \times 3N \text{ sized matrix}$$

"Big Hessian matrix" J

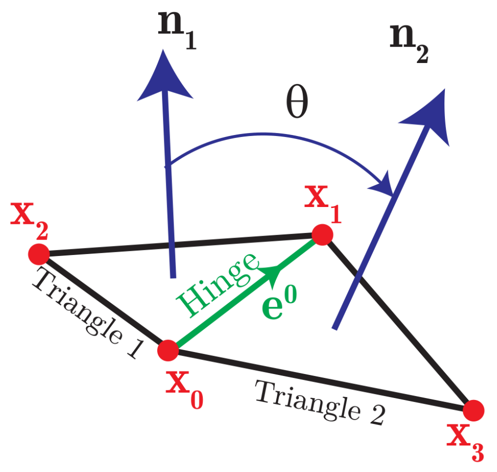
Non-zero component of Hessian of discrete energies
Code in Appendix of the Course Notes

```
F (ind) = F(ind) + dF
```

Repeat for $k = 1, \ldots, N_{\text{edge}}$ (stretching)
Repeat for $k = 1, \ldots, N_{\text{hinge}}$ (bending)

```
J (ind, ind) = J(ind, ind) + dJ
```

Repeat for $k = 1, \ldots, N_{\text{edge}}$ (stretching)
Repeat for $k = 1, \ldots, N_{\text{hinge}}$ (bending)



Algorithm 1 Gradient and Hessian of Elastic Energy Calculation

```
Require: $\mathbf{q}$                                   ▷ Degrees of Freedom
Ensure: F               ▷ $3N$ sized elastic gradient vector, $\frac{\partial E_{\text{elastic}}}{\partial q_i}$
Ensure: J               ▷ $3N \times 3N$ sized elastic Hessian matrix, $\mathbb{J}^{\text{elastic}}$

function GRAD_HESS_ELASTIC($\mathbf{q}$)
    F ← zeros(3N,1)
    J ← zeros(3N,3N)
    for $k \leftarrow 1$ to $N_{\text{edge}}$ do
        $\mathbf{x}_0$ ← nodal coordinates of first node of the $k$-th edge
        $\mathbf{x}_1$ ← nodal coordinates of second node of the $k$-th edge
        ind ← locations of the two nodes in the DOF vector
        [dF, dJ] ← gradEs_hessEs_Shell($\mathbf{x}_0, \mathbf{x}_1$)        ▷ Appendix
        F (ind) = F (ind) + dF
        J (ind) = J (ind, ind) + dJ
    end for

    for $k \leftarrow 1$ to $N_{\text{hinge}}$ do
        $\mathbf{x}_0$ ← nodal coordinates of first node of the $k$-th hinge
        $\mathbf{x}_1$ ← nodal coordinates of second node of the $k$-th hinge
        $\mathbf{x}_2$ ← nodal coordinates of the third node on Triangle 1
        $\mathbf{x}_3$ ← nodal coordinates of the third node on Triangle 2
        ind ← locations of the four nodes in the DOF vector
        [dF, dJ] ← gradEb_hessEb_Shell($\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$)   ▷ Appendix
        F (ind) = F (ind) + dF
        J (ind) = F (ind, ind) + dJ
    end for
    return F and J
end function
```

Stretching Energy

Bending Energy

Assignments

- Complete the assignments in Chapter 8: *Discrete Elastic Plates and Shells*

STRUCTURES-COMPUTER
INTERACTION LABORATORY

Module 25

Discrete Elastic Shells (DES) Algorithm

Goal

- Augment Discrete Elastic Plate (DEP) simulation to include shell-like structures
  - In undeformed configuration, shells have curvature. Plates do not have curvature.
  - All plates are shells but all shells are not plates.

Resources

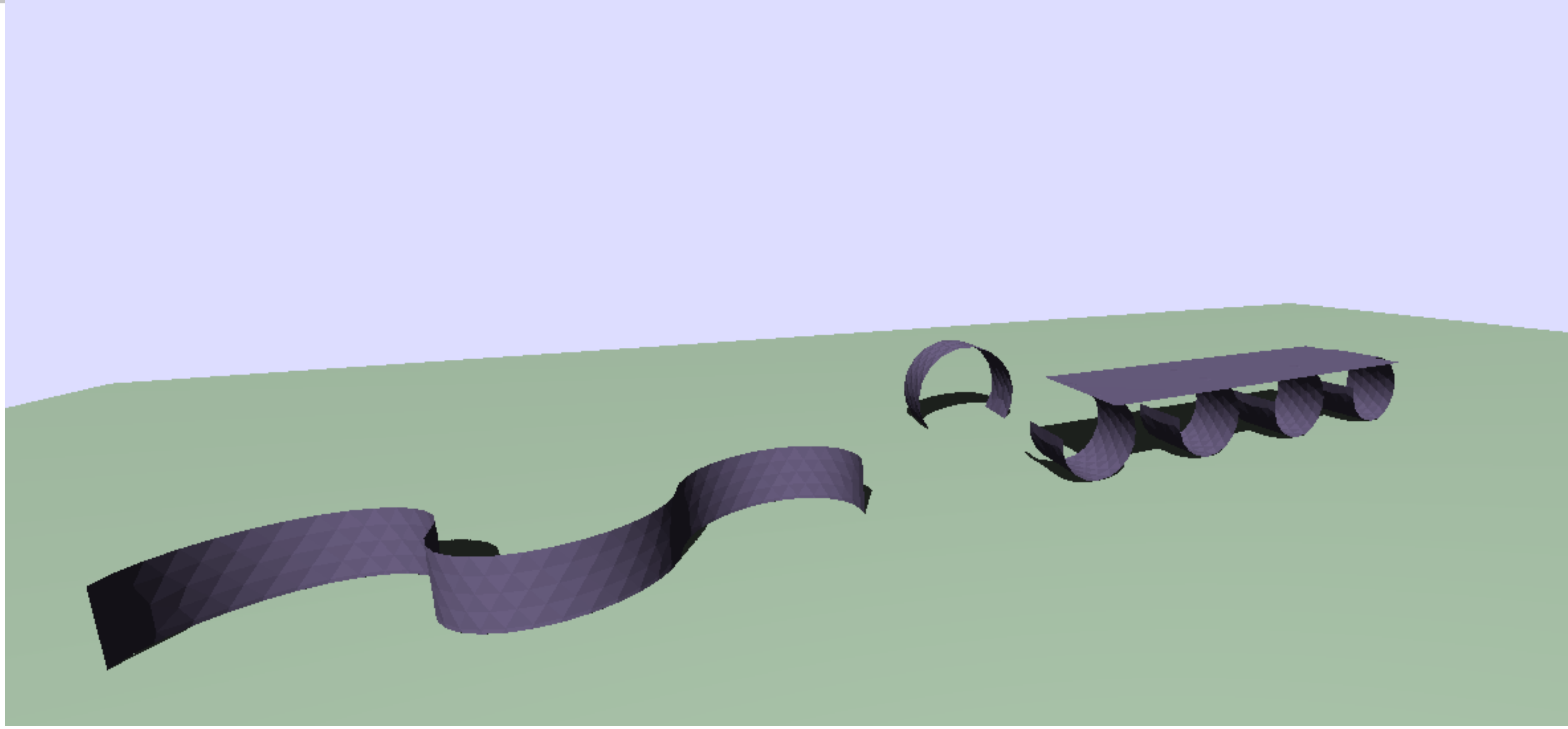
- Read *“Chapter 8: Discrete Elastic Plates and Shells”*
- Review the following papers:
  - Grinspun et al., “Discrete shells.” *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2003)
  - R. Tamstorf and E. Grinspun, “Discrete bending forces and their jacobians,” *Graphical Models* (2013) 75(6) 362–370

Degrees of Freedom (DOF)

$3N$ DOF for a shell with N nodes

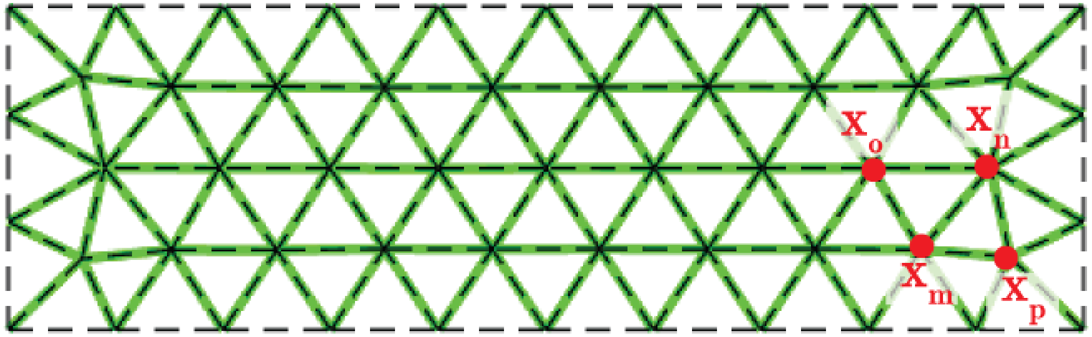
Degrees of Freedom (DOF) vector,

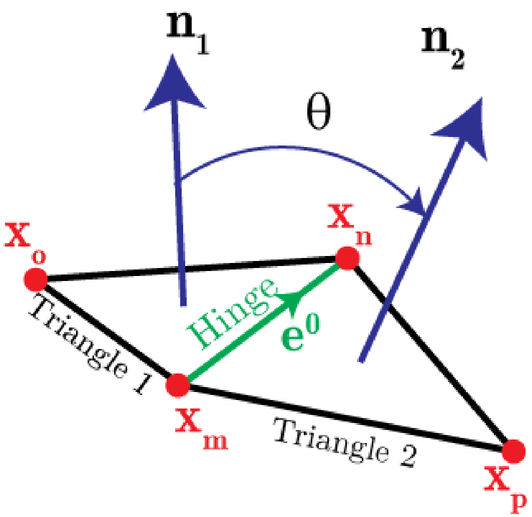
$$\mathbf{q} = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ x_2 \\ y_2 \\ z_2 \\ \dots \\ \dots \\ \dots \\ x_N \\ y_N \\ z_N \end{bmatrix} = \begin{bmatrix} \boxed{\mathbf{x}_1} \\ \boxed{\mathbf{x}_2} \\ \dots \\ \dots \\ \dots \\ \mathbf{x}_N \end{bmatrix}$$



k-th node corresponds to DOF # $[3k-2, 3k-1, 3k]$
i.e. $\mathbf{x}_k = \mathbf{q}\left([3k-2, 3k-1, 3k]\right)$

Only difference between discrete plate and discrete shell





$$E_{\mathrm{b}}^{k} = \frac{1}{2} k_b \theta^2$$

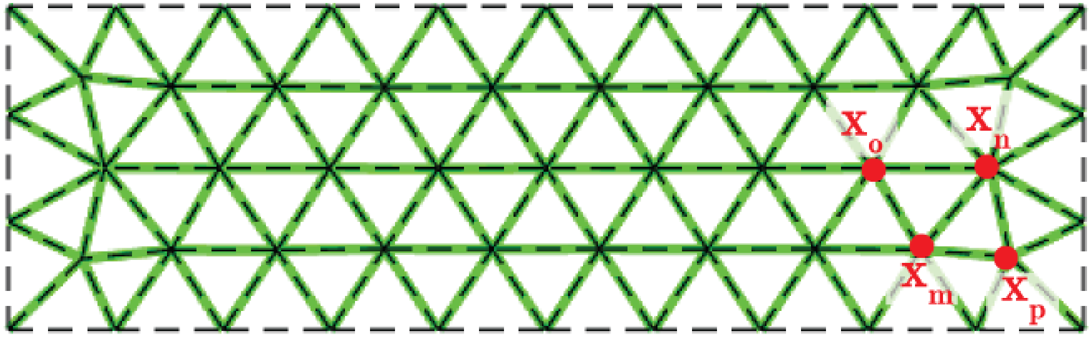
Bending stiffness, $k_b = \frac{2}{\sqrt{3}} \frac{Y h^3}{12}$

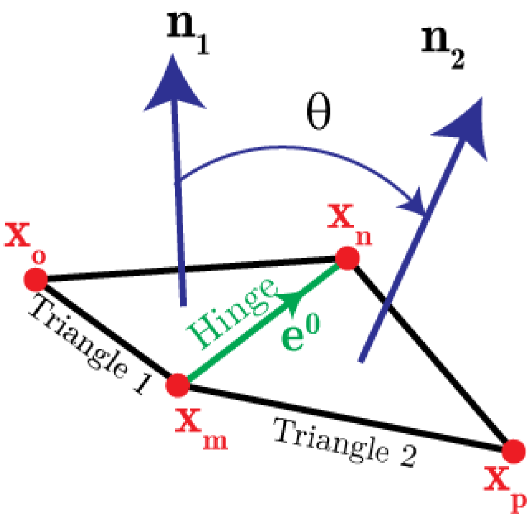
Y is Young's modulus

h is the plate thickness

θ is a function of $\mathbf{x}_m, \mathbf{x}_n, \mathbf{x}_o$, and $\mathbf{x}_p$

Only difference between discrete plate and discrete shell





$$E_{\mathrm{b}}^k = \frac{1}{2} k_b \theta^2$$

Bending stiffness, $k_b = \frac{2}{\sqrt{3}} \frac{Y h^3}{12}$

Y is Young's modulus

h is the plate thickness

θ is a function of $\mathbf{x}_m, \mathbf{x}_n, \mathbf{x}_o$, and $\mathbf{x}_p$

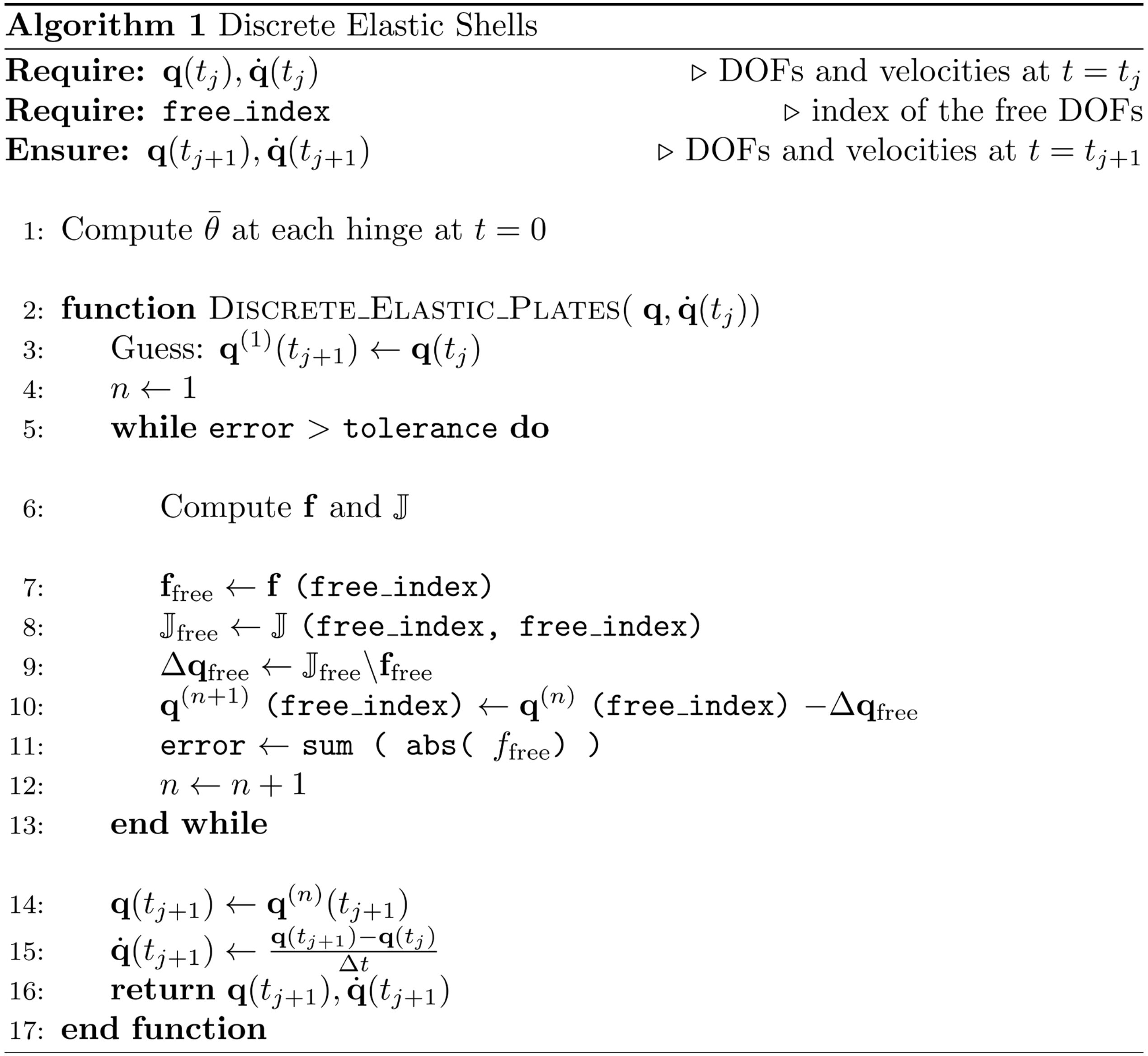
$$E_{\mathrm{b}}^k = \frac{1}{2} k_b \left(\theta - \bar{\theta}\right)^2$$

$\bar{\theta}$ is the hinge angle in undeformed configuration ($\bar{\theta} = 0$ for a plate)

Be careful about the sign. Hinge angle θ is a signed angle from $\mathbf{n}_1$ to $\mathbf{n}_2$ about the hinge vector

Discrete Elastic Shells (DES)

STRUCTURES-COMPUTER INTERACTION LABORATORY



Algorithm 1 Discrete Elastic Shells

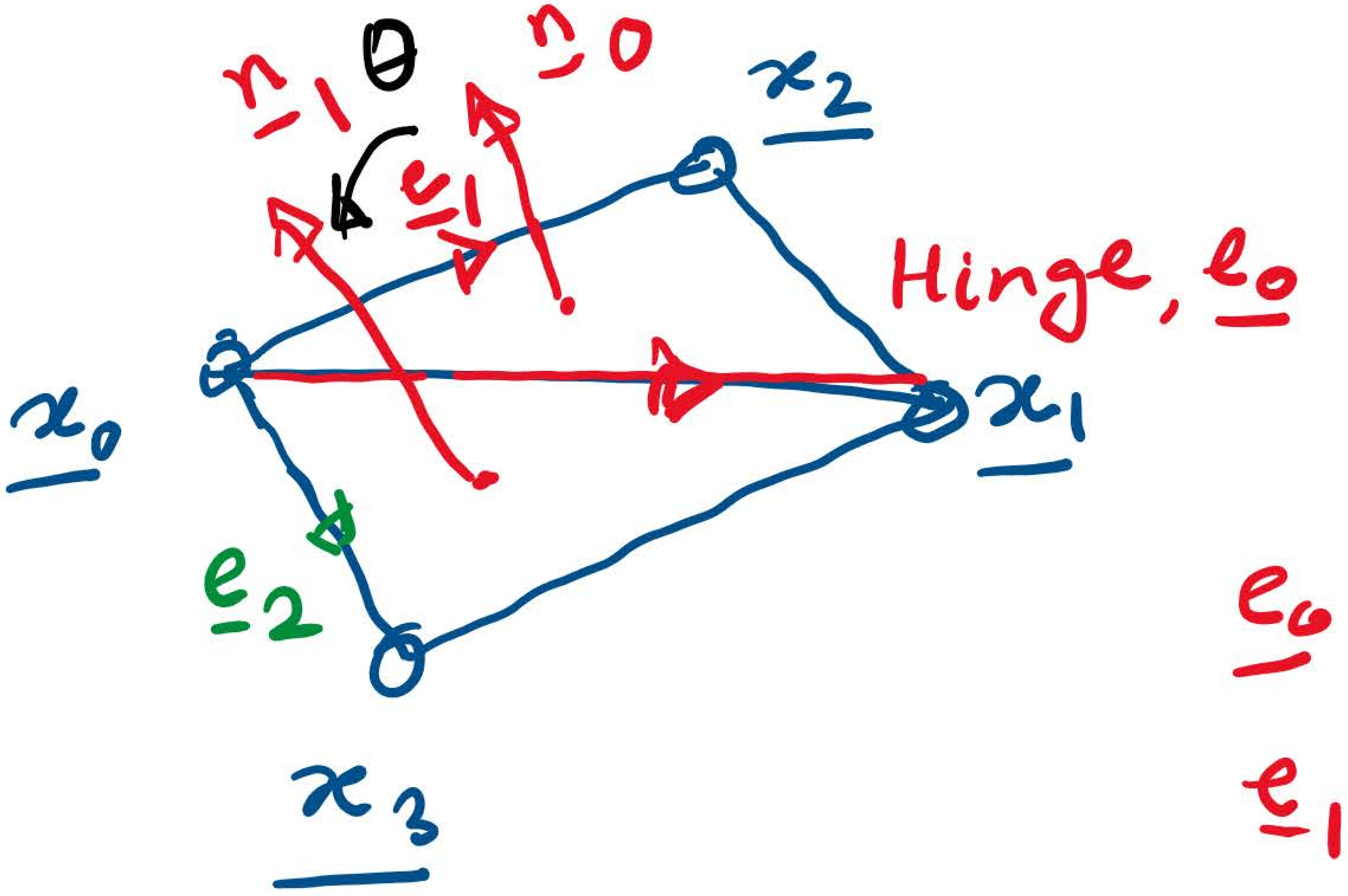
```
Require: $\mathbf{q}(t_j), \dot{\mathbf{q}}(t_j)$                ▷ DOFs and velocities at $t = t_j$
Require: free_index                                ▷ index of the free DOFs
Ensure: $\mathbf{q}(t_{j+1}), \dot{\mathbf{q}}(t_{j+1})$          ▷ DOFs and velocities at $t = t_{j+1}$

Compute $\bar{\theta}$ at each hinge at $t = 0$

function DISCRETE_ELASTIC_PLATES( $\mathbf{q}, \dot{\mathbf{q}}(t_j)$)
    Guess: $\mathbf{q}^{(1)}(t_{j+1}) \leftarrow \mathbf{q}(t_j)$
    $n \leftarrow 1$
    while error $>$ tolerance do

        Compute $\mathbf{f}$ and $\mathbb{J}$

        $\mathbf{f}_{\text{free}} \leftarrow \mathbf{f}$ (free_index)
        $\mathbb{J}_{\text{free}} \leftarrow \mathbb{J}$ (free_index, free_index)
        $\Delta\mathbf{q}_{\text{free}} \leftarrow \mathbb{J}_{\text{free}} \backslash \mathbf{f}_{\text{free}}$
        $\mathbf{q}^{(n+1)}$ (free_index) $\leftarrow \mathbf{q}^{(n)}$ (free_index) $-\Delta\mathbf{q}_{\text{free}}$
        error $\leftarrow$ sum ( abs( $f_{\text{free}}$) )
        $n \leftarrow n + 1$
    end while

    $\mathbf{q}(t_{j+1}) \leftarrow \mathbf{q}^{(n)}(t_{j+1})$
    $\dot{\mathbf{q}}(t_{j+1}) \leftarrow \frac{\mathbf{q}(t_{j+1}) - \mathbf{q}(t_j)}{\Delta t}$
    return $\mathbf{q}(t_{j+1}), \dot{\mathbf{q}}(t_{j+1})$
end function
```

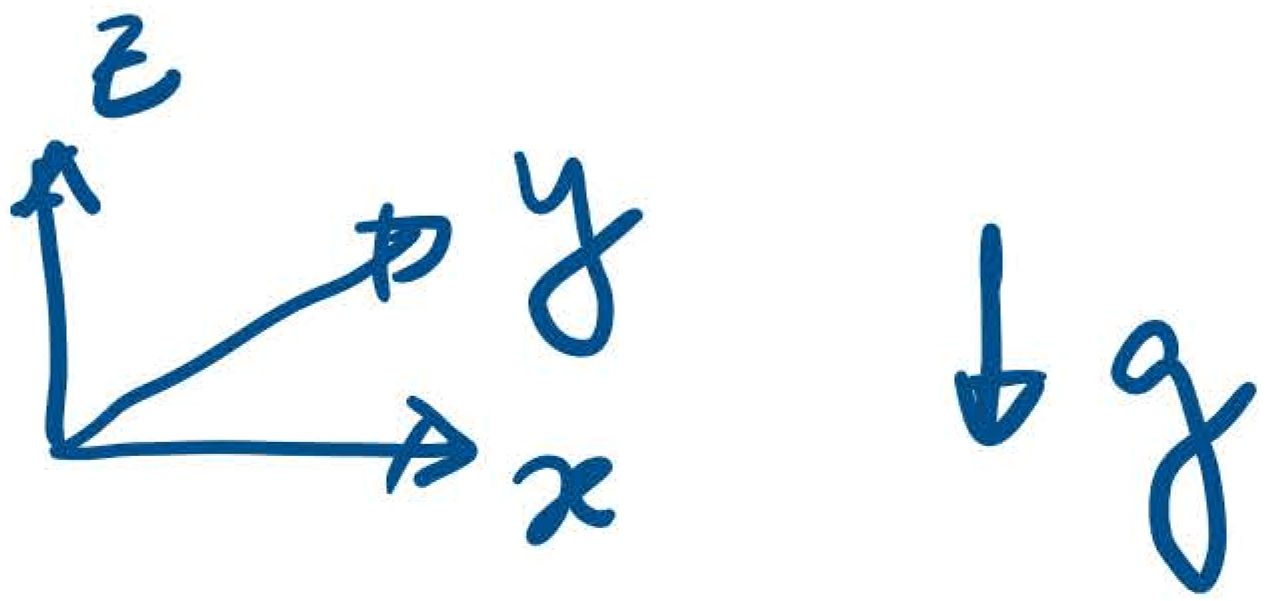
Next Step

1. Incorporate external forces, e.g. hydrodynamics, into the simulation
2. Simulate contact and collision
3. Include kinematic constraints, e.g. rigid bodies
4. Model complex systems consisting of beams, rods, plates, shells, and rigid bodies

Lecture 9

Monday, October 28, 2024 12:03 PM





4 nodes

$$\left.\begin{aligned}\underline{e}_0 &= \underline{x}_1 - x_0 \quad \text{(Hinge)}\\ \underline{e}_1 &= \underline{x}_2 - x_0\\ \underline{e}_2 &= \underline{x}_3 - \underline{x}_0\end{aligned}\right\} \text{3 edges}$$

$$\left.\begin{aligned}\underline{n}_0 &= \underline{e}_0 \times \underline{e}_1\\ \underline{n}_1 &= \underline{e}_2 \times \underline{e}_0\end{aligned}\right\} \text{may normalize to make it unit}$$

$\theta = \text{signedAngle}(\underline{n}_0, \underline{n}_1, \underline{e}_0)$

from to about

$\text{getTheta}\left(\begin{matrix}\text{12 numbers}\\ \underline{x}_0, \underline{x}_1, \underline{x}_2, \underline{x}_3\end{matrix}\right)$

gradTheta : $\nabla \theta$ [12 array]

hessTheta $\nabla^2 \theta$ [12x12 array]

$$E_b^i = \frac{1}{2} k_b \boxed{(\theta - \bar{\theta})^2} \xrightarrow{} (\tan(\theta - \bar{\theta}))^2$$

$\bar{\theta} = 0$ (Plate)

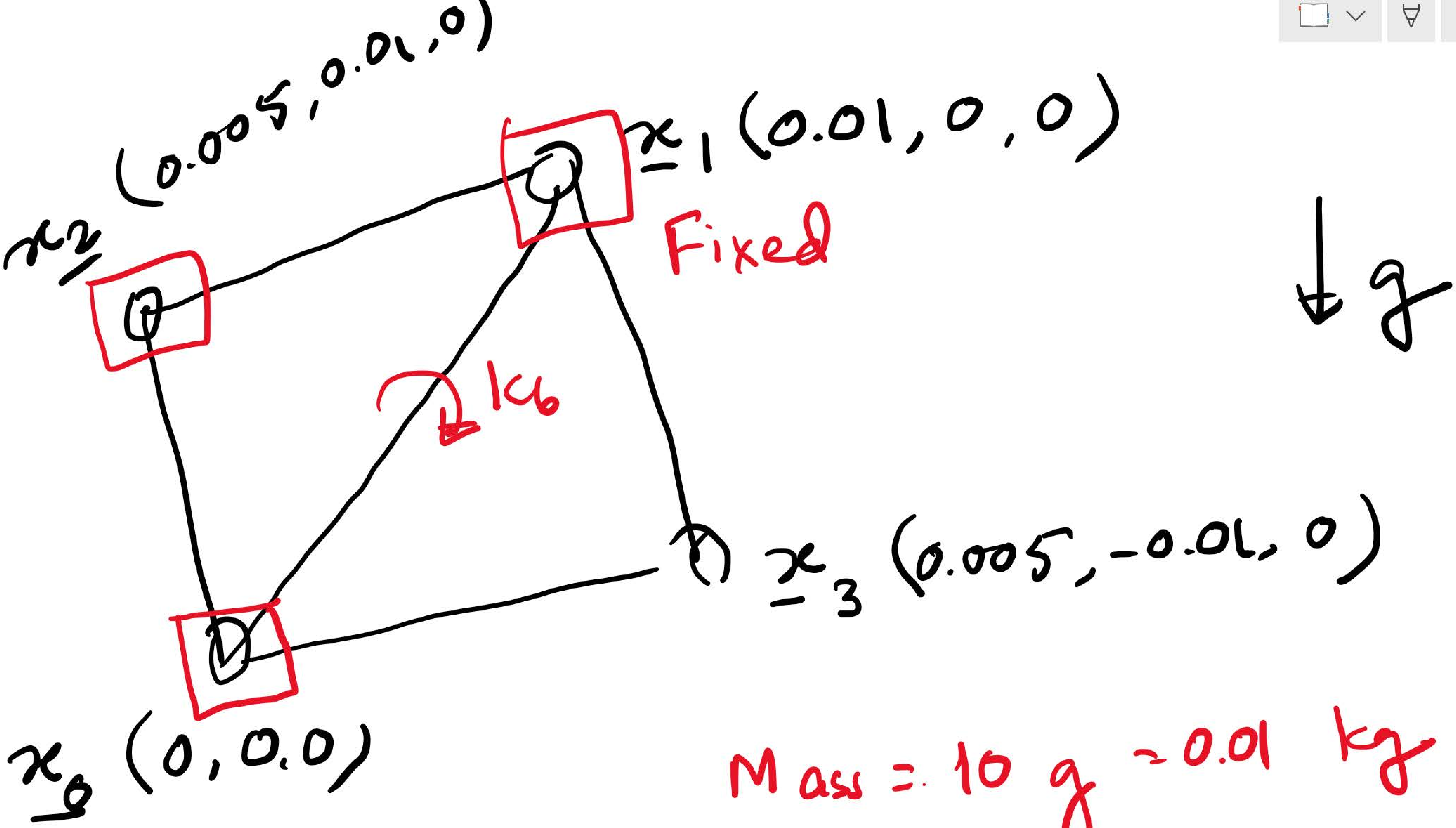
Gradient

$$\nabla E_b^i = \frac{\partial}{\partial \underline{q}} E_b^i = \frac{1}{2} k_b \cdot 2(\theta - \bar{\theta}) \nabla \theta = k_b (\theta - \bar{\theta}) \overbrace{\nabla \theta}^{\text{gradTheta}}$$

Hessian

$$\nabla(\nabla E_b^i) = \nabla\left(k_b(\theta - \bar{\theta}) \nabla \theta\right)$$

$$= k_b(\theta - \bar{\theta}) \underbrace{\nabla^2 \theta}_{\text{hessTheta}} + k_b \overbrace{\underbrace{\nabla \theta \otimes \nabla \theta}_{\text{np.outer}}}^{\nabla \theta . \nabla \theta^T}$$


$\underline{x}_2$ (0.005, 0.01, 0)
$\underline{x}_1$ (0.01, 0, 0)
Fixed
g
k_6
$\underline{x}_3$ (0.005, -0.01, 0)
$\underline{x}_0$ (0, 0.0)

Mass = 10 g = 0.01 kg

Weight = 0.01 * 9.81 N

Young's modulus = 10 MPa

h = 0.001 m (thickness)

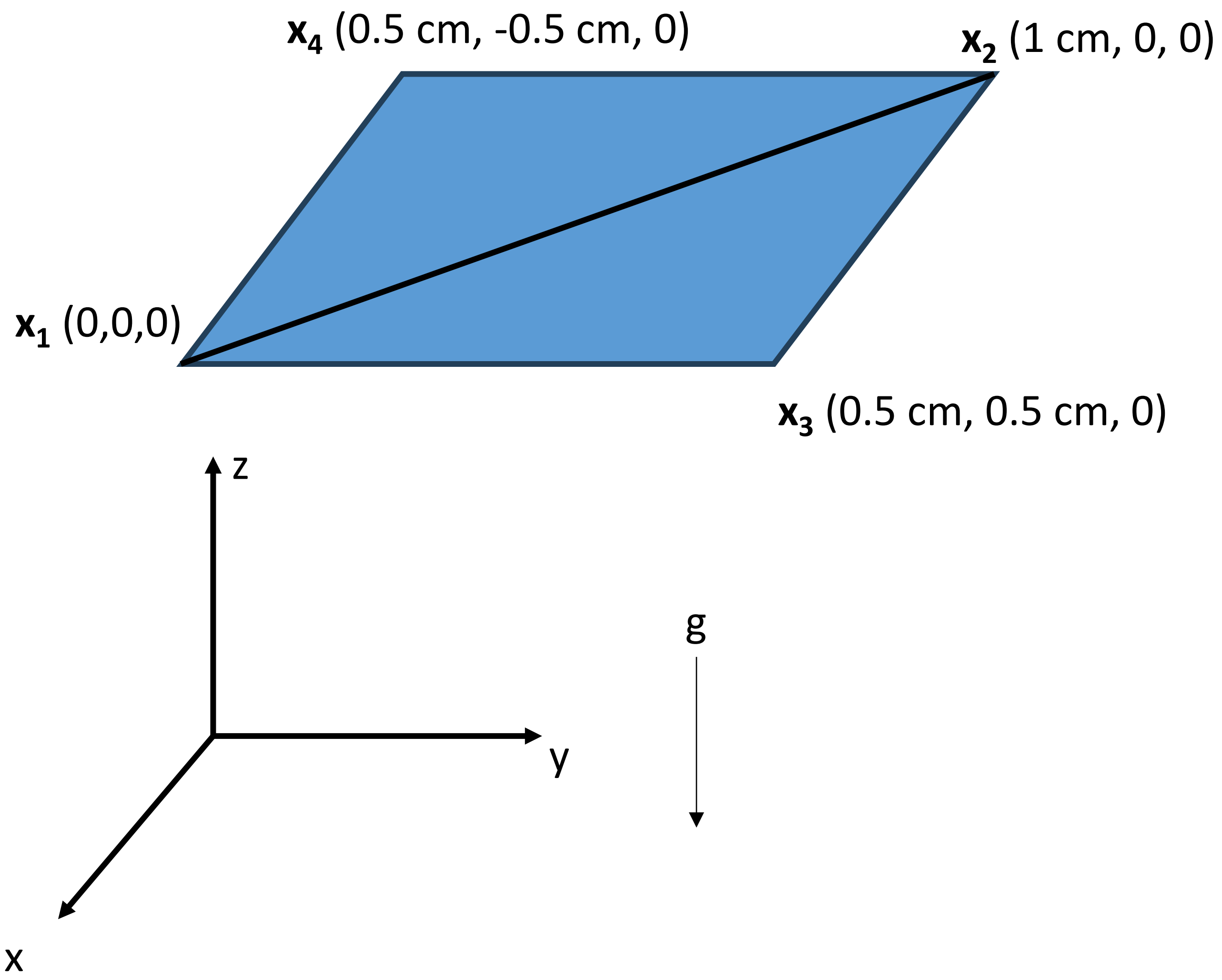
STRUCTURES-COMPUTER
INTERACTION LABORATORY

Module : Plate Example

Discrete Elastic Plates (DEP) Algorithm

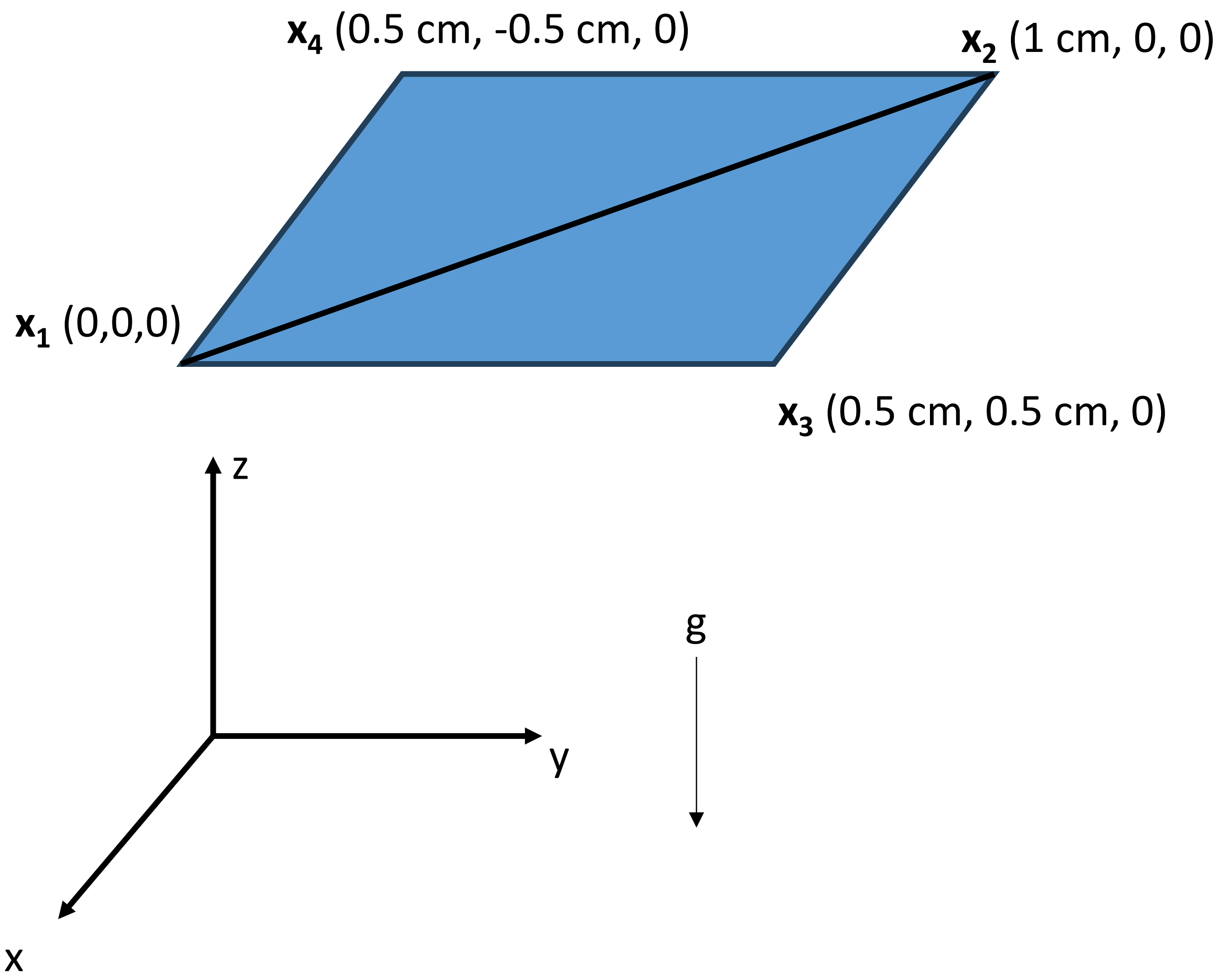
Problem Setup

- 4 nodes
- 3 nodes are fixed, one is free
- What is the deformed shape under gravity?


$\mathbf{x}_4$ (0.5 cm, -0.5 cm, 0)
$\mathbf{x}_2$ (1 cm, 0, 0)
$\mathbf{x}_1$ (0,0,0)
$\mathbf{x}_3$ (0.5 cm, 0.5 cm, 0)
z
y
x
g

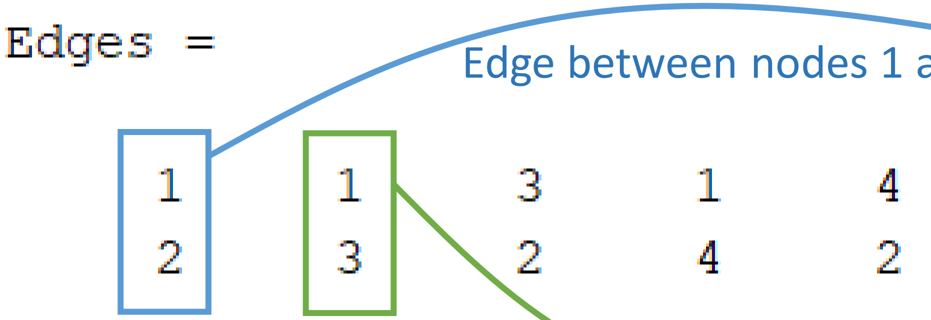
Parameters

- Thickness = 1 mm
- Young's modulus = 10 MPa
- Total mass = 10 g
- Gravity = -9.81 m/s^2
- Total time = 5 seconds
- Time step size = 0.01 second


$\mathbf{x}_4$ (0.5 cm, -0.5 cm, 0)
$\mathbf{x}_2$ (1 cm, 0, 0)
$\mathbf{x}_1$ (0,0,0)
$\mathbf{x}_3$ (0.5 cm, 0.5 cm, 0)
z
y
x
g

Problem Setup

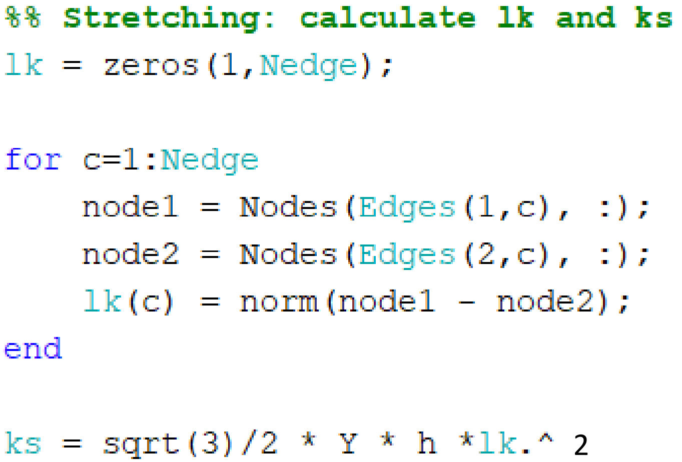
Edges: 5x (Stretching)



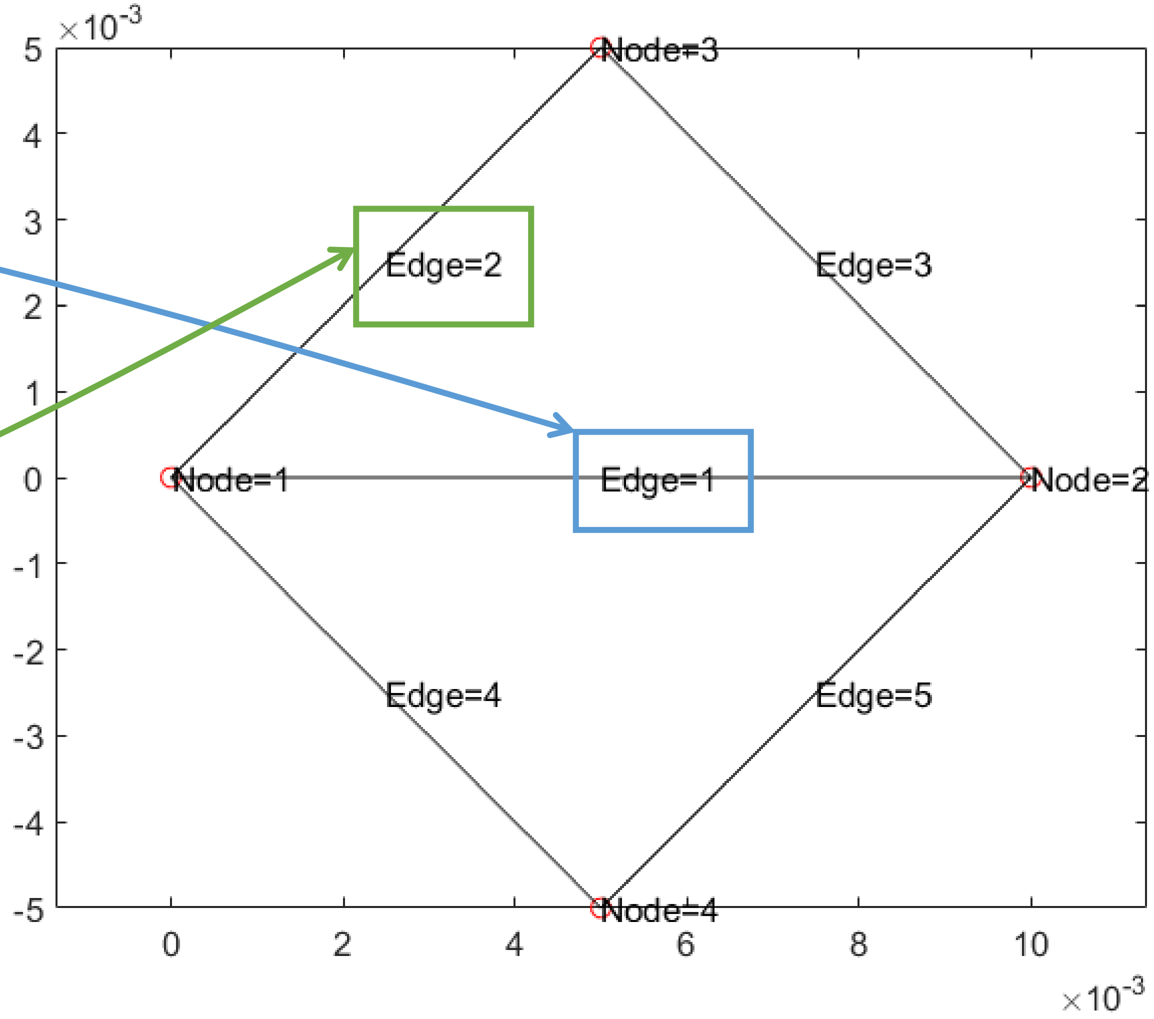
```
Edges =

     1     1     3     1     4
     2     3     2     4     2
```

Edge between nodes 1 and 2

Edge between nodes 1 and 2

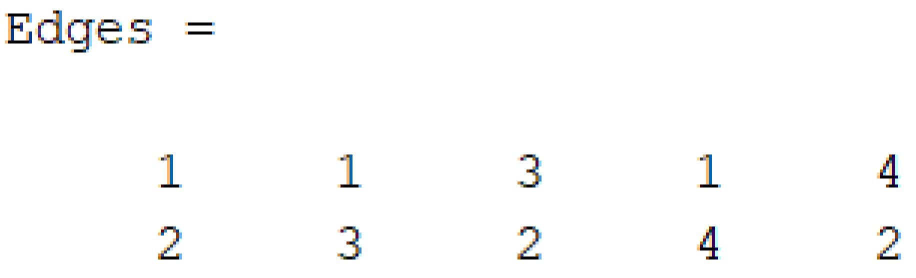


```
%% Stretching: calculate lk and ks
lk = zeros(1,Nedge);

for c=1:Nedge
    node1 = Nodes(Edges(1,c), :);
    node2 = Nodes(Edges(2,c), :);
    lk(c) = norm(node1 - node2);
end

ks = sqrt(3)/2 * Y * h *lk.^ 2
```



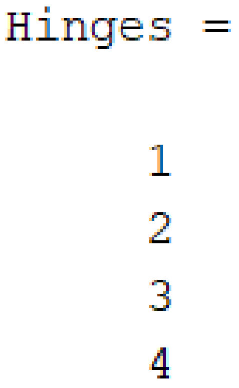
Problem Setup

Edges: 5x (Stretching)

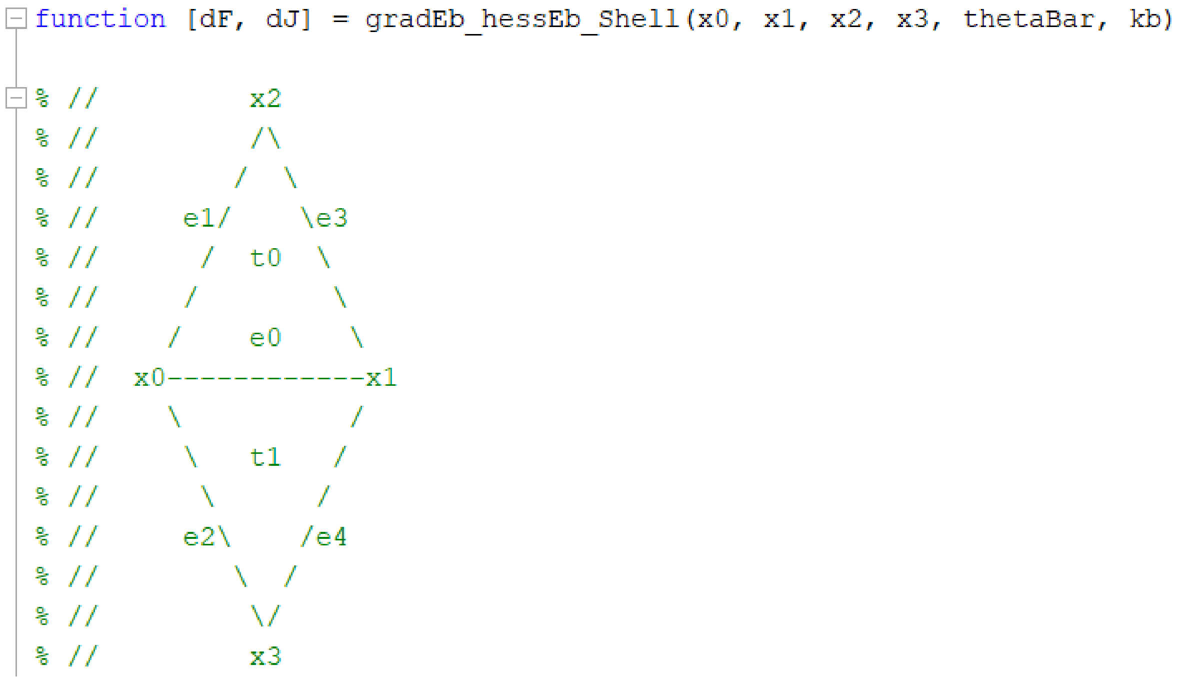


```
Edges =

     1     1     3     1     4
     2     3     2     4     2
```

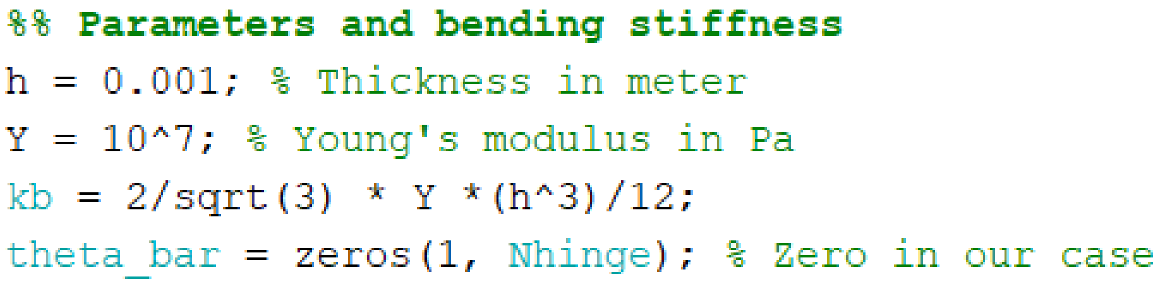
Hinges: 1x (Bending)



```
Hinges =

     1
     2
     3
     4
```



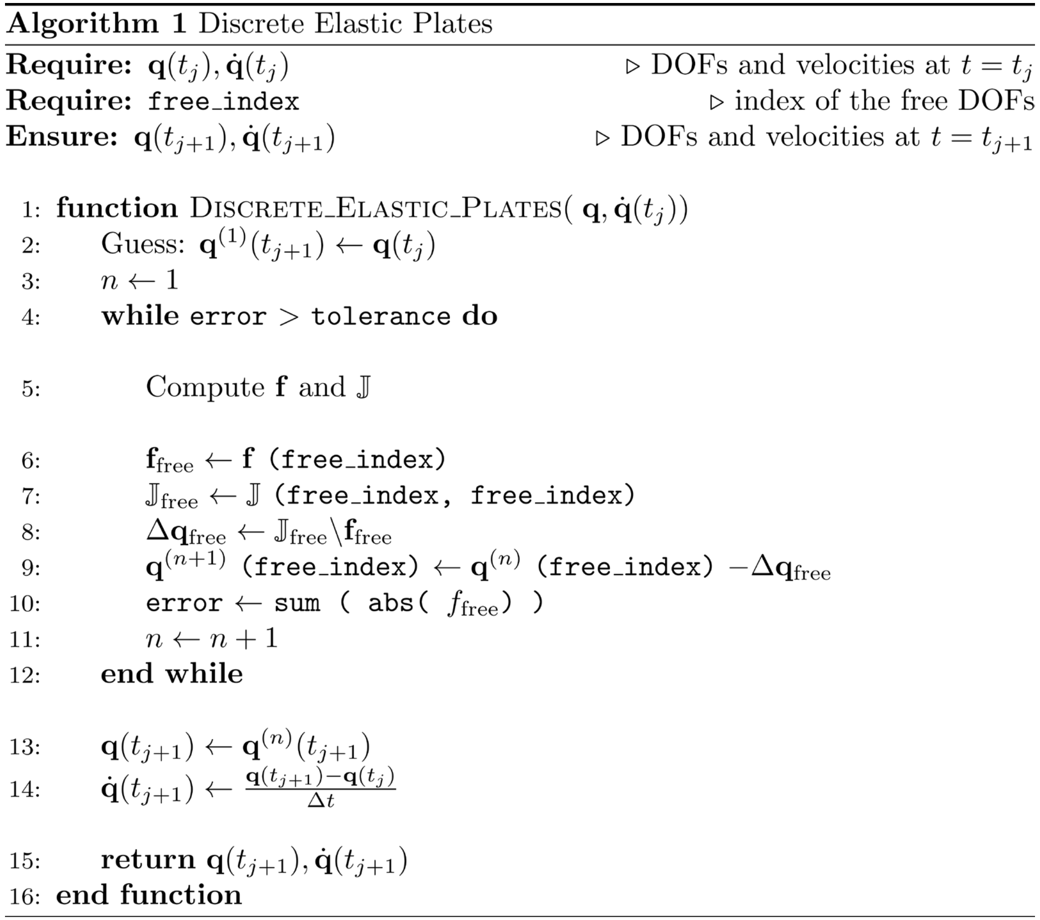
```
function [dF, dJ] = gradEb_hessEb_Shell(x0, x1, x2, x3, thetaBar, kb)

% //         x2
% //         /\
% //        /  \
% //     e1/    \e3
% //      /  t0  \
% //     /        \
% //    /    e0    \
% //  x0------------x1
% //    \          /
% //     \   t1   /
% //      \      /
% //     e2\    /e4
% //        \  /
% //         \/
% //         x3
```



```
%% Parameters and bending stiffness
h = 0.001; % Thickness in meter
Y = 10^7; % Young's modulus in Pa
kb = 2/sqrt(3) * Y *(h^3)/12;
theta_bar = zeros(1, Nhinge); % Zero in our case
```

Time Stepping

```
for t=1:1:Nsteps
    [q,v] = DiscreteElasticPlates(q,v);
end
```



Algorithm 1 Discrete Elastic Plates

```
Require: $\mathbf{q}(t_j), \dot{\mathbf{q}}(t_j)$          ▷ DOFs and velocities at $t = t_j$
Require: free_index                                     ▷ index of the free DOFs
Ensure: $\mathbf{q}(t_{j+1}), \dot{\mathbf{q}}(t_{j+1})$    ▷ DOFs and velocities at $t = t_{j+1}$

function DISCRETE_ELASTIC_PLATES( $\mathbf{q}, \dot{\mathbf{q}}(t_j)$)
    Guess: $\mathbf{q}^{(1)}(t_{j+1}) \leftarrow \mathbf{q}(t_j)$
    $n \leftarrow 1$
    while error > tolerance do

        Compute $\mathbf{f}$ and $\mathbb{J}$

        $\mathbf{f}_{\text{free}} \leftarrow \mathbf{f}$ (free_index)
        $\mathbb{J}_{\text{free}} \leftarrow \mathbb{J}$ (free_index, free_index)
        $\Delta\mathbf{q}_{\text{free}} \leftarrow \mathbb{J}_{\text{free}} \backslash \mathbf{f}_{\text{free}}$
        $\mathbf{q}^{(n+1)}$ (free_index) $\leftarrow \mathbf{q}^{(n)}$ (free_index) $-\Delta\mathbf{q}_{\text{free}}$
        error $\leftarrow$ sum ( abs( $f_{\text{free}}$) )
        $n \leftarrow n + 1$
    end while

    $\mathbf{q}(t_{j+1}) \leftarrow \mathbf{q}^{(n)}(t_{j+1})$
    $\dot{\mathbf{q}}(t_{j+1}) \leftarrow \frac{\mathbf{q}(t_{j+1}) - \mathbf{q}(t_j)}{\Delta t}$

    return $\mathbf{q}(t_{j+1}), \dot{\mathbf{q}}(t_{j+1})$
end function
```