

# Module 13

Fixed and Free Degrees of Freedom

- What happens when boundary conditions are applied?



# Goal

- Understand fixed and free degrees of freedom
- Adapt our method to incorporate boundary conditions



# Resources

- Read “*Chapter 4: Conservative force and potential energy*” of Course Notes

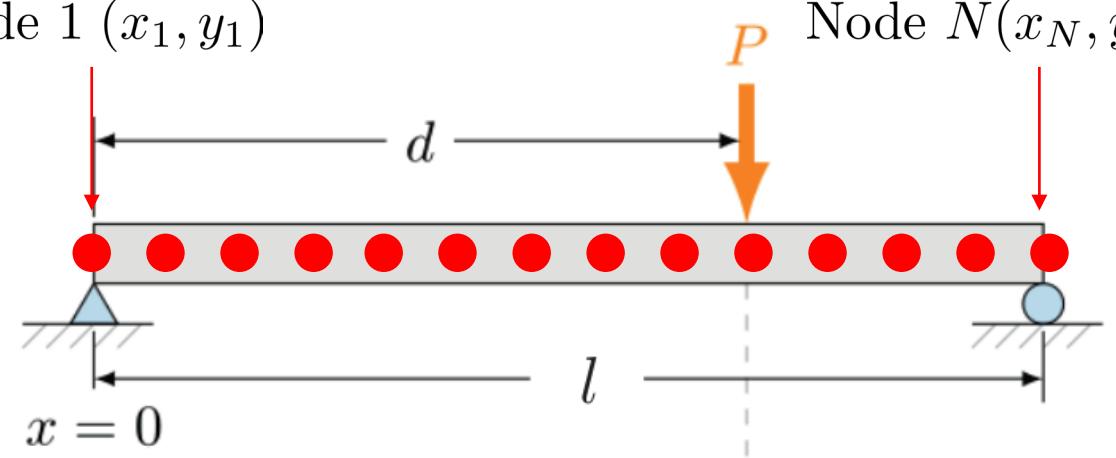


# Boundary conditions

DOF Vector with size  $2N$ :

$$\mathbf{q} = \begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ \dots \\ \dots \\ x_k \\ y_k \\ \dots \\ \dots \\ x_N \\ y_N \end{bmatrix}$$

Node 1 ( $x_1, y_1$ )



Fixed DOF:

$$x_1 = 0$$

$$y_1 = 0$$

$$y_N = 0$$



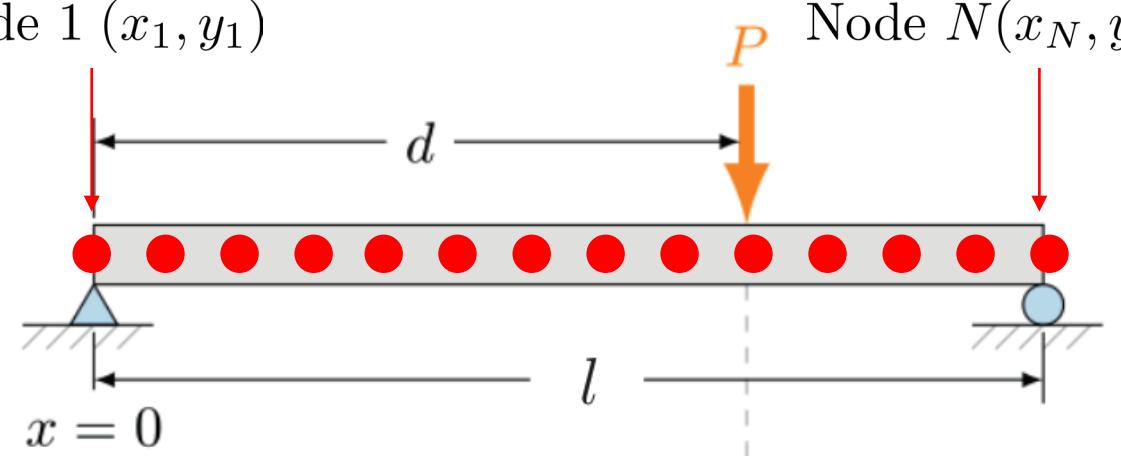
# Fixed and Free Degrees of Freedom

DOF Vector with size  $2N$ :

$$\mathbf{q} = \begin{bmatrix} x_1 \\ y_1 \\ \vdots \\ \vdots \\ x_k \\ y_k \\ \vdots \\ \vdots \\ x_N \\ y_N \end{bmatrix}$$

$\mathbf{q}_{\text{fixed}}$        $\mathbf{q}_{\text{free}}$

Node 1 ( $x_1, y_1$ )





# Fixed and Free Degrees of Freedom

DOF Vector with size  $2N$ :

Equations of motion are not applicable to fixed DOFs. Instead, we have the following equations:

Fixed DOF:

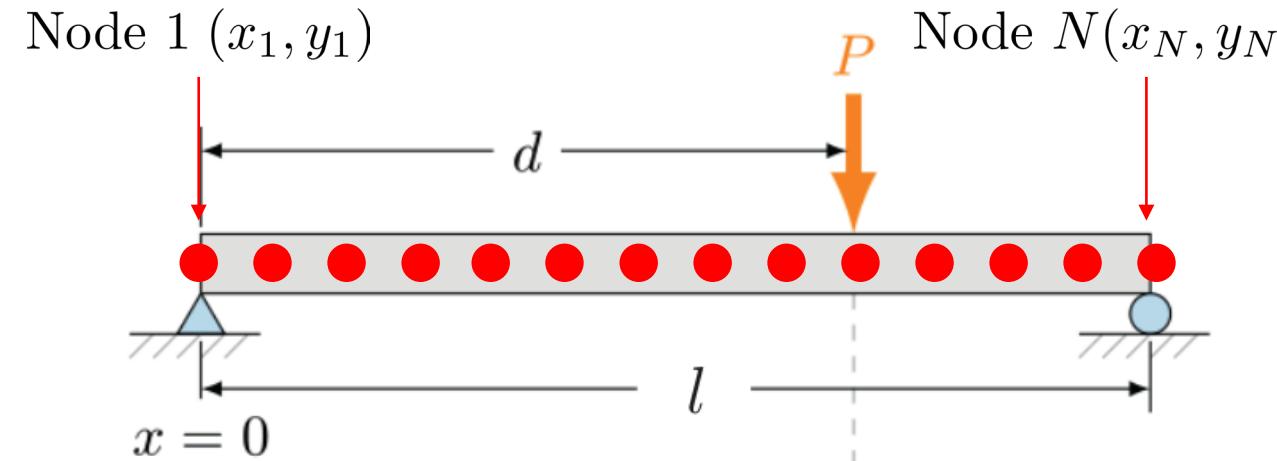
$$x_1 = 0$$

$$y_1 = 0$$

$$y_N = 0$$

$$\mathbf{q} = \begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ \dots \\ \dots \\ x_k \\ y_k \\ \dots \\ \dots \\ x_N \\ y_N \end{bmatrix}$$

$\mathbf{q}_{\text{fixed}}$        $\mathbf{q}_{\text{free}}$



Fixed index:  $[1, 2, 2N]$

Free index:  $[3, 4, \dots, 2N - 1]$

We will need this later.



# Programming Implementation

$[\mathbf{q}(t_{k+1}), \dot{\mathbf{q}}(t_{k+1})] = \text{SimulationLoop}([\mathbf{q}(t_k), \dot{\mathbf{q}}(t_k)])$

When all DOFs are free

Compute  $\mathbf{f} = \mathbf{f}(\mathbf{q}(t_{k+1}))$   
Compute  $\mathbb{J} = \mathbb{J}(\mathbf{q}(t_{k+1}))$

$\Delta\mathbf{q} = \mathbb{J}^{-1}\mathbf{f}$

New estimate  $\mathbf{q} = -\Delta\mathbf{q} + \mathbf{q}$

When some DOFs are fixed

Compute  $\mathbf{f} = \mathbf{f}(\mathbf{q}(t_{k+1}))$   
Compute  $\mathbb{J} = \mathbb{J}(\mathbf{q}(t_{k+1}))$

$\mathbf{f}_{\text{free}} = \mathbf{f}(\text{free index})$   
 $\mathbb{J}_{\text{free}} = \mathbb{J}(\text{free index}, \text{free index})$

$\Delta\mathbf{q}_{\text{free}} = \mathbb{J}_{\text{free}}^{-1}\mathbf{f}_{\text{free}}$

New estimate  $\mathbf{q}_{\text{free}} = -\Delta\mathbf{q}_{\text{free}} + \mathbf{q}_{\text{free}}$

New estimate  $\mathbf{q}(\text{free index}) = \mathbf{q}_{\text{free}}$

New estimate  $\mathbf{q}(\text{fixed index}) = \mathbf{q}_{\text{fixed}}$



# Programming Implementation

$$\mathbf{f} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_k \\ \vdots \\ f_N \end{bmatrix}$$

$\mathbf{f}_{\text{free}} = \mathbf{f}(\text{free index})$

$\mathbb{J} = \begin{bmatrix} J_{11} & J_{12} & \cdots & J_{1j} & \cdots & J_{1N} \\ J_{21} & J_{22} & \cdots & J_{2j} & \cdots & J_{2N} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ J_{N1} & J_{N2} & \cdots & J_{Nj} & \cdots & J_{NN} \end{bmatrix}$

$\mathbb{J}_{\text{free}} = \mathbb{J}(\text{free index})$

“Delete” the rows and columns corresponding to fixed index

$$\mathbf{q} = \begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ \vdots \\ \vdots \\ x_k \\ y_k \\ \vdots \\ \vdots \\ x_N \\ y_N \end{bmatrix}$$

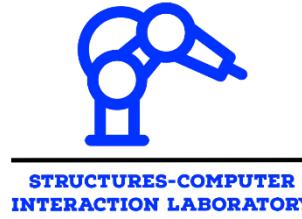
$\mathbf{q}_{\text{fixed}}$

$\mathbf{q}_{\text{free}}$



# Take-away Message

- Don't be afraid of fixed degrees of freedom
  - More fixed degrees of freedom mean faster computation time due to smaller linear system



# Module 14

---

Starter on Beam Bending

- Example on beam bending
- Comparison with Euler-Bernoulli beam theory



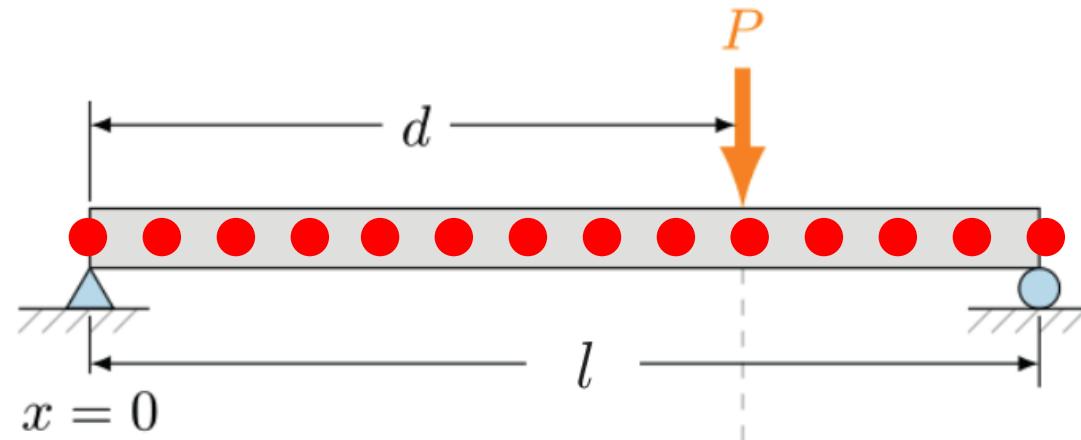
# Goal

- Setup a problem with an Euler-Bernoulli type beam bending problem
- Understand fixed and free degrees of freedom
- Compare discrete simulation with Euler-Bernoulli beam theory



# Resources

- Read “*Chapter 4: Conservative force and potential energy*” of Course Notes

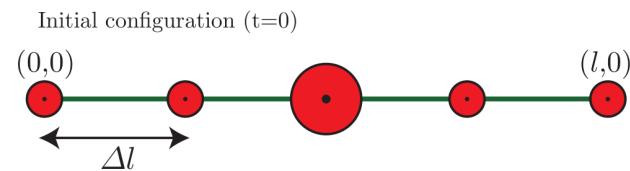
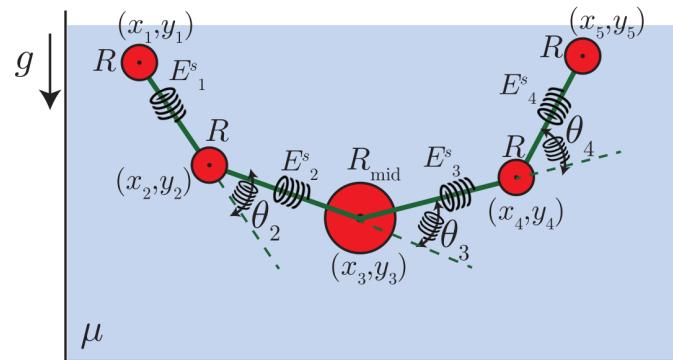




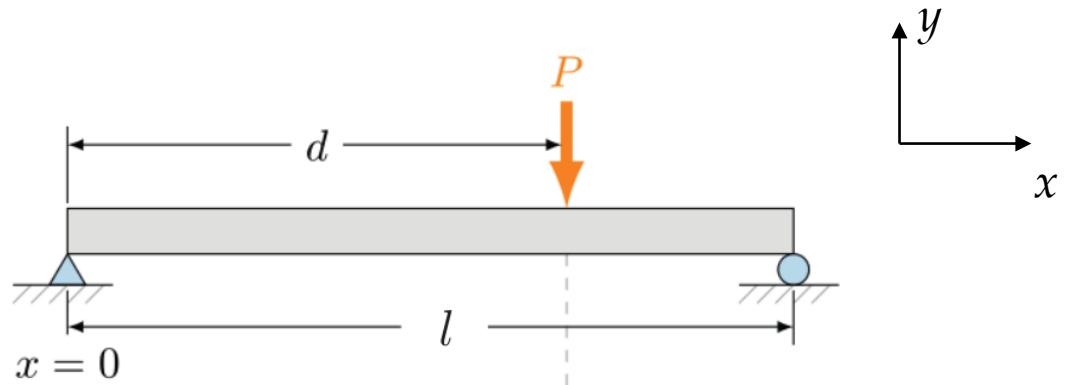
# Equations of Motion (EOM)

$$\mathbf{M} \ddot{\mathbf{q}} + \frac{\partial E^{\text{elastic}}}{\partial \mathbf{q}} + \mathbf{C} \dot{\mathbf{q}} - \mathbf{W} = 0$$

$$\mathbf{M} \ddot{\mathbf{q}} + \frac{\partial E^{\text{elastic}}}{\partial \mathbf{q}} - \mathbf{P} = 0$$



Old Problem

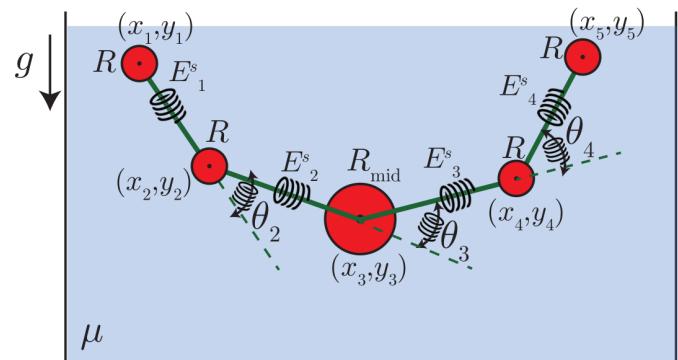


New Problem

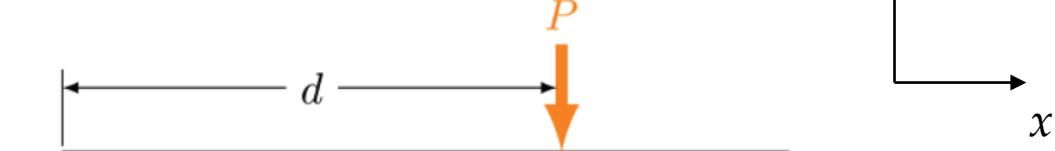


# Equations of Motion (EOM)

$$\mathbf{M} \ddot{\mathbf{q}} + \frac{\partial E^{\text{elastic}}}{\partial \mathbf{q}} + \mathbf{C} \dot{\mathbf{q}} - \mathbf{W} = 0$$



$$\mathbf{M} \ddot{\mathbf{q}} + \frac{\partial E^{\text{elastic}}}{\partial \mathbf{q}} - \mathbf{P} = 0$$



$$\mathbf{q} = \begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ \vdots \\ \vdots \\ x_k \\ y_k \\ \vdots \\ \vdots \\ x_N \\ y_N \end{bmatrix} \quad \mathbf{P} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ \vdots \\ 0 \\ -P \\ \vdots \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$



# Programming Implementation

$$[\mathbf{q}(t_{k+1}), \dot{\mathbf{q}}(t_{k+1})] = \text{SimulationLoop}([\mathbf{q}(t_k), \dot{\mathbf{q}}(t_k)])$$

Old Problem

Compute  $\mathbf{f} = \mathbf{f}(\mathbf{q}(t_{k+1}))$   
Compute  $\mathbb{J} = \mathbb{J}(\mathbf{q}(t_{k+1}))$

$$\Delta\mathbf{q} = \mathbb{J} \setminus \mathbf{f}$$

$$\text{New estimate } \mathbf{q} = \Delta\mathbf{q} + \mathbf{q}$$

New Problem

Compute  $\mathbf{f} = \mathbf{f}(\mathbf{q}(t_{k+1}))$   
Compute  $\mathbb{J} = \mathbb{J}(\mathbf{q}(t_{k+1}))$

$\mathbf{f}_{\text{free}} = \mathbf{f}(\text{free index})$   
 $\mathbb{J}_{\text{free}} = \mathbb{J}(\text{free index}, \text{free index})$

$$\Delta\mathbf{q}_{\text{free}} = \mathbb{J}_{\text{free}} \setminus \mathbf{f}_{\text{free}}$$

$$\text{New estimate } \mathbf{q}_{\text{free}} = \Delta\mathbf{q}_{\text{free}} + \mathbf{q}_{\text{free}}$$

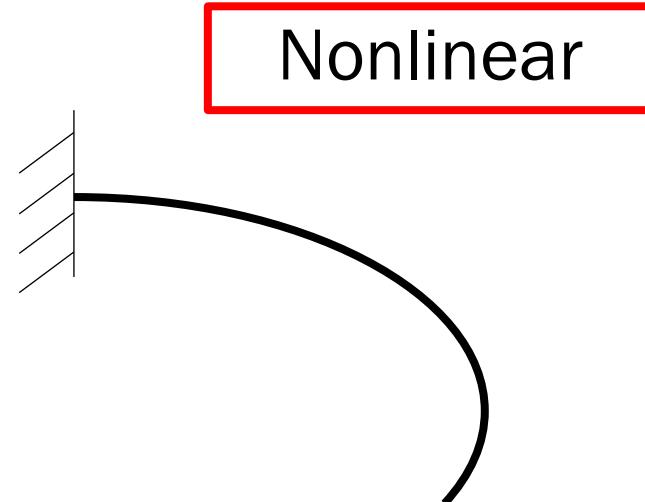
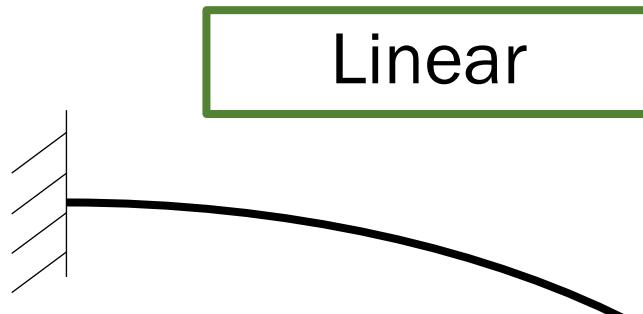
$$\text{New estimate } \mathbf{q}(\text{free index}) = \mathbf{q}_{\text{free}}$$

New estimate  $\mathbf{q}(\text{fixed index}) = \mathbf{q}_{\text{fixed}}$   
//Calculate error based on  $\mathbf{f}_{\text{free}}$

# Comparison with Euler-Bernoulli Beam Theory

- Euler-Bernoulli beam theory is *geometrically linear*
- Only applicable when the slope is small

Curvature of a function  $y = y(x)$  is  $\kappa = \frac{y''}{(1+y'^2)^{3/2}} \approx y''$

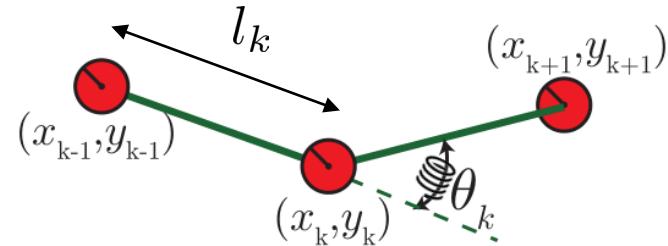


# Comparison with Euler-Bernoulli Beam Theory

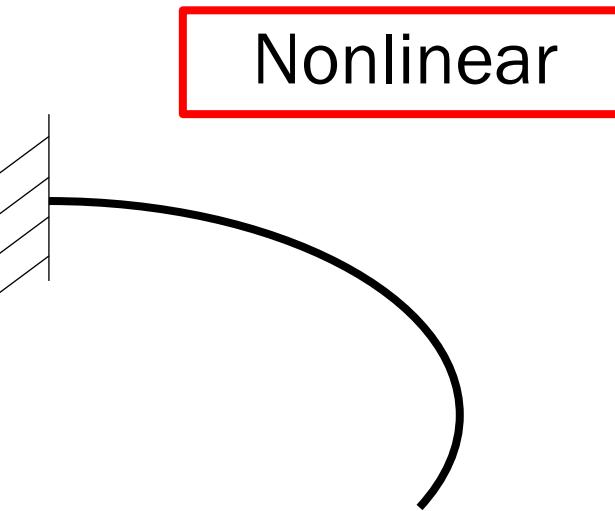
- Our discrete simulation is *geometrically nonlinear*
- It can handle “arbitrary” slopes

Curvature at node  $k$  is

$$\kappa = 2 \tan\left(\frac{\theta_k}{2}\right) / l_k$$



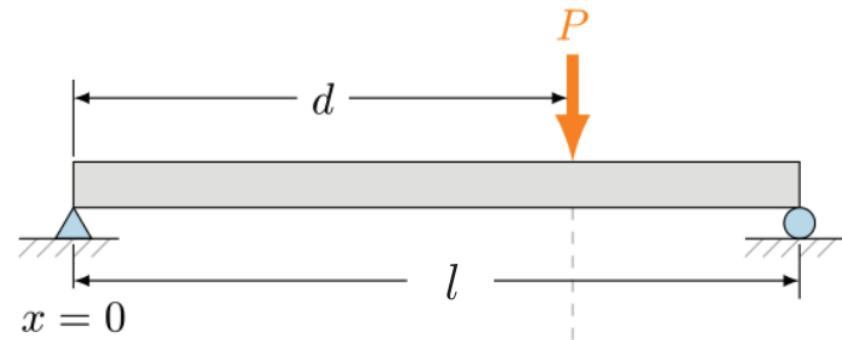
- More details in “A primer on the kinematics of discrete elastic rods” by MK Jawed, A Novelia, OM O'Reilly
- Available for free on the course website

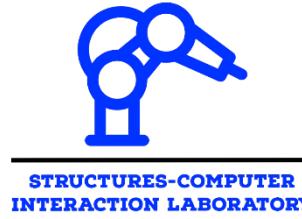




# Assignments

- Complete Assignment 3 in Chapter 4: Generalized case of elastic beam falling in viscous flow





# Module 15

Discrete Elastic Energies of a Beam

- Continuous and Discrete Elastic Energies of a Beam
- Natural Curvature



# Goal

- Differences between curvature and “discrete integrated” curvature
- Natural curvature
- Discrete representation of the elastic (bending and stretching) energies in a beam



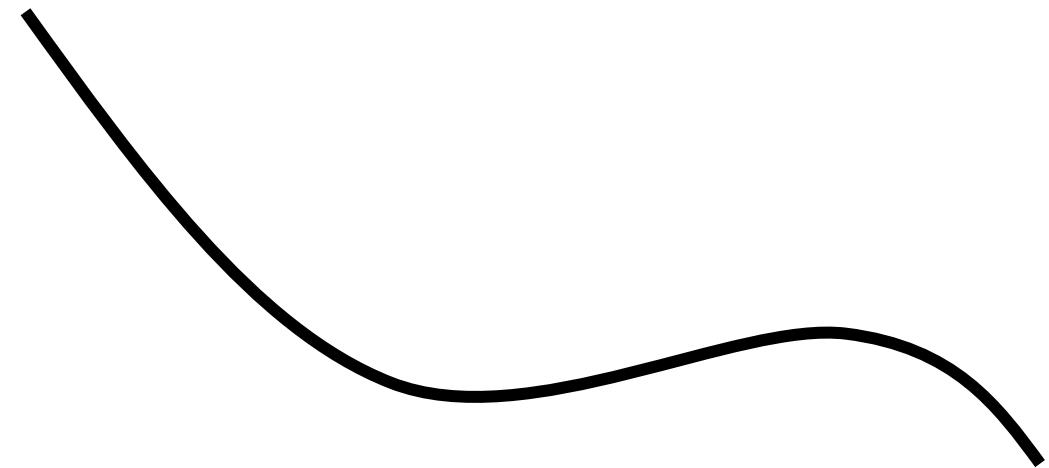
# Resources

- Read “*Chapter 5: Discrete simulation of an elastic beam*” of Course Notes
- Recall “*Module: Elastic Beam in Viscous Flow*” and “*Module: Starter on Beam Bending*”

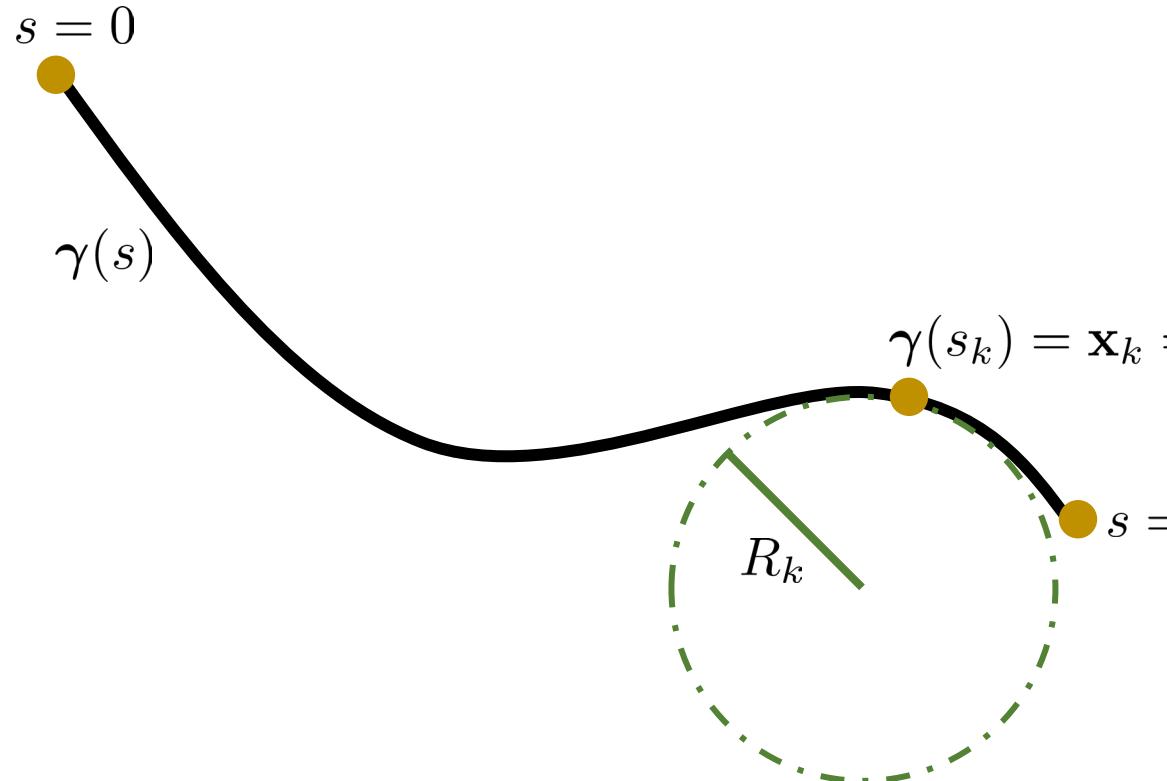


# Elastic Energies of a Beam

- A beam has two types of elastic energies:
  - Bending energy
  - Stretching energy (often neglected)



# Curvature



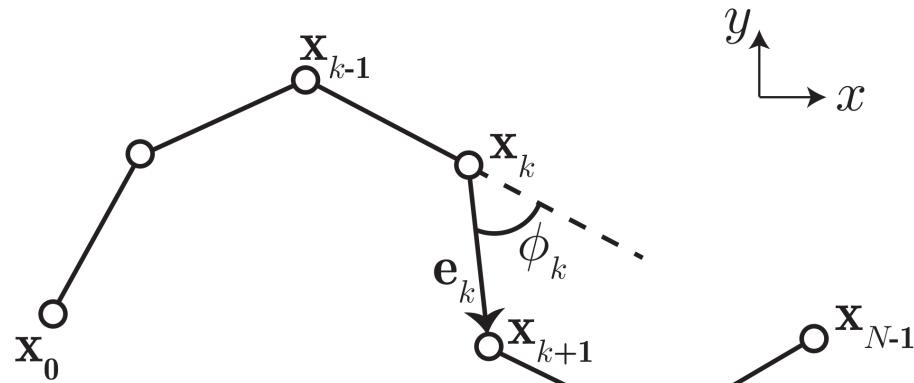
A curve  $\gamma$  with an arc-length parameter  $s$

$$\text{Curvature, } \kappa_k = \frac{1}{R_k}$$

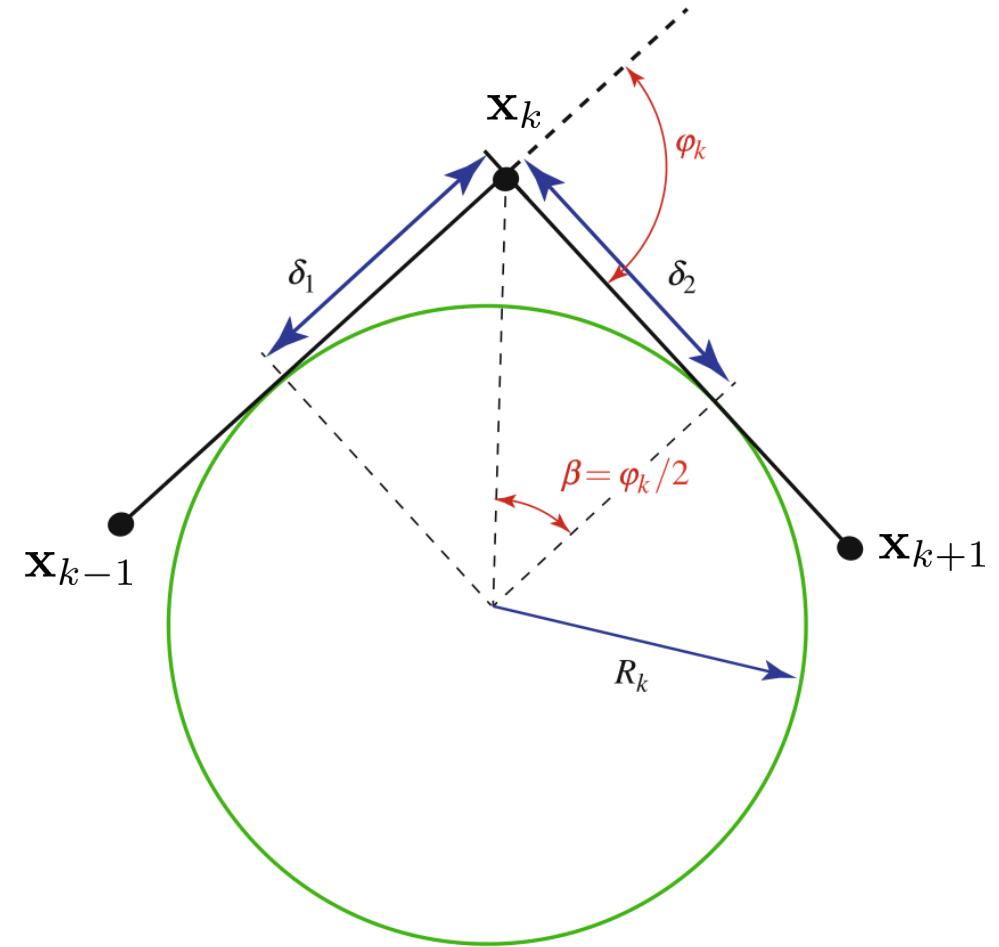
$$\gamma''(s_k) \equiv \frac{\partial^2 \gamma}{\partial s^2} |_{s=s_k} = \frac{1}{R_k}$$



# Discrete Integrated Curvature

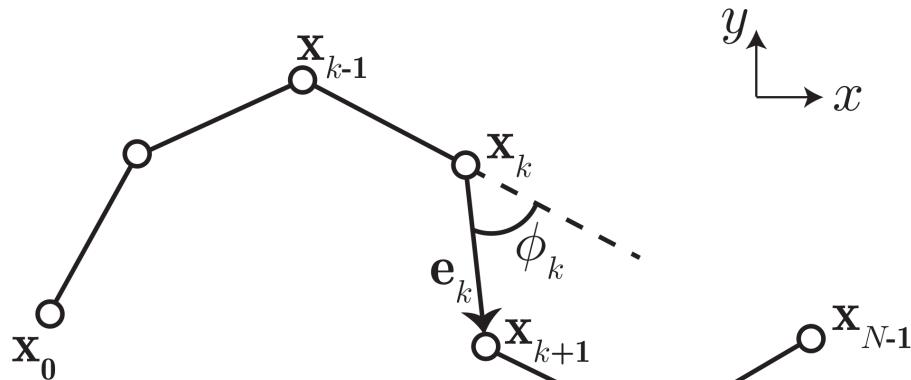


$$R_k = ?$$





# Discrete Integrated Curvature



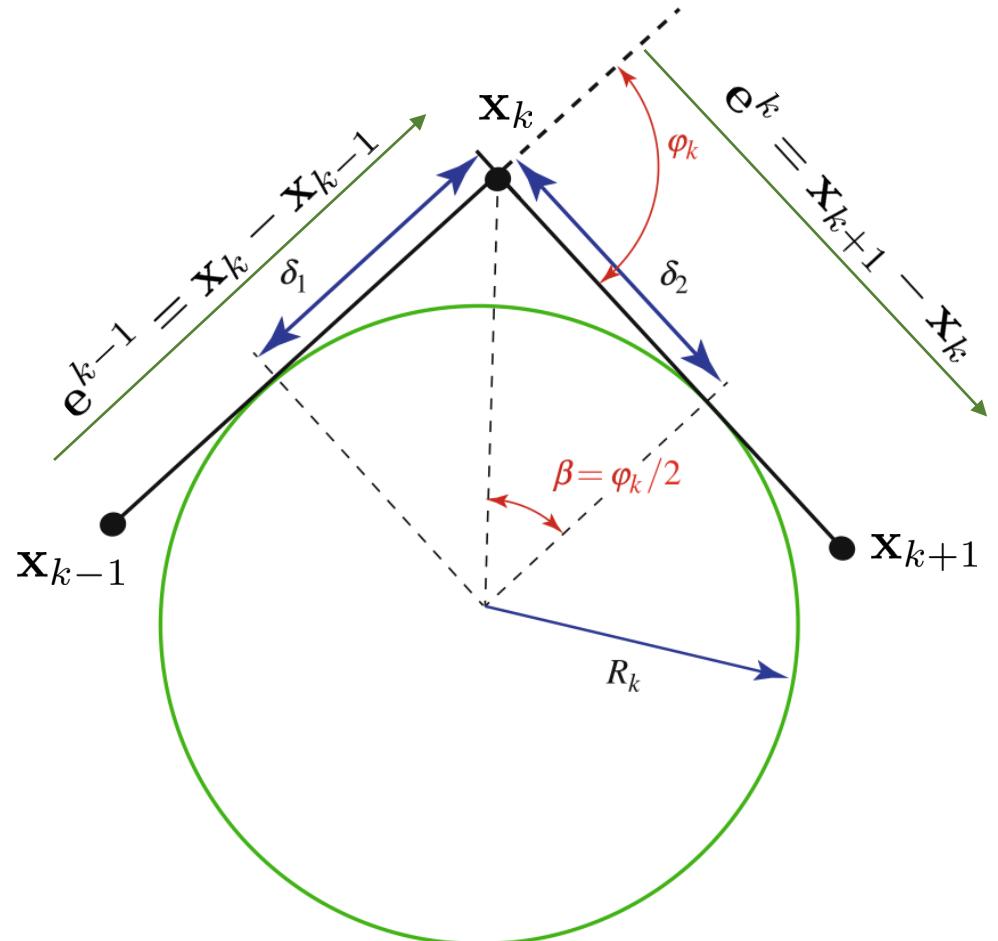
$$\delta_1 = \delta_2 \equiv \frac{dl}{2}$$

Simple geometry:  $R_k = \frac{dl}{2 \tan(\phi_k/2)}$

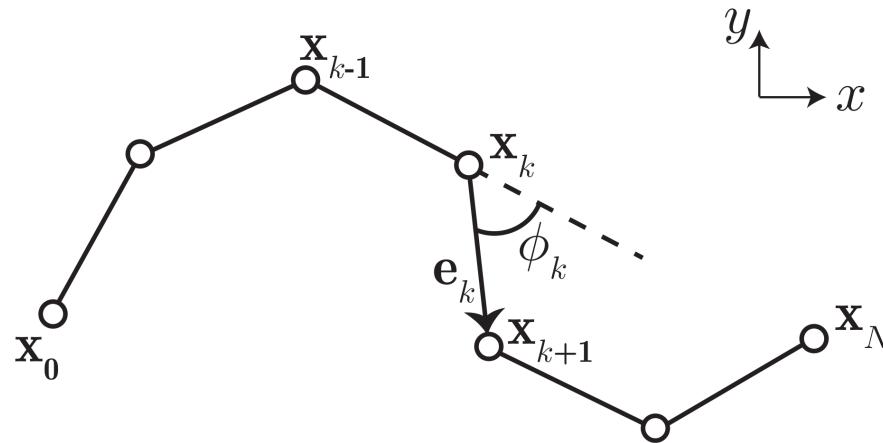
$$\phi_k = \tan^{-1} \frac{(\mathbf{e}^{k-1} \times \mathbf{e}^k) \cdot \hat{\mathbf{e}}_z}{\mathbf{e}^{k-1} \cdot \mathbf{e}^k}$$

Discrete integrated curvature:  $\kappa_k \equiv 2 \tan(\phi_k/2)$

Standard definition of curvature =  $\frac{\kappa_k}{dl} = \frac{1}{R_k}$



# Summary: Discrete Integrated Curvature

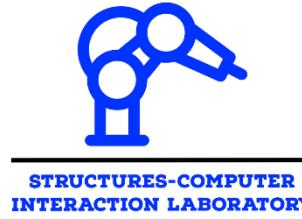


We learned how to compute the curvature along the beam from the degrees of freedom (DOFs)  $x_k, y_k$ .

Notation:

Subscript indicates node-based quantities, e.g.  $\kappa_k$

Superscript indicates edge-based quantities, e.g.  $e^k$



# Module 16

---

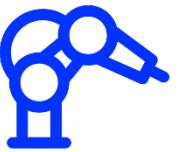
Discrete Elastic Beams

- Continuous and Discrete Equations of Motion
- Programming Implementation



# Goal

- Equations of motion for an elastic beam
- Programming implementation



# Resources

- Read “*Chapter 5: Discrete simulation of an elastic beam*” of Course Notes
- Recall “*Module: Elastic Beam in Viscous Flow*” and “*Module: Starter on Beam Bending*”
  - Only new concept is *natural curvature*



# Continuous Equations of Motion (EOM)

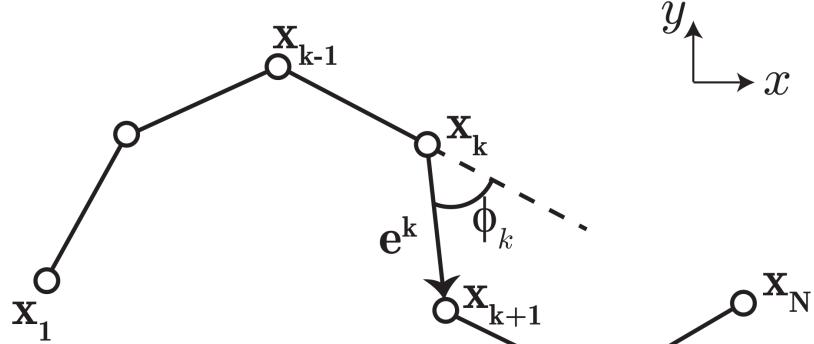
Degrees of Freedom (DOF) vector,

$$\mathbf{q} = \begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ \dots \\ \dots \\ x_k \\ y_k \\ \dots \\ \dots \\ x_N \\ y_N \end{bmatrix}$$

$2N$  equations of motion for  $N$  nodes:

$$\mathbf{M} \ddot{\mathbf{q}} + \frac{\partial E_{\text{elastic}}}{\partial \mathbf{q}} - \mathbf{F}_{\text{external}} = \mathbf{0}$$

$$\implies \mathbf{M} \ddot{\mathbf{q}} + \boxed{\frac{\partial E_b}{\partial \mathbf{q}} + \frac{\partial E_s}{\partial \mathbf{q}}} - \mathbf{F}_{\text{external}} = \mathbf{0}$$



Elastic Energy = Bending + Stretching



# Discrete Equations of Motion (EOM)

$$\mathbf{M} \ddot{\mathbf{q}} + \frac{\partial E_b}{\partial \mathbf{q}} + \frac{\partial E_s}{\partial \mathbf{q}} - \mathbf{F}_{\text{external}} = \mathbf{0}$$

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

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

where

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

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

$$\mathbb{J}_{ij}^{\text{external}} = -\frac{\partial F_{i,\text{external}}}{\partial q_j}.$$

# Programming Implementation

$t = 0 :$

Compute natural discrete curvature  $\kappa_k^0$  and  
undeformed length of each edge  $dl$

Move from one time step to the next one:

$[\mathbf{q}(t_{k+1}), \dot{\mathbf{q}}(t_{k+1})] = \text{SimulationLoop}([\mathbf{q}(t_k), \dot{\mathbf{q}}(t_k)] )$

Compute  $\mathbf{f} = \mathbf{f}(\mathbf{q}(t_{k+1}))$

Compute  $\mathbb{J} = \mathbb{J}(\mathbf{q}(t_{k+1}))$

$\mathbf{f}_{\text{free}} = \mathbf{f}(\text{free index})$

$\mathbb{J}_{\text{free}} = \mathbb{J}(\text{free index}, \text{free index})$

$\Delta\mathbf{q}_{\text{free}} = \mathbb{J}_{\text{free}} \backslash \mathbf{f}_{\text{free}}$

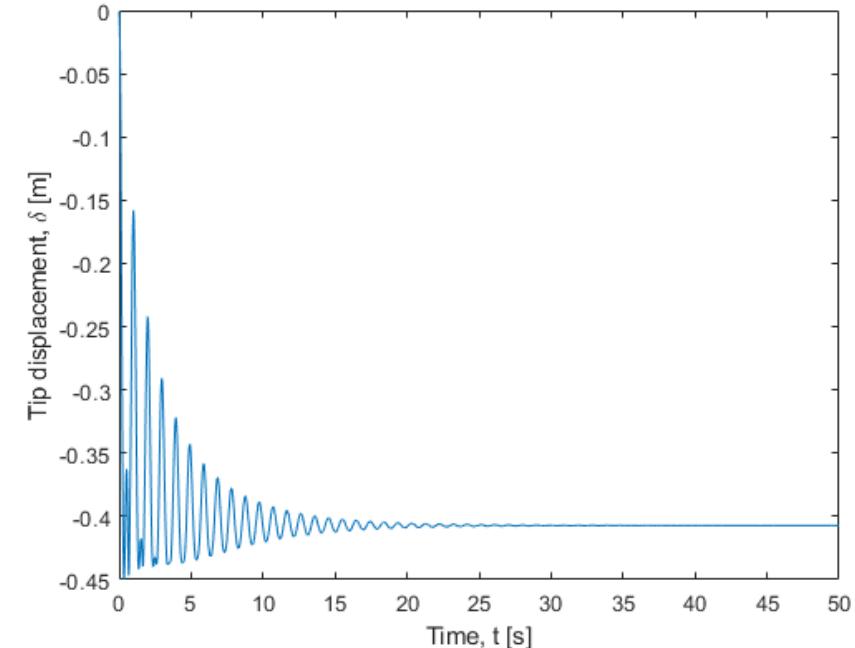
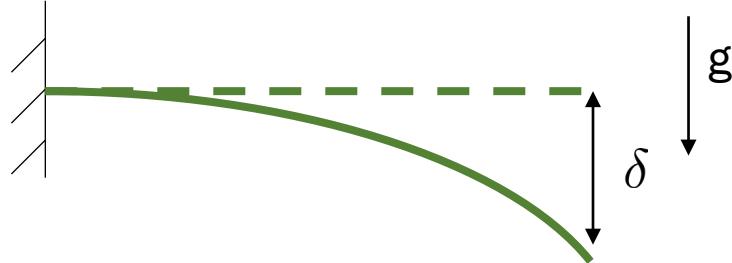
New estimate  $\mathbf{q}_{\text{free}} = \Delta\mathbf{q}_{\text{free}} + \mathbf{q}_{\text{free}}$

New estimate  $\mathbf{q}(\text{free index}) = \mathbf{q}_{\text{free}}$

New estimate  $\mathbf{q}(\text{fixed index}) = \mathbf{q}_{\text{fixed}}$

# Dynamic Simulation vs Static Simulation

- Conventional Euler-Bernoulli beam theory has no time dependence
- Discrete simulations presented here include *dynamics*
  - Example: clamped beam under self-weight



Caution: Implicit Euler artificially dissipates energy. Careful about physical accuracy.

Reference: "Newmark-Beta Method in Discrete Elastic Rods Algorithm to Avoid Energy Dissipation", W Huang, MK Jawed, Journal of Applied Mechanics 86 (8)



# Discrete Elastic Bending Energy

$$E_b = \sum_{k=2}^{N-1} E_{b,k}$$

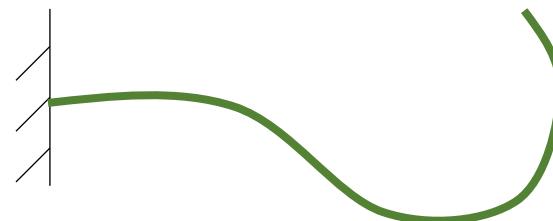
where  $E_{b,k} = \frac{1}{2} \frac{EI}{dl} (\kappa_k)^2$

What is the beam is naturally curved?



Straight beam

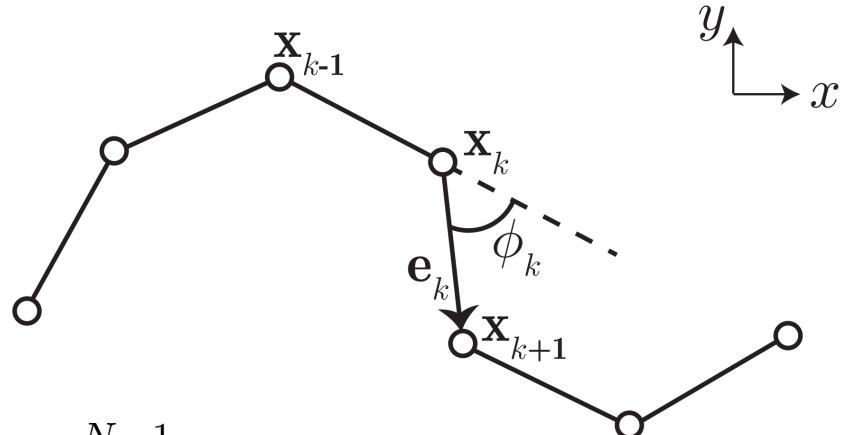
vs.

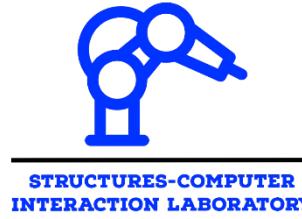


Naturally curved beam

$$E_b = \sum_{k=2}^{N-1} E_{b,k}$$

where  $E_{b,k} = \frac{1}{2} \frac{EI}{dl} (\kappa_k - \kappa_k^0)^2$





# Module 17

## Networks of Beams

- Simulation of Networks of Beams
- Application to Soft Robotics



# Goal

- Discrete Beam simulation for complex structures
- Application to soft robotics



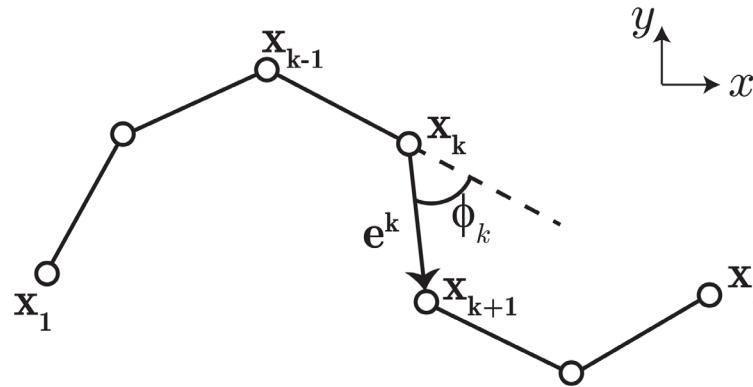
STRUCTURES-COMPUTER  
INTERACTION LABORATORY

# Sea-star Inspired Soft Robot



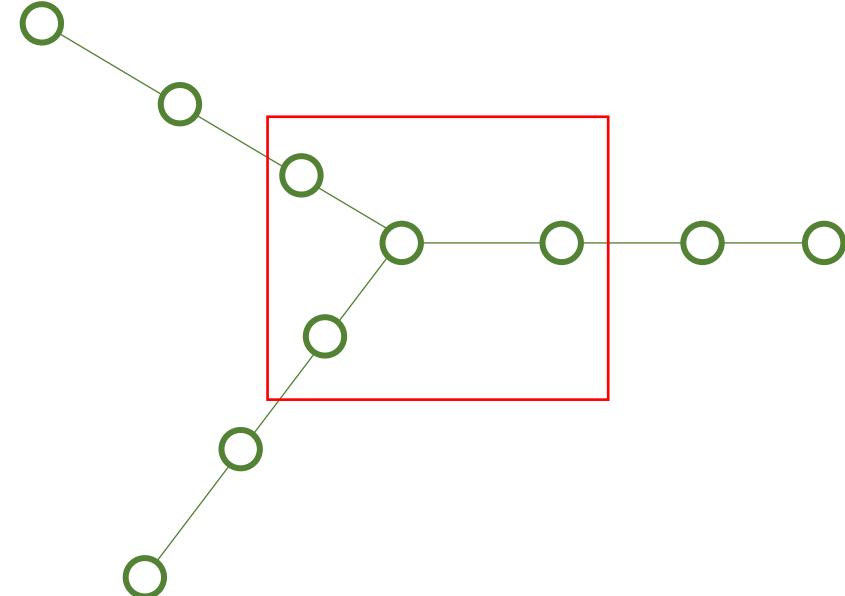


# Beam vs Soft Robot (Sea-Star)



## Elastic energy

- Bending energy between three consecutive nodes
- Stretching energy between two consecutive nodes



## Soft robot

- “Joint” needs special treatment
- Actuation has to be incorporated

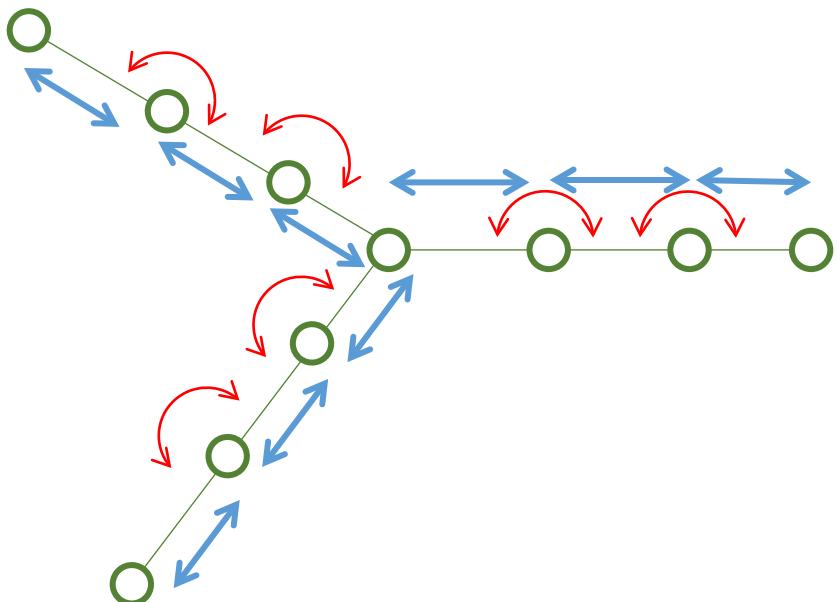


# Energies of a Soft Robot (Sea-Star)

$2N$  equations of motion for  $N$  nodes:  
$$\mathbf{M} \ddot{\mathbf{q}} + \frac{\partial E^{\text{elastic}}}{\partial \mathbf{q}} - \mathbf{F}_{\text{external}} = \mathbf{0}$$

Bending energies

Stretching energies

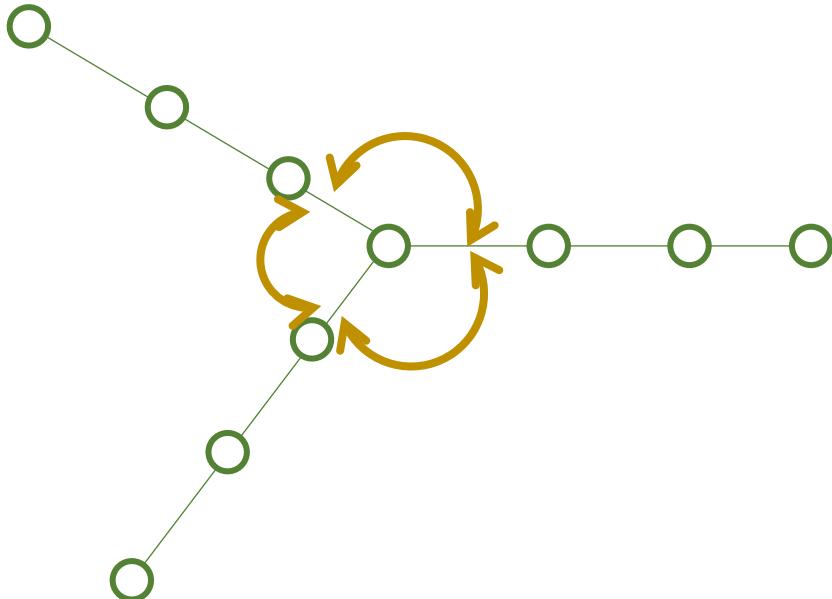




# Energies of a Soft Robot (Sea-Star)

$2N$  equations of motion for  $N$  nodes:

$$\mathbf{M} \ddot{\mathbf{q}} + \frac{\partial E^{\text{elastic}}}{\partial \mathbf{q}} - \mathbf{F}_{\text{external}} = \mathbf{0}$$



Joint bending energy

- One node has three discrete bending energies



# Energies of a Soft Robot (Sea-Star)

$2N$  equations of motion for  $N$  nodes:

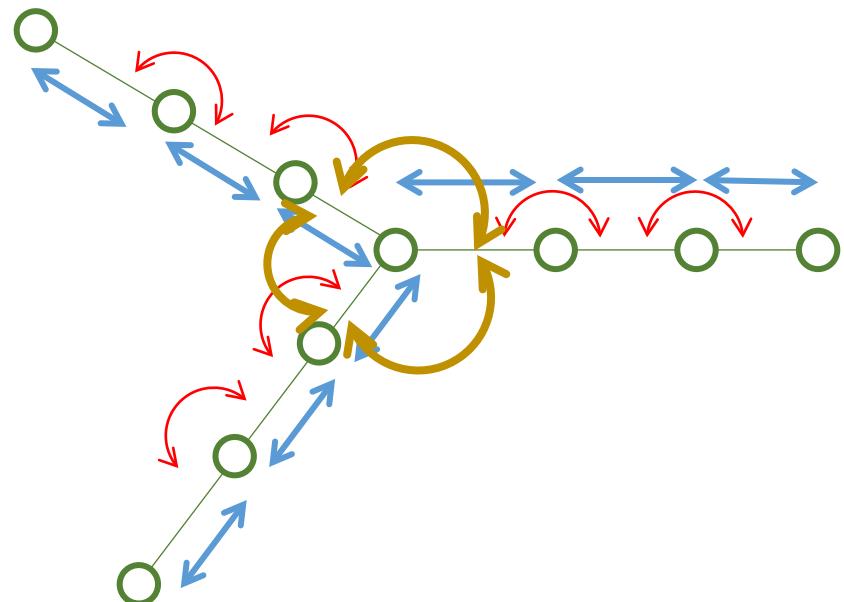
$$\mathbf{M} \ddot{\mathbf{q}} + \frac{\partial E^{\text{elastic}}}{\partial \mathbf{q}} - \mathbf{F}_{\text{external}} = \mathbf{0}$$

Bending energies

Stretching energies

Summary:

Elastic energy = bending energy + stretching energy,  
with special treatment for the joints.



Joint bending energy

- One node has three discrete bending energies

# Actuation: vary natural curvature with time

Recall bending energy formulation:

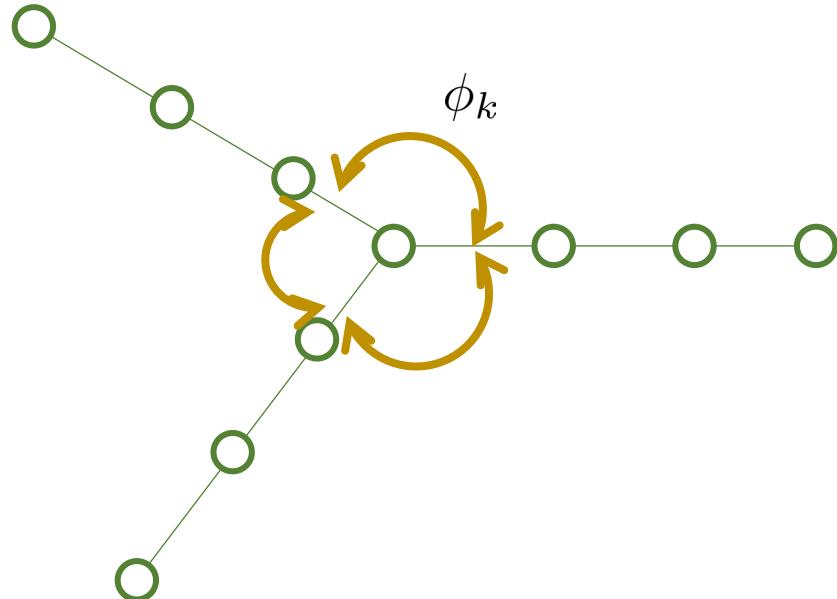
Total bending energy,  $E_b = \sum E_{b,k}$

$$E_{b,k} = \frac{1}{2} \frac{EI}{dl} (\kappa_k - \kappa_k^0)^2$$

Discrete curvature,  $\kappa_k = 2 \tan(\phi_k/2)$

Natural curvature varies with time for the bending energies associated with the head:

$$E_{b,k} = \frac{1}{2} \frac{EI}{dl} (\kappa_k - \boxed{\kappa_k^0(t)})^2$$





# External Forces

$2N$  equations of motion for  $N$  nodes:

$$\mathbf{M} \ddot{\mathbf{q}} + \frac{\partial E^{\text{elastic}}}{\partial \mathbf{q}} - \mathbf{F}_{\text{external}} = \mathbf{0}$$

- External forces, e.g. hydrodynamic forces, can be incorporated using the  $\mathbf{F}_{\text{external}}$  term in the equations of motion
  - A simple force model at low Reynolds number:

$$\mathbf{F}_{\text{external}} = \begin{bmatrix} \dots \\ \dots \\ -\mu dl \left( \frac{q_i(t_{k+1}) - q_i(t_k)}{\Delta t} \right) \\ \dots \\ \dots \end{bmatrix},$$

where  $\mu$  is viscosity (Pa-s) and  $dl$  is the length of an edge



# Recap: Beam to Soft Robots

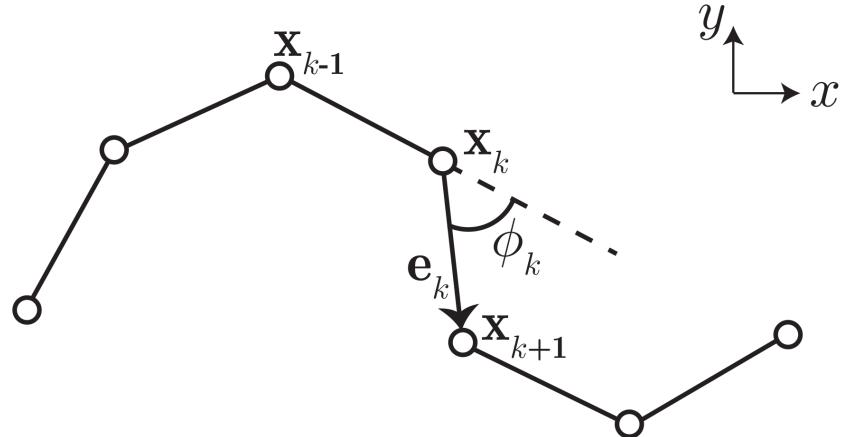
1. Special treatment for the joints
2. Actuation
3. External forces



# Discrete Elastic Bending Energy

$$E_b = \sum_{k=2}^{N-2} E_{b,k}$$

where  $E_{b,k} = \frac{1}{2} \frac{EI}{dl} (\kappa_k - \boxed{\kappa_k^0})^2$



Curvature computed in *undeformed* (i.e. *stress-free*) configuration

In most examples, undeformed configuration is the same as initial configuration  
(i.e. shape of the beam at  $t = 0$ )

However, *undeformed* and *initial* are not necessarily the same



Naturally curved beam

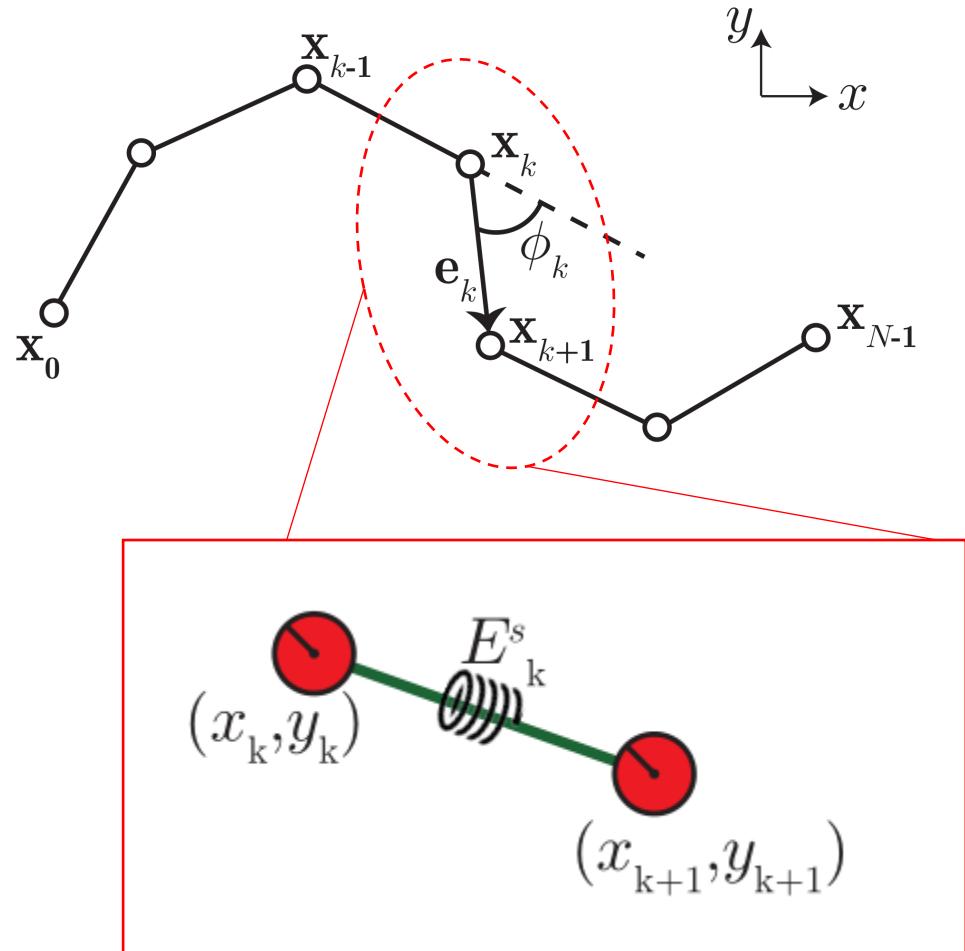


# Discrete Elastic Stretching Energy

Total elastic stretching energy:  $E_s = \sum_{k=1}^{N-1} E_s^k$

Stretching energy per edge:  $E_s^k = \frac{1}{2} EA\varepsilon^2 dl$

Axial stretch:  $\varepsilon = 1 - \frac{\sqrt{(x_{k+1}-x_k)^2+(y_{k+1}-y_k)^2}}{dl}$ ,  
where  $dl$  is the *undeformed* length of the edge

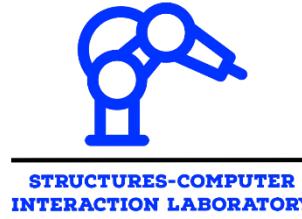




# From a Beam to a Rod

- Rod
  - 3D configuration (Beam: 2D configuration)
  - Bending energy
  - Stretching energy
  - **Twisting energy** (Beam does not twist)





# Module 18

---

Twist in a Discrete Rod

- Continuous Definition of Twist
- Discrete Definition of Twist
- Parallel Transport



# Goal

- Kinematics of a rod and the need for a material frame
- Twist of a rod
- Discrete representation of twist
- Orthonormal Adapted Frame
- Parallel Transport

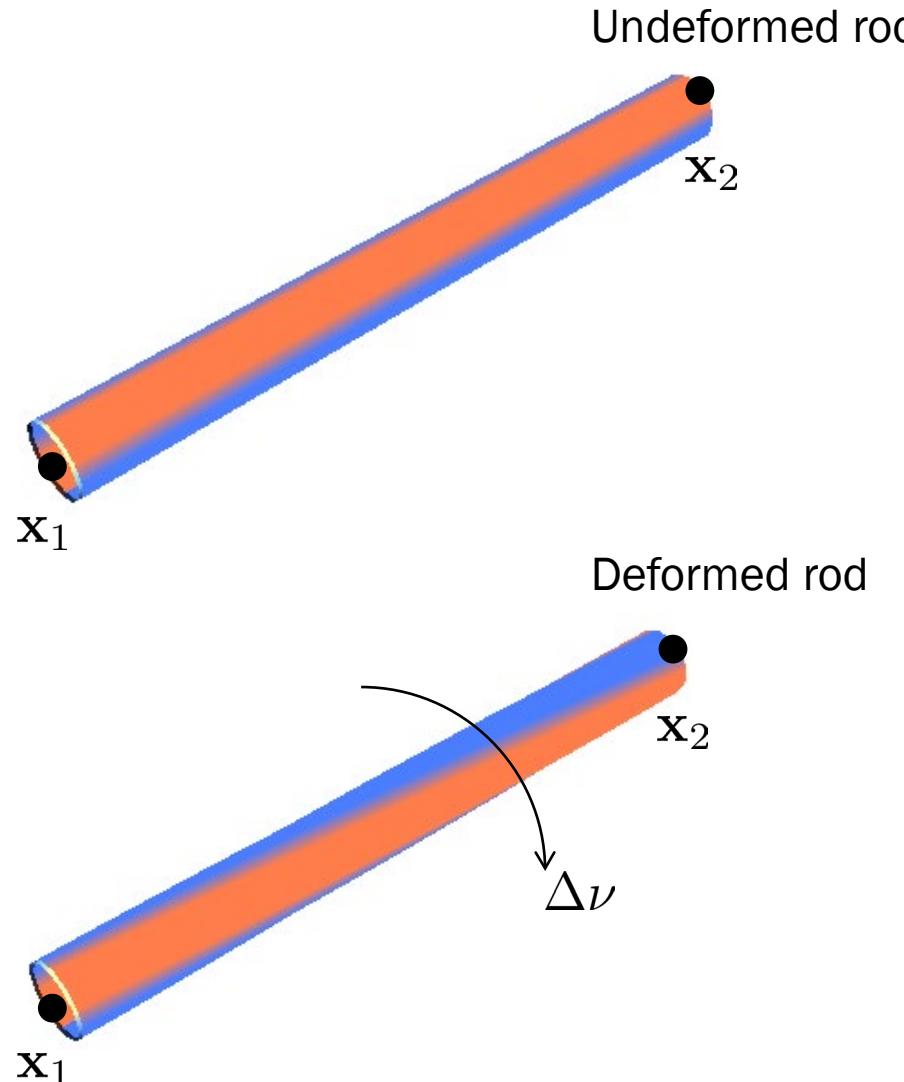


# Resources

- Read “*Chapter 6: Discrete Twist*”
- Review undergraduate-level mechanics
  - *Chapter 6: Torsion of Prismatic Bars of Advanced Mechanics of Materials and Applied Elasticity* by A. C. Ugural and Saul Fenster



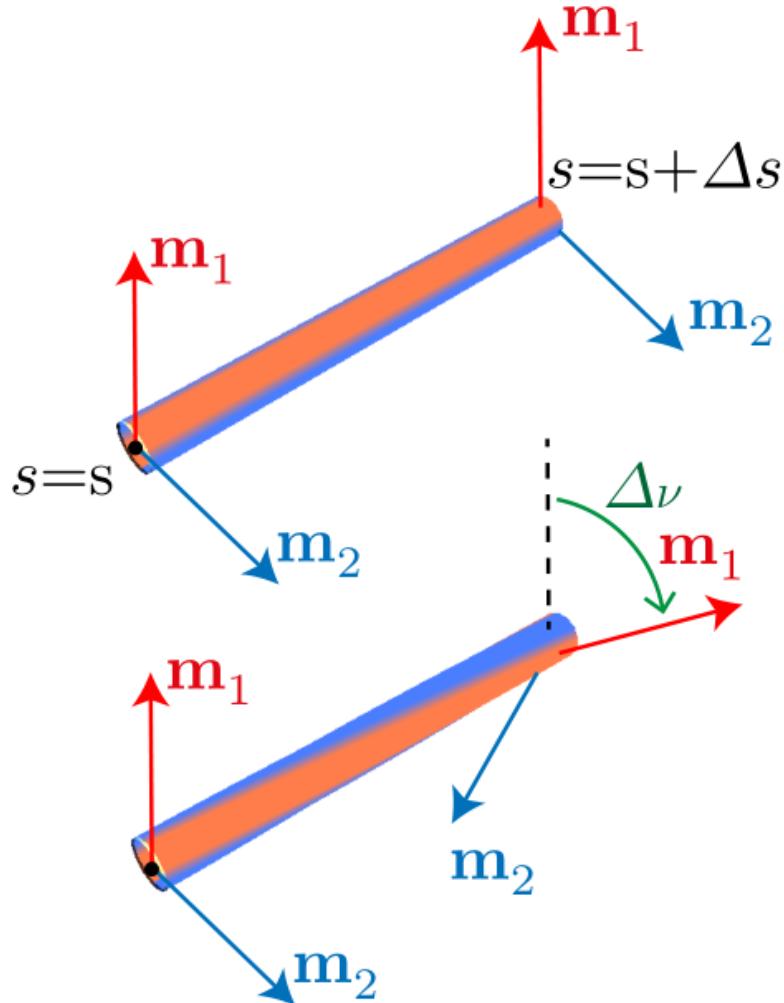
# Nodes and its coordinates are not enough!



If only nodes on the centerline are used to describe the kinematics of a rod, The energy will remain the same between the undeformed and deformed configurations.



# Twist: Continuous Case



Continuous twist [unit: 1/meter]:

$$\tau(s) = \lim_{\Delta s \rightarrow 0} \frac{\Delta\nu}{\Delta s},$$

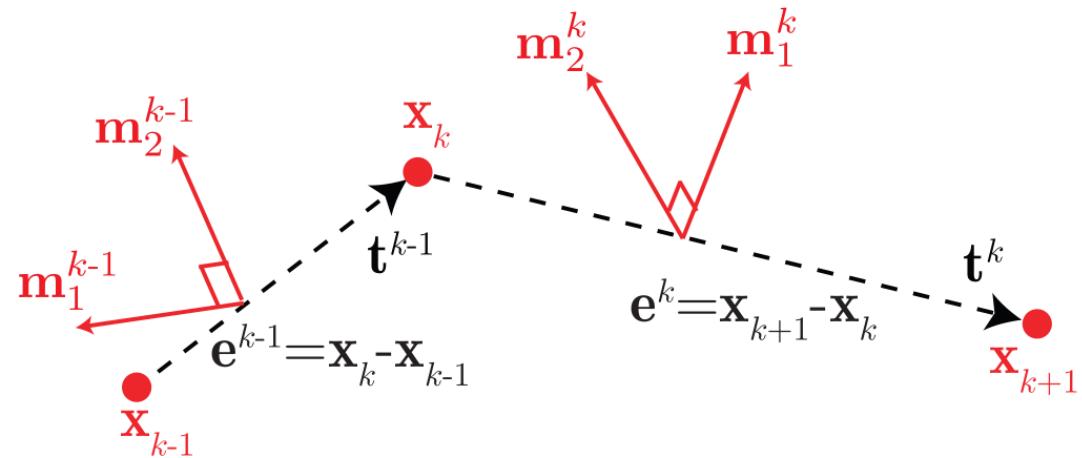


# Twist: Discrete Case

- We will need three preliminary concepts before explaining the twist in a discrete rod
  1. Orthonormal adapted frame
  2. Signed angle
  3. Parallel transport



# 1. Orthonormal adapted frame



$N$  nodes long the centerline,  $\mathbf{x}_k$  ( $1 \leq k \leq N$ )

$N - 1$  edges long the centerline,  $\mathbf{e}^k = \mathbf{x}_{k+1} - \mathbf{x}_k$

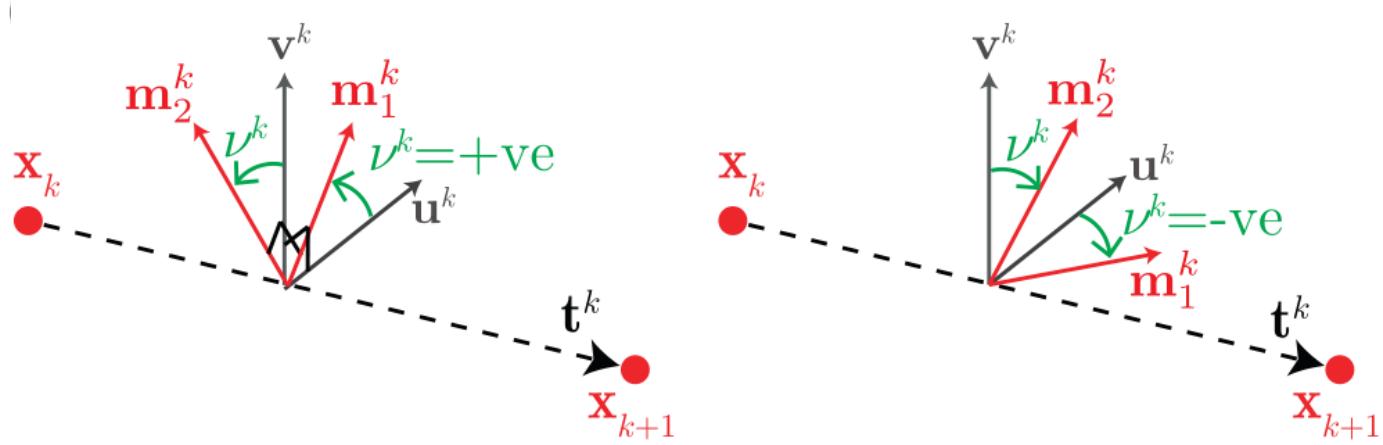
Tangent on the  $k$ -th edge is  $\mathbf{t}^k = \frac{\mathbf{e}^k}{\|\mathbf{e}^k\|}$

Adapted frame is any frame  $(\mathbf{m}_1^k, \mathbf{m}_2^k, \mathbf{t}^k)$  that shares the tangent as the third director

Orthonormal means the three directors  $(\mathbf{m}_1^k, \mathbf{m}_2^k, \mathbf{t}^k)$  are unit vectors and perpendicular to one another



## 2. Signed angle



Consider two frames:

1. Material frame  $(m_1^k, m_2^k, t^k)$
2. Reference frame  $(u^k, v^k, t^k)$  (this frame is also orthonormal adapted)

The angle from  $u^k$  to  $m_1^k$  is  $\nu^k$ .

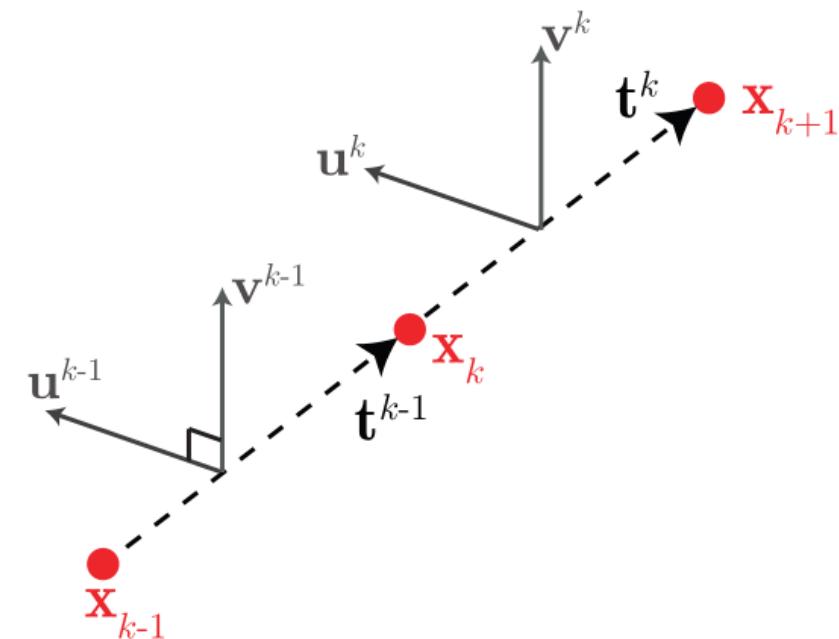
This angle should have a “direction”  $t^k$  associated with it.

The sign of the angle,  $\nu^k$ , can be determined using right hand rule.



# 3. Parallel transport

- What is the *most natural* or *twist free* way of moving an adapted frame one edge ( $e^{k-1}$ ) to the next ( $e^k$ )?
  - Easy if the two edges are aligned
  - $(u^{k-1}, v^{k-1}, t^{k-1})$  and  $(u^k, v^k, t^k)$  are the same

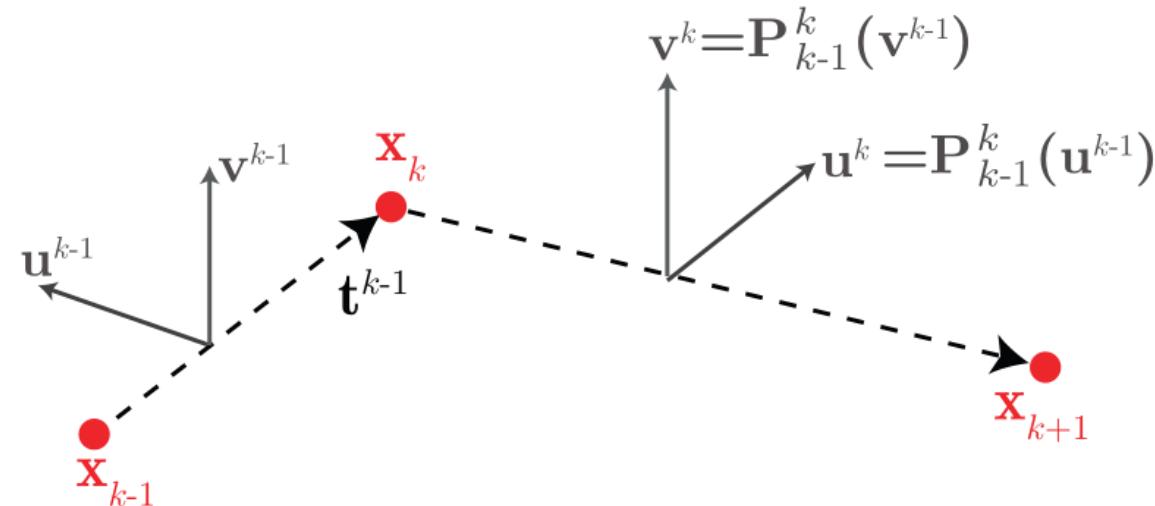


# 3. Parallel transport

- What is the *most natural* or *twist free* way of moving an adapted frame one edge ( $e^{k-1}$ ) to the next ( $e^k$ )?
  - If the two edges are not aligned, we need *parallel transport*.

$$P_{k-1}^k(\mathbf{u}) \equiv \text{parallel\_transport}(\mathbf{u}, \mathbf{t}^{k-1}, \mathbf{t}^k)$$

$P_{k-1}^k(\mathbf{u})$  is the vector obtained after moving  $\mathbf{u}$  from  $\mathbf{t}^{k-1}$  to  $\mathbf{t}^k$  without any twist (tangential angular velocity).  $P_{k-1}^k(\mathbf{u})$  is perpendicular to  $\mathbf{t}^k$  and  $\mathbf{u}$  is perpendicular to  $\mathbf{t}^{k-1}$ .



# 3. Parallel transport

- What is the *most natural* or *twist free* way of moving an adapted frame one edge ( $e^{k-1}$ ) to the next ( $e^k$ )?
  - If the two edges are not aligned, we need *parallel transport*.

$$P_{k-1}^k(\mathbf{u}) \equiv \text{parallel\_transport}(\mathbf{u}, \mathbf{t}^{k-1}, \mathbf{t}^k)$$

$$\mathbf{b} = \mathbf{t}^{k-1} \times \mathbf{t}^k,$$

$$\hat{\mathbf{b}} = \frac{\mathbf{b}}{|\mathbf{b}|},$$

$$\mathbf{n}_1 = \mathbf{t}^{k-1} \times \hat{\mathbf{b}},$$

$$\mathbf{n}_2 = \mathbf{t}^k \times \hat{\mathbf{b}},$$

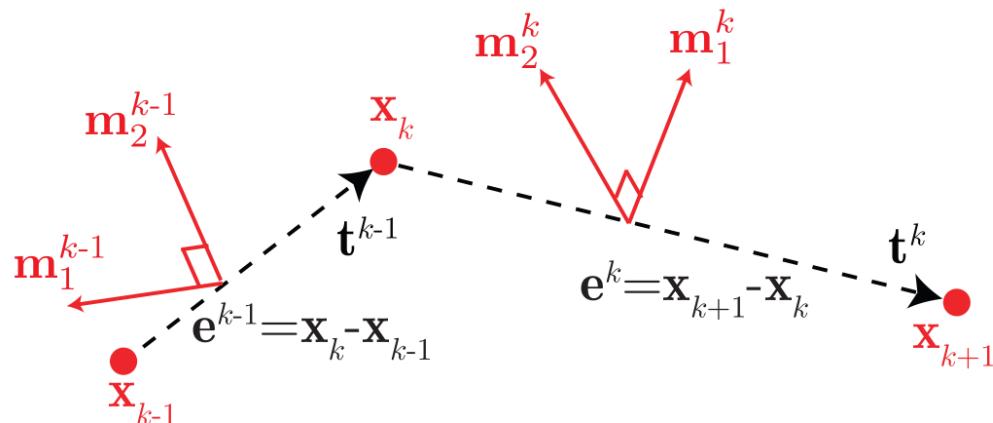
$$P_{k-1}^k(\mathbf{u}) = (\mathbf{u} \cdot \mathbf{t}^{k-1})\mathbf{t}^k + (\mathbf{u} \cdot \mathbf{n}_1)\mathbf{n}_2 + (\mathbf{u} \cdot \hat{\mathbf{b}})\hat{\mathbf{b}}.$$

# Twist: Discrete Case

The integrated discrete twist (or, more simply, *twist*) of an adapted frame [e.g.  $(\mathbf{m}_1^k, \mathbf{m}_2^k, \mathbf{t}^k)$  with  $k = 1, \dots, N - 1$ ] at node  $\mathbf{x}_k$  is the signed angle from  $P_{k-1}^k(\mathbf{m}_1^{k-1})$  to  $\mathbf{m}_1^k$  about the axis  $\mathbf{t}^k$ .

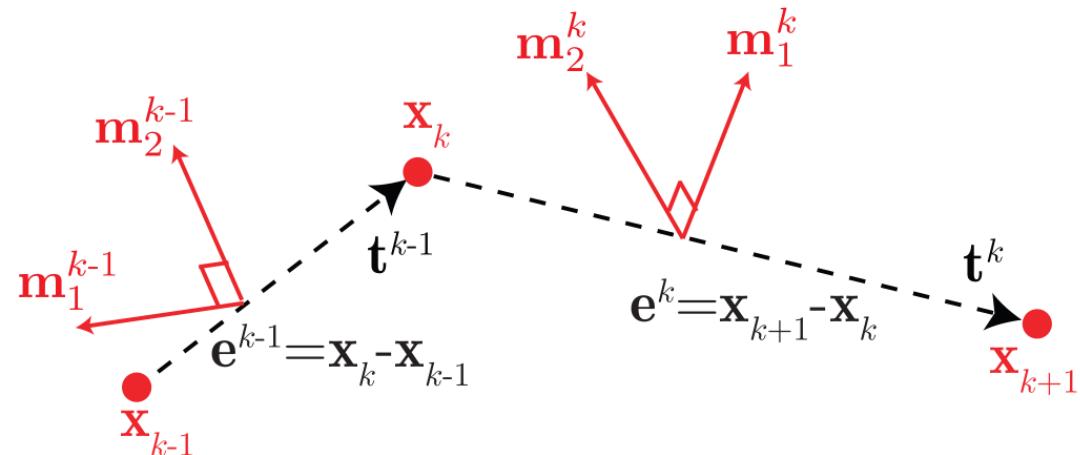
The word *integrated*, which may often be omitted, is used to distinguish this dimensionless measure of twist from the conventional definition of twist with unit of 1/length.

The integrated twist can be divided by the *Voronoi length* of the node to obtain the conventional twist, where the Voronoi length  $\bar{l}_k = (|\bar{\mathbf{e}}^{k-1}| + |\bar{\mathbf{e}}^k|) / 2$  and  $(\bar{\ })$  represents evaluation in undeformed state.





# Discrete Elastic Twisting Energy



$$E_t = \sum_{k=2}^{N-1} E_{t,k}$$

$$\text{where } E_{t,k} = \frac{1}{2} \frac{GJ}{l_k} (\tau_k)^2$$

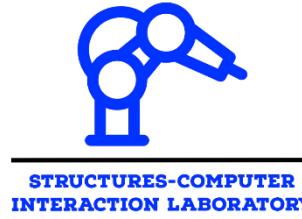
- $G$  = shear modulus =  $\frac{\text{Young's modulus}}{2(1+\text{Poisson ratio})}$
- $J$  = polar moment of area ( $= \frac{\pi r_0^4}{2}$  if circular)
- $\tau_k$  = signed angle from  $P_{k-1}^k(\mathbf{m}_1^{k-1})$  to  $\mathbf{m}_1^k$  about the axis  $\mathbf{t}^k$



# Recall: From a Beam to a Rod

- Rod
  - 3D configuration (Beam: 2D configuration)
  - Bending energy
  - Stretching energy
  - **Twisting energy** (Beam does not twist)



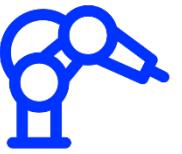


# Module 19

---

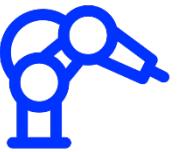
Space Parallel Reference Frame

- Definition of twist using space parallel reference frame



# Goal

- Formulate a reference frame on the rod centerline using parallel transport along the arc-length (space parallel transport)
- Compute the twist along the rod using space parallel reference frame
- At the end of this module, you should be able to compute the twisting energy of a rod given its centerline and material frame

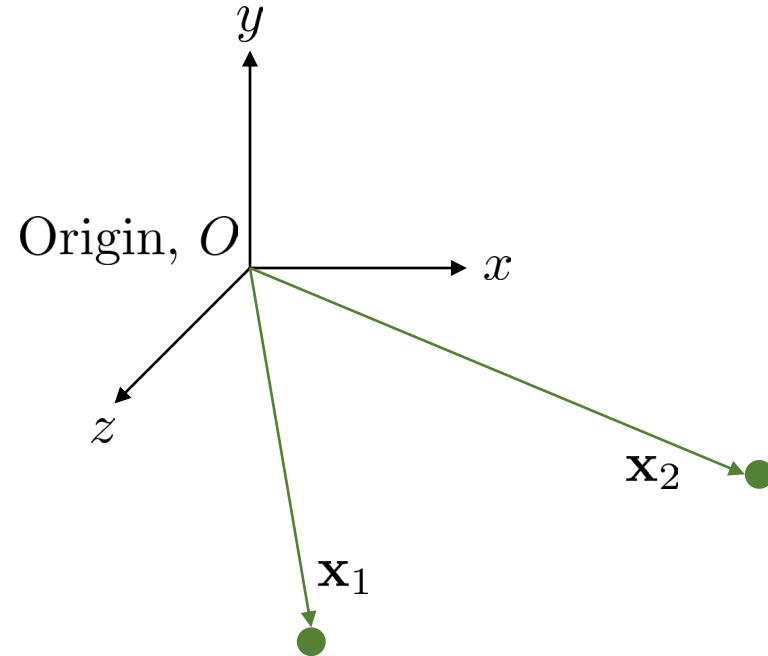


# Resources

- Read “*Chapter 6: Discrete Twist*”
- Review Bergou, Miklós, et al. “Discrete elastic rods” ACM SIGGRAPH (2008) 1-12.



# Relative Position



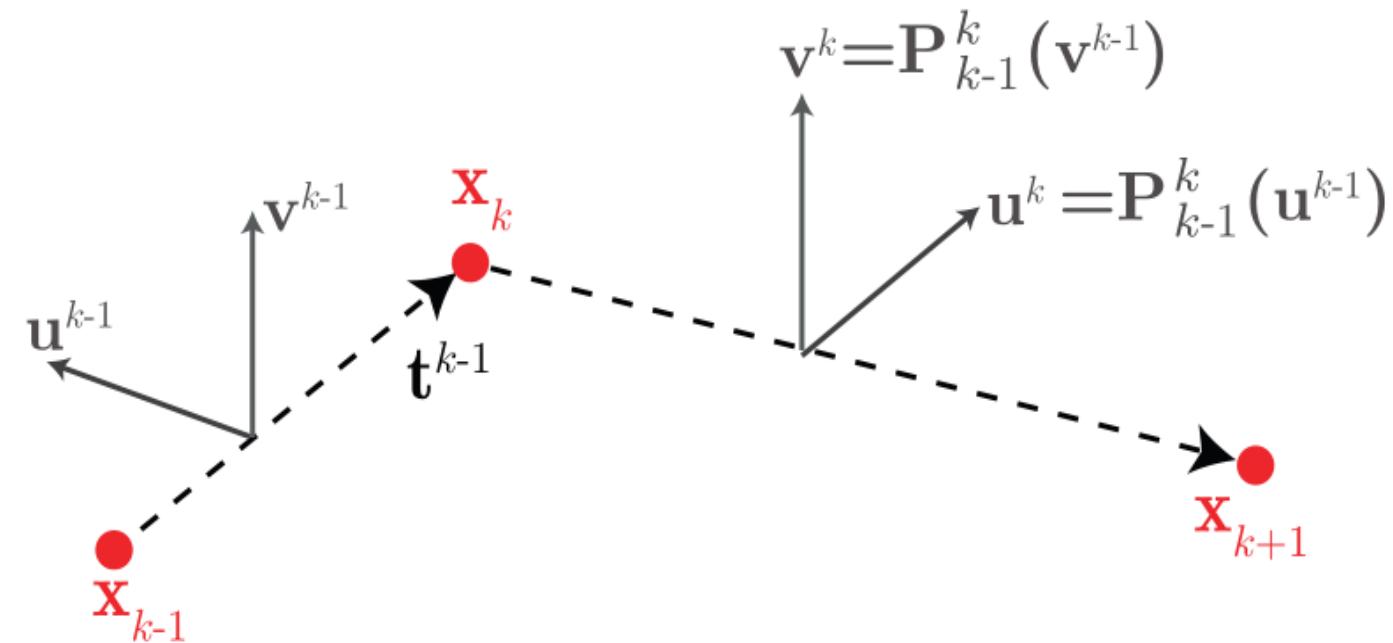
When describing the nodal coordinates, e.g.  $x_1$ , we implicitly assume a reference frame with origin  $O$ .



# Relative Rotation - Twist

When describing the rotation of the material frame, we need a reference frame

- How to compute the reference frame,  $(\mathbf{u}^k, \mathbf{v}^k, \mathbf{t}^k)$ ?
- Known quantities: the nodal coordinates,  $\mathbf{x}_k$





# Reference Frame

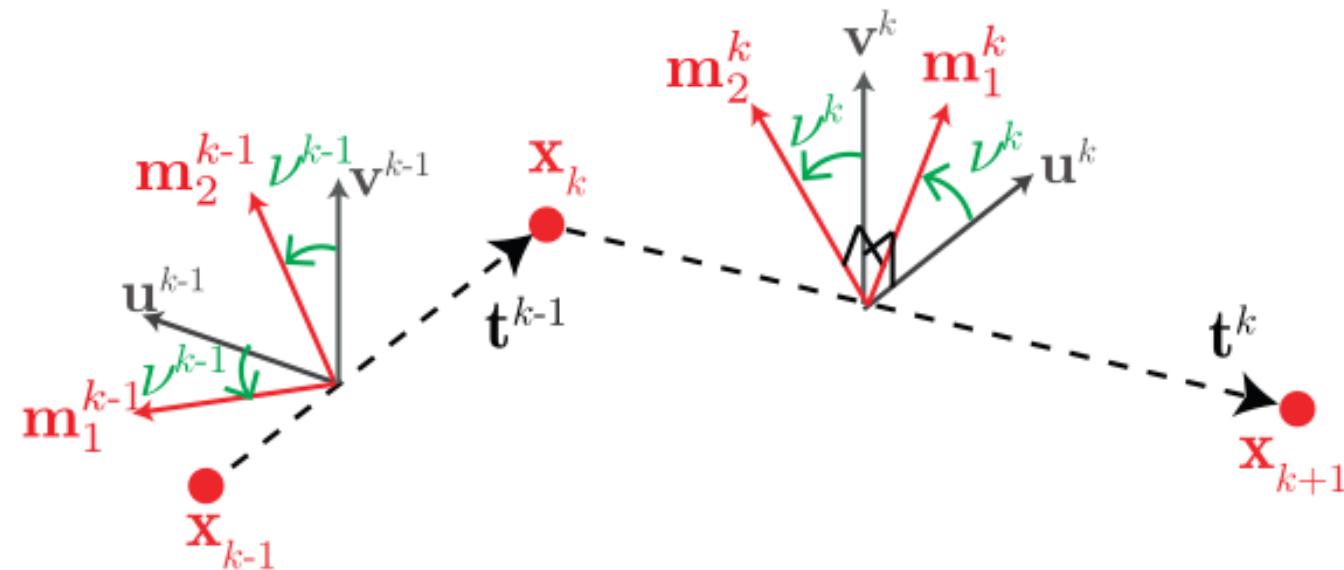
- Since the positions of the nodes are known, the tangent on each edge is known:

$$\mathbf{t}^k = \frac{\mathbf{e}^k}{\|\mathbf{e}^k\|} = \frac{\mathbf{x}_{k+1} - \mathbf{x}_k}{\|\mathbf{x}_{k+1} - \mathbf{x}_k\|}$$

- Start with an arbitrary (but adapted) orthonormal frame at the first edge,  $\mathbf{e}^1$ . For that purpose, choose any unit vector  $\mathbf{u}^1$  that is orthogonal to  $\mathbf{t}^1$ . The second director is  $\mathbf{v}^1 = \mathbf{t}^1 \times \mathbf{u}^1$  and the frame is  $(\mathbf{u}^1, \mathbf{v}^1, \mathbf{t}^1)$ .
- Sequentially compute the reference frames for the subsequent edges using  $\mathbf{u}^k = P_{k-1}^k(\mathbf{u}^{k-1}) \equiv \text{parallel\_transport}(\mathbf{u}^{k-1}, \mathbf{t}^{k-1}, \mathbf{t}^k)$  and  $\mathbf{v}^k = \mathbf{t}^k \times \mathbf{u}^k$ .

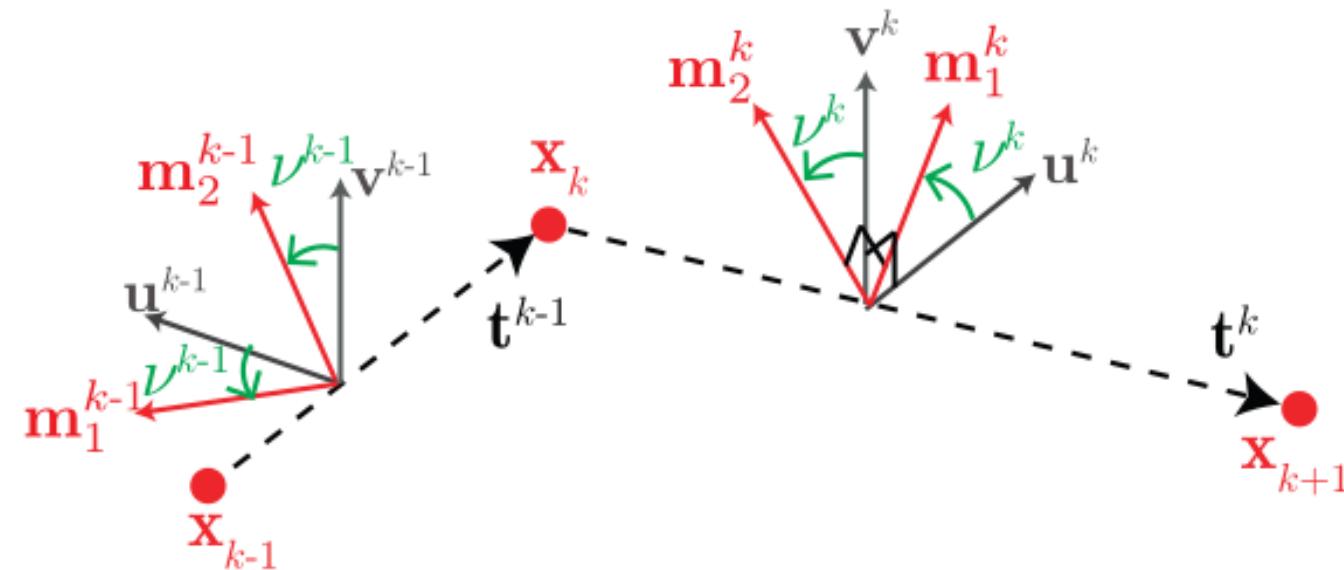
# How to compute the discrete integrated twist?

- Known quantities: configuration of the rod
  - Nodal coordinates,  $x_k$
  - Material frame on each edge,  $(m_1^k, m_2^k, t^k)$
  - We learned how to compute the reference frame,  $(u^k, v^k, t^k)$ , using space parallel transport



# How to compute the discrete integrated twist?

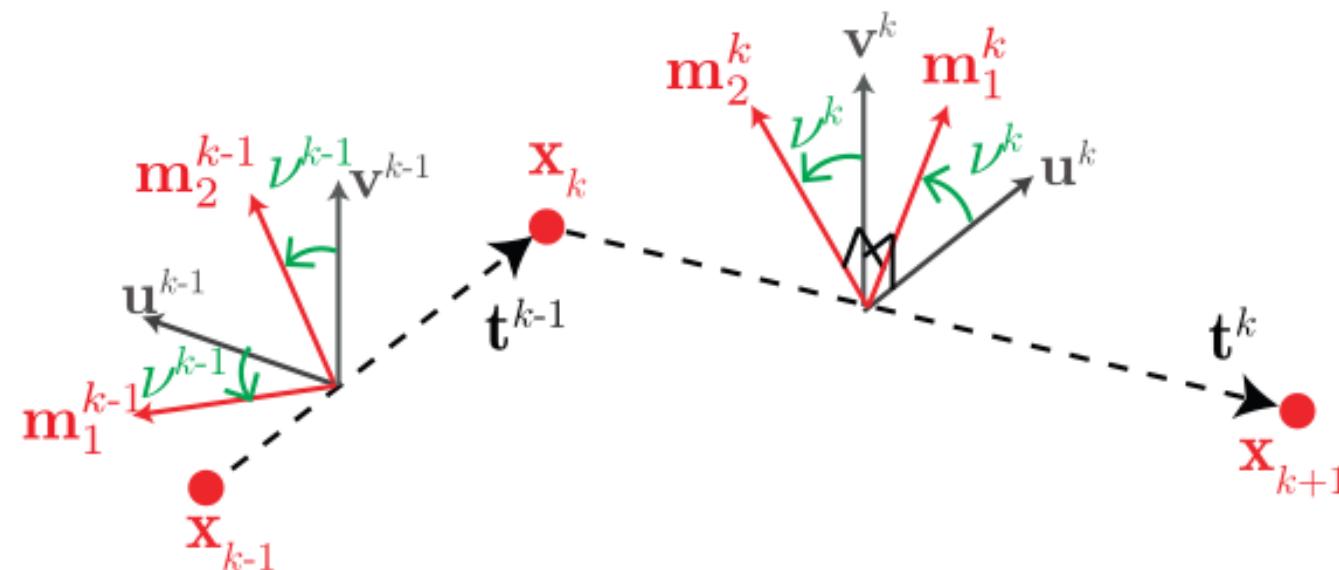
- Steps to compute the twist at the node,  $x_k$ 
  1. Compute the signed angles from the reference frame to the material frame,  $\nu^k$
  2. Discrete integrated twist is the difference between the signed rotation angles of two consecutive edges:  $\tau_k = \nu^k - \nu^{k-1}$





# Degrees of Freedom of a Rod

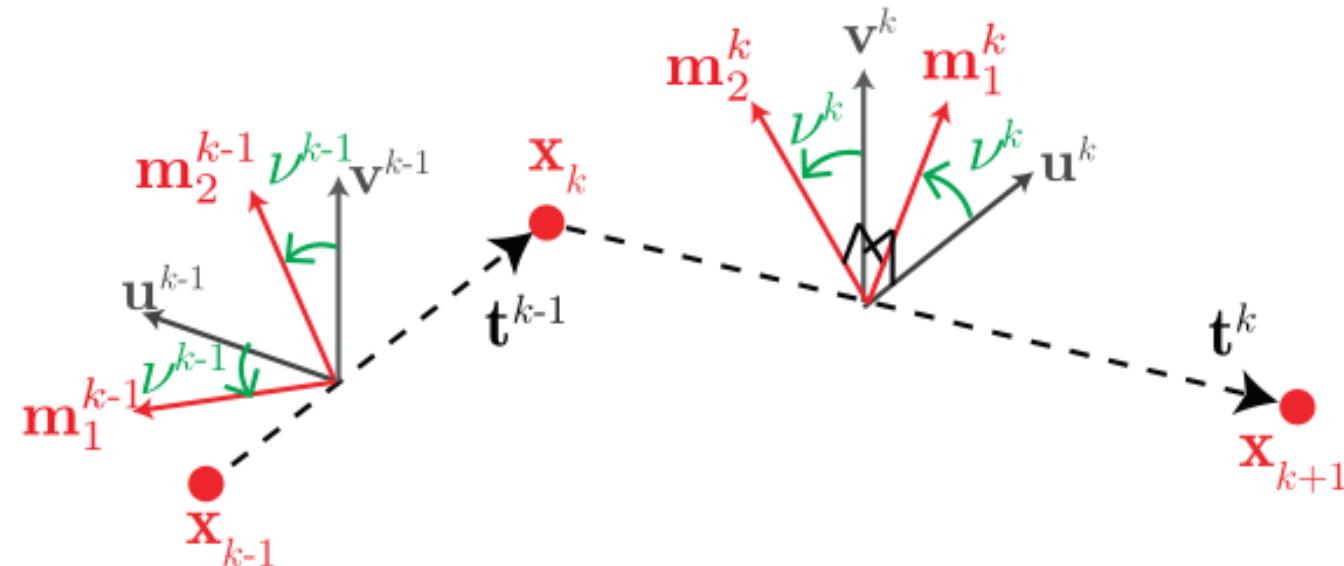
- If only the nodal coordinates,  $x_k$ , and twist angles,  $\nu_k$ , are known, the complete configuration of the rod can be determined
  - First, determine the space parallel reference frame
  - Rotate the reference frame director,  $u^k$ , by an angle  $\nu_k$  to get the material frame director,  $m_1^k$ . Recall that  $m_2^k = t^k \times m_1^k$ .





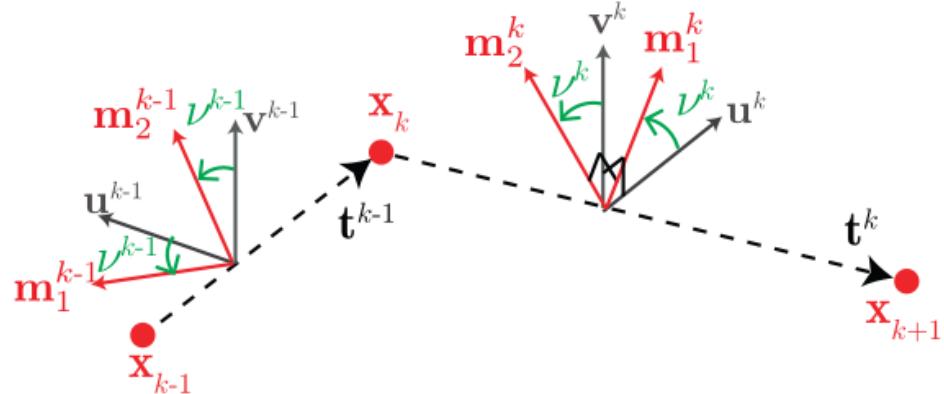
# Degrees of Freedom (DOF) of a Rod

- A rod with  $N$  nodes has  $(4N-1)$  DOFs
  1.  $3N$  DOFs associated with the positions of  $N$  nodes,  $x_k$  ( $1 \leq k \leq N$ )
  2.  $(N-1)$  DOFs for the twist angles of the edges,  $\nu^k$  ( $1 \leq k \leq N - 1$ )





# Discrete Elastic Twisting Energy



$$E_t = \sum_{k=2}^{N-1} E_{t,k}$$

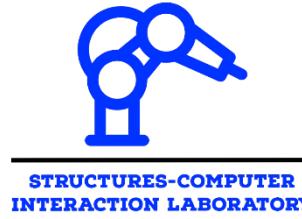
$$\text{where } E_{t,k} = \frac{1}{2} \frac{GJ}{l_k} (\tau_k)^2 = \frac{1}{2} \frac{GJ}{l_k} (\nu^k - \nu^{k-1})^2$$

- $G$  = shear modulus =  $\frac{\text{Young's modulus}}{2(1+\text{Poisson ratio})}$
- $J$  = polar moment of area ( $= \frac{\pi r_0^4}{2}$  if circular)
- $\tau_k$  = signed angle from  $P_{k-1}^k(\mathbf{m}_1^{k-1})$  to  $\mathbf{m}_1^k$  about the axis  $\mathbf{t}^k$



# Future Plan

- Bergou et al. *SIGGRAPH* (2008) used space-parallel reference frame
- In 2010 (Bergou et al. *SIGGRAPH* (2010)), the authors formulated another formulation of reference frame: time-parallel reference frame
  - This leads to significant speed-up in computation
  - In this class, we will follow the formulation in the 2010 version and discuss time-parallel reference frame in next module



# Module 20

---

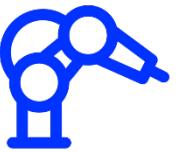
Time Parallel Reference Frame

- Definition of twist using time parallel reference frame



# Goal

- Formulate a reference frame on the rod centerline using parallel transport along time (time parallel transport)
- Compute the twist along the rod using time parallel reference frame
- At the end of this module, you should be able to compute the twisting energy of a rod using time parallel reference frame



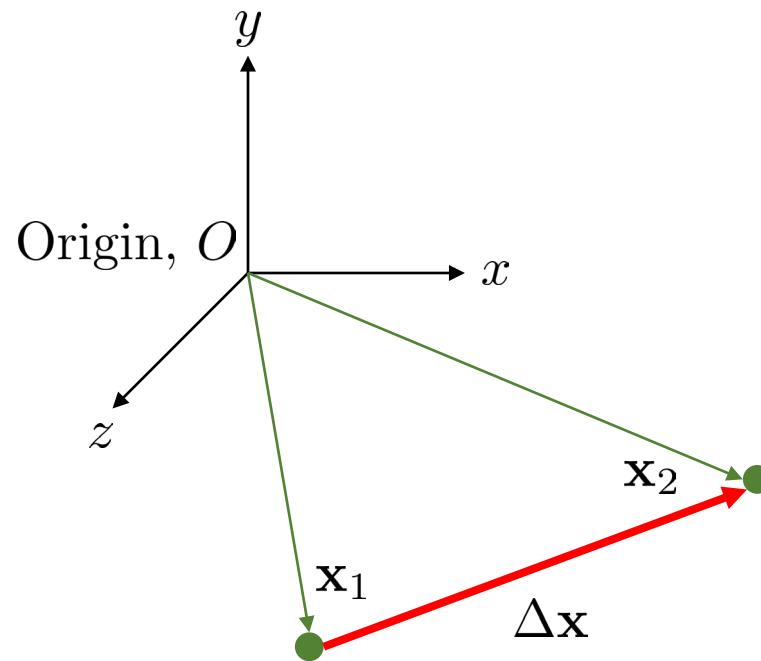
# Resources

- Read “*Chapter 6: Discrete Twist*”
- Review Bergou, Miklós, et al. “Discrete viscous threads” SIGGRAPH (2010) 29(4) 1-10



# Warm up: Relative Position

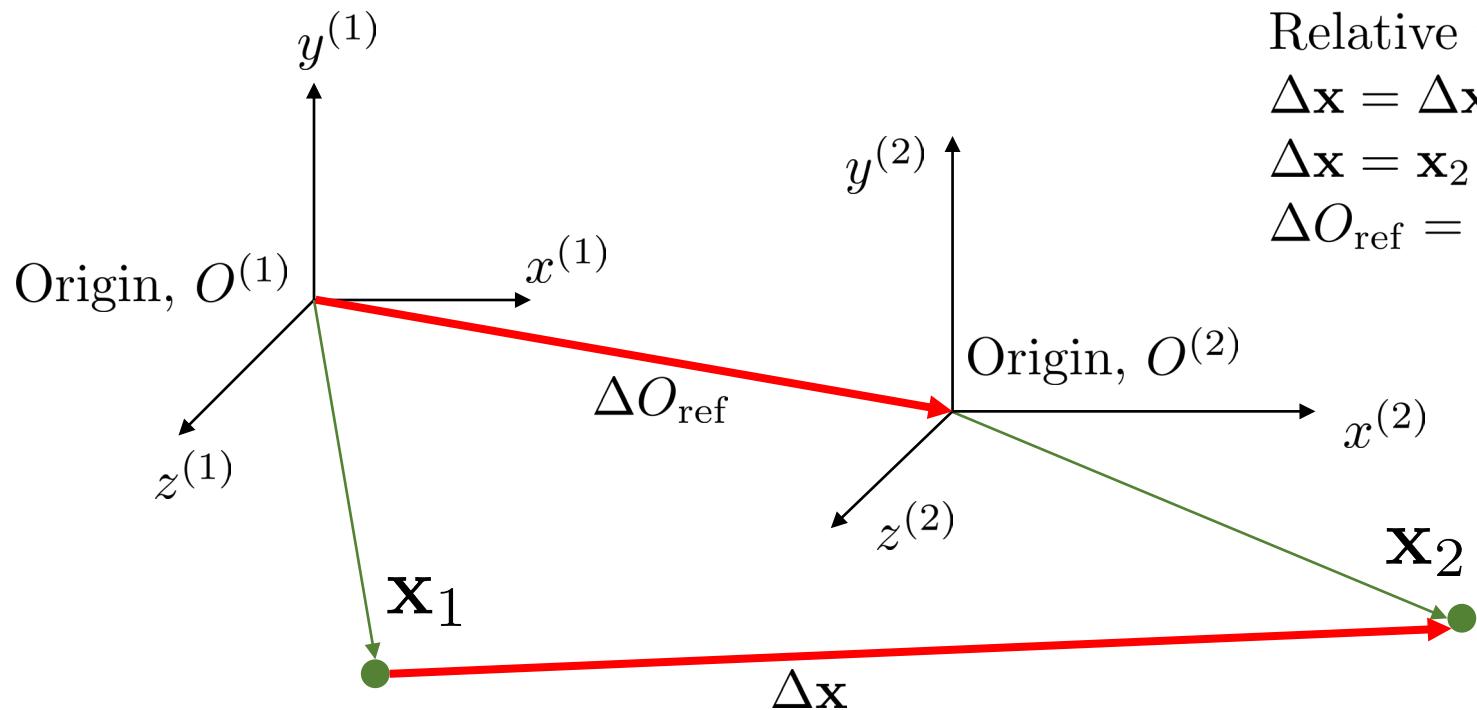
Relative position of node # 2 with respect to # 1:  
 $\Delta\mathbf{x} = \mathbf{x}_2 - \mathbf{x}_1$





# Warm up: Relative Position

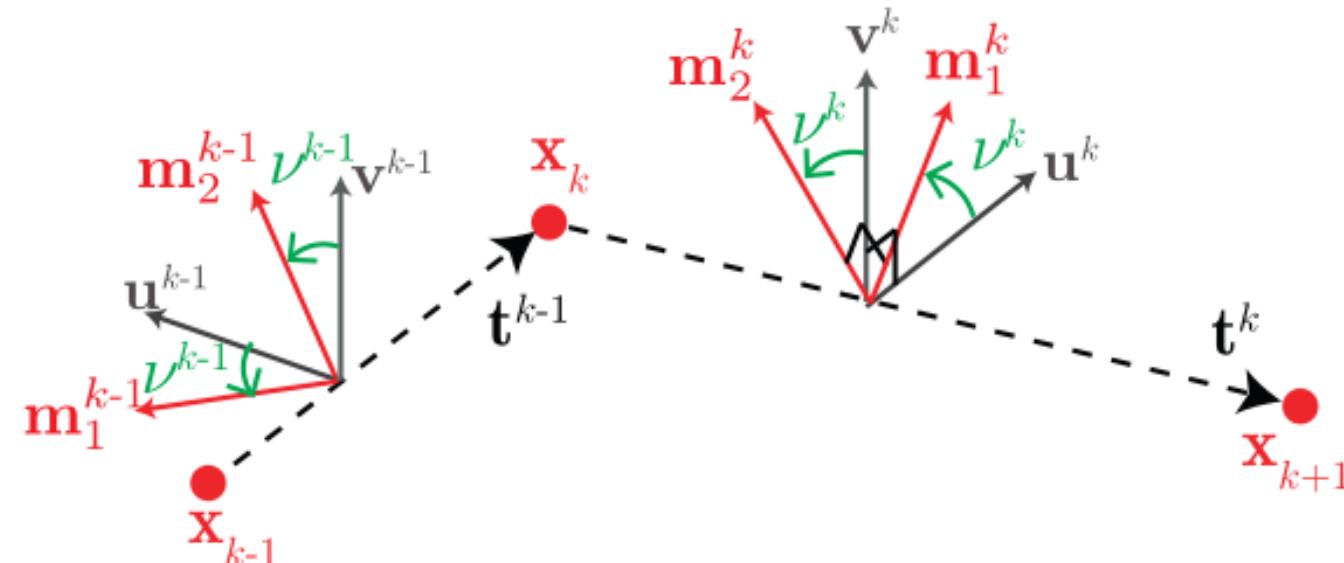
What if  $\mathbf{x}_1$  and  $\mathbf{x}_2$  are measured in two different Cartesian frames?



Relative position of node # 2 with respect to # 1:  
 $\Delta\mathbf{x} = \mathbf{x}_2 - \mathbf{x}_1$  and  
 $\Delta O_{\text{ref}} = \text{"reference" displacement}$

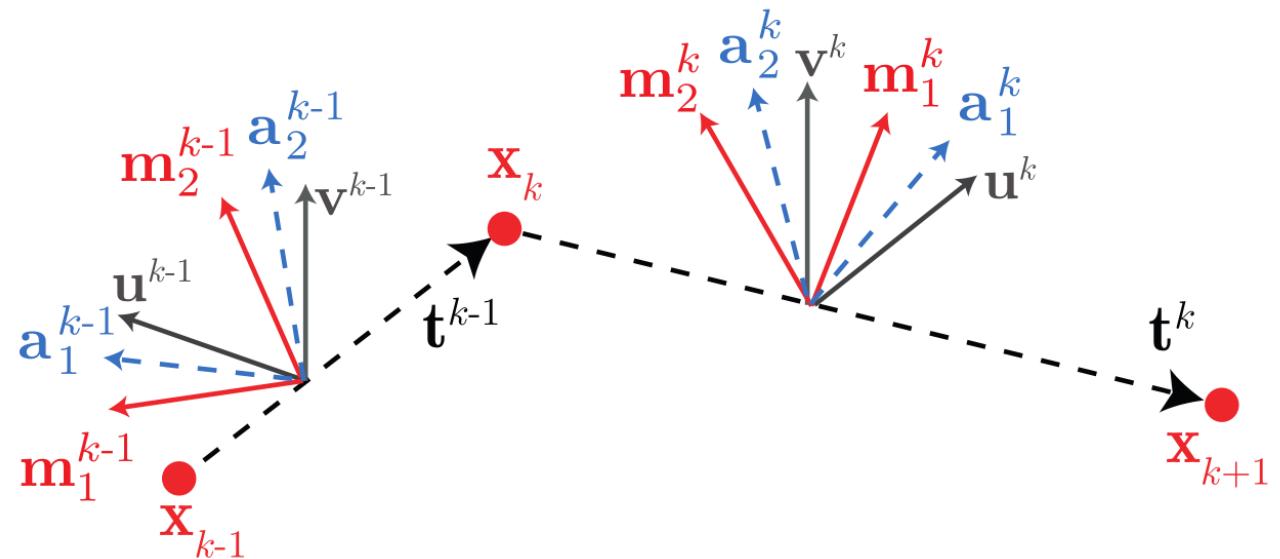
# Recall: Twist using Space-Parallel Reference Frame

- The frame from the  $(k - 1)$ -th edge is transported *without twist* to the  $k$ -th edge
- Twist between  $(\mathbf{u}^{k-1}, \mathbf{v}^{k-1}, \mathbf{t}^{k-1})$  and  $(\mathbf{u}^k, \mathbf{v}^k, \mathbf{t}^k)$  is zero (analogous to  $\Delta O_{\text{ref}} = 0$ )
- Discrete twist at the  $k$ -th node ( $\mathbf{x}_k$ ) is  $\tau_k = \nu^k - \nu^{k-1}$  (analogous to  $\Delta x = \mathbf{x}_2 - \mathbf{x}_1$ )



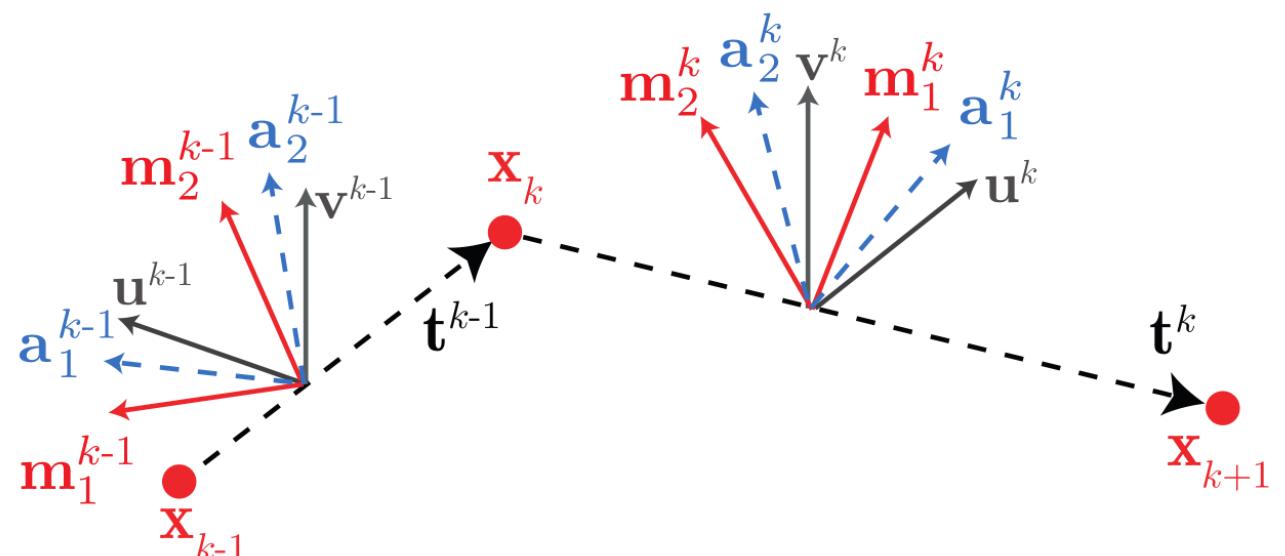
# Time-Parallel Reference Frame

- At time  $t = t_{j+1}$ , space-parallel reference frame is  
 $\mathbf{u}^k(t_{j+1}) = \text{parallel\_transport}(\mathbf{u}^{k-1}(t_{j+1}), \mathbf{t}^{k-1}(t_{j+1}), \mathbf{t}^k(t_{j+1}))$  and  
 $\mathbf{v}^k(t_{j+1}) = \mathbf{t}^k(t_{j+1}) \times \mathbf{u}^k(t_{j+1})$
- At time  $t = t_{j+1}$ , time-parallel reference frame is  
 $\mathbf{a}_1^k(t_{j+1}) = \text{parallel\_transport}(\mathbf{a}_1^k(t_j), \mathbf{t}^k(t_j), \mathbf{t}^k(t_{j+1}))$  and  
 $\mathbf{a}_2^k(t_{j+1}) = \mathbf{t}^k(t_{j+1}) \times \mathbf{a}_1^k(t_{j+1})$



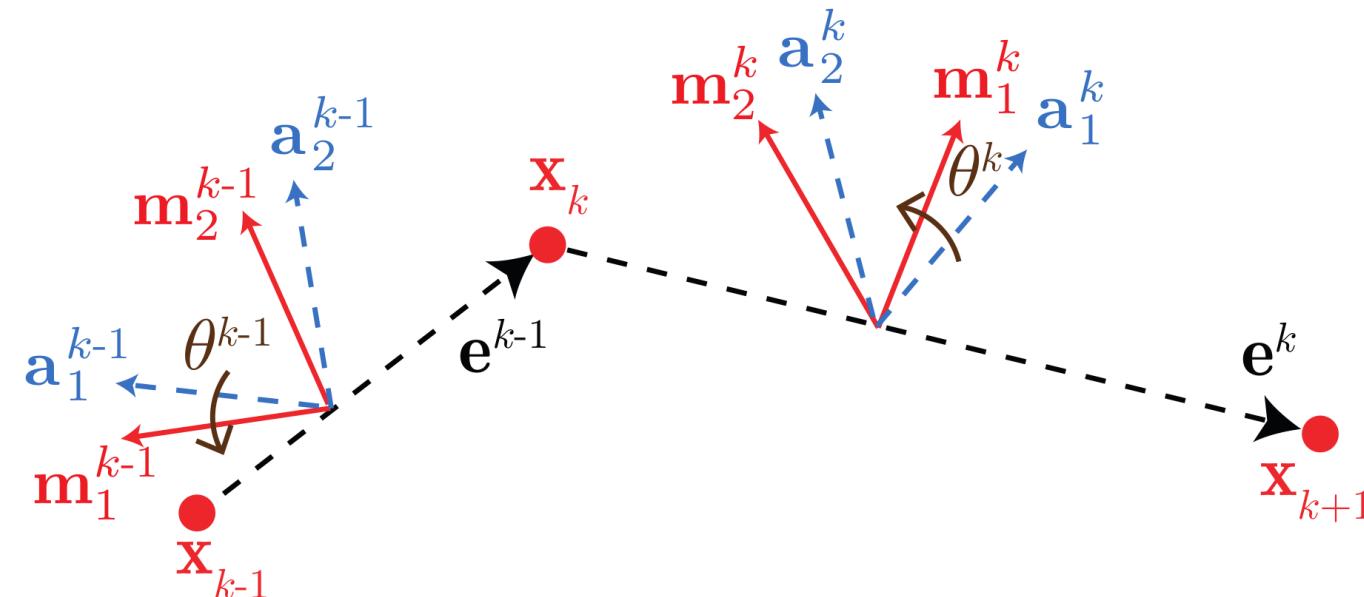
# Time-Parallel Reference Frame

- Space-parallel frame  $\mathbf{u}^k(t_{j+1}), \mathbf{v}^k(t_{j+1}), \mathbf{t}^k(t_{j+1})$  moves forward in *space*
- Time-parallel frame  $\mathbf{a}_1^k(t_{j+1}), \mathbf{a}_2^k(t_{j+1}), \mathbf{t}^k(t_{j+1})$  moves forward in *time*
- Space-parallel frame has no *reference twist*
- Time-parallel frame may have *reference twist* (analogous to  $\Delta O_{\text{ref}} \neq 0$ )



# New Definition of Twist Angle

- Recall: signed angle from space-parallel frame director  $\mathbf{u}^k(t_{j+1})$  to the material frame director  $\mathbf{m}_1^k(t_{j+1})$  is the *twist angle*  $\nu^k$
- Signed angle from time-parallel frame director  $\mathbf{a}_1^k(t_{j+1})$  to the material frame director  $\mathbf{m}_1^k(t_{j+1})$  is the new *twist angle*  $\theta^k$
- We will use  $\theta^k$  ( $1 \leq k < N$ ) as the DOFs of the rod

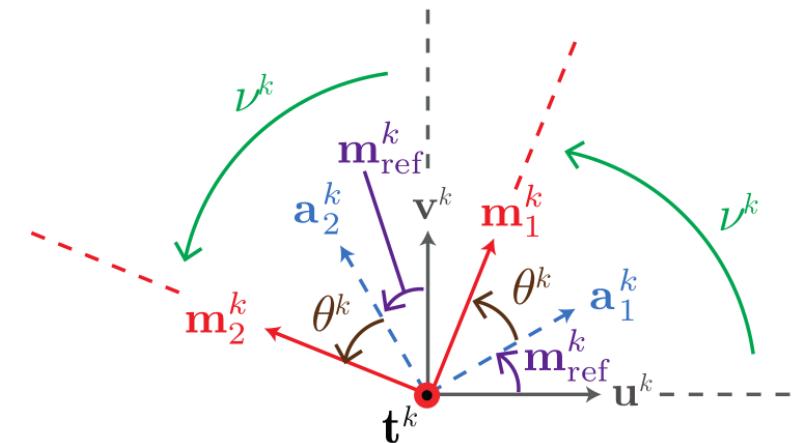
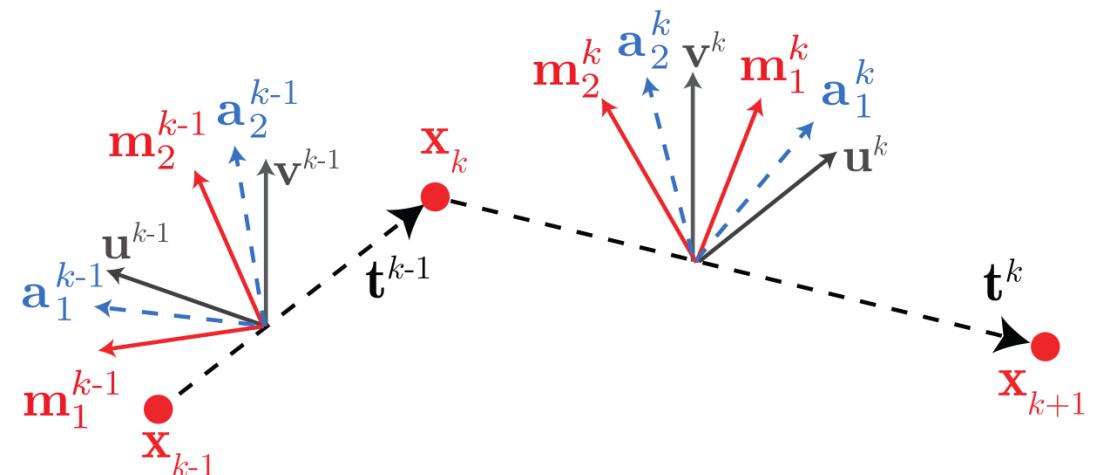


# Reference Twist

- $m_{\text{ref}}^k$  is the signed angle from space-parallel frame director,  $\mathbf{u}^k$ , to time-parallel frame director,  $\mathbf{a}_1^k$

$$\nu^k = \theta^k + m_{\text{ref}}$$

- Reference twist at the  $k$ -th node is  $\Delta m_{\text{ref}} = m_{\text{ref}}^k - m_{\text{ref}}^{k-1}$



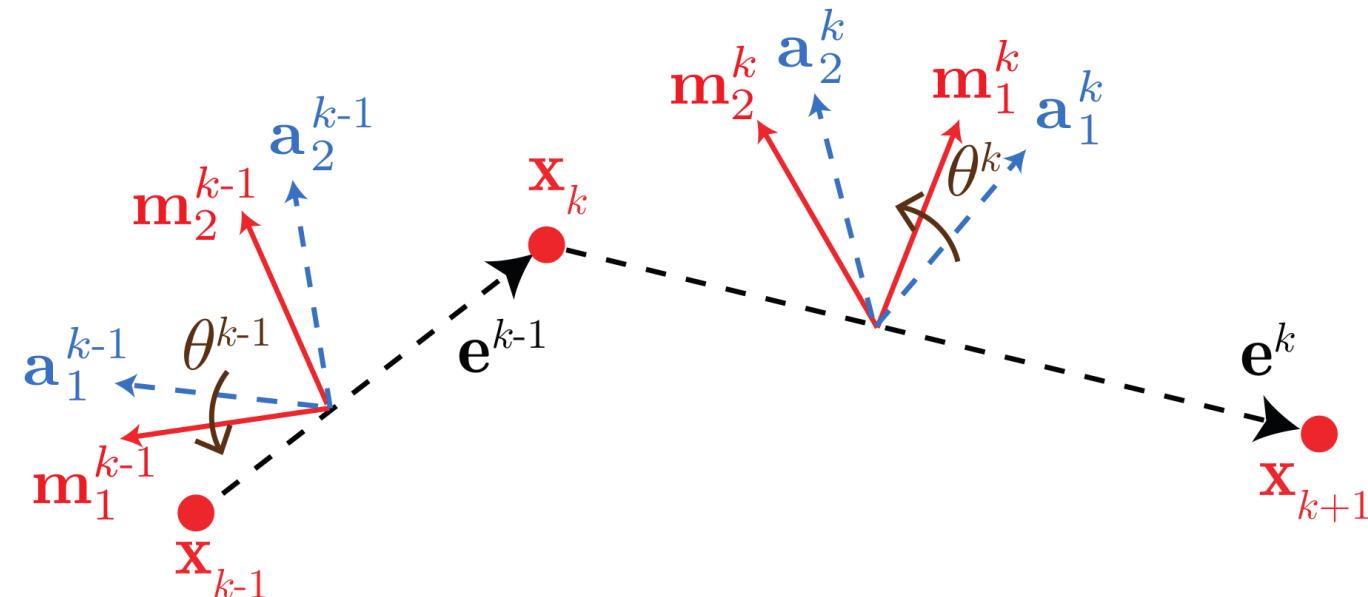
# Measurement of Discrete Twist using Time-Parallel Reference Frame

Twist at the  $k$ -th node is

$$\tau_k = \nu^k - \nu^{k-1} = (\theta^k + m_{\text{ref}}^k) - (\theta^{k-1} + m_{\text{ref}}^{k-1}) = \Delta\theta_k + \Delta m_{k,\text{ref}}$$

The reference twist,  $\Delta m_{k,\text{ref}}$ , at node  $\mathbf{x}_k$  is the signed angle from  $P_{k-1}^k(\mathbf{a}_1^{k-1}(t_{j+1}))$  to  $\mathbf{a}_1^k(t_{j+1})$  about the axis  $\mathbf{t}^k(t_{j+1})$ , i.e.

$$\Delta m_{k,\text{ref}} = \text{signedAngle}\left(P_{k-1}^k(\mathbf{a}_1^{k-1}(t_{j+1})), \mathbf{a}_1^k(t_{j+1}), \mathbf{t}^k(t_{j+1})\right)$$

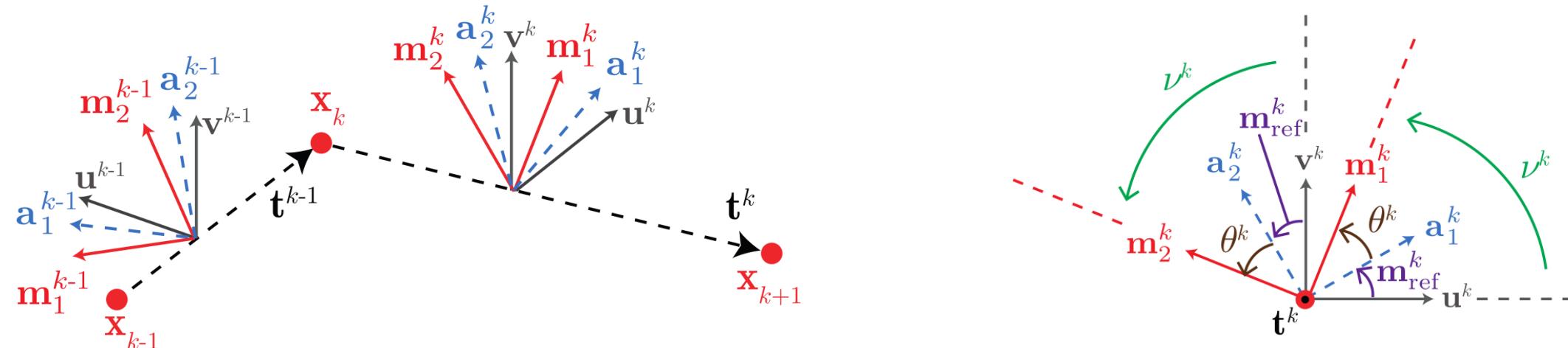


# Summary: Time-Parallel Reference Frame and Twist

- At time  $t = 0$ , start with a space-parallel reference frame on each edge of the rod. This is also our time-parallel frame at  $t = 0$
- Parallel transport the frame from previous time step ( $t = t_j$ ) to the next time step ( $t = t_{j+1}$ )
- Twist at the  $k$ -th node is

$$\tau_k = \nu^k - \nu^{k-1} = (\theta^k + m_{\text{ref}}^k) - (\theta^{k-1} + m_{\text{ref}}^{k-1}) = \Delta\theta_k + \Delta m_{k,\text{ref}}$$

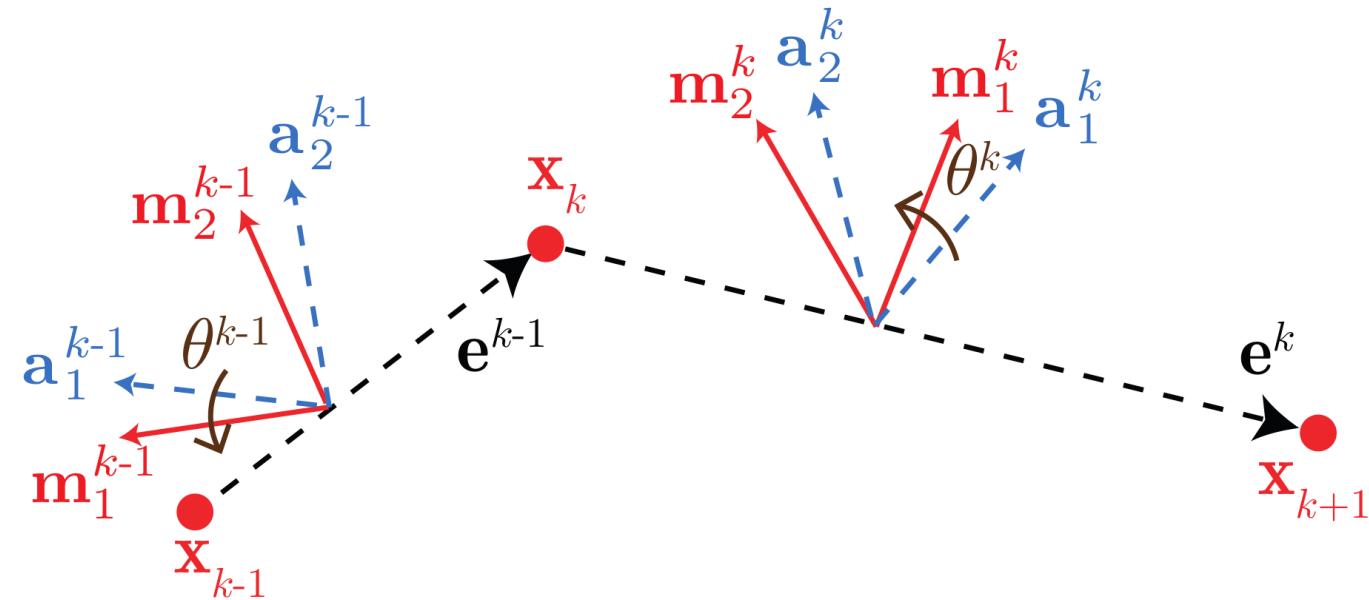
where  $\Delta m_{k,\text{ref}} = \text{signedAngle}\left(P_{k-1}^k(\mathbf{a}_1^{k-1}(t_{j+1})), \mathbf{a}_1^k(t_{j+1}), \mathbf{t}^k(t_{j+1})\right)$





# Degrees of Freedom (DOF) of a Rod

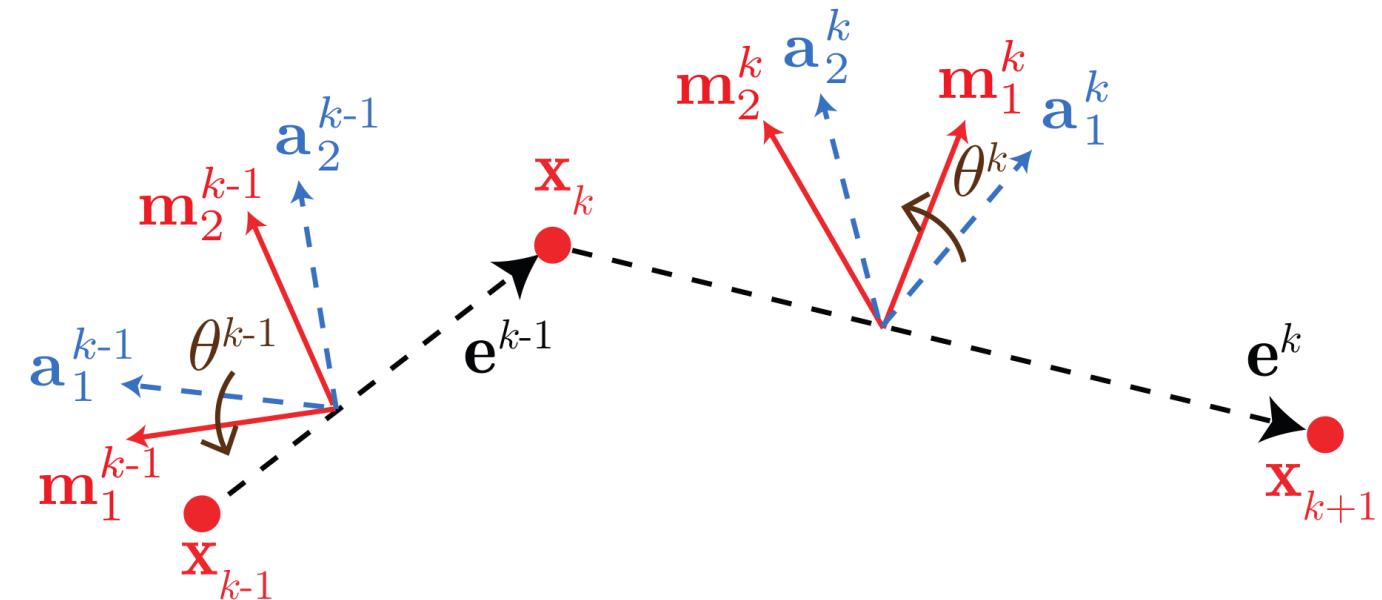
- A rod with  $N$  nodes has  $(4N-1)$  DOFs
  1.  $3N$  DOFs associated with the positions of  $N$  nodes,  $x_k$  ( $1 \leq k \leq N$ )
  2.  $(N-1)$  DOFs for the twist angles of the edges,  $\theta^k$  ( $1 \leq k \leq N - 1$ )



# Discrete Elastic Twisting Energy

$$E_t = \sum_{k=2}^{N-1} E_{t,k}$$

where  $E_{t,k} = \frac{1}{2} \frac{GJ}{l_k} (\tau_k)^2 = \frac{1}{2} \frac{GJ}{l_k} (\Delta\theta_k + \Delta m_{k,\text{ref}})^2$



- $G$  = shear modulus =  $\frac{\text{Young's modulus}}{2(1+\text{Poisson ratio})}$
- $J$  = polar moment of area ( $= \frac{\pi r_0^4}{2}$  if circular)



# Future Plan

- Follow Bergou et al. *SIGGRAPH* (2010) to formulate a discrete simulation algorithm for elastic rods