香 港 大 學
THE UNIVERSITY OF HONG KONG

Physics-based animation

Tutorial: Finite Element Method (FEM)

Floyd M. Chitalu and Prof. Taku Komura

Lecture overview

- Discrete formulations (practice)
  - Energy and force discretization
  - Linear tetrahedral elements
  - Implementation

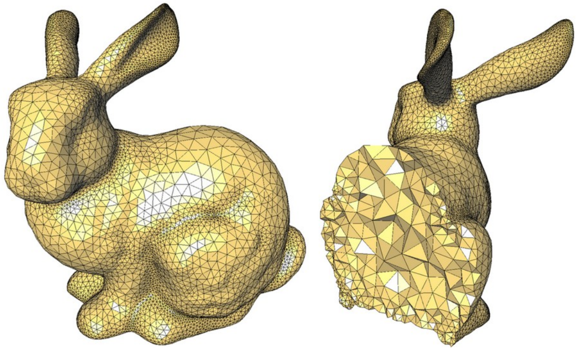
Energy and force discretization

- When modeling a deformable body on the computer we only store the values of the deformation map

$$\phi(\vec{X})$$

  - ... on a finite number of points corresponding to the vertices of a discretization mesh.

$$\vec{X}_1, \vec{X}_2, \ldots, \vec{X}_N$$



Discrete degrees of freedom

- The respective deformed vertex locations
$$\vec{x}_i = \phi(\vec{X}_i),\ \ i = 1, 2, \ldots, N$$
… are our discrete degrees of freedom.
- We can also write
$$\mathbf{x} = \left(\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_N\right)$$
… which is a useful way to represent the aggregate state of our model.

The 1st step to discretiztion

- As a first step, we need to specify a method for reconstructing a continuous deformation map

$$\hat{\phi}$$

... from our discrete samples.

$$\vec{x}_i = \phi(\vec{X}_i)$$

The 1st step to discretiztion

- As a first step, we need to specify a method for reconstructing a continuous deformation map

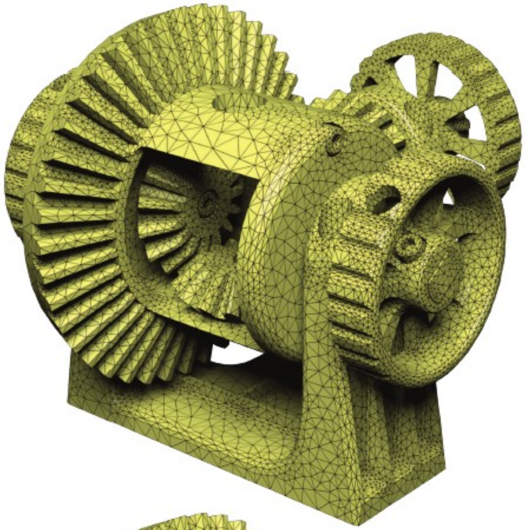
$$\hat{\phi}$$

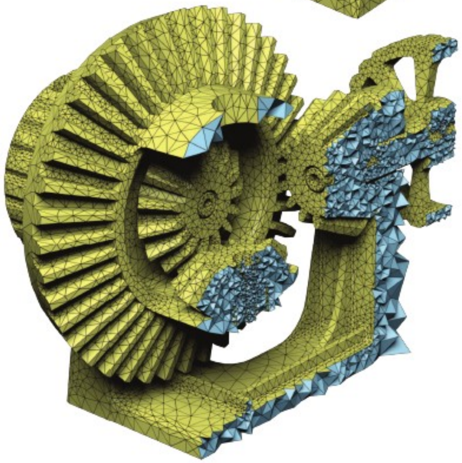
… from our discrete samples.

$$\vec{x}_i = \phi(\vec{X}_i)$$

- *This comes down to a choice of an interpolation scheme.*

One example of an interpolation scheme





- The tetrahedral mesh
  - If we use such a mesh to describe the deforming body, barycentric interpolation will extend the nodal deformations to the entire interior mesh
- (Trilinear interpolation would be a natural choice for lattice discretizations)

Interpolated deformation map

- At any rate, we denote the interpolated deformation map by

$$\hat{\phi}(\vec{X}; \mathbf{x})$$

  - … which emphasizes that this interpolated deformation is dependent on the discrete state **x**

Defining a discrete energy

- For a hyperelastic material, the strain energy of any given deformation φ(X) is computed by integrating the energy density Ψ over the entire body Ω

$$E[\phi] := \int_{\Omega} \Psi(\mathbf{F}) d\vec{X}$$

- From this, we will define a discrete energy, expressed as a function of the DOFs **x**.

Defining a discrete energy

- We define a descrete energy by plugging the interpolated deformation φ̂ into the definition of the strain energy

$$E(\mathbf{x}) := E\left[\hat{\phi}(\vec{X};\mathbf{x})\right] = \int_\Omega \Psi\left(\hat{\mathbf{F}}(\vec{X};\mathbf{x})\right) d\vec{X}$$

- For our practical purposes today, we will look at how both the energy Ψ(F) and the interpolated map φ̂ are defined using tetrahedral meshes.

Discrete forces

- Assuming (for now) that we have defined the discrete energy E(x), we can then now compute the elastic forces associated with individual mesh nodes
- We do this by taking the (negative) gradient of the elastic energy w.r.t the corresponding DOF

$$\vec{f}_i(\mathbf{x}) = -\frac{\partial E(\mathbf{x})}{\partial \vec{x}_i} \quad \text{or} \quad \mathbf{f} := (\vec{f}_1, \vec{f}_2, \ldots, \vec{f}_N) = -\frac{\partial E(\mathbf{x})}{\partial \mathbf{x}}$$

$$\mathbf{f} := (\vec{f}_1, \vec{f}_2, \ldots, \vec{f}_N) = -\frac{\partial E(\mathbf{x})}{\partial \mathbf{x}}$$

Discrete forces

- In practice, prior to computing each force, we first separate the energy integral into the contributions of individual elements Ω_e using:

$$E(\mathbf{x}) = \sum_e E^e(\mathbf{x}) = \sum_e \int_{\Omega_e} \Psi\left(\hat{\mathbf{F}}(\vec{X}; \mathbf{x})\right) d\vec{X}$$

- Thus, nodal forces can be computed by adding the contributions of all coincident elements:

$$\vec{f}_i(\mathbf{x}) = \sum_{e \in \mathcal{N}_i} \vec{f}_i^e(\mathbf{x})$$

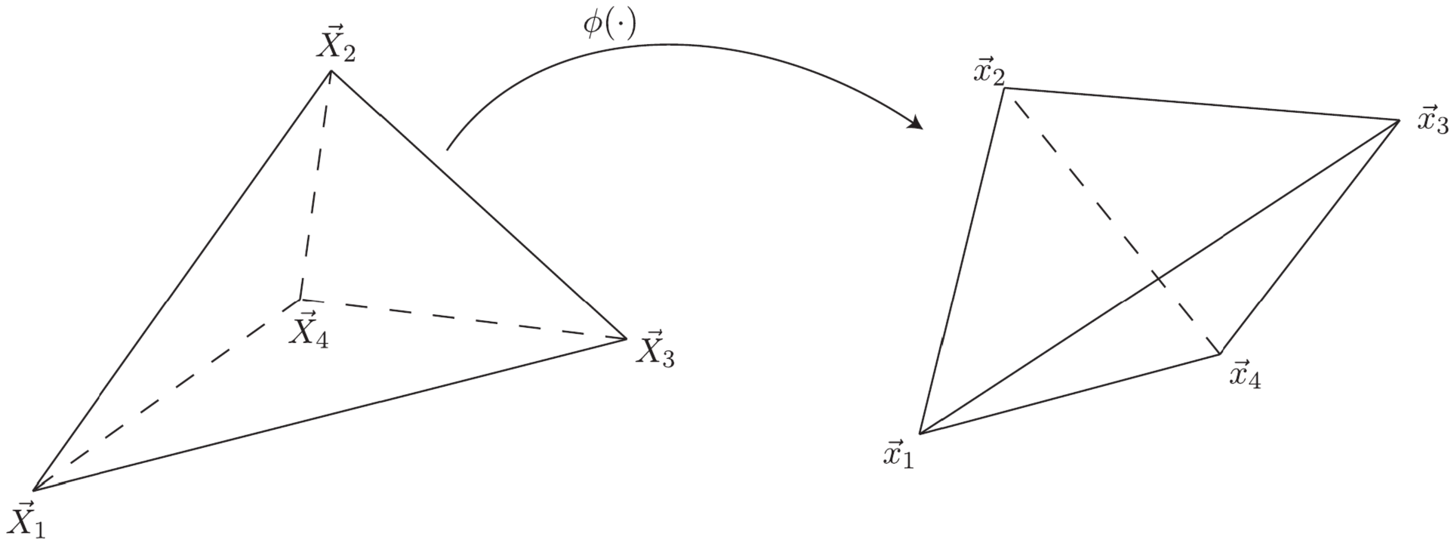
$$\vec{f}_i^e(\mathbf{x}) = -\frac{\partial E^e(\mathbf{x})}{\partial \vec{x}_i}$$

$$\vec{f}_i(\mathbf{x}) = \sum_{e \in \mathcal{N}_i} \vec{f}_i^e(\mathbf{x})$$

Discrete forces

- *For simplicity, we will compute the nodal forces on an element-by-element basis.*

Linear tetrahedral elements



Linear tetrahedral elements

$$\hat{\phi}(\vec{X}) = \mathbf{A}_i \vec{X} + \vec{b}_i$$

For all $\vec{X} \in \mathcal{T}_i$

$\hat{\phi}(\vec{X}) = \mathbf{A}_i \vec{X} + \vec{b}_i$

Linear tetrahedral elements

$$\mathbf{F} = \partial\hat{\phi}/\partial\vec{X} = \mathbf{A}_i$$

Linear tetrahedral elements

$$\mathbf{F} = \partial\hat{\phi}/\partial\vec{X} = \mathbf{A}_i$$

NOTE: *This is constant on each element, and as a consequence so will be any discrete strain measure and stress tensor.*

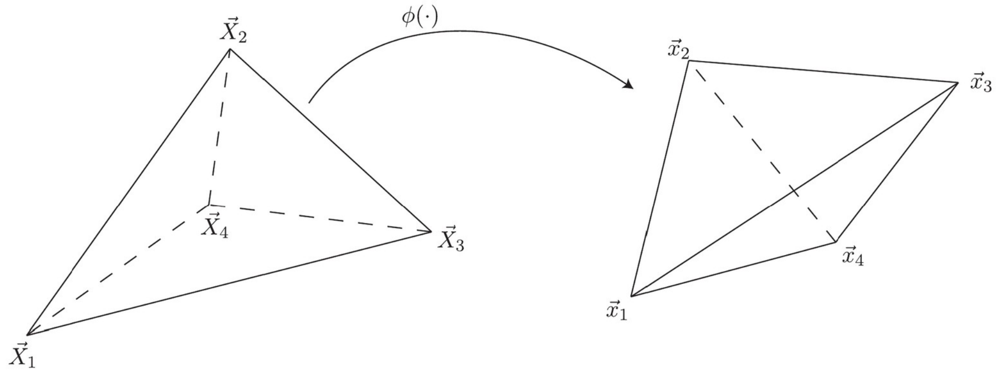
Linear tetrahedral elements

- For simplicity of notation, we will write

$$\phi(\vec{X}) = \mathbf{F}\vec{X} + \vec{b}$$

Deriving the discrete form of F

- In practice, it is possible to determine **F** directly from the locations of the tetrahedron vertices, without involving any reasoning related to barycentric interpolation



Deriving the discrete form of F

- First, we observe that each vertex must satisfy

$$\vec{x}_i = \phi(\vec{X}_i)$$

... or

$$\left\{ \begin{array}{l} \vec{x}_1 = \mathbf{F}\vec{X}_1 + \vec{b} \\ \vec{x}_2 = \mathbf{F}\vec{X}_2 + \vec{b} \\ \vec{x}_3 = \mathbf{F}\vec{X}_3 + \vec{b} \\ \vec{x}_4 = \mathbf{F}\vec{X}_4 + \vec{b} \end{array} \right\} \Rightarrow \left\{ \begin{array}{l} \vec{x}_1 - \vec{x}_4 = \mathbf{F}\left(\vec{X}_1 - \vec{X}_4\right) \\ \vec{x}_2 - \vec{x}_4 = \mathbf{F}\left(\vec{X}_2 - \vec{X}_4\right) \\ \vec{x}_3 - \vec{x}_4 = \mathbf{F}\left(\vec{X}_3 - \vec{X}_4\right) \end{array} \right\}$$

Deriving the discrete form of F

- First, we observe that each vertex must satisfy

$$\vec{x}_i = \phi(\vec{X}_i)$$

... or

$$\left\{ \begin{array}{l} \vec{x}_1 = \mathbf{F}\vec{X}_1 + \vec{b} \\ \vec{x}_2 = \mathbf{F}\vec{X}_2 + \vec{b} \\ \vec{x}_3 = \mathbf{F}\vec{X}_3 + \vec{b} \\ \vec{x}_4 = \mathbf{F}\vec{X}_4 + \vec{b} \end{array} \right\} \Rightarrow \left\{ \begin{array}{l} \vec{x}_1 - \vec{x}_4 = \mathbf{F}\left(\vec{X}_1 - \vec{X}_4\right) \\ \vec{x}_2 - \vec{x}_4 = \mathbf{F}\left(\vec{X}_2 - \vec{X}_4\right) \\ \vec{x}_3 - \vec{x}_4 = \mathbf{F}\left(\vec{X}_3 - \vec{X}_4\right) \end{array} \right\}$$

Eliminate ‘b’ by subtracting x4 from x1, x2 and x3.

Deriving the discrete form of F

- … we can then group the last three (vector) equations as a single matrix equation

$$\left\{ \begin{aligned} \vec{x}_1 - \vec{x}_4 = \mathbf{F}\left(\vec{X}_1 - \vec{X}_4\right) \\ \vec{x}_2 - \vec{x}_4 = \mathbf{F}\left(\vec{X}_2 - \vec{X}_4\right) \\ \vec{x}_3 - \vec{x}_4 = \mathbf{F}\left(\vec{X}_3 - \vec{X}_4\right) \end{aligned} \right\}$$

$$\Downarrow$$

$$\left[\begin{array}{ccc} \vec{x}_1 - \vec{x}_4 & \vec{x}_2 - \vec{x}_4 & \vec{x}_3 - \vec{x}_4 \end{array}\right] = \left[\begin{array}{ccc} \mathbf{F}\left(\vec{X}_1 - \vec{X}_4\right) & \mathbf{F}\left(\vec{X}_2 - \vec{X}_4\right) & \mathbf{F}\left(\vec{X}_3 - \vec{X}_4\right) \end{array}\right]$$

$$\left[\begin{array}{ccc} \vec{x}_1 - \vec{x}_4 & \vec{x}_2 - \vec{x}_4 & \vec{x}_3 - \vec{x}_4 \end{array}\right] = \mathbf{F}\left[\begin{array}{ccc} \vec{X}_1 - \vec{X}_4 & \vec{X}_2 - \vec{X}_4 & \vec{X}_3 - \vec{X}_4 \end{array}\right]$$

$$\begin{bmatrix} \vec{x}_1 - \vec{x}_4 & \vec{x}_2 - \vec{x}_4 & \vec{x}_3 - \vec{x}_4 \end{bmatrix} = \begin{bmatrix} \mathbf{F}\left(\vec{X}_1 - \vec{X}_4\right) & \mathbf{F}\left(\vec{X}_2 - \vec{X}_4\right) & \mathbf{F}\left(\vec{X}_3 - \vec{X}_4\right) \end{bmatrix}$$

$$\begin{bmatrix} \vec{x}_1 - \vec{x}_4 & \vec{x}_2 - \vec{x}_4 & \vec{x}_3 - \vec{x}_4 \end{bmatrix} = \mathbf{F}\begin{bmatrix} \vec{X}_1 - \vec{X}_4 & \vec{X}_2 - \vec{X}_4 & \vec{X}_3 - \vec{X}_4 \end{bmatrix}$$

Deriving the discrete form of F

- ... which represents $\mathbf{D}_s = \mathbf{F}\mathbf{D}_m$ where

$$\mathbf{D}_s := \begin{bmatrix} x_1 - x_4 & x_2 - x_4 & x_3 - x_4 \\ y_1 - y_4 & y_2 - y_4 & y_3 - y_4 \\ z_1 - z_4 & z_2 - z_4 & z_3 - z_4 \end{bmatrix} \qquad \mathbf{D}_m := \begin{bmatrix} X_1 - X_4 & X_2 - X_4 & X_3 - X_4 \\ Y_1 - Y_4 & Y_2 - Y_4 & Y_3 - Y_4 \\ Z_1 - Z_4 & Z_2 - Z_4 & Z_3 - Z_4 \end{bmatrix}$$

Computing F

$$\mathbf{F} = \mathbf{D}_s \mathbf{D}_m^{-1}$$

... or

$$\mathbf{F}(\mathbf{x}) = \mathbf{D}_s(\mathbf{x}) \mathbf{D}_m^{-1}$$

Constant strain energy

- Since **F** is constant over the linear tetrahedron, the strain energy of this element reduces to

$$E_i = \int_{T_i} \Psi(\mathbf{F}) d\vec{X} = \Psi(\mathbf{F}_i) \int_{T_i} d\vec{X} = W \cdot \Psi(\mathbf{F}_i) \quad \text{or} \quad E(\mathbf{x}) = W \cdot \Psi(\mathbf{F}(\mathbf{x}))$$

$$E_i = \int_{T_i} \Psi(\mathbf{F}) d\vec{X} = \Psi(\mathbf{F}_i) \int_{T_i} d\vec{X} = W \cdot \Psi(\mathbf{F}_i) \quad \text{or} \quad E(\mathbf{x}) = W \cdot \Psi(\mathbf{F}(\mathbf{x}))$$

Discrete nodal forces

- The constant strain energy function is then used to derive the contribution of element to the elastic forces on its vertices $\vec{f}_k^i = -\partial E_i(\mathbf{x}) / \partial \vec{x}_k$
- Computing the forces on all four vertices comes down to

$$\mathbf{H} = \left[\vec{f}_1 \quad \vec{f}_2 \quad \vec{f}_3\right] = -W\mathbf{P}(\mathbf{F})\mathbf{D}_m^{-T} \qquad \vec{f}_4 = -\vec{f}_1 - \vec{f}_2 - \vec{f}_3$$

Implementation

Gist: ”How to compute of all elastic forces in a tetrahedral mesh”.

Implementation

procedure PRECOMPUTATION($\mathbf{x}, \mathbf{B}_m[1 \ldots M], W[1 \ldots M]$)

  for each $\mathcal{T}_e = (i, j, k, l) \in \mathcal{M}$ **do** $\triangleright$ M is the number of tetrahedra

    $\mathbf{D}_m \leftarrow \begin{bmatrix} X_i - X_l & X_j - X_l & X_k - X_l \\ Y_i - Y_l & Y_j - Y_l & Y_k - Y_l \\ Z_i - Z_l & Z_j - Z_l & Z_k - Z_l \end{bmatrix}$

    $\mathbf{B}_m[e] \leftarrow \mathbf{D}_m^{-1}$

    $W[e] \leftarrow \frac{1}{6}\det(\mathbf{D}_m)$ $\triangleright$ W is the undeformed volume of $\mathcal{T}_e$

  end for

end procedure

procedure COMPUTEELASTICFORCES($\mathbf{x}, \mathbf{f}, \mathcal{M}, \mathbf{B}_m[], W[]$)

  $\mathbf{f} \leftarrow \mathbf{0}$ $\triangleright$ $\mathcal{M}$ is a tetrahedral mesh

  for each $\mathcal{T}_e = (i, j, k, l) \in \mathcal{M}$ **do**

    $\mathbf{D}_s \leftarrow \begin{bmatrix} x_i - x_l & x_j - x_l & x_k - x_l \\ y_i - y_l & y_j - y_l & y_k - y_l \\ z_i - z_l & z_j - z_l & z_k - z_l \end{bmatrix}$

    $\mathbf{F} \leftarrow \mathbf{D}_s \mathbf{B}_m[e]$

    $\mathbf{P} \leftarrow \mathbf{P}(\mathbf{F})$ $\triangleright$ From the constitutive law

    $\mathbf{H} \leftarrow -W[e]\mathbf{P}\left(\mathbf{B}_m[e]\right)^T$

    $\vec{f}_i \mathrel{+}= \vec{h}_1,\ \vec{f}_j \mathrel{+}= \vec{h}_2,\ \vec{f}_k \mathrel{+}= \vec{h}_3$ $\triangleright$ $\mathbf{H} = \left[\vec{h}_1\ \vec{h}_2\ \vec{h}_3\right]$

    $\vec{f}_l \mathrel{+}= (-\vec{h}_1 - \vec{h}_2 - \vec{h}_3)$

  end for

end procedure

Getting started

- A template will be provided to you on by Monday.

References

- Bonet, J., & Wood, R. (2008). Nonlinear Continuum Mechanics for Finite Element Analysis (2nd ed.). Cambridge: Cambridge University Press.
- Eftychios Sifakis and Jernej Barbic. 2012. FEM simulation of 3D deformable solids: a practitioner's guide to theory, discretization and model reduction. In ACM SIGGRAPH 2012 Courses (SIGGRAPH '12).
- Theodore Kim and David Eberle. 2020. Dynamic deformables: implementation and production practicalities. In ACM SIGGRAPH 2020 Courses (SIGGRAPH '20)