Basics of deformation, elasticity, and finite elements

Yuanming Hu

MIT CSAIL

June 15, 2020

Overview

Simulating elastic materials is a lot of fun!

- Cool visual effects
- Not too hard to implement (using Taichi **(Demos)**)
- Base of other materials (viscoelastic, elastoplastic, viscoplastic...)

Recommended reading

1. The classical FEM method and discretization methodology by *Eftychios Sifakis*[1]
2. The Material Point Method for Simulating Continuum Materials by Chenfanfu Jiang et al.[2]

[1] E. Sifakis and J. Barbic (2012). "FEM simulation of 3D deformable solids: a practitioner's guide to theory, discretization and model reduction". In: *Acm siggraph 2012 courses*, pp. 1–50.

[2] C. Jiang et al. (2016). "The material point method for simulating continuum materials". In: *ACM SIGGRAPH 2016 Courses*, pp. 1–52.

Table of Contents

Deformation

Deformation map ϕ: a (vector to vector) function that relates rest material position and deformed material position.

$$\mathbf{x}_{\text{deformed}} = \phi(\mathbf{x}_{\text{rest}})$$

Deformation gradient $\mathbf{F}$

$$\mathbf{F} := \frac{\partial \mathbf{x}_{\text{deformed}}}{\partial \mathbf{x}_{\text{rest}}}$$

Deformation gradients are translational invariant

$\phi_1 = \phi(\mathbf{x}_{\text{rest}})$ and $\phi_2 = \phi(\mathbf{x}_{\text{rest}}) + \mathbf{c}$ have the same deformation gradients!

Deform/rest volume ratio $J = \det(\mathbf{F})$

Table of Contents

Hyperelasticity

Hyperelastic materials: materials whose stress–strain relationship is defined by a **strain energy density function**

$$\psi = \psi(\mathbf{F})$$

Intuitive understanding: ψ is a potential function that penalizes deformation.
"Stress": the material's internal elastic forces.
"Strain": just replace it with deformation gradients $\mathbf{F}$ for now.

Be careful

We use ψ as the strain energy density function and ϕ as the deformation map. They are completely **different**.

Stress tensor

Stress stands for internal forces that infinitesimal material components exert on their neighborhood.

Based on our need, we use different measures of **stress**

- The First Piola-Kirchhoff stress tensor (PK1): $\mathbf{P}(\mathbf{F}) = \frac{\partial \psi(\mathbf{F})}{\partial \mathbf{F}}$ (easy to compute, but in rest space)
- Kirchhoff stress: $\boldsymbol{\tau}$
- Cauchy stress tensor: $\boldsymbol{\sigma}$ (symmetric, because of conservation of angular momentum)

Relationship: $\boldsymbol{\tau} = J\boldsymbol{\sigma} = \mathbf{P}\mathbf{F}^T$ $\quad \mathbf{P} = J\boldsymbol{\sigma}\mathbf{F}^{-T}$ $\quad$ Traction $\mathbf{t} = \boldsymbol{\sigma}^T\mathbf{n}$.

Intuition of $\mathbf{P} = J\boldsymbol{\sigma}\mathbf{F}^{-T}$: $\mathbf{F}^{-T}$ compensates for material deformation. (Note that it's $\mathbf{F}^{-T}$ instead of $\mathbf{F}^{-1}$ since we transform the normal $\mathbf{n}$ instead of $\mathbf{x}$.)

Elastic moduli (isotropic materials)

- Young's modulus $E = \frac{\sigma}{\varepsilon}$
- Bulk modulus $K = -V\frac{dP}{dV}$
- Poisson's ratio $\nu \in [0.0, 0.5)$ (Auxetics have negative Poisson's ratio)

Lamé parameters:

- Lamé's first parameter μ
- Lamé's second parameter λ (aka. shear modulus, denoted by G)

Useful conversion formula:

$$K = \frac{E}{3(1-2\nu)} \quad \lambda = \frac{E\nu}{(1+\nu)(1-2\nu)} \quad \mu = \frac{E}{2(1+\nu)}$$

Hyperelastic material models

Popular ones in graphics:

- Linear elasticity (small deformation only)
- Neo-Hookean:
  - $\psi(\mathbf{F}) = \frac{\mu}{2}\sum_i[(\mathbf{F}^T\mathbf{F})_{ii} - 1] - \mu\log(J) + \frac{\lambda}{2}\log^2(J)$.
  - $\mathbf{P}(\mathbf{F}) = \frac{\partial\psi}{\partial\mathbf{F}} = \mu(\mathbf{F} - \mathbf{F}^T) + \lambda\log(J)\mathbf{F}^{-T}$
- (Fixed) Corotated:
  - $\psi(\mathbf{F}) = \mu\sum_i(\sigma_i - 1)^2 + \frac{\lambda}{2}(J-1)^2$. σ_i are singular values of $\mathbf{F}$.
  - $\mathbf{P}(\mathbf{F}) = \frac{\partial\psi}{\partial\mathbf{F}} = 2\mu(\mathbf{F} - \mathbf{R}) + \lambda(J-1)J\mathbf{F}^{-T}$

More details: The Material Point Method for Simulating Continuum Materials[3]

[3] C. Jiang et al. (2016). "The material point method for simulating continuum materials". In: *ACM SIGGRAPH 2016 Courses*, pp. 1–52.

Table of Contents

The finite element method

Finite element method: Galerkin discretization scheme that builds discrete equations using weak formulations of continuous PDEs. (More details later in this course.)

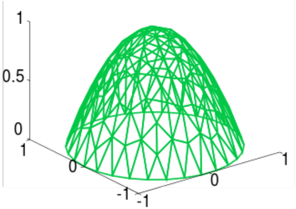

Figure: A solution to a discretized partial differential equation, obtained with FEM (source: Wikipedia)

Linear tetrahedral (triangular) FEM

Linear tetrahedral finite elements (for elasticity) assume **the deformation map ϕ is affine and thereby deformation gradient $\mathbf{F}$ is constant** within a single tetrahedral element:

$$\mathbf{x}_{\text{deformed}} = \mathbf{F}\mathbf{x}_{\text{rest}} + \mathbf{p}.$$

For every element e, its elastic potential energy

$$U(e) = \int_e \psi(\mathbf{F}(\mathbf{x}))\mathbf{x} = V_e \psi(\mathbf{F}_e).$$

Question: how to compute $\mathbf{F}_e(\mathbf{x})$?

Computing $\mathbf{F}_e$ in linear triangular finite elements (1)

Recall that

$$\mathbf{x}_{\textsf{deformed}} = \mathbf{F}\mathbf{x}_{\textsf{rest}} + \mathbf{p}.$$

In 2D triangular elements (3D would be tetrahedral elements), assuming the **rest** positions of the vertices are $\mathbf{a}_{\textsf{rest}}, \mathbf{b}_{\textsf{rest}}, \mathbf{c}_{\textsf{rest}}$ and deformed positions are $\mathbf{a}_{\textsf{deformed}}, \mathbf{b}_{\textsf{deformed}}, \mathbf{c}_{\textsf{deformed}}$. Since within an linear triangular element $\mathbf{F}$ is constant, we have

$$\begin{aligned}
\mathbf{a}_{\textsf{deformed}} &= \mathbf{F}\mathbf{a}_{\textsf{rest}} + \mathbf{p} && (1)\\
\mathbf{b}_{\textsf{deformed}} &= \mathbf{F}\mathbf{b}_{\textsf{rest}} + \mathbf{p} && (2)\\
\mathbf{c}_{\textsf{deformed}} &= \mathbf{F}\mathbf{c}_{\textsf{rest}} + \mathbf{p} && (3)
\end{aligned}$$

Let's eliminate $\mathbf{p}$:

$$\begin{aligned}
(\mathbf{a}_{\textsf{deformed}} - \mathbf{c}_{\textsf{deformed}}) &= \mathbf{F}(\mathbf{a}_{\textsf{rest}} - \mathbf{c}_{\textsf{rest}}) && (4)\\
(\mathbf{b}_{\textsf{deformed}} - \mathbf{c}_{\textsf{deformed}}) &= \mathbf{F}(\mathbf{b}_{\textsf{rest}} - \mathbf{c}_{\textsf{rest}}) && (5)
\end{aligned}$$

Computing $\mathbf{F}_e$ in linear triangular finite elements (2)

$$
\begin{aligned}
(\mathbf{a}_{\textsf{deformed}} - \mathbf{c}_{\textsf{deformed}}) &= \mathbf{F}(\mathbf{a}_{\textsf{rest}} - \mathbf{c}_{\textsf{rest}}) \quad (6)\\
(\mathbf{b}_{\textsf{deformed}} - \mathbf{c}_{\textsf{deformed}}) &= \mathbf{F}(\mathbf{b}_{\textsf{rest}} - \mathbf{c}_{\textsf{rest}}) \quad (7)
\end{aligned}
$$

Note that $\mathbf{F}_{2\times 2}$ now has four linear constraints (equations).

$$
\begin{aligned}
\mathbf{B} &= [\mathbf{a}_{\textsf{rest}} - \mathbf{c}_{\textsf{rest}} | \mathbf{b}_{\textsf{rest}} - \mathbf{c}_{\textsf{rest}}]^{-1} \quad (8)\\
\mathbf{D} &= [\mathbf{a}_{\textsf{deformed}} - \mathbf{c}_{\textsf{deformed}} | \mathbf{b}_{\textsf{deformed}} - \mathbf{c}_{\textsf{deformed}}] \quad (9)\\
\mathbf{F} &= \mathbf{D}\mathbf{B} \quad (10)
\end{aligned}
$$

($\mathbf{B}$ is **constant** through out the physical process. Therefore it should be pre-computed.)

Explicit linear triangular FEM simulation

Recall the Semi-implicit Euler (aka. symplectic Euler) time integration

$$\begin{aligned}\mathbf{v}_{t+1,i} &= \mathbf{v}_{t,i} + \Delta t \frac{\mathbf{f}_{t,i}}{m_i} \\ \mathbf{x}_{t+1,i} &= \mathbf{x}_{t,i} + \Delta t \mathbf{v}_{t+1,i}\end{aligned}$$

Note that $\mathbf{x}_{t,i}$ and $\mathbf{v}_{t,i}$ are stored on the **vertices** of finite elements (triangles/tetrahedrons).

$$\mathbf{f}_{t,i} = -\frac{\partial U}{\partial \mathbf{x}_i} = -\sum_e \frac{\partial U(e)}{\partial \mathbf{x}_i} = -\sum_e V_e \frac{\partial \psi(\mathbf{F}_e)}{\partial \mathbf{F}_e} \frac{\partial \mathbf{F}_e}{\partial \mathbf{x}_i} = -\sum_e V_e \mathbf{P}(\mathbf{F}_e) \frac{\partial \mathbf{F}_e}{\partial \mathbf{x}_i}$$

Don't want to compute $\mathbf{P}(\mathbf{F}_e)$? Use Taichi's AutoDiff system.

Implicit linear triangular FEM simulation

Recall backward Euler time integration:

$$\left[\mathbf{I} - \Delta t^2 \mathbf{M}^{-1} \frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\mathbf{x}_t)\right] \mathbf{v}_{t+1} = \mathbf{v}_t + \Delta t \mathbf{M}^{-1} \mathbf{f}(\mathbf{x}_t) \tag{11}$$

Want implicit time integration? Compute force differentials $\frac{\partial \mathbf{f}}{\partial \mathbf{x}} = -\frac{\partial^2 \psi}{\partial \mathbf{x}^2}$.

Question: in both explicit and implicit schemes, how to compute m_i?
Use mass lumping (or any other convenient approximation you want...)