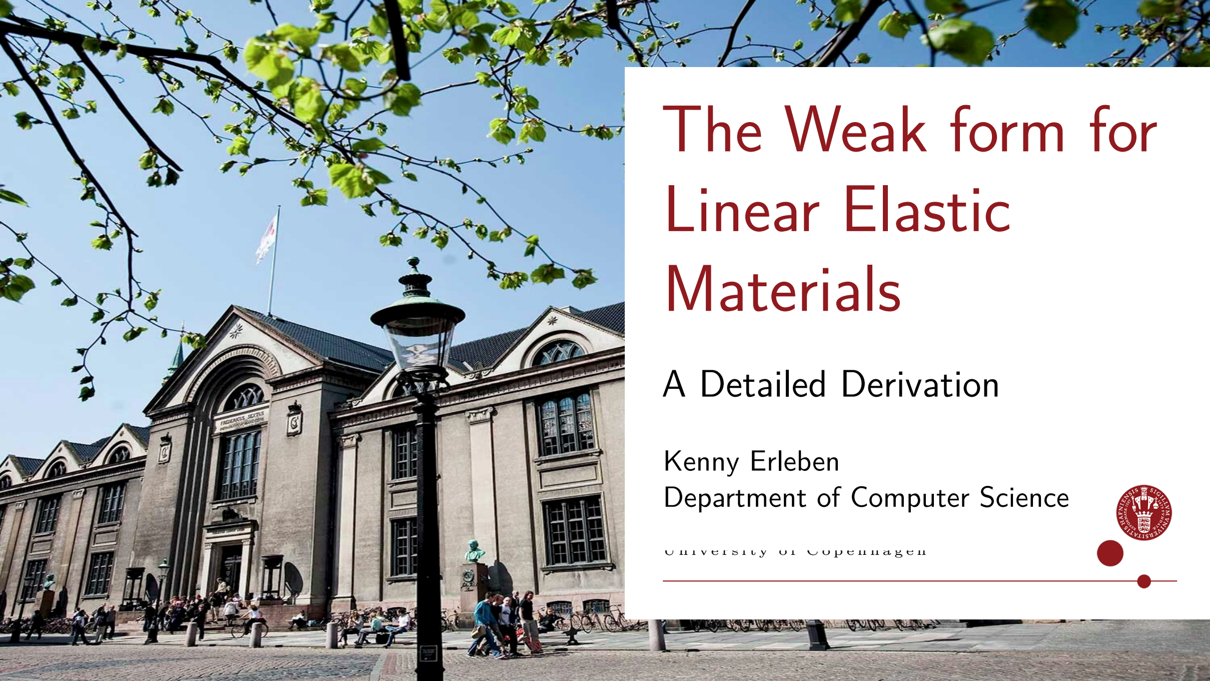

The Weak form for Linear Elastic Materials

A Detailed Derivation

Kenny Erleben
Department of Computer Science

University of Copenhagen

An Elastic Solid

- Reference frame called the material coordinates $\boldsymbol{X}$.
- Deformed into spatial coordinates $\boldsymbol{x}$ by the spatial-temporal deformation $\boldsymbol{x} = \Phi(\boldsymbol{X}, t)$. Material coordinates are the spatial coordinates at time $t = 0$. That is $\boldsymbol{X} = \Phi(\boldsymbol{X}, 0)$.
- The material volume is V and the spatial volume is v
- The material/spatial boundaries are ∂V and ∂v

The Physical Model – Governing Equations (1/2

The Cauchy equation at some spatial point $\boldsymbol{x}$

$$\rho\ddot{\boldsymbol{x}} = \nabla \cdot \boldsymbol{\sigma} + \boldsymbol{b},$$

where $\boldsymbol{b}$ is body force density, ρ is spatial mass density, and $\boldsymbol{\sigma}$ is the Cauchy stress tensor as defined by Cauchy's stress hypothesis

$$\boldsymbol{t} = \boldsymbol{\sigma}\boldsymbol{n},$$

where $\boldsymbol{n}$ is an unit normal vector of a plane and $\boldsymbol{t}$ is the corresponding traction. Conservation of angular momentum implies

$$\boldsymbol{\sigma}^T = \boldsymbol{\sigma}.$$

The Physical Model – Governing Equations (2/2)

We write a model of our problem as

$$\begin{aligned} \rho\ddot{\boldsymbol{x}} &= \nabla \cdot \boldsymbol{\sigma} + \boldsymbol{b}, & \forall \boldsymbol{x} \in v, \\ \boldsymbol{t} &= \boldsymbol{\sigma}\boldsymbol{n}, & \forall \boldsymbol{x} \in \partial v_t. \end{aligned}$$

Here $\partial v_t \subseteq \partial v$ is the subset of the boundary where a known non-zero surface traction is applied.

The Weak Form Reformulation

We multiply with an admissible test function $\boldsymbol{w}$ and integrate over the material volume V

$$\int_V (\rho \ddot{\mathbf{x}} - \nabla \cdot \boldsymbol{\sigma} - \boldsymbol{b}) \cdot \boldsymbol{w} dV = 0$$

which we split into terms

$$0 = \int_V \rho \ddot{\mathbf{x}} \cdot \boldsymbol{w} dV - \int_V (\nabla \cdot \boldsymbol{\sigma}) \cdot \boldsymbol{w} dV - \int_V \boldsymbol{b} \cdot \boldsymbol{w} dV.$$

The next step involves the tensor equivalent of the product rule for the term $\nabla \cdot (\boldsymbol{\sigma} \boldsymbol{w}) = (\nabla \cdot \boldsymbol{\sigma}) \cdot \boldsymbol{w} + \boldsymbol{\sigma} : \nabla \boldsymbol{w}^T$ and the fact that $\boldsymbol{\sigma}$ is symmetric,

$$0 = \int_V \rho \ddot{\mathbf{x}} \cdot \boldsymbol{w} dV + \int_V \boldsymbol{\sigma} : \nabla \boldsymbol{w} dV - \int_V \nabla \cdot (\boldsymbol{\sigma} \boldsymbol{w}) dV - \int_V \boldsymbol{b} \cdot \boldsymbol{w} dV,$$

The Weak Form Reformulation

Next, we apply the Gauss divergence theorem to rewrite the volume integral of $\nabla \cdot (\boldsymbol{\sigma} \boldsymbol{w})$ and use $\boldsymbol{t} = \boldsymbol{\sigma} \boldsymbol{n}$,

$$\begin{aligned}
\int_V \nabla \cdot (\boldsymbol{\sigma} \boldsymbol{w}) dV &= \int_{\partial V} (\boldsymbol{\sigma} \boldsymbol{w}) \cdot \boldsymbol{n} dS, \\
&= \int_{\partial V} \boldsymbol{w} \cdot (\boldsymbol{\sigma} \boldsymbol{n}) dS, \\
&= \int_{\partial V} \boldsymbol{t} \cdot \boldsymbol{w} dS, \\
&= \int_{\partial V_t} \boldsymbol{t} \cdot \boldsymbol{w} dS.
\end{aligned}$$

In the final step, we have applied our boundary condition $\boldsymbol{\sigma} \boldsymbol{n} = \boldsymbol{t}$ on ∂V_t so $\boldsymbol{t} = 0$ everywhere else.

The Weak Form Reformulation

We introduce the symbols

$$
\begin{aligned}
P_\rho &= \int_V \rho \ddot{\boldsymbol{x}} \cdot \boldsymbol{w} dV, \\
P_e &= \int_V \boldsymbol{\sigma} : \nabla \boldsymbol{w} dV, \\
P_t &= -\int_{\partial V_t} \boldsymbol{t} \cdot \boldsymbol{w} dS, \\
P_b &= -\int_V \boldsymbol{b} \cdot \boldsymbol{w} dV.
\end{aligned}
$$

We now have

$$0 = P_\rho + P_e + P_t + P_b.$$

For now, we will consider free-floating objects so $P_t = 0$.

The Body Power Term

We use viscous damping as our body forces

$$P_b = -\int_V -c\dot{\boldsymbol{x}} \cdot \boldsymbol{w} dV = \int_V c\dot{\boldsymbol{x}} \cdot \boldsymbol{w} dV$$

where $c > 0$ is a viscous damping coefficient.

The Elastic Power Term (1/2)

The virtual elastic power term is given as

$$P_e = \int_V \boldsymbol{\sigma} : \nabla \boldsymbol{w} dV.$$

Since the Cauchy stress tensor is symmetric we have

$$P_e = \int_V \boldsymbol{\sigma} : \frac{1}{2}\left(\nabla \boldsymbol{w} + \nabla \boldsymbol{w}^T\right) dV.$$

The Elastic Power Term (2/2)

Recall the infinitesimal strain tensor is by definition

$$\varepsilon = \frac{1}{2}\left(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T\right),$$

so we define

$$\varepsilon_w = \frac{1}{2}\left(\nabla \boldsymbol{w} + \nabla \boldsymbol{w}^T\right).$$

That means using the virtual infinitesimal stress tensor we have

$$P_e = \int_V \boldsymbol{\sigma} : \varepsilon_w dV.$$

The Constitutive Equation

The stress-strain relation comes form $\boldsymbol{\sigma} = \frac{\partial \Psi}{\partial \boldsymbol{\varepsilon}}$ where Ψ is the strain energy function per unit initial volume. It is defined as

$$\boldsymbol{\varepsilon} = \frac{1}{2}(\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T),$$

where $\boldsymbol{u} = \boldsymbol{x} - \boldsymbol{X}$ is the displacement field. In the case of isotropic linear elastic (compressible Neo–Hookean) solids the strain energy function is given as

$$\Psi = \frac{\lambda}{2} \operatorname{tr}(\boldsymbol{\varepsilon})^2 + \mu \boldsymbol{\varepsilon} : \boldsymbol{\varepsilon},$$

where λ and μ are the Lamé constants. Thus, the stress-strain relation has the form

$$\boldsymbol{\sigma} = \lambda \operatorname{tr}(\boldsymbol{\varepsilon})\, \boldsymbol{I} + 2\mu \boldsymbol{\varepsilon}.$$

The Constitutive Equation – In Matrix Notation (1/4)

Writing out each component of the Cauchy stress tensor yields

$$\begin{aligned}
\sigma_{xx} &= \lambda\left(\varepsilon_{xx} + \varepsilon_{yy} + \varepsilon_{zz}\right) + 2\mu\varepsilon_{xx},\\
\sigma_{yy} &= \lambda\left(\varepsilon_{xx} + \varepsilon_{yy} + \varepsilon_{zz}\right) + 2\mu\varepsilon_{yy},\\
\sigma_{zz} &= \lambda\left(\varepsilon_{xx} + \varepsilon_{yy} + \varepsilon_{zz}\right) + 2\mu\varepsilon_{zz},\\
\sigma_{xy} &= 2\mu\varepsilon_{xy},\\
\sigma_{xz} &= 2\mu\varepsilon_{xz},\\
\sigma_{yz} &= 2\mu\varepsilon_{yz}.
\end{aligned}$$

The Constitutive Equation – In Matrix Notation (2/4)

This can be written in matrix form by defining the vectors

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{xx} & \sigma_{yy} & \sigma_{zz} & \sigma_{xy} & \sigma_{xz} & \sigma_{yz} \end{bmatrix}^T$$

and

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_{xx} & \varepsilon_{yy} & \varepsilon_{zz} & 2\varepsilon_{xy} & 2\varepsilon_{xz} & 2\varepsilon_{yz} \end{bmatrix}^T$$

The Constitutive Equation – In Matrix Notation (3/4)

then

$$\begin{bmatrix}\sigma_{xx}\\ \sigma_{yy}\\ \sigma_{zz}\\ \sigma_{xy}\\ \sigma_{xz}\\ \sigma_{yz}\end{bmatrix} = \begin{bmatrix}\lambda+2\mu & \lambda & \lambda & 0 & 0 & 0\\ \lambda & \lambda+2\mu & \lambda & 0 & 0 & 0\\ \lambda & \lambda & \lambda+2\mu & 0 & 0 & 0\\ 0 & 0 & 0 & \mu & 0 & 0\\ 0 & 0 & 0 & 0 & \mu & 0\\ 0 & 0 & 0 & 0 & 0 & \mu\end{bmatrix} \begin{bmatrix}\varepsilon_{xx}\\ \varepsilon_{yy}\\ \varepsilon_{zz}\\ 2\varepsilon_{xy}\\ 2\varepsilon_{xz}\\ 2\varepsilon_{yz}\end{bmatrix}.$$

In terms of Youngs moduls E and Poisson ratio ν the Lamé constants are

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)},$$
$$\mu = \frac{E}{2(1+\nu)}.$$

The Constitutive Equation – In Matrix Notation (4/4)

Using all these relations we get the final form of the constitutive equation that we will use,

$$\boldsymbol{\sigma} = \boldsymbol{D}\boldsymbol{\varepsilon}$$

where

$$\boldsymbol{D} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} d_0 & d_1 & d_1 & 0 & 0 & 0 \\ d_1 & d_0 & d_1 & 0 & 0 & 0 \\ d_1 & d_1 & d_0 & 0 & 0 & 0 \\ 0 & 0 & 0 & d_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & d_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & d_2 \end{bmatrix}$$

and $d_0 = (1-\nu)$, $d_1 = \nu$, and $d_2 = \frac{(1-2\nu)}{2}$.

The Finite Element Discretization

Changing from tensor notation to matrix (Voigt) notation our weak form integral reads

$$P_e = \int_V \boldsymbol{\sigma} \cdot \varepsilon_w dV = \int_V \varepsilon_w^T \boldsymbol{D} \varepsilon dV.$$

Our mathematical model can be stated as

$$0 = P_\rho + P_e + P_b.$$

Our final versions are,

$$P_\rho = \int_V \rho \ddot{\boldsymbol{x}} \cdot \boldsymbol{w} dV, \quad P_e = \int_V \varepsilon_w^T \boldsymbol{D} \varepsilon dV, \quad P_b = \int_V c \dot{\boldsymbol{x}} \cdot \boldsymbol{w} dV.$$

Now we can discretize our model using our shape functions.

The Shape Functions

A volume integral over the solid can be written as a sum of element-wise volume integrals

$$\int_V (\ldots) dV = \sum_e \int_{V^e} (\ldots) dV.$$

We use this without loss of generality to simplify our equations by only considering one tetrahedral element in our derivations.

The Shape Functions

We have a point $\boldsymbol{X}$ inside a tetrahedron e with nodal labels i, j, k, and m using linear shape functions

$$\begin{aligned}\boldsymbol{X} &= N_i\boldsymbol{X}_i + N_j\boldsymbol{X}_j + N_k\boldsymbol{X}_k + N_m\boldsymbol{X}_m,\\ \boldsymbol{u} &= N_i\boldsymbol{u}_i + N_j\boldsymbol{u}_j + N_k\boldsymbol{u}_k + N_m\boldsymbol{u}_m,\end{aligned}$$

where the N_a's for $a \in e \equiv \{i, j, k, m\}$ are the barycentric/volume coordinates of the point $\boldsymbol{X}$ with respect to the e^{th} tetrahedron.

The Shape Functions

The interpolation weights are equivalent to the local hat functions and easily computed from the geometry

$$
\begin{aligned}
N_i &= \frac{V_{jkm}}{V^e} \\
&= \frac{(\boldsymbol{X} - \boldsymbol{X}_j) \cdot ((\boldsymbol{X}_k - \boldsymbol{X}_j) \times (\boldsymbol{X}_m - \boldsymbol{X}_k))}{(\boldsymbol{X}_m - \boldsymbol{X}_i) \cdot ((\boldsymbol{X}_j - \boldsymbol{X}_i) \times (\boldsymbol{X}_k - \boldsymbol{X}_i))}
\end{aligned}
$$

where $\boldsymbol{V}^e$ is material volume and V_{jkm} is the volume of the tetrahedron spanned by $\boldsymbol{X}$ and nodes j, k, and m.

The Shape Functions

The material gradients are easily computed

$$\nabla_0 N_i = \frac{(\boldsymbol{X}_k - \boldsymbol{X}_j) \times (\boldsymbol{X}_m - \boldsymbol{X}_k)}{6V^e},$$

Where $\nabla_0 \equiv \frac{\partial}{\partial \boldsymbol{X}} = \begin{bmatrix} \partial_X & \partial_Y & \partial_Z \end{bmatrix}^T$ is the material gradient operator whereas $\nabla \equiv \frac{\partial}{\partial \boldsymbol{x}} = \begin{bmatrix} \partial_x & \partial_y & \partial_z \end{bmatrix}^T$ is the spatial gradient operator.

The Problem and Solution

Reexamining our model equation we observe that the spatial gradients are used but integrals are taken over material coordinates. Our assumption of infinitesimal displacements means $\boldsymbol{x} \approx \boldsymbol{X}$, $\| \nabla \boldsymbol{u} \| \ll 1$ and so it follows $\nabla_0 \boldsymbol{u} \approx \nabla \boldsymbol{u}$. This means that we can think of all spatial gradients as material gradients.

The Matrix Notation of Shape Functions

For the displacement field, we have

$$\boldsymbol{u} \approx \sum_{a\in\{i,j,k,m\}} N_a \boldsymbol{u}_a = \boldsymbol{N}^e \begin{bmatrix} \tilde{\boldsymbol{u}}_i \\ \tilde{\boldsymbol{u}}_j \\ \tilde{\boldsymbol{u}}_k \\ \tilde{\boldsymbol{u}}_m \end{bmatrix} = \boldsymbol{N}^e \tilde{\boldsymbol{u}}^e .$$

Similar for $\boldsymbol{w} \approx \boldsymbol{N}^e \tilde{\boldsymbol{w}}^e$ the spatial positions $\boldsymbol{x} \approx \boldsymbol{N}^e \tilde{\boldsymbol{x}}^e$. Where

$$\boldsymbol{N}^e = \begin{bmatrix} N_i \boldsymbol{I} & N_j \boldsymbol{I} & N_k \boldsymbol{I} & N_l \boldsymbol{I} \end{bmatrix} \quad \text{and} \quad \boldsymbol{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} .$$

The Matrix Notation of Shape Functions

Recall that the strain tensor was defined as $\varepsilon = \frac{1}{2}(\nabla_0 \boldsymbol{u} + \nabla_0 \boldsymbol{u}^T)$ so

$$\boldsymbol{\varepsilon} \approx \underbrace{\begin{bmatrix} \partial_X & 0 & 0 \\ 0 & \partial_Y & 0 \\ 0 & 0 & \partial_Z \\ \partial_Y & \partial_X & 0 \\ \partial_Z & 0 & \partial_X \\ 0 & \partial_Z & \partial_Y \end{bmatrix}}_{\boldsymbol{S}} \boldsymbol{N}^e \tilde{\boldsymbol{u}}^e = \underbrace{\boldsymbol{S}\boldsymbol{N}^e}_{\boldsymbol{B}^e} \tilde{\boldsymbol{u}}^e = \boldsymbol{B}^e \tilde{\boldsymbol{u}}^e,$$

similar for the test strain $\boldsymbol{\varepsilon}_w^e \approx \boldsymbol{B}^e \tilde{\boldsymbol{w}}^e$.

The Matrix Notation of Shape Functions

Writing out we have

$$\boldsymbol{B}^e = \begin{bmatrix} \boldsymbol{B}_i^e & \boldsymbol{B}_j^e & \boldsymbol{B}_k^e & \boldsymbol{B}_m^e \end{bmatrix}$$

where

$$\boldsymbol{B}_i^e = \begin{bmatrix} \partial_X N_i & 0 & 0 \\ 0 & \partial_Y N_i & 0 \\ 0 & 0 & \partial_Z N_i \\ \partial_Y N_i & \partial_X N_i & 0 \\ \partial_Z N_i & 0 & \partial_X N_i \\ 0 & \partial_Z N_i & \partial_Y N_i \end{bmatrix}.$$

Similar for $\boldsymbol{B}_j^e$, $\boldsymbol{B}_k^e$, and $\boldsymbol{B}_m^e$.

Applying the Shape Functions to the Power Terms

Applying all approximations we have

$$P_\rho = (\tilde{\boldsymbol{w}}^e)^T \left(\int_{V^e} \rho (\boldsymbol{N}^e)^T \boldsymbol{N}^e dV \right) \ddot{\tilde{\boldsymbol{x}}}^e,$$

$$P_e = (\tilde{\boldsymbol{w}}^e)^T \left(\int_{V^e} (\boldsymbol{B}^e)^T \boldsymbol{D} \boldsymbol{B}^e dV \right) \tilde{\boldsymbol{u}}^e,$$

$$P_b = (\tilde{\boldsymbol{w}}^e)^T \left(\int_{V^e} c (\boldsymbol{N}^e)^T \boldsymbol{N}^e dV \right) \dot{\tilde{\boldsymbol{x}}}^e,$$

Simplifying

The equation $P_\rho + P_e + P_b = 0$ must hold for arbitrary values of $\tilde{\boldsymbol{w}}^e$ so we end up with

$$\boldsymbol{M}^e \ddot{\tilde{\boldsymbol{x}}}^e + \boldsymbol{C}^e \dot{\tilde{\boldsymbol{x}}}^e + \boldsymbol{K}^e \tilde{\boldsymbol{u}}^e = 0,$$

where

$$\begin{aligned}
\boldsymbol{M}^e &= \int_{V^e} \rho (\boldsymbol{N}^e)^T \boldsymbol{N}^e dV, \\
\boldsymbol{C}^e &= \int_{V^e} c (\boldsymbol{N}^e)^T \boldsymbol{N}^e dV, \\
\boldsymbol{K}^e &= \int_{V^e} (\boldsymbol{B}^e)^T \boldsymbol{D} \boldsymbol{B}^e dV.
\end{aligned}$$

For computational efficiency, one may apply lumping for the mass and damping matrices.

Assembly Process

Summing up over all tetrahedra, the assembly process, we have the global system

$$\boldsymbol{M}\ddot{\tilde{\boldsymbol{x}}} + \boldsymbol{C}\dot{\tilde{\boldsymbol{x}}} + \boldsymbol{K}\tilde{\boldsymbol{u}} = 0.$$

If we had included gravity in the body forces or similar then we may add an extra $\boldsymbol{f}$ term to the right-hand side to account for this constant body force term.
Observe

- The infinitesimal assumption allowed us to interchange spatial gradients with material gradients
- The consequence is that the $\boldsymbol{K}$ matrix is a constant matrix and can be precomputed

Time Discretization

The finite element method resulted in the 2nd order differential equation

$$\boldsymbol{M}\ddot{\boldsymbol{x}} + \boldsymbol{C}\dot{\boldsymbol{x}} + \boldsymbol{K}\boldsymbol{u} = \boldsymbol{f}.$$

We have dropped the tilde notation. The time derivative of the velocity is the acceleration, $\dot{\boldsymbol{v}} = \ddot{\boldsymbol{x}}$, and so using first-order Euler,

$$\dot{\boldsymbol{v}} \approx \frac{\boldsymbol{v}^{t+\Delta t} - \boldsymbol{v}^{t}}{\Delta t},$$

and

$$\boldsymbol{x}^{t+\Delta t} \approx \boldsymbol{x}^{t} + \Delta t \boldsymbol{v}^{t+\Delta t}.$$

Time Discretization

Substituting the finite difference approximations we have

$$\boldsymbol{M}\frac{\boldsymbol{v}^{t+\Delta t}-\boldsymbol{v}^{t}}{\Delta t}+\boldsymbol{K}\left(\boldsymbol{x}^{t+\Delta t}-\boldsymbol{X}\right)+\boldsymbol{C}\boldsymbol{v}^{t+1}=\boldsymbol{f}.$$

Further manipulation

$$\boldsymbol{M}\boldsymbol{v}^{t+\Delta t}-\boldsymbol{M}\boldsymbol{v}^{t}+\Delta t\boldsymbol{K}\left(\boldsymbol{x}^{t}+\Delta t\boldsymbol{v}^{t+\Delta t}-\boldsymbol{X}\right)+\Delta t\boldsymbol{C}\boldsymbol{v}^{t+1}=\Delta t\boldsymbol{f}.$$

Time Discretization

Finally

$$\boldsymbol{A}\boldsymbol{v}^{t+\Delta t} = \boldsymbol{b},$$

where

$$\begin{aligned}\boldsymbol{A} &= \boldsymbol{M} + \Delta t\boldsymbol{C} + \Delta t^2\boldsymbol{K},\\ \boldsymbol{b} &= \boldsymbol{M}\boldsymbol{v}^t + \Delta t\left(\boldsymbol{f} - \boldsymbol{K}\boldsymbol{x}^t + \boldsymbol{f}_0\right).\end{aligned}$$

Here $\boldsymbol{f}_0 = \boldsymbol{K}\boldsymbol{X}$. $\boldsymbol{A}$ is a sparse block symmetric and positive definite. Having found $\boldsymbol{v}^{t+\Delta t}$ we may do the position update

$$\boldsymbol{x}^{t+\Delta t} = \boldsymbol{x}^t + \Delta t\boldsymbol{v}^{t+\Delta t}.$$

The Corotational Formulation

Problem

- If we drop infinitesimal assumptions $\boldsymbol{K}$ is a non-linear function of $\boldsymbol{x}$. The elastic force term is $\boldsymbol{K}^e(\boldsymbol{x}^e)\boldsymbol{u}^e$.

We seek an approximation of the term $\boldsymbol{K}^e(\boldsymbol{x}^e)\boldsymbol{u}^e$

- Idea: large deformations are due to the current spatial rotation.

The Corotational Formulation

Thus, we write

$$\boldsymbol{K}^e(\boldsymbol{x}^e)\boldsymbol{u}^e \approx \boldsymbol{O}^e\boldsymbol{K}^e\left(\left(\boldsymbol{O}^e\right)^T\boldsymbol{x}^e - \boldsymbol{X}^e\right),$$

where $\boldsymbol{O}^e$ is the block rotation matrix

$$\boldsymbol{O}^e = \begin{bmatrix} \boldsymbol{R}^e & 0 & 0 & 0 \\ 0 & \boldsymbol{R}^e & 0 & 0 \\ 0 & 0 & \boldsymbol{R}^e & 0 \\ 0 & 0 & 0 & \boldsymbol{R}^e \end{bmatrix}$$

and $\boldsymbol{R}^e$ is the rotation matrix from the material coordinates to spatial coordinates.

The Rotation Matrix

In the least square error sense, the rotation matrix could be understood as the problem

$$\boldsymbol{R}^e = \arg\min_{\boldsymbol{R}} \sum_{a=\{i,j,k,m\}} \| \boldsymbol{x}_a - \boldsymbol{R}\boldsymbol{X}_a \|^2$$

subject to the constraint

$$\boldsymbol{R}^T\boldsymbol{R} = \boldsymbol{I}.$$

Rather than solving this constrained minimization problem one may exploit the polar decomposition of the deformation gradient. We will omit the element superscript for readability.

The Deformation Gradient

We define the edge vectors

$$\begin{aligned} \boldsymbol{e}_{ab} &= \boldsymbol{x}_a - \boldsymbol{x}_b, \\ \boldsymbol{E}_{ab} &= \boldsymbol{X}_a - \boldsymbol{X}_b, \end{aligned}$$

for all $a \neq b$ and $a, b \in \{i, j, k, m\}$. By definition of the deformation gradient

$$\boldsymbol{e}_{ab} = \boldsymbol{F}\boldsymbol{E}_{ab}.$$

The Deformation Gradient

We form the linear system

$$\underbrace{\begin{bmatrix} \boldsymbol{e}_{ji} & \boldsymbol{e}_{ki} & \boldsymbol{e}_{mi} \end{bmatrix}}_{\boldsymbol{E}} = \boldsymbol{F} \underbrace{\begin{bmatrix} \boldsymbol{E}_{ji} & \boldsymbol{E}_{ki} & \boldsymbol{E}_{mi} \end{bmatrix}}_{\boldsymbol{E}_0} .$$

The deformation gradient is then easily computed as

$$\boldsymbol{F} = \boldsymbol{E}\boldsymbol{E}_0^{-1}.$$

This is efficient as $\boldsymbol{E}_0^{-1}$ is in material coordinates.

The Polar Decomposition

There exists a polar decomposition of the deformation gradient

$$\boldsymbol{F} = \boldsymbol{R}\boldsymbol{U},$$

where $\boldsymbol{R}$ is an orthogonal matrix and $\boldsymbol{U}$ is a symmetric stretch matrix. From the right Cauchy–Green strain tensor we have

$$\boldsymbol{C} = \boldsymbol{F}^T\boldsymbol{F} = \boldsymbol{U}^2.$$

The Polar Decomposition

Eigenvalue decomposition of $\boldsymbol{C}$ gives the diagonal matrix $\boldsymbol{\Lambda}$ of eigenvalues and $\boldsymbol{\Omega}$ the orthogonal matrix of eigenvectors

$$\boldsymbol{C} = \boldsymbol{\Omega}\boldsymbol{\Lambda}\boldsymbol{\Omega}^T$$

so

$$\boldsymbol{U} = \boldsymbol{\Omega}\sqrt{\boldsymbol{\Lambda}}\boldsymbol{\Omega}^T.$$

Finally we can compute $\boldsymbol{R}$ as

$$\boldsymbol{R} = \boldsymbol{F}\boldsymbol{U}^{-1} = \boldsymbol{F}\left(\sqrt{\boldsymbol{\Lambda}^{-1}}\right)\boldsymbol{\Omega}^T.$$

The Co-rotational Final Warp

Before performing the velocity update by solving $\boldsymbol{A}\boldsymbol{v}^{t+\Delta t} = \boldsymbol{b}$ one must compute $\boldsymbol{R}^{e}$ for each tetrahedron according to the above equation and then update each local stiffness element leading to

$$\begin{aligned} \boldsymbol{K}^{e'} &= \boldsymbol{O}^{e}\boldsymbol{K}^{e}(\boldsymbol{O}^{e})^{T}, \\ \boldsymbol{f}_{0}^{e'} &= \boldsymbol{O}^{e}\boldsymbol{K}^{e}\boldsymbol{X}^{e}. \end{aligned}$$

The primed quantities are now used in place of the unprimed quantities in the assembly before computing the velocity update. Observe that a new assembly process must be done as the first thing in each iteration of the simulation loop.