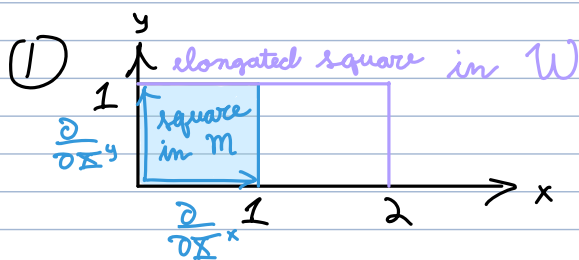


12. HW 3

Continuous Theory

(1) 2D Square Elongation
(youtube example)

- Calculate F .



$$m \xrightarrow{\phi} W$$

$$\frac{\partial}{\partial x^a} \quad \frac{\partial}{\partial x^i}$$

$$\vec{x} = \phi(\vec{X}) = \begin{bmatrix} 2X^x \\ X^y \end{bmatrix} \begin{matrix} \frac{\partial}{\partial x^x} \\ \frac{\partial}{\partial x^y} \end{matrix} \downarrow i$$

$$F = \frac{\partial \phi^i}{\partial X^a} = \begin{matrix} \downarrow i \end{matrix} \begin{matrix} \xrightarrow{a} \end{matrix} \begin{bmatrix} \frac{\partial x^x}{\partial X^x} & \frac{\partial x^x}{\partial X^y} \\ \frac{\partial x^y}{\partial X^x} & \frac{\partial x^y}{\partial X^y} \end{bmatrix}$$

$$= \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$$

(2) 3D Abstract Time-Dependent Motion (youtube ex.)

- Calculate F .

(2) Assume:

$$\phi(\vec{X}, t) = \begin{bmatrix} 3X^x X^y + tX^z \\ X^{z^2} - tX^x \\ 5X^y X^z \end{bmatrix} \downarrow i$$

$$F = \frac{\partial \phi^i}{\partial X^a} = \begin{matrix} \downarrow i \end{matrix} \begin{matrix} \xrightarrow{a} \end{matrix} \begin{bmatrix} \frac{\partial x^x}{\partial X^x} & \frac{\partial x^x}{\partial X^y} & \frac{\partial x^x}{\partial X^z} \\ \frac{\partial x^y}{\partial X^x} & \frac{\partial x^y}{\partial X^y} & \frac{\partial x^y}{\partial X^z} \\ \frac{\partial x^z}{\partial X^x} & \frac{\partial x^z}{\partial X^y} & \frac{\partial x^z}{\partial X^z} \end{bmatrix}$$

$$= \begin{bmatrix} 6X^x X^y & 3X^{x^2} & 3tX^{z^2} \\ -t & 0 & 2X^z \\ 0 & 5X^z & 5X^y \end{bmatrix}$$

- F may not be symmetric.
- F is a fn of \vec{X} and t .

Discrete Theory

- ① What are the steps involved in doing an elastic body simulation?

Note: the subscript "c" means "per cell" and the subscript "v" means "per vertex".

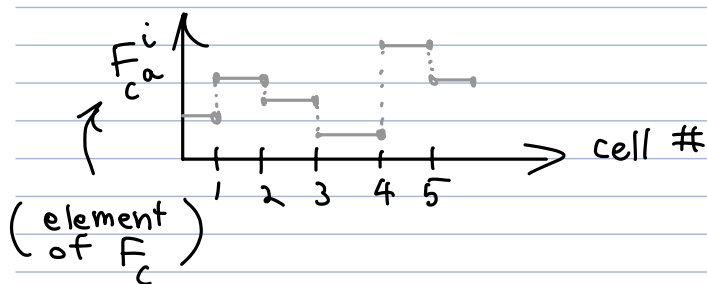
Equivalent approach:

$$C_c \rightarrow U_c \rightarrow U_{\text{tot}} = \sum_c U_c$$

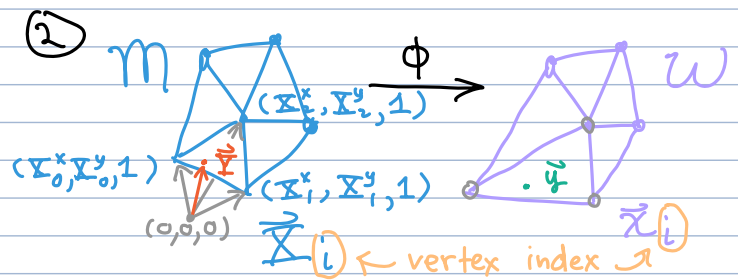
$$f = \frac{\partial U_{\text{tot}}}{\partial x_i}$$

- ② Calculate F_c for a 2D mesh.

Note: We have a different F_c for each cell. Thus, F_c is piecewise constant.



- ① (i) \vec{x}_v (flow map)
 (ii) F_c (deformation grad.)
 (iii) $C_c = F_c^T F_c$ (induced metric)
 (iv) $E_c = \frac{1}{2} (C_c - I)$ (strain tensor)
 (v) Find appropriate stress-strain relation. One possible choice:
 $S_c = 2\mu E_c + \lambda \text{tr}(E_c) I$ (1st and 2nd Piola stress)
 (vi) $P_c = F S_c$ (1st Piola stress)
 (vii) $\vec{F}_v = \frac{1}{h} \sum_{c \ni v} P_c \hat{n}_{c,v} A_{c,v}$
 (= $\text{div}(P_c)$ = discrete divergence)
 (ix) $\vec{v}_v = \vec{F}_v / m_v$, $\dot{\vec{x}}_v = \vec{v}_v$



$\vec{y} = \phi(\vec{Y})$ is the continuous (infinite-dimensional) flow map.

$$F_c = \frac{\partial \phi_i}{\partial Y_a}$$

\leftarrow increment over $\dim(W)=2$
 \leftarrow increment over $\dim(M)=2$

$$\text{Given } \begin{bmatrix} y^x \\ y^y \\ 1 \end{bmatrix} = \phi_c(\vec{Y}) = AB \begin{bmatrix} Y^x \\ Y^y \\ 1 \end{bmatrix}$$

$$A = \begin{bmatrix} x_0^x & x_1^x & x_2^x \\ x_0^y & x_1^y & x_2^y \\ 1 & 1 & 1 \end{bmatrix}$$

$$B = \begin{bmatrix} x_0^x & x_1^x & x_2^x \\ x_0^y & x_1^y & x_2^y \\ 1 & 1 & 1 \end{bmatrix}^{-1}$$

$$Q = AB = \begin{bmatrix} Q_{11}^x & Q_{12}^x & Q_{13}^x \\ Q_{21}^x & Q_{22}^x & Q_{23}^x \\ Q_{31}^x & Q_{32}^x & Q_{33}^x \end{bmatrix}$$

we care about this part:

$$\therefore \phi_c(\vec{Y}) = \begin{bmatrix} a_{a1}^x Y^x + a_{b1}^x Y^y + a_{c1}^x \\ a_{a2}^x Y^x + a_{b2}^x Y^y + a_{c2}^x \\ a_{a3}^x Y^x + a_{b3}^x Y^y + a_{c3}^x \end{bmatrix} \downarrow i$$

$$F_c = \frac{\partial \phi_c^i}{\partial \vec{Y}} = \begin{bmatrix} a_{a1}^x & a_{b1}^x \\ a_{a2}^x & a_{b2}^x \end{bmatrix} \quad \begin{matrix} i=0,1 \\ \alpha=0,1 \end{matrix}$$

(per cell)

Alternative approach for getting F_c (produces equivalent result):

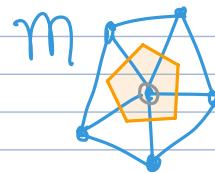
$$F_c = -\frac{1}{nV_c} \sum_{j=0}^n \begin{bmatrix} 1 \\ \chi_j \end{bmatrix} [-A_{c,j} \eta_{j,j}^T]$$

We use this to derive our discrete divergence operator in $\vec{f}_v = \text{div}(P_c)$.

(3) Calculate m_v .

Note: ρ is the mass density.

$$(3) \quad m_v = \rho \sum_{c \in v} \frac{1}{n+1} V_c$$



m_v is a diagonal matrix (which makes matrix inversion easier).