Sensitive Deformables

Tutorial A3

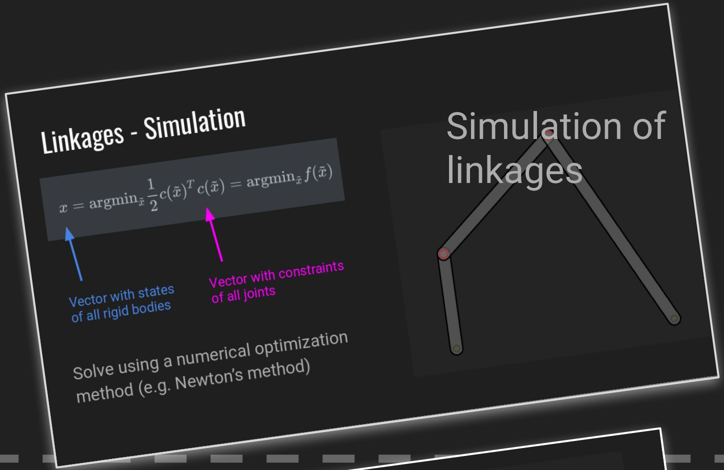
Organizational stuff

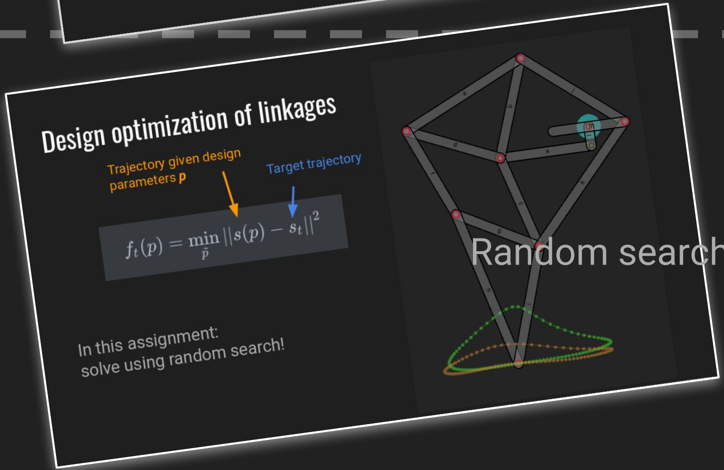
- Printed linkages after easter break
- No tutorial next week
  Questions?
  → https://github.com/computational-robotics-lab/comp-fab-a3/issues
  → if you use your private repo's issues, make sure to mention me: @moritzge
- After easter break: Project Q&A sessions during tutorials
- Don't forget to hand-in your project proposal! (moritzge@inf.ethz.ch)
- Solution to A1 and A2 will be pushed asap

Simulation

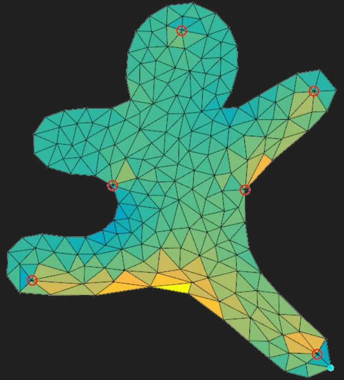
Shape Optimization

Last assignment:

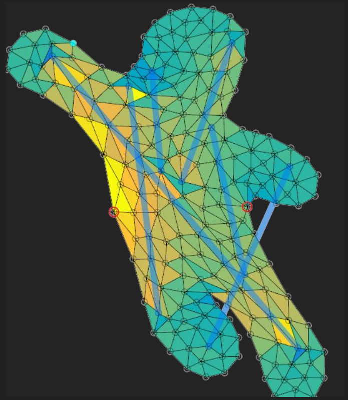




This assignment:



Simulation of deformables



Sensitivity Analysis
Steepest Descent
(SASD)

Finite Element

Deformation energy:

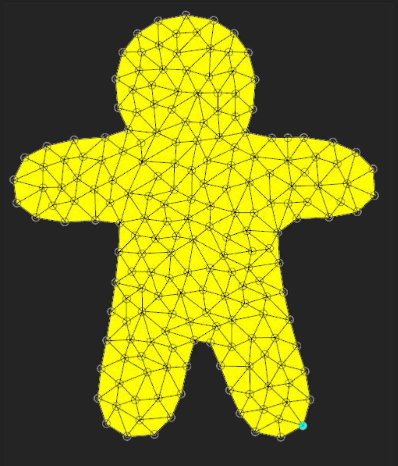
$$W_d(x) = \int_{e\in E} \Psi_{NH}(\mathbf{F})dA_e$$

Strain energy density: (Neo-Hookean)

$$\Psi_{NH}(\mathbf{F}) = \frac{\mu}{2}(tr(\mathbf{C}) - 2) - \mu \ln(J) + \frac{\lambda}{2}\ln(J)^2$$

$$\mathbf{C} = \mathbf{F}^T\mathbf{F}$$

$$J = \det(\mathbf{F})$$



Discretization: (linear basis fctns → const $\mathbf{F}$)

$$W_d(x) = \int_{e\in E} \Psi_{NH}(\mathbf{F})dA_e = \sum_{e\in E} \Psi_{NH}(\mathbf{F})A_e$$

FEM Simulation

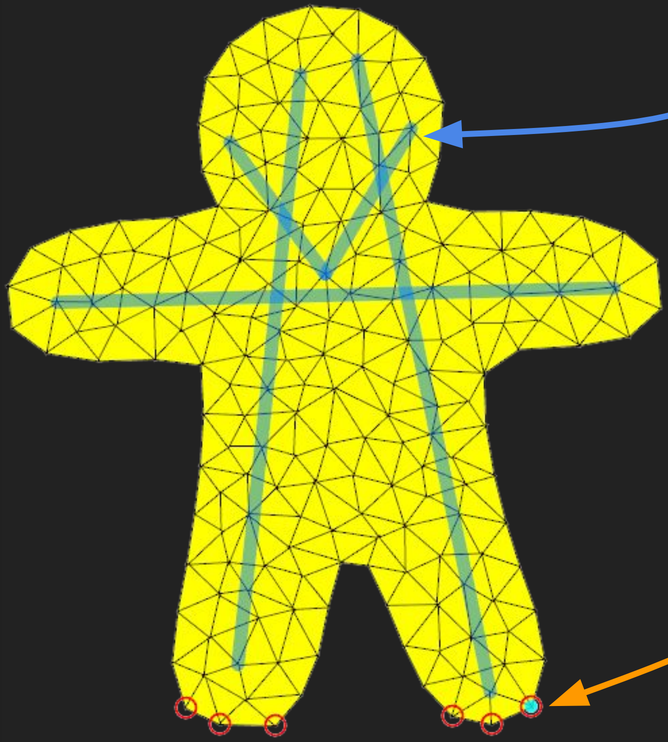
$$x = \operatorname{argmin}_{\tilde{x}} W_d(\tilde{x})$$

$$x = \operatorname{argmin}_{\tilde{x}} W_d(\tilde{x}) + W_f(\tilde{x}) + W_m(\tilde{x})$$

Fixed points

Muscles

Muscles & fixed point elements



Muscles:

$$W_{m,i}(x_1, x_2) = \frac{k}{2}(||x_1 - x_2|| - l)^2$$

parameters

Fixed point elements:

$$W_{f,i}(x_j) = \frac{k}{2}||x_j - p_t||^2$$

Sensitivity Analysis

Given an optimization problem

$$x(p) = \operatorname{argmin}_{\tilde{x}} W(\tilde{x}, p)$$

How does **x** change w.r.t. **p**?

$$\frac{dx}{dp} = ?$$

$$f = \nabla_x W$$

$$f(x(p)) = 0, \ f(x(p + \Delta p)) = 0$$

$$\Rightarrow \lim_{\Delta p \to 0} \frac{f(x(p+\Delta p)) - f(x(p))}{\Delta p} = \frac{df}{dp} = 0$$

$$\frac{df}{dp} = 0 = \frac{\partial f}{\partial x}\frac{dx}{dp} + \frac{\partial f}{\partial p}$$

$$\frac{dx}{dp} = -\frac{\partial f}{\partial x}^{-1}\frac{\partial f}{\partial p}$$

Shape Optimization

We want to minimize an objective function

$$p = \text{argmin}_{\tilde{p}} T(\tilde{p}, x(\tilde{p}))$$

Given the simulation:

$$x(p) = \text{argmin}_{\tilde{x}} W(\tilde{x}, p)$$

Random search? → slow!

Gradient descent? → need $\dfrac{dT}{dp}$

Sensitivity Analysis to the rescue!

$$\frac{dT}{dp} = \frac{dx}{dp}^T \frac{\partial T}{\partial x} + \frac{\partial T}{\partial p}$$

Sensitivity Analysis Steepest Descent

$$\frac{dT}{dp} = \frac{dx}{dp}^T \frac{\partial T}{\partial x} + \frac{\partial T}{\partial p}$$

$$\frac{dx}{dp} = -\frac{\partial f}{\partial x}^{-1} \frac{\partial f}{\partial p}$$

Sensitivity Analysis

$$\frac{dT}{dp} = -\frac{\partial f}{\partial p}^T \underbrace{\frac{\partial f}{\partial x}^{-1} \frac{\partial T}{\partial x}}_{g} + \frac{\partial T}{\partial p}$$

$$g = \frac{\partial f}{\partial x}^{-1} \frac{\partial T}{\partial x} \Rightarrow \frac{\partial f}{\partial x} g = \frac{\partial T}{\partial x}$$

Adjoint method

Sensitivity Analysis Steepest Descent

$$\frac{dT}{dp} = \frac{dx}{dp}^T \frac{\partial T}{\partial x} + \frac{\partial T}{\partial p}$$

$$\frac{dx}{dp} = -\frac{\partial f}{\partial x}^{-1} \frac{\partial f}{\partial p}$$

Sensitivity Analysis

easy!

$$\frac{dT}{dp} = -\frac{\partial f}{\partial p}^T \underbrace{\frac{\partial f}{\partial x}^{-1} \frac{\partial T}{\partial x}}_{g} + \frac{\partial T}{\partial p}$$

Analytical form!

$$W_{m,i}(x_1, x_2) = \frac{k}{2}(||x_1 - x_2|| - l)^2$$

Hessian of ***W*** easy!

$$g = \frac{\partial f}{\partial x}^{-1} \frac{\partial T}{\partial x} \Rightarrow \frac{\partial f}{\partial x} g = \frac{\partial T}{\partial x}$$

Adjoint method

The Eigen library

Documentation: http://eigen.tuxfamily.org/dox/

```
Matrix<double, N, M> A;                          // matrix of size NxM
Matrix<double, -1, -1> B(N, M);                  // dynamically sized matrix
double y = A(i,j);                               // element at (i,j)
auto C = A.block<n, m>(p, q);                    // submatrix at A(p,q), size NxM
B.block<n,m>(i,j) = A.block<n, m>(p, q);// can also write to block-matrix!
A.setZero();                                     // sets all elements to zero
auto D = A.transpose() * B;                      // operator overloading
                                                 // for common operations
```

Libraries for your project

- 2D rendering: nanovg
- 3D rendering: OpenGL ← example code after easter break
- GUI: ImGui
- File I/O: nlohmann::json ← A3
- File dialogs: portable-file-dialogs ← A3
- Obj reading: tinyobjloader ← A3
- Simulation & optimization: this class!

After easter break we'll have a closer look at all this :)

→ Code Review

starter code:

github.com/computational-robotics-lab/comp-fab-a3

post issues there!

Questions

- Questions about assignments on corresponding issues page: https://github.com/computational-robotics-lab/comp-fab-a3/issues

- Other questions: in your personal repo, or moritzge@inf.ethz.ch