MAE 259B Group 2 Final Presentation

Siyuan Chen, Xiangzhou Kong, Long Chen
06/06/2018

Elasticity and inner pressure of a ball is related to the rebounding motion of the ball.

Deflegate scandal of New England Patriots

Background

Finite Element Method	Discrete Elastic Rod Model
Widely used in industry Abaqus, CATIA, ANSYS, etc.	What we are doing in this project (A lot faster)

Review of our midterm presentation

Started from simplest 2D DER

- Performance optimization (pre-computing reused terms)
- Initial curvature
- Circular structure
- Inflation pressure (models as force-per-unit-length)
- Surface contact (naïve predictor-corrector)
- Damping (relative to center-of-mass)
- Adaptive time stepping (limit momentum change per step)
- Dynamic friction (partially Euler-forward)

Review of our midterm presentation

Bouncing ring structure on surface w/o and w/ friction

🎥 no-friction.mp4	🎥 friction-0.2.mp4

Not all PDF viewers can play videos. Videos files can be alternatively found at

https://github.com/kmxz/mae259b/tree/master/final-presentation/video.

Review of our midterm presentation

Damping

Naïve damping $\underline{F}_{d,i} \propto -\underline{u}_i$ ruins bulk motion (as our bulk velocity is high)

Use velocity relative to center of mass: $\underline{F}_{d,i} \propto -(\underline{u}_i - \underline{u}_{cm})$

No damping	Naïve damping	CM-Relative damping
🎥 no-damp.mp4	🎥 bad-damp.mp4	🎥 good-damp.mp4

Adaptive time stepping

After trying Δx, Δv and contact as criteria for determining time step size, we resulted in using momentum change as the optimal criteria.

Step size should be inverse proportional to the maximum nodal reaction force from the ground.
We limit $(F_{max})(dt)$ to a threshold ($\approx 0.005\,\mathrm{N} \cdot \mathrm{s}$) for momentum change per step.
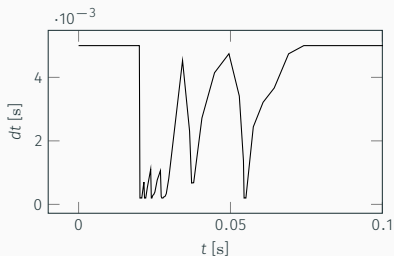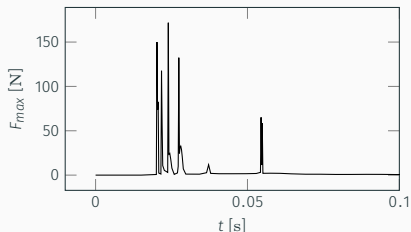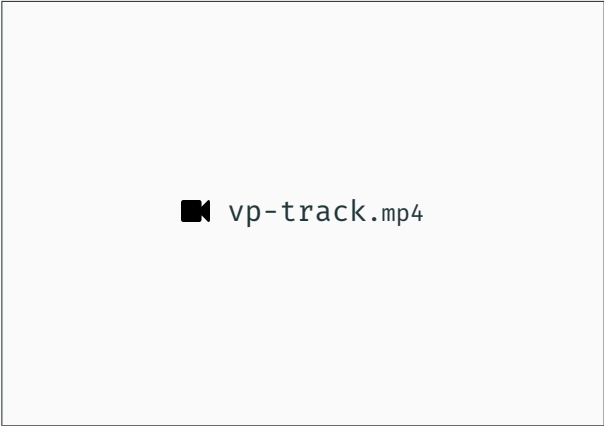
Adaptive time stepping:

🎥 ats.mp4

We spent quite some time trying to write a 3-D discrete shell simulation in C++.

But we gave up, considering the limited time and difficulty in debugging.

We continued improving the 2-D DER model.

Let viewport track the ring:

🎥 vp-track.mp4

Baraff-Witkin mass modification

For surface contact, we used Baraff-Witkin mass modification method to replace our naïve predictor-corrector method.

Why?

- Our previous implementation need to extract sub-matrices of different sizes in each time step. With mass modification method, implementation can be neater.
- Our previous implementation only handles horizontal surface. With mass modification method, sloped surface can be included.

Baraff-Witkin mass modification

- Detect DOFs to be constrained, just as before.
- Compute "imposed velocity change" $\underline{z}$ for those DOFs to stay above contact surface.
- In time marching, use v_i instead of x_i:

$$\frac{\delta v_i}{\delta t} = \frac{x_i(t_{k+1}) - x_i(t_k)}{(dt)^2} - \frac{v_i(t_k)}{dt}$$

- Therefore, EOM becomes: $\Delta \underline{v} - (dt)\underline{\underline{w}}\,\underline{F} - \underline{z} = 0$
- Inversed mass matrix $\underline{\underline{w}}$ is composed by 2×2 component matrices $\underline{\underline{w}}_i$, which is $\frac{1}{m_i}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ for unconstrained nodes, $\frac{1}{m_i}(\underline{\underline{I}} - \underline{n}\,\underline{n}^T)$ for nodes constrained in $\underline{n}$ direction.

Newton-Raphson iteration

Since we are solving for $\Delta \underline{v}$, in each iteration:

$$\Delta \underline{v} \leftarrow \Delta \underline{v} - \underline{\underline{J}} \backslash \underline{f}$$

where $\underline{f} = \Delta \underline{v} - (dt) \underline{\underline{w}} \, \underline{F} - z$ and $\underbrace{\underline{\underline{J}}}_{\frac{\partial \underline{F}}{\partial \Delta \underline{v}}} = \underline{\underline{I}} - (dt)^2 \underbrace{\underline{\underline{J}}^{old}}_{\frac{\partial \underline{F}}{\partial \underline{x}}}$

Yeah, sloped surface.

🎥 sloped.mp4

Baraff-Witkin mass modification

Yeah, sloped surface.

🎥 sloped-track.mp4

Baraff-Witkin mass modification

The procedure of handling **non-zero friction**, as covered in class, is:

1. Constrain both directions (x, y) of a node (to emulate static friction)
2. Decompose reaction force $\underline{f}_i$ into normal force $\underline{f}_i^{normal}$ and $\underline{f}_i^{friction}$ (by $\underline{f}^{normal} = \underline{f}_i \cdot \underline{n}_{wall}$ and $\underline{f}^{friction} = \underline{f}_i - \underline{f}_i^{normal}$)
3. Test if $f_i^{friction} > \mu_{static} f_i^{normal}$. If so, release x-constraint and apply dynamic friction $\mu_{dynamic} f_i^{normal}$ instead.

However, in our scenario, normal force is always relatively small. $f_i^{friction} > \mu_{static} f_i^{normal}$ happens all the time. Static friction never get applied. Dynamic pressure is simply added to $F^{external}$.

Variable inflation pressure

Recall that we discretized the inflation pressure as below:

However, instead of constant f, we may expect the pressure to get higher when the volume inside the structure is reduced.

For ideal gas:

$$pV = nRT$$

Assume the temperature is constant. Then $p \propto \frac{1}{V}$.

"Rolling ribbon" (no internal "pressure")

Different Young's modulus, other parameters kept the same:

🎥 roll-2e6.mp4

🎥 roll-3e6.mp4

🎥 roll-4e6.mp4

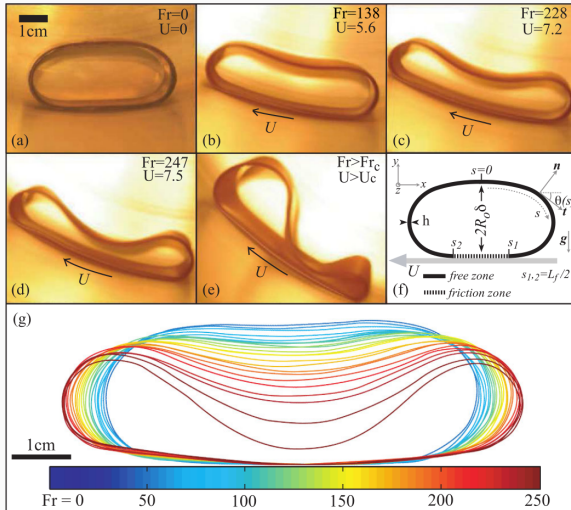🎥 roll-8e6.mp4

"Rolling ribbon" (no internal "pressure")

Figure from "Rolling Ribbons" (P. S. Raux *et al.*, 2010)

Variable inflation pressure

For computing the internal volume, we take the area of the 2D ring. Since we already discretized the structure into a polygon, we can use shoelace formula to calculate area:

$$A = \frac{1}{2} \left[\sum_{i=1}^{n-1} x_i y_{i+1} + x_n y_1 - \sum_{i=1}^{n-1} x_{i+1} y_i - x_1 y_n \right]$$

Then, for each time frame, $p(t_k) = \frac{A(t_0)}{A(t_k)} p(t_0)$

We simple ignore this effects' contribution to Jacobian (otherwise the Jacobian will no longer be banded), it turns out we can still get convergence.

Variable inflation pressure

Variable inflation pressure does make results more realistic!

◼ pressure-inv.mp4

◼ pressure-var.mp4