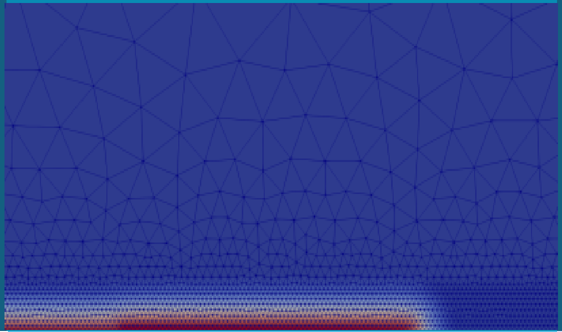
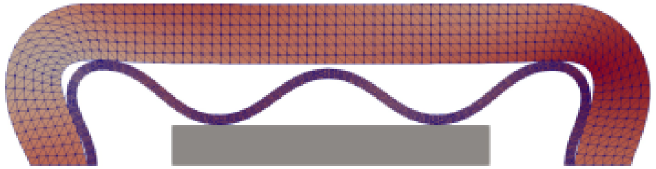
SAND2021-8944C

Sandia National Laboratories

An optimization-inspired solver for inequality constrained nonlinear solid mechanics:

frictional contact, buckling, and phase-field fracture





Michael Tupek, Brandon Talamini

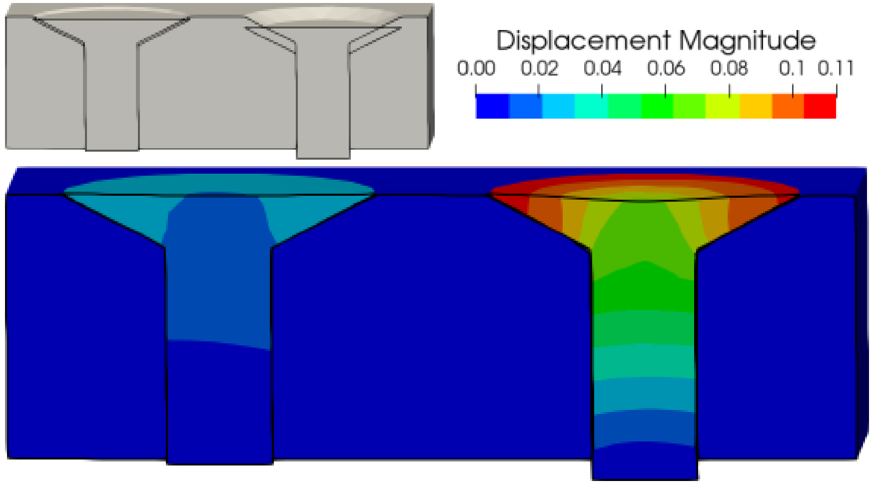
U.S. DEPARTMENT OF ENERGY NNSA

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

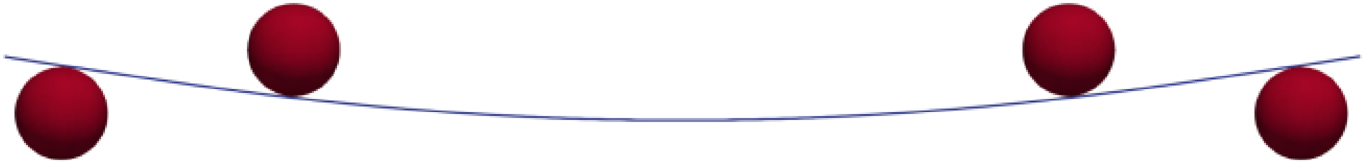
Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Key goal: robust and accurate implicit quasi-statics

- Design/qualification of nonlinear mechanical systems, want robust and credible capability
- Desire tight tolerances and **mechanically stable** solutions
- Efficient and robust contact enforcement (with friction)
- Handle material instabilities (buckling, failure, necking)

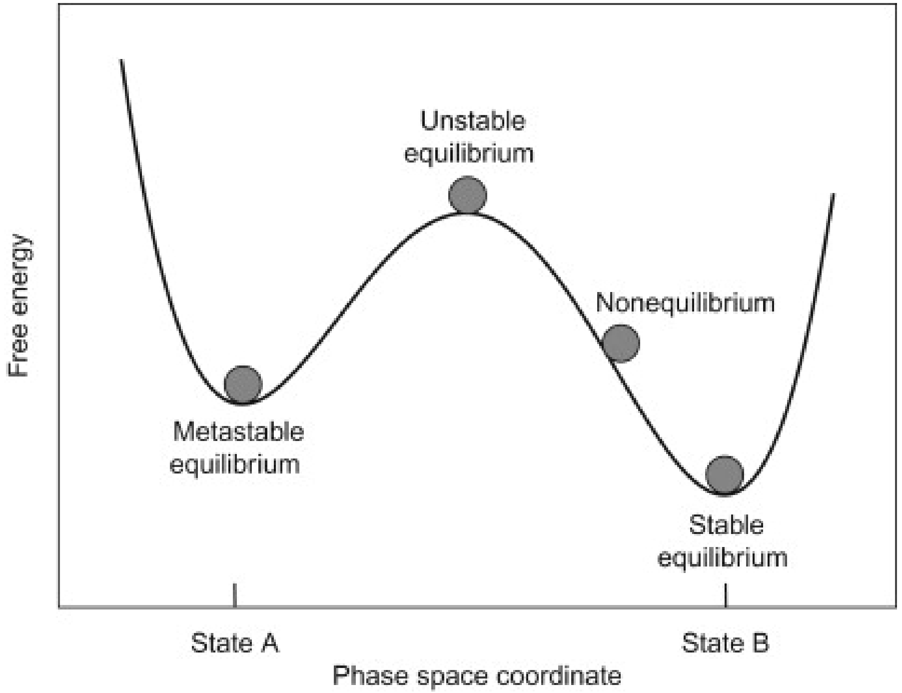


Overlap removal and bolt pre-load

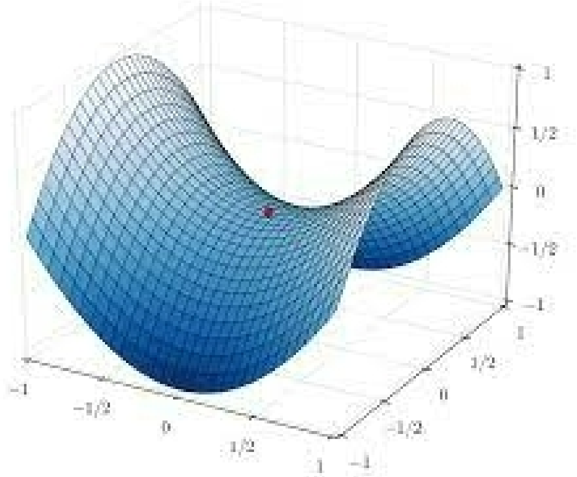


Four-point bend

Isn't statics just 'F=0'? **No!**



*



Unstable second order saddle

Higher order saddle**

Stable equilibrium satisfy
second order optimality conditions:

$$\mathbf{K} \succeq 0$$

(stiffness eigenvalues are non-negative)

The 3 reasons for non-convexity***

1) Non-unique solutions: real materials can buckle
2) Frame invariance
3) Positive material densities: $\psi(\mathbf{F}) \to \infty, \text{ as } \det \mathbf{F} \to 0$

* *Nucleation in Condensed Matter*. *Kelton and Greer*. *2010*.
** mathcurve.com
*** *Mathematical Foundations of Elasticity*. Marsden, Hughes. 1994. Just be glad I'm not discussing Hadamard Ellipticity.

Can we always define an energy?

No (but it's close)

The big challenges:

1. Friction
2. Following loads (pressure BCs)
3. Legacy material models (hypo-elastic)

These have asymmetric stiffnesses.

What if we don't even know the energy?

Use incremental work as a surrogate model for the energy: $\frac{1}{2}\left(\mathbf{f}^n + \mathbf{f}^{n+1}\right) \cdot \left(\mathbf{u}^{n+1} - \mathbf{u}^n\right)$.

Only accept solution steps with negative work increments.

A second order approximation to the work increment, so quadratic convergence when possible.

What is the best way to minimize energy?

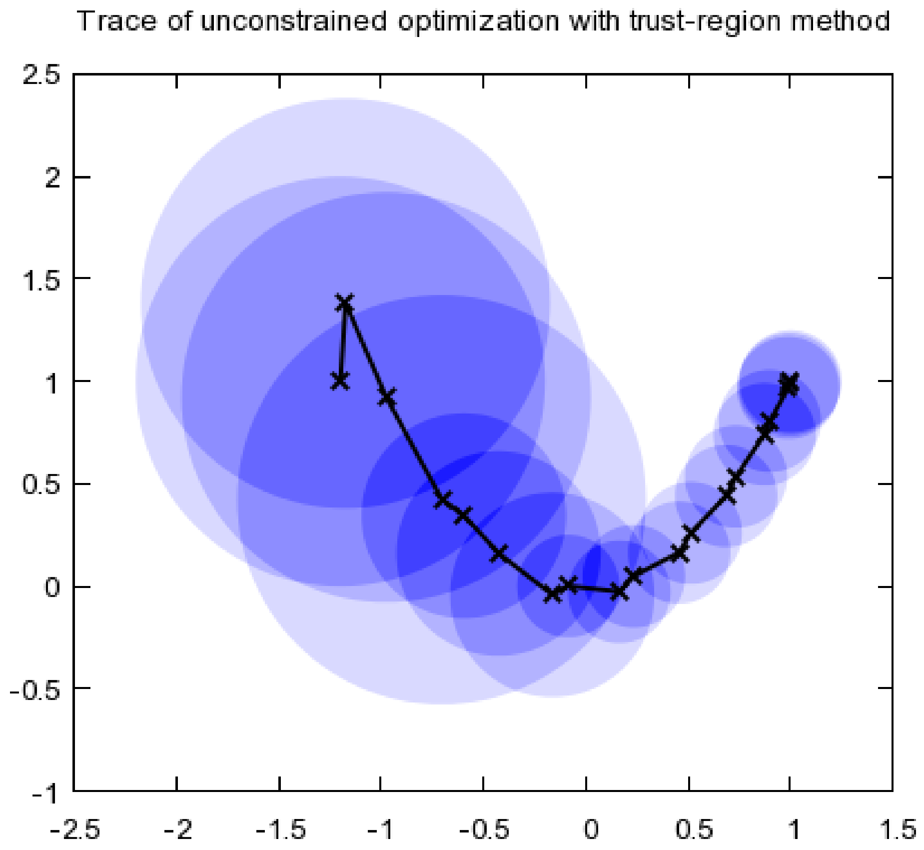
Probably trust-region solvers

Important ideas:

1. Local quadratic model
2. Inner iterations use preconditioned linear CG
3. Quickly identity saddle points and directions of negative curvature

Algorithm 1 Trust-region algorithm

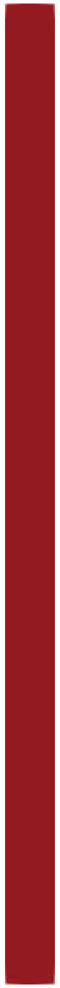
Initialize: $\mathbf{x}$, $\Delta > 0$, $\epsilon > 0$, $0 \le \eta_1 \le \eta_2 < \eta_3 < 1$, $0 < t_1 < 1 < t_2$
while $\|\nabla f(\mathbf{x})\| > \epsilon$ **do**
  $\mathbf{z} := \arg\min_{\mathbf{z}'} m(\mathbf{z}'; \mathbf{x})$, s.t. $\|\mathbf{z}'\| < \Delta$ (the minimization can be approximate)
  $\mathbf{x}' := \mathbf{x} + \mathbf{z}$
  $\rho := \frac{f(\mathbf{x}) - f(\mathbf{x}')}{m(\mathbf{0};\mathbf{x}) - m(\mathbf{z};\mathbf{x})}$
  if $\rho > \eta_2$ **then**
    if $\rho > \eta_3$ **and** $\|\mathbf{z}\| = \Delta$ **then**
      $\Delta := t_2 \Delta$
    end if
  else
    $\Delta := t_1 \Delta$
  end if
  if $\rho > \eta_1$ **then**
    $\mathbf{x} := \mathbf{x}'$
  end if
end while



Machine learning has led to a recent growth in non-convex optimization research, which we hope to leverage

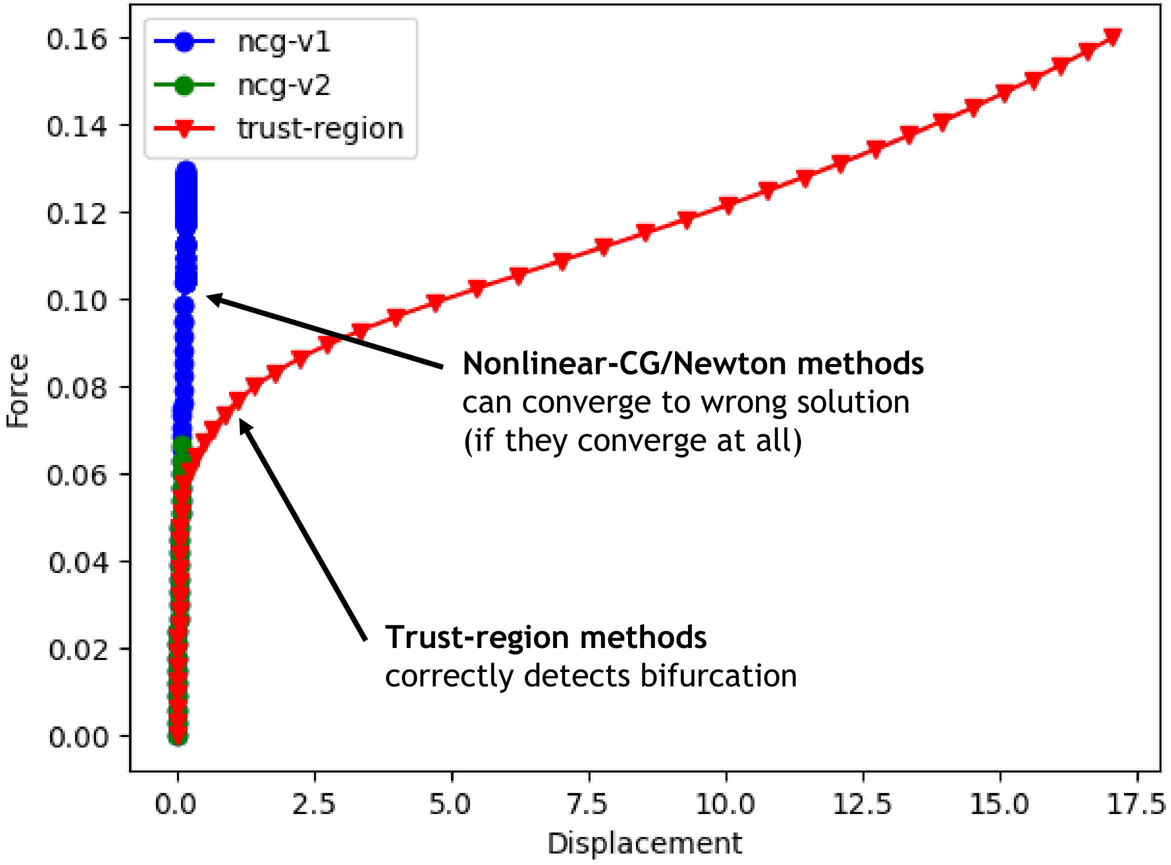
Can we solve problems with material instabilities?

Yes, lets try load-controlled Euler beam buckling

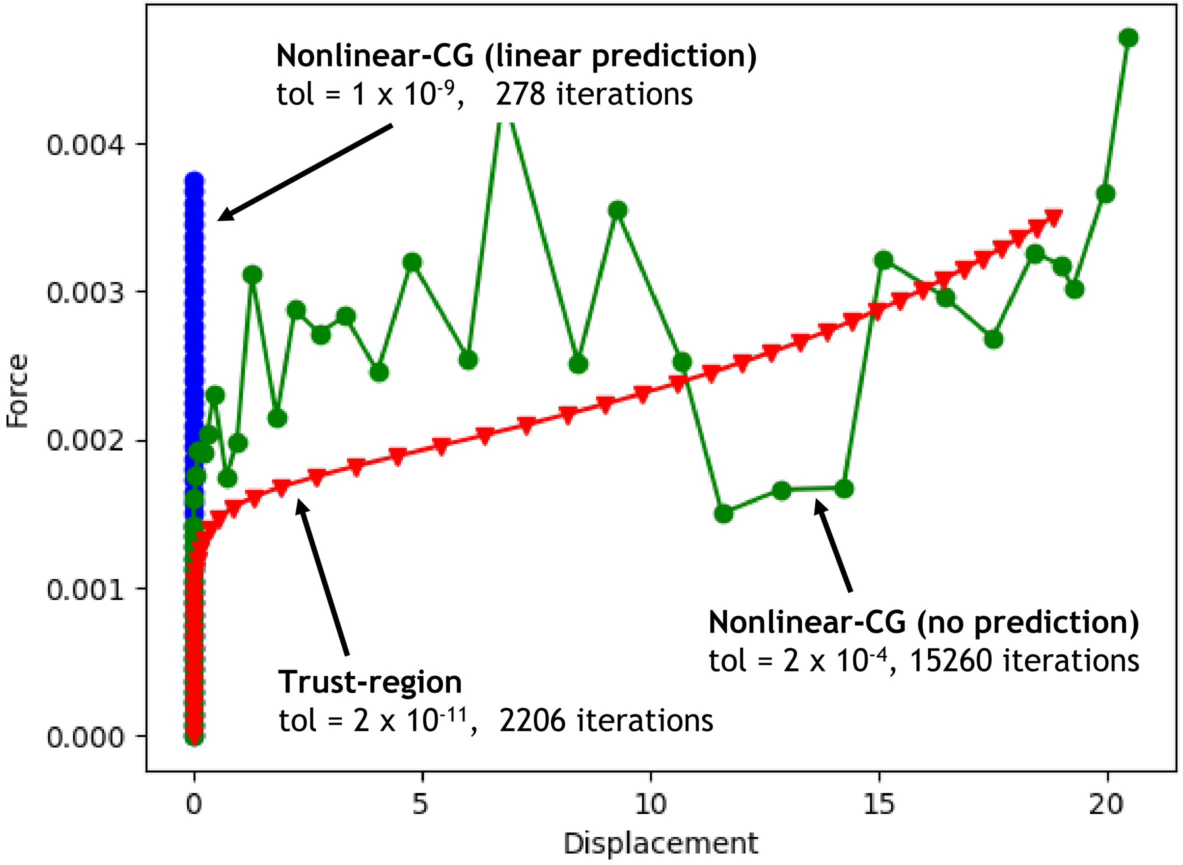


Comparison vs nonlinear conjugate gradient (force controlled)

Wide beam

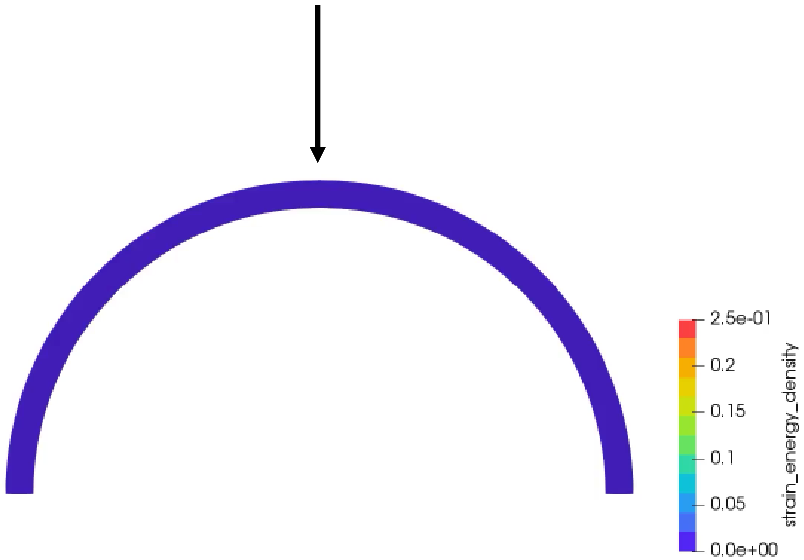


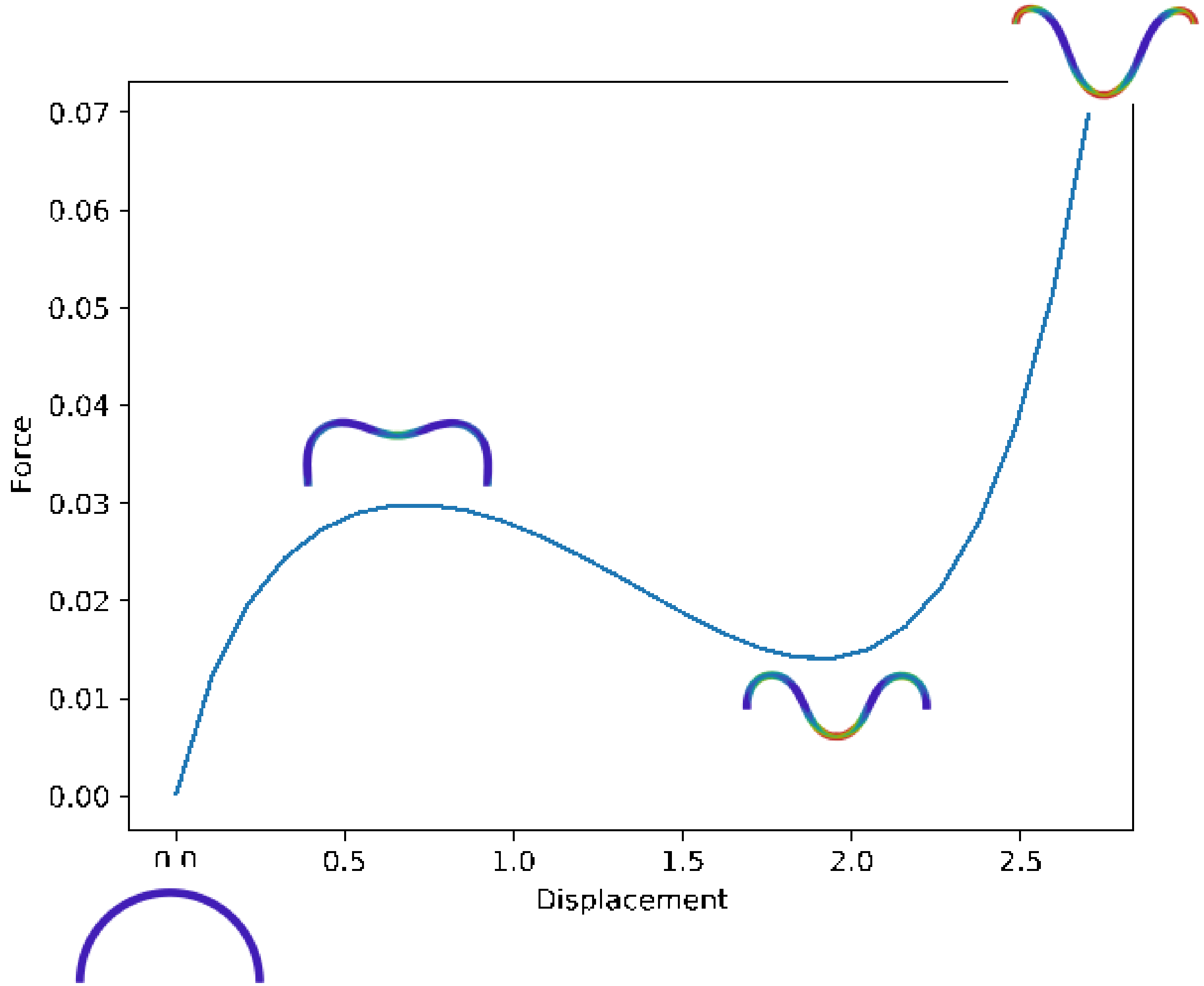
Thin beam



How hard is displacement-controlled loading with softening?

Not very hard

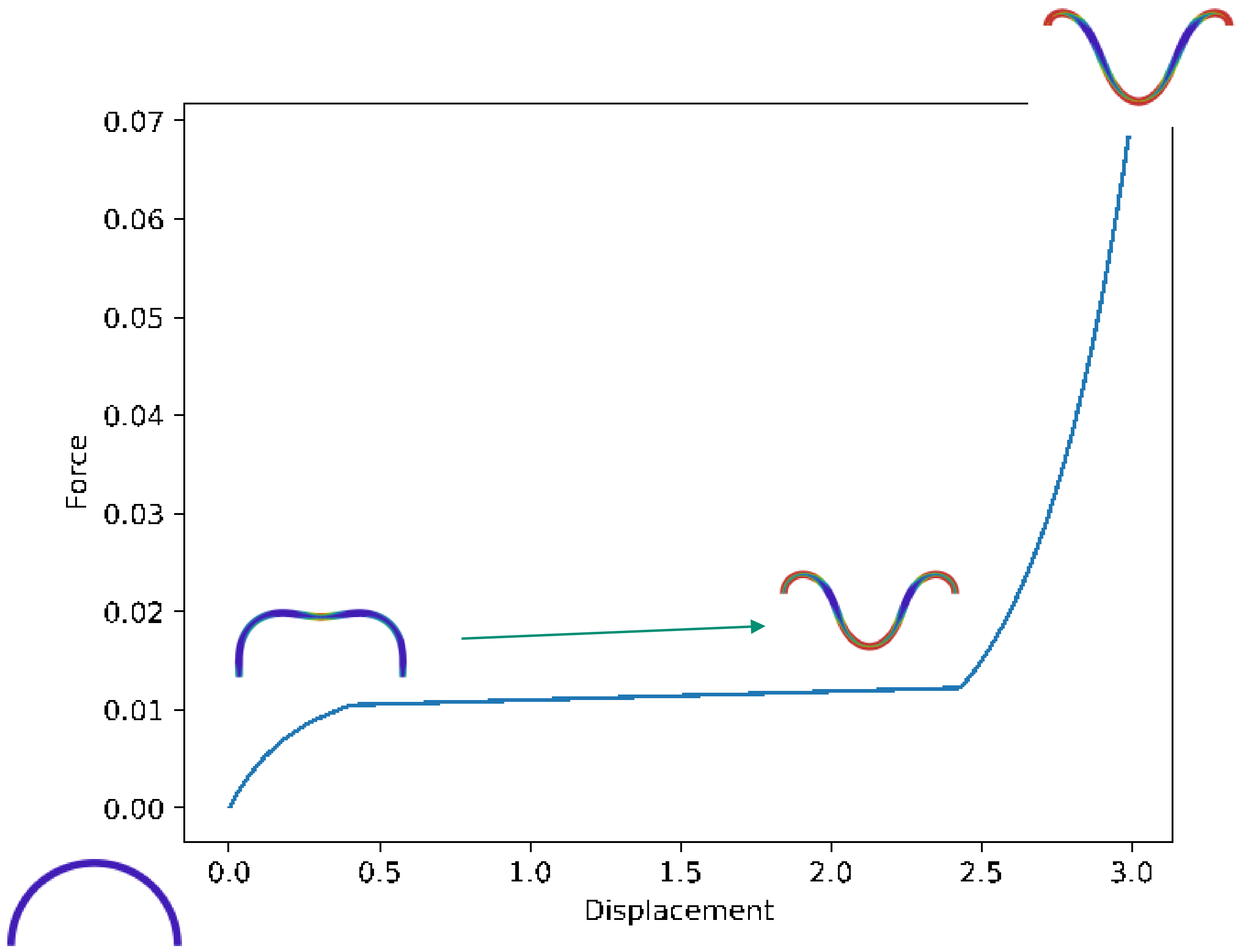




Neo-Hookean material, 256 quadratic triangles

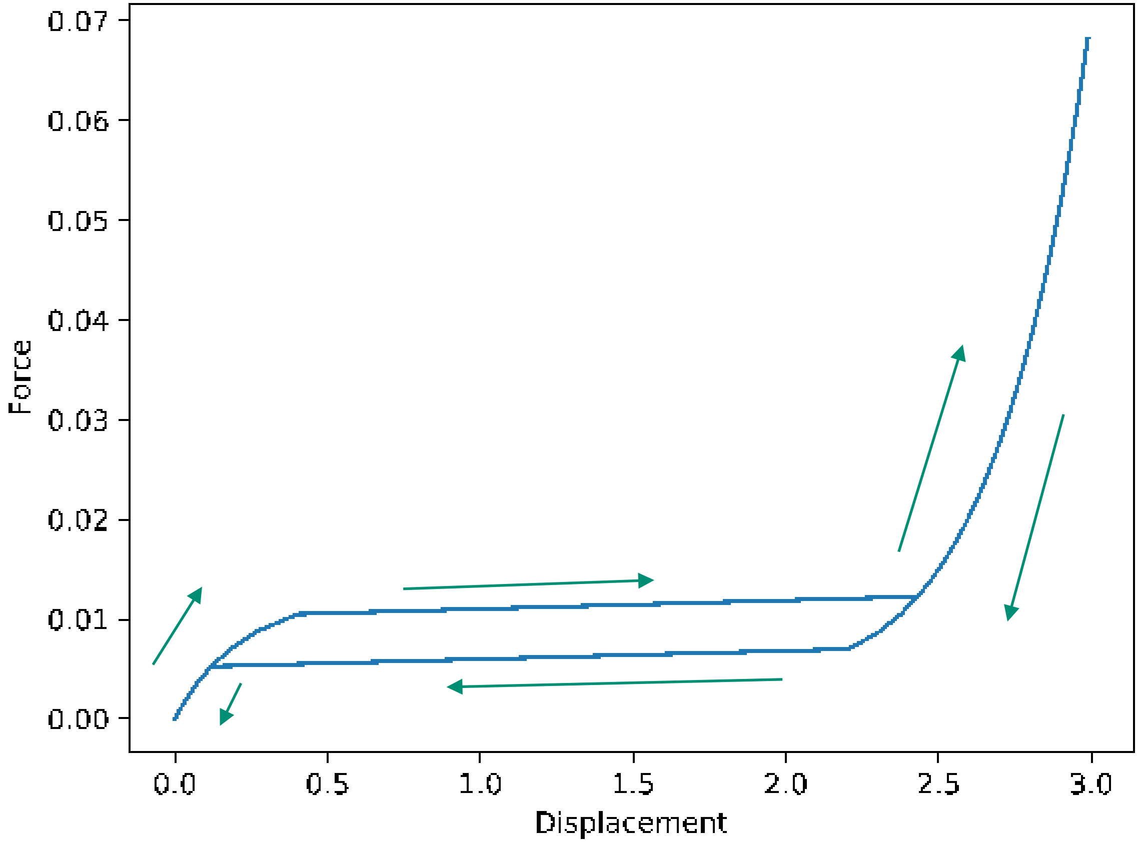
What about load-controlled snap-through?

Also not very hard



- Solver finds correct solution across large jump in configuration space
- No regularization used (implicit dynamics, artificial viscosity, etc.)

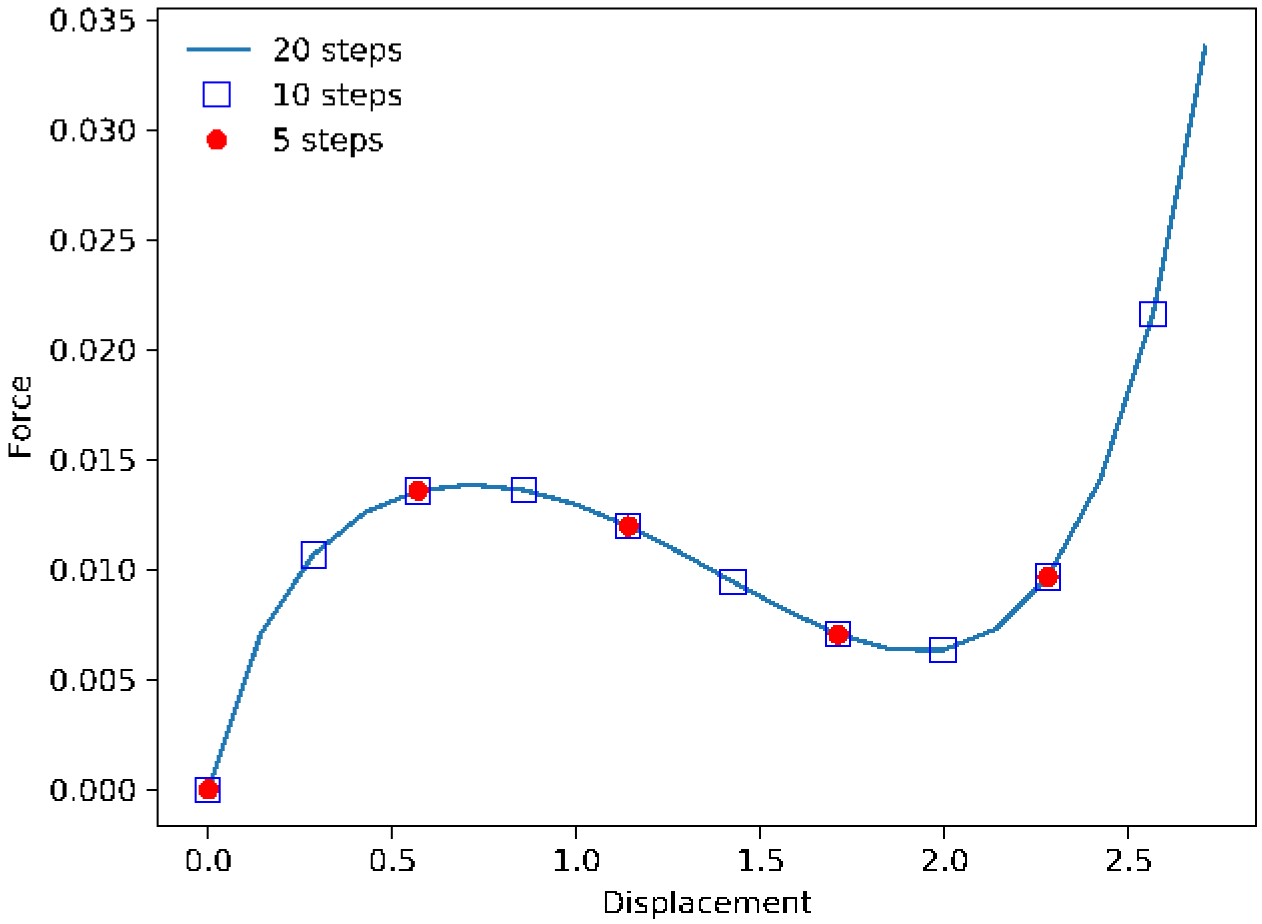
What if we reverse the loading?



- Robustness of trust-region solver allows it to accurately follow path-dependent behavior
- Necessary to resolve dissipation
- Note material model is elastic – the hysteresis is a structural effect

Can we take large load-steps?

Massive

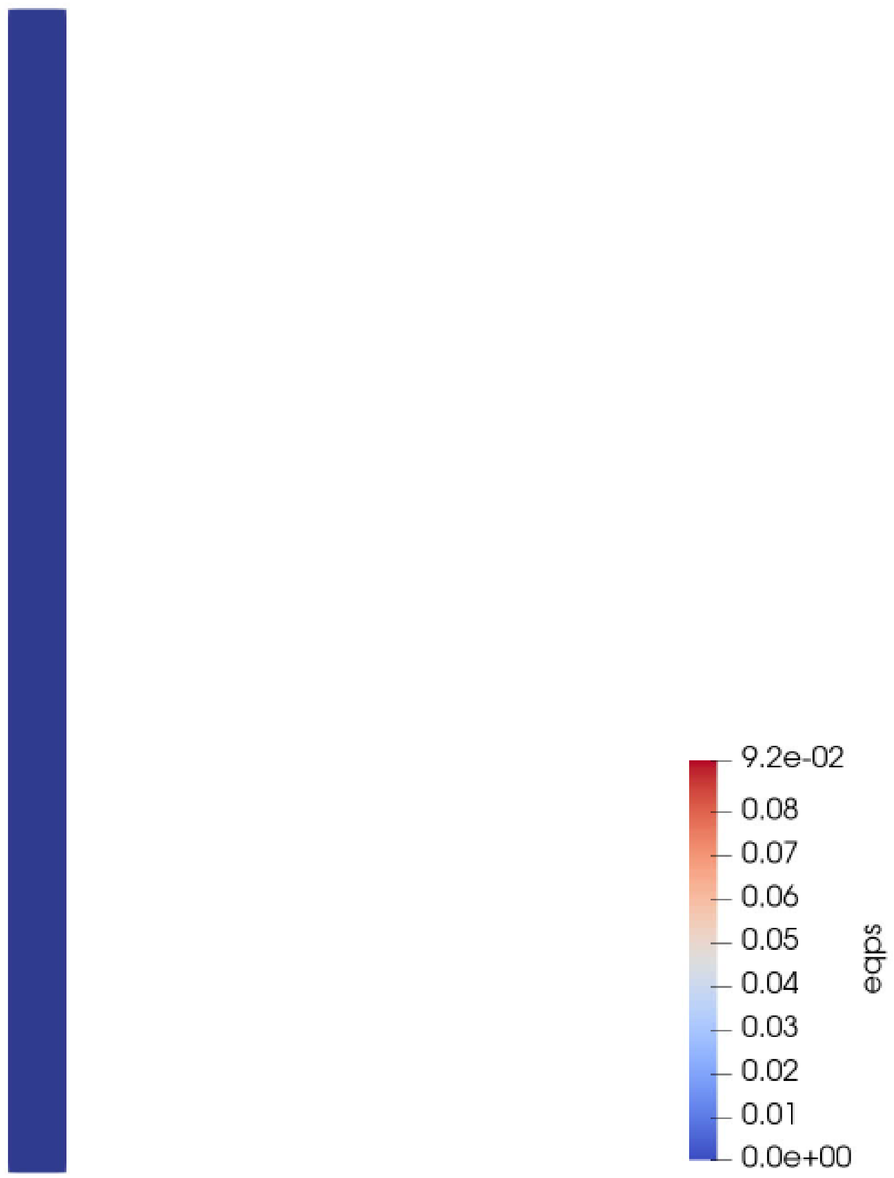


Trust-region solver follows path accurately even with very coarse load-steps

Each load-step uses a linearized prediction to avoid element inversion

What about plasticity?

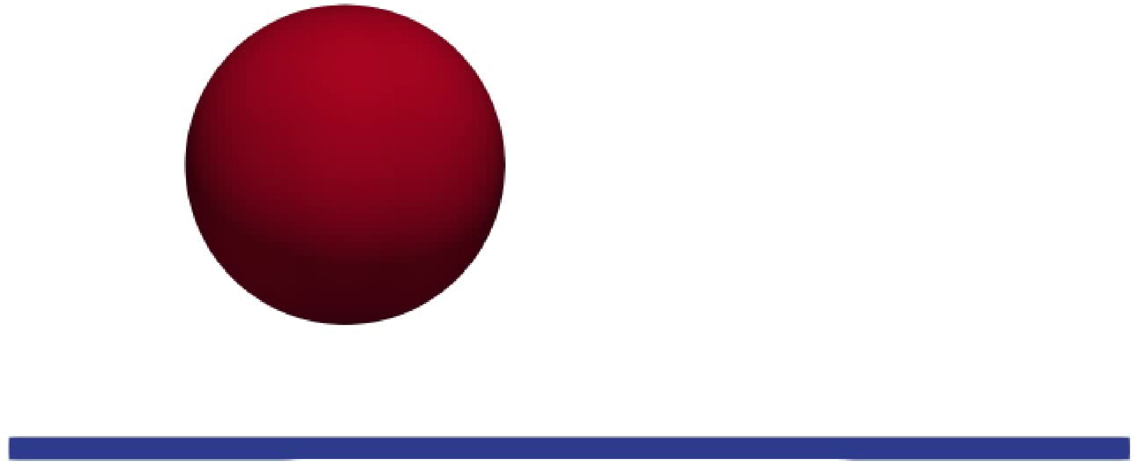
Sure



What about contact?

Just a constrained minimization problem*

$$
\begin{aligned}
\mathbf{u}^{\star} &= \arg\min_{\mathbf{u}} a(\mathbf{u}), \;\; \text{s.t.} \\
\mathbf{u} = \bar{\mathbf{u}} &\quad \text{for } \mathbf{x} \in \Gamma_D, \\
\phi(\mathbf{x}+\mathbf{u}) \geq 0 &\quad \text{for } \mathbf{x} \in \Gamma_N,
\end{aligned}
$$



*OK, there are inf-sup concerns too and constraint smoothness is essential

How do you enforce the inequality constraints?

$$\mathbf{u}^\star = \arg\min_{\mathbf{u}} a(\mathbf{u}), \ \text{s.t.}$$

$$\mathbf{c}(\mathbf{u}) \geq \mathbf{0}$$

Common options:

1. Penalty methods (wrong answer, stiff system of equations)
2. Interior point methods (stiff, cannot use initial guesses, must be always feasible)
3. **Augmented Lagrangian (simple, robust, but 1st order solver convergence)**

Solve sub-problem to second-order stationary point:

$$\bar{L}(\mathbf{u}; \boldsymbol{\lambda}^n, \boldsymbol{\kappa}^n) = a(\mathbf{u}) + \sum_{i=1}^{N_c} \frac{1}{2\kappa_i^n} \langle \lambda_i^n - \kappa_i^n c_i(\mathbf{u}) \rangle^2$$

Macauley bracket

$$\langle x \rangle = \begin{cases} x \text{ if } x > 0 \\ 0 \text{ if } x \leq 0 \end{cases}$$

First-order Lagrange multipliers update:

$$\lambda_i^{n+1} = \langle \lambda_i^n - \kappa_i^n c_i(\mathbf{u}^{n+1}) \rangle$$

Increase penalties κ_i^n when sufficient progress is not being made on constraints

Can we get faster convergence on the Lagrange multipliers?

Often

Fisher-Burmeister function $c \geq 0,\ \lambda \geq 0,\ \lambda c = 0 \iff \sqrt{k^2c^2 + \lambda^2} - kc - \lambda = 0$

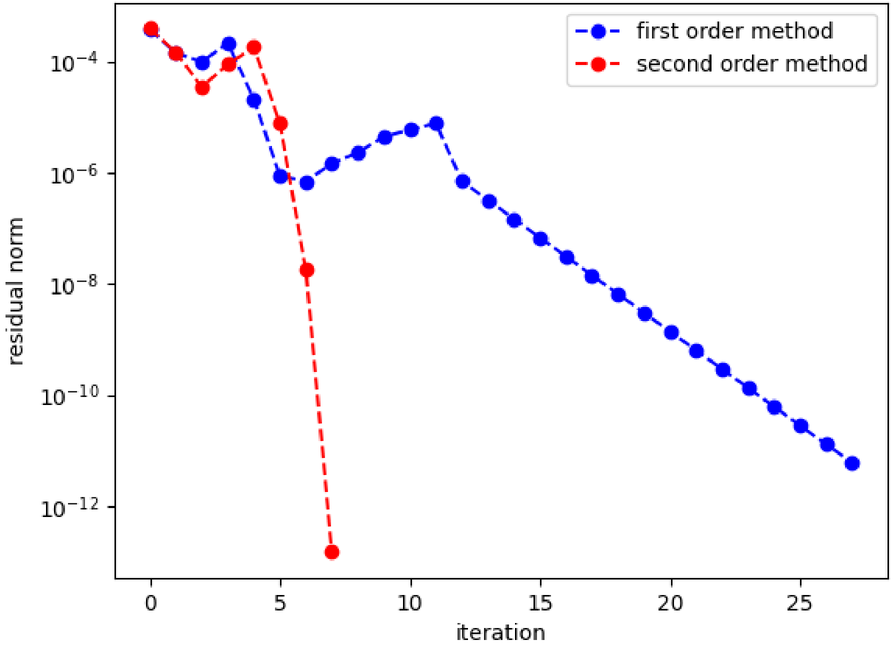
NCP residual: $\omega_{FB}(c, \lambda) = \sqrt{k^2c^2 + \lambda^2} - kc - \lambda$

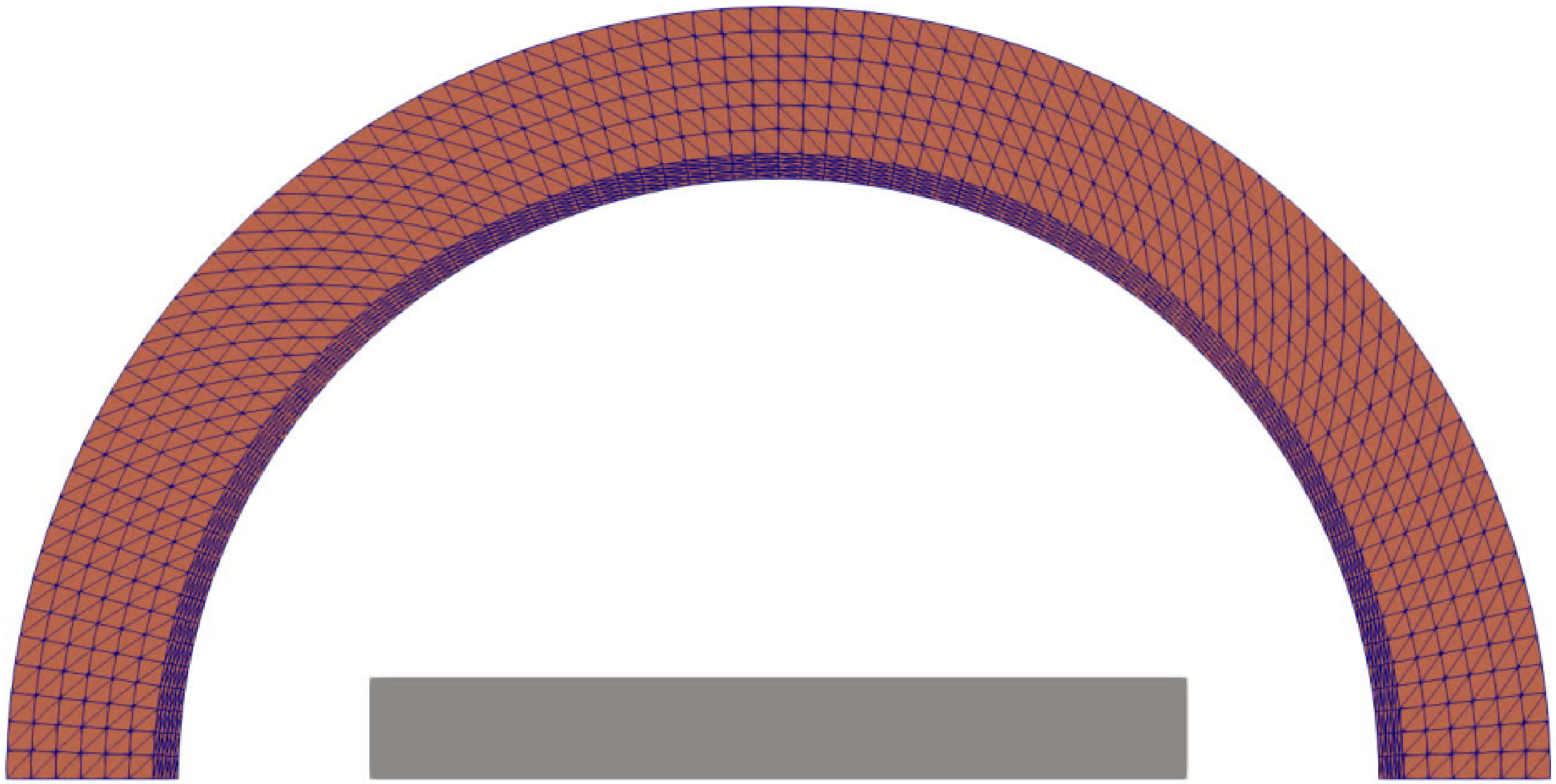
Combine NCP with mechanical residual: $\mathbf{R}(\mathbf{u}, \boldsymbol{\lambda}) := \begin{bmatrix} \mathbf{r}(\mathbf{u}; \boldsymbol{\lambda}, \mathbf{0}) \\ \boldsymbol{\omega}(\mathbf{u}; \boldsymbol{\lambda}) \end{bmatrix}$

Make sure to avoid saddle-points in u (for fixed λ).

Key idea:

1. Alternative between minimizing and doing a newton step
2. Preconditioned GMRES for full system solves





How about friction?

A minimization problem … if we knew the Lagrange multipliers!
(also known as a quasi-variational inequality)

$$\mathbf{u}^\star = \arg\min_{\mathbf{u}} a(\mathbf{u}) + \boldsymbol{\lambda}^\star \cdot \Upsilon(\mathbf{s}_\perp(\mathbf{u})), \text{ s.t.}$$

$$\mathbf{c}(\mathbf{u}) \geq \mathbf{0},$$

Smooth Friction potential:

$$\Upsilon(\mathbf{s}_\perp) = \begin{cases} \frac{\mu}{2 v_c \Delta t} \mathbf{s}_\perp \cdot \mathbf{s}_\perp & \text{for } \|\mathbf{s}_\perp\| \leq v_c \Delta t, \\ \mu \sqrt{\mathbf{s}_\perp \cdot \mathbf{s}_\perp} - \frac{v_c \Delta t}{2} & \text{for } \|\mathbf{s}_\perp\| > v_c \Delta t, \end{cases}$$

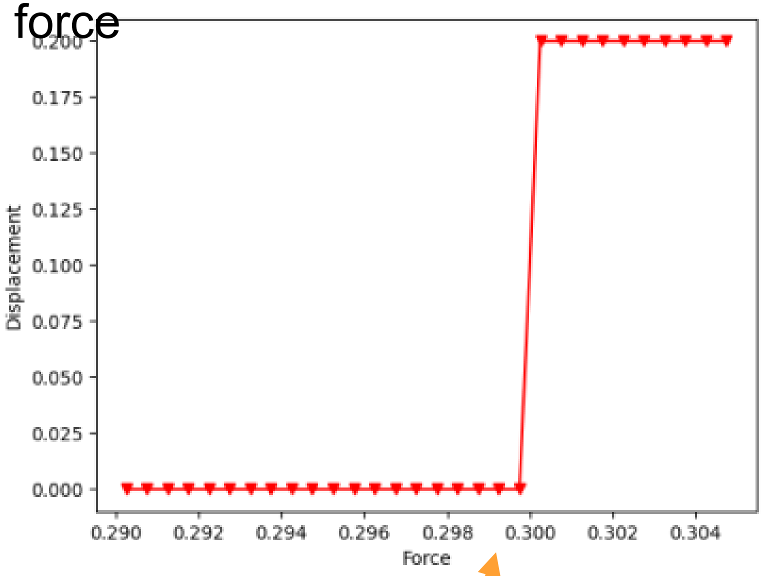
Strong form:

$$\begin{aligned} \mathbf{f}_m(\mathbf{u}) + \mathbf{f}_f(\mathbf{u}, \boldsymbol{\lambda}^\star) - \boldsymbol{\lambda}^T \mathbf{c}_{,\mathbf{u}}(\mathbf{u}) &= \mathbf{0}, \\ \mathbf{c}(\mathbf{u}) &\geq \mathbf{0}, \\ \boldsymbol{\lambda} &\geq \mathbf{0}, \\ c_i \lambda_i &= 0, \text{ for } i = 1, ..., N_c \\ \boldsymbol{\lambda}^\star &= \boldsymbol{\lambda}, \end{aligned}$$

Solve with augmented Lagrangian method, second-order version works too

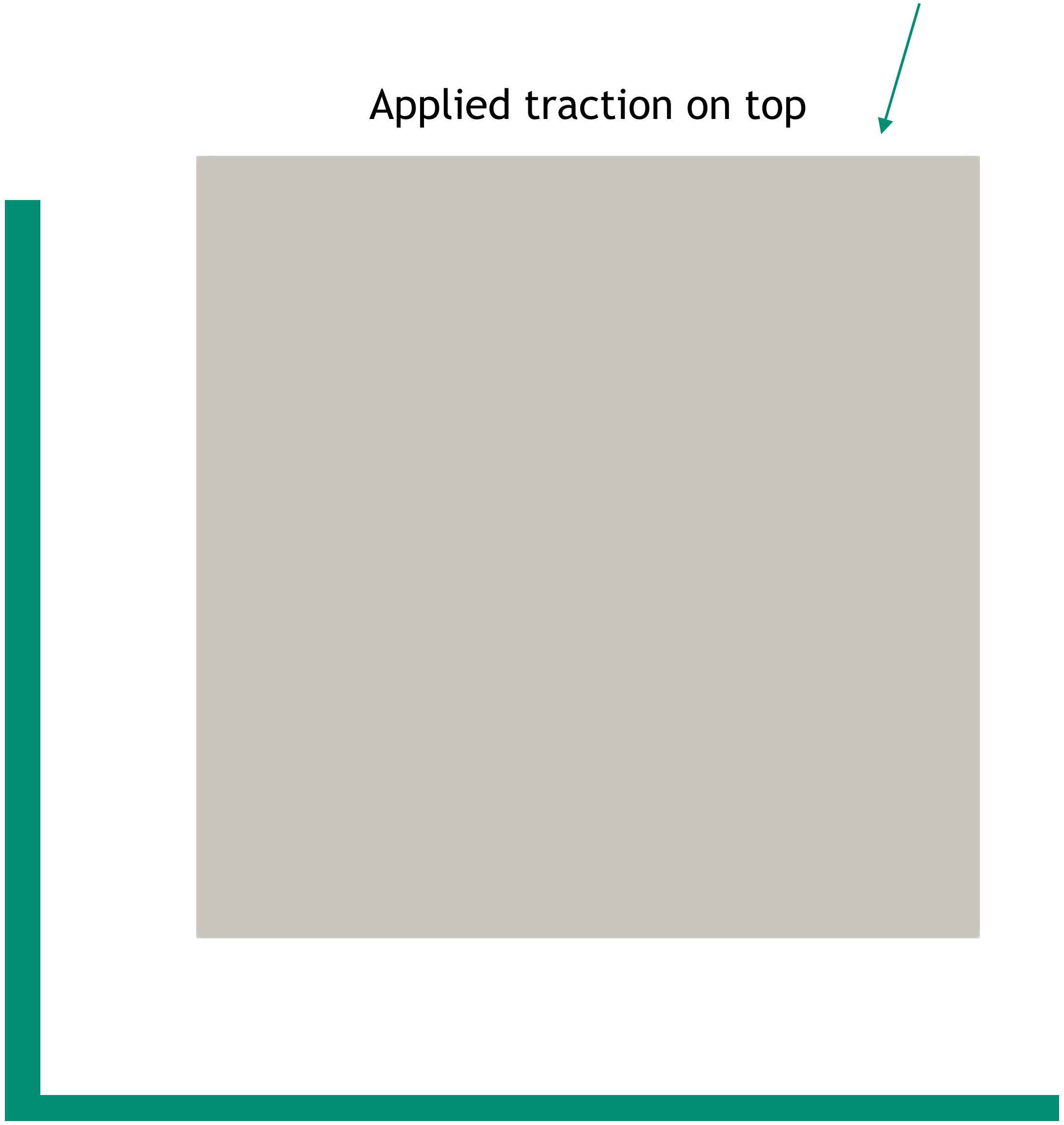
Frictional contact with no Dirichlet boundary conditions

Tangential displacement vs tangential force

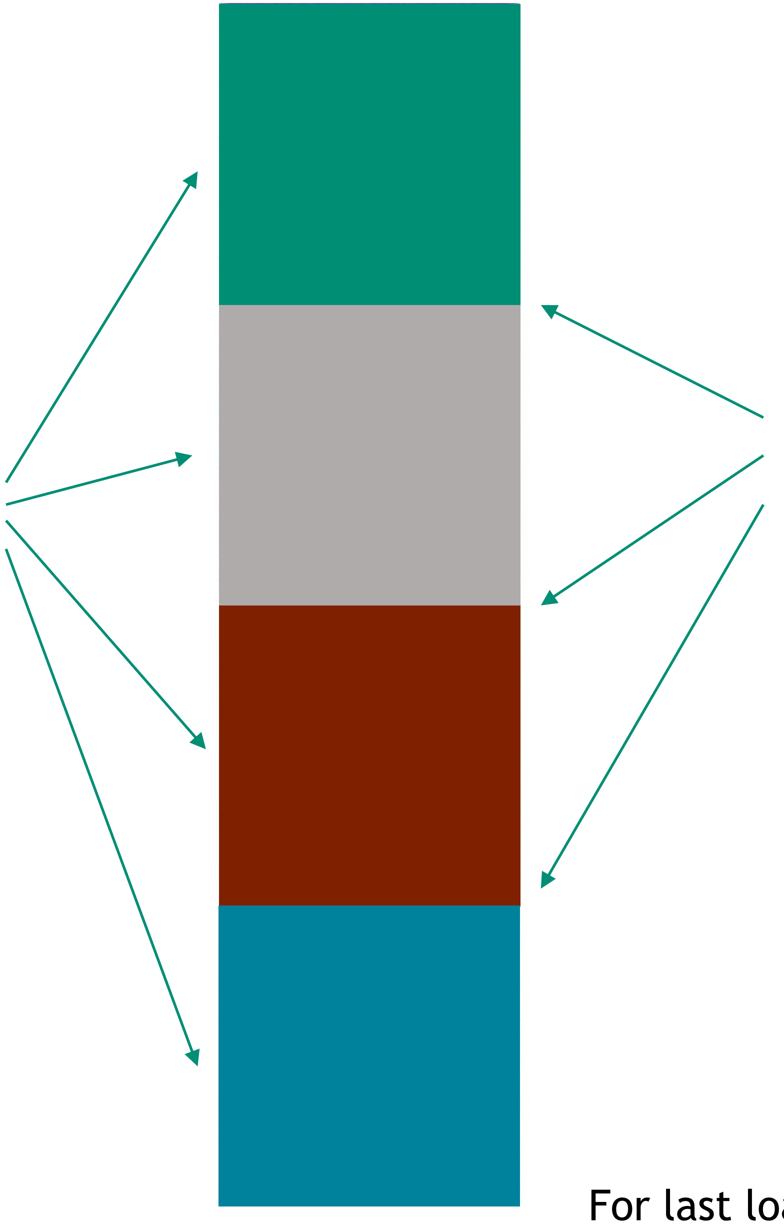


1. Slipping occurs at the expected tangential force corresponding to friction coefficient $\mu = 0.3$

2. This required solving to very tight tolerances on the mechanics and NCP



Contact with friction and eventual buckling



For last load-step, no stable solution could be found!

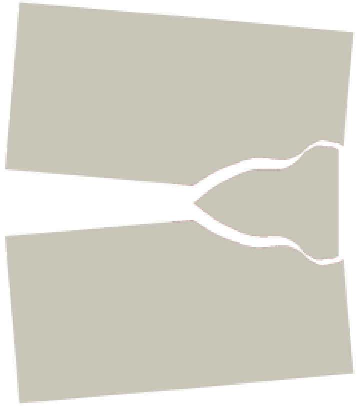
What about material failure?

Working on it, turns out failure is highly non-convex!

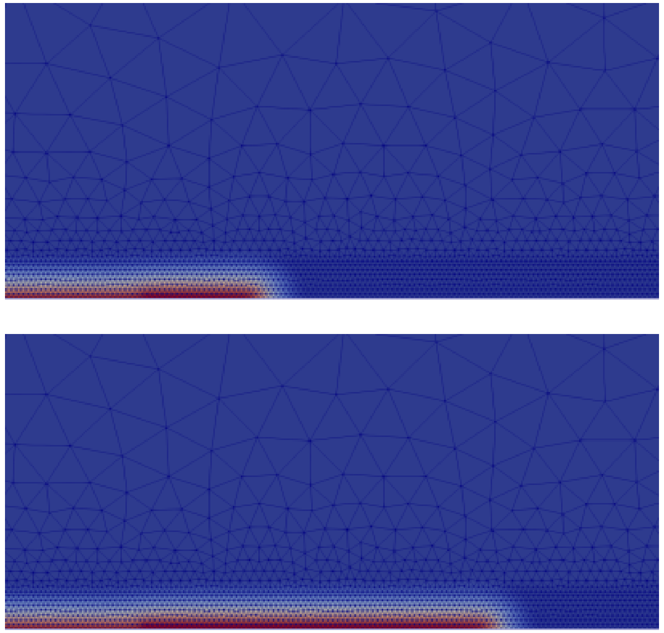
Phase-field fracture, minimize over displacement and damage

$$\int_\Omega g(d)\psi(\mathbf{u}) + \frac{G_c}{l}h(d) + \alpha G_c l|\nabla d|^2 \, d\mathbf{X}$$

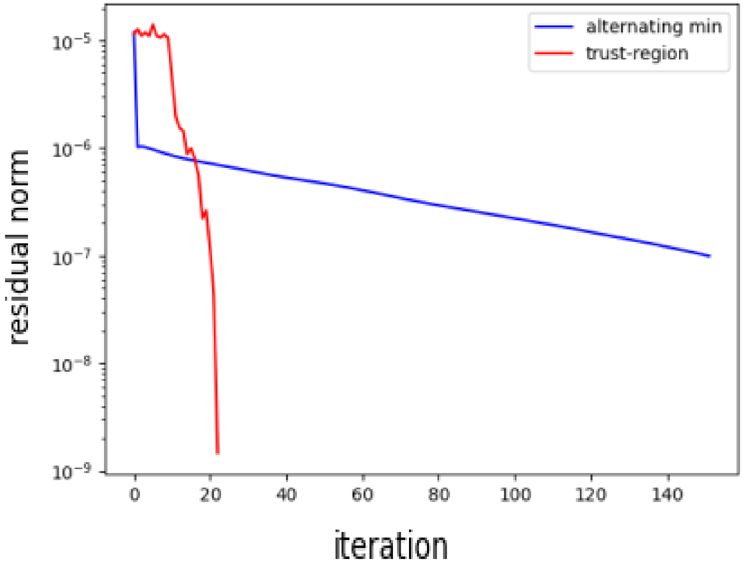
$$\text{s.t. } d \geq 0$$



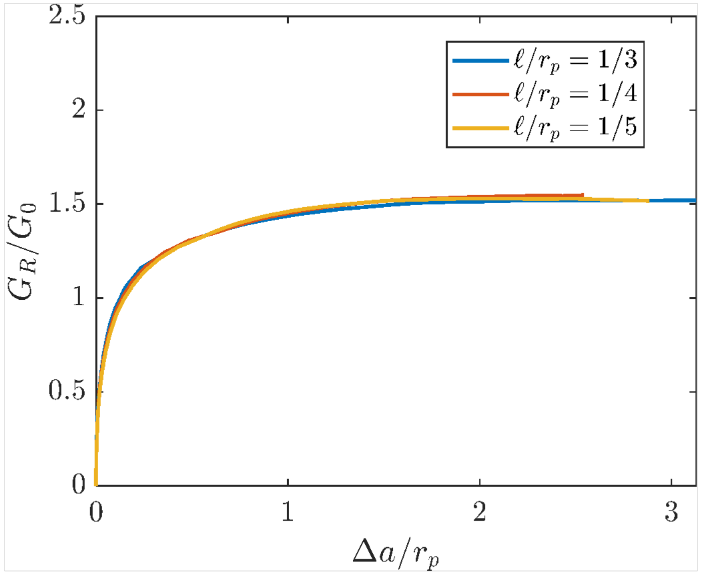
Dynamic phase-field image



Robustly propagates damage across multiple elements in 1 load-step



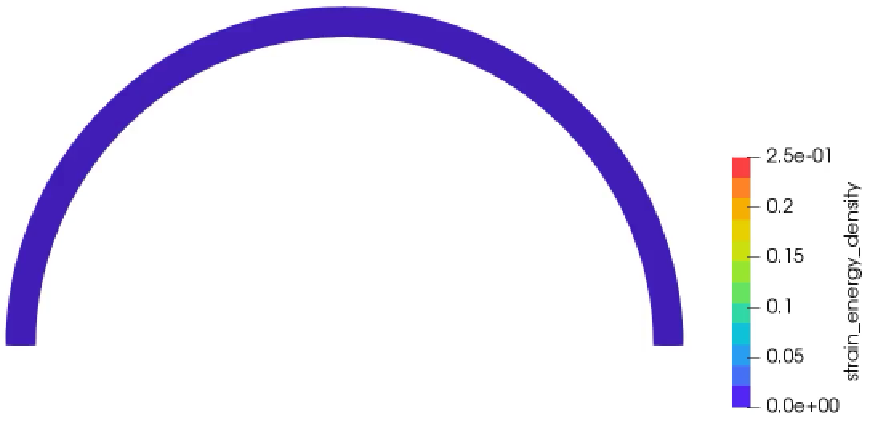
Super-linear solver convergence

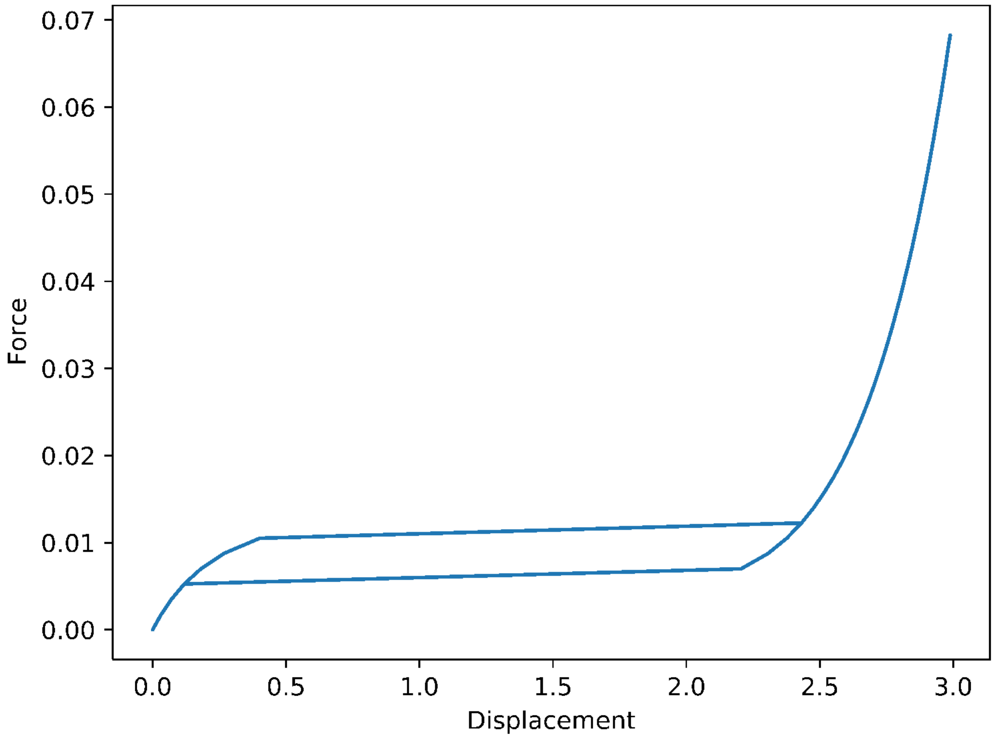


Mesh convergent stable crack growth*

*Attaining regularization length insensitivity in phase-field models of ductile failure, Talamini, et al. CMAME. 2021.

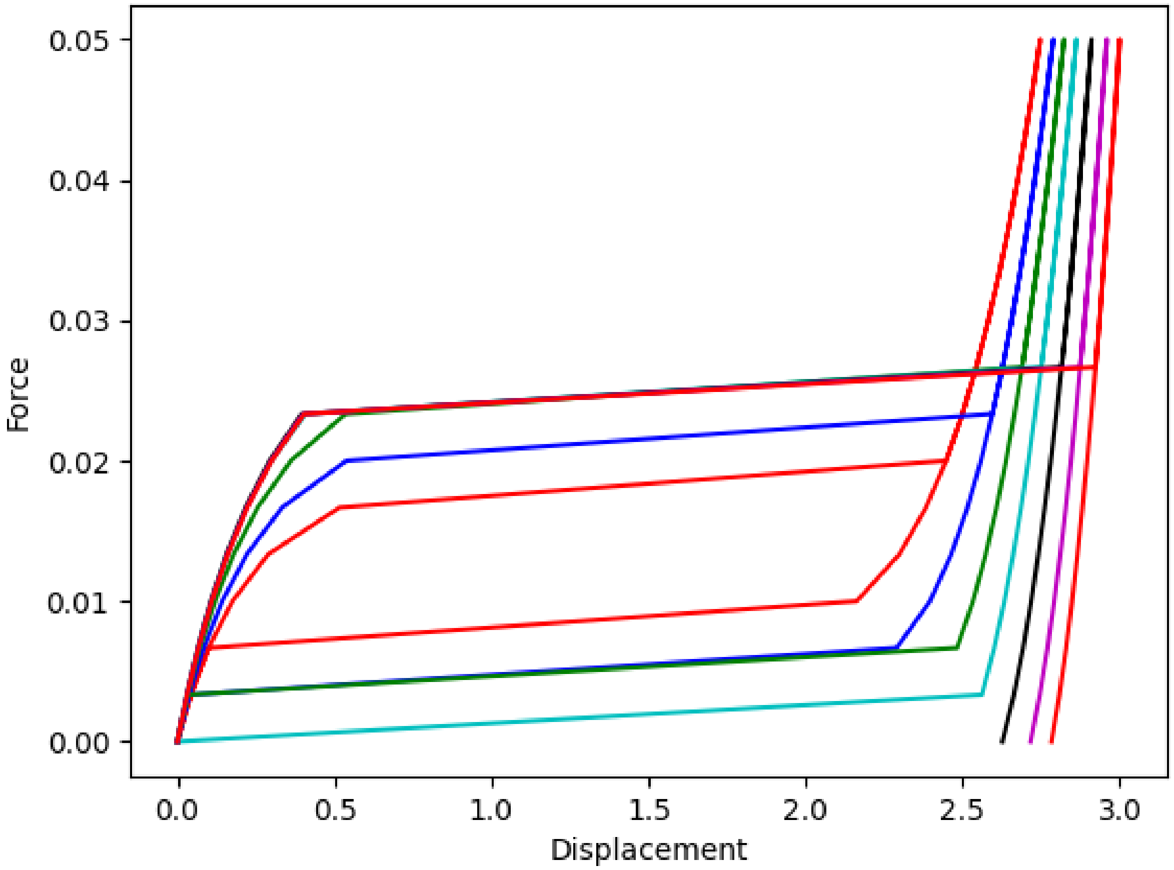
Can we automatically design a structure to maximize the hysteresis?



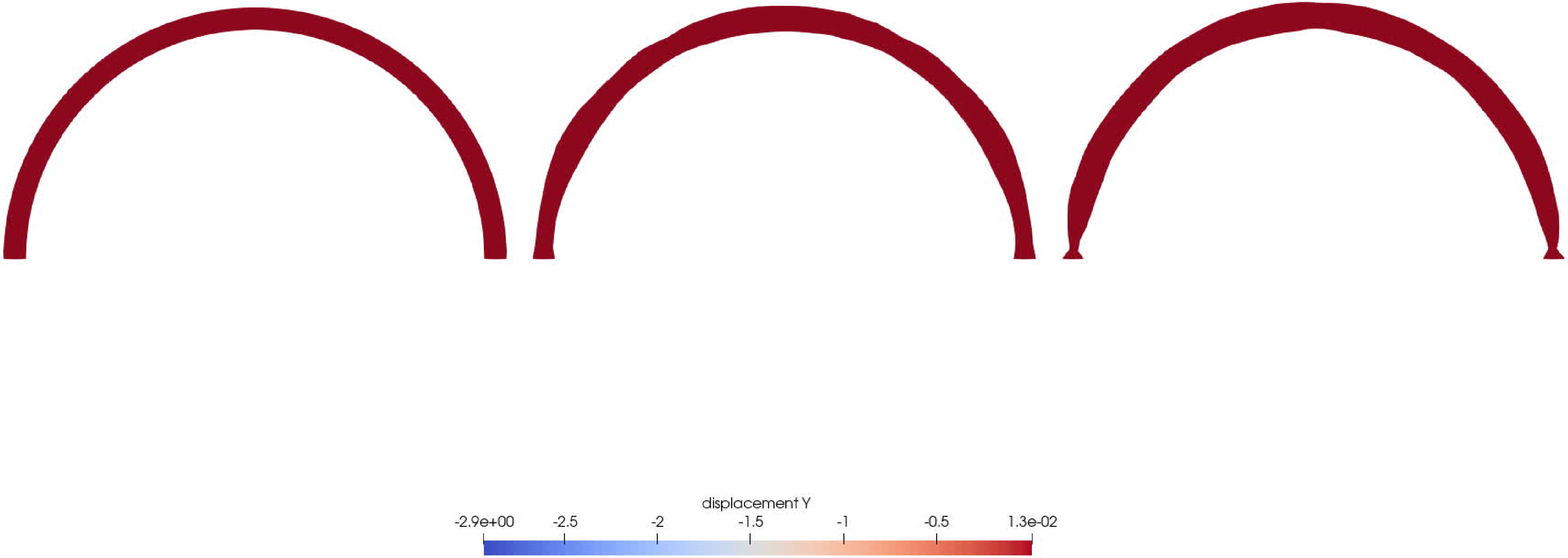


Can we maximize the hysteresis?

Maximize the integrated external work $\int_t \mathbf{f} \cdot \mathbf{v} \, dt$ over a cycle for a force-controlled loading



Can we maximize the hysteresis?



Conclusions: why use optimization-inspired solvers?

- **Robust**
  - If a stable solution exists, it will almost certainly be found
- **Faster turn-around times**
  - Super-linear solver convergence rates
  - Exploit negative curvature information
  - Infrequent matrix assembly
- **Credibility**
  - Avoid unstable equilibrium
  - Solve to significantly tighter tolerances

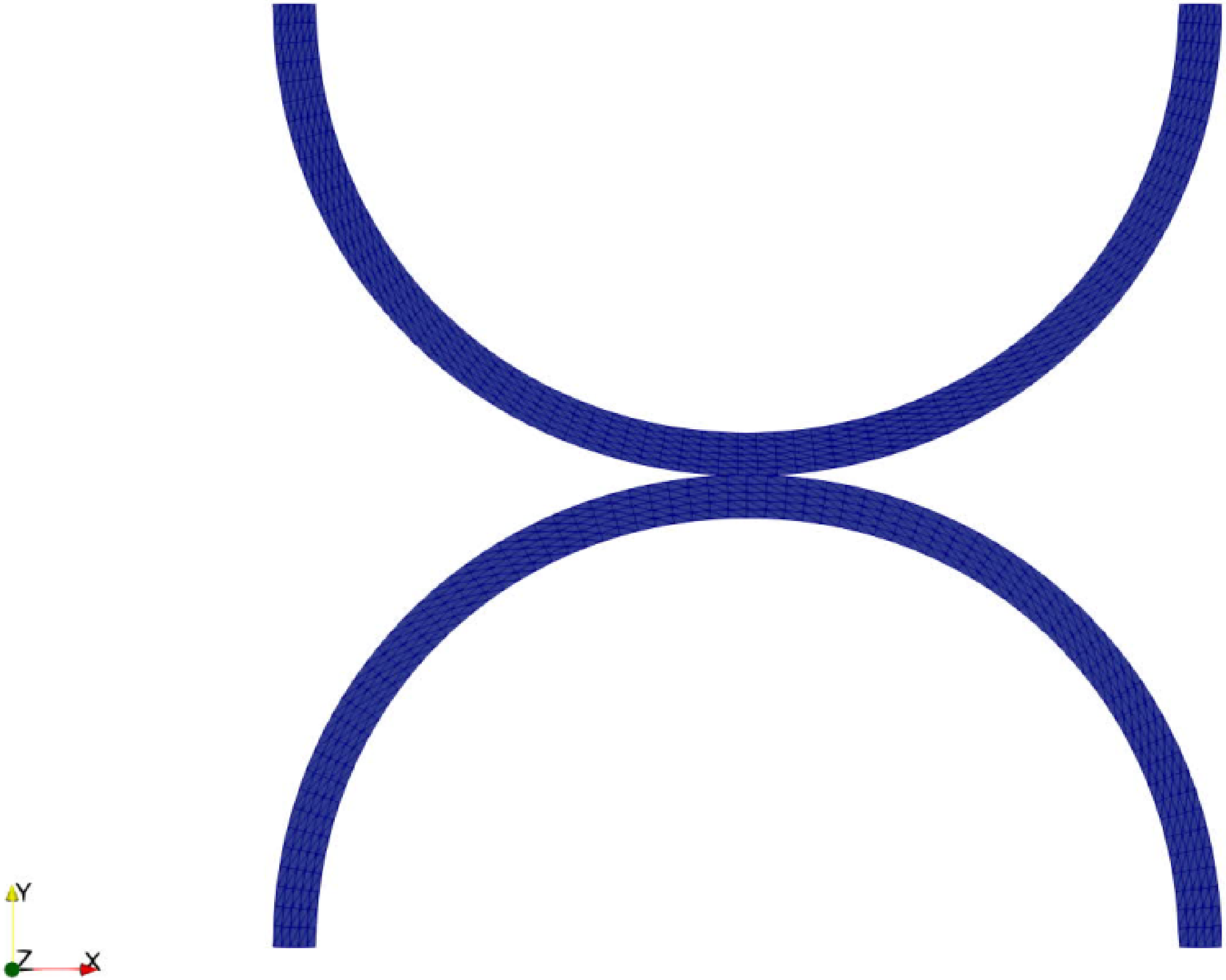
Special thanks to google's JAX library for incredibly powerful and efficient automatic differentiation

No time for

1. Discussion on setting solver tolerances and pre-scaling the degrees of freedom
2. Discussion on when to update preconditioner
3. Details of the trust region solver
4. Warm-start (a.k.a. linearized load-step predictor) which is essential for robust and large load steps
5. Saddle point preconditioner
6. Jax optimizations, tips and tricks
7. Variational plasticity and phase-field
8. Lesson’s learned doing design problems in jax
9. Equations for smooth signed distance field
10. Topology optimization example

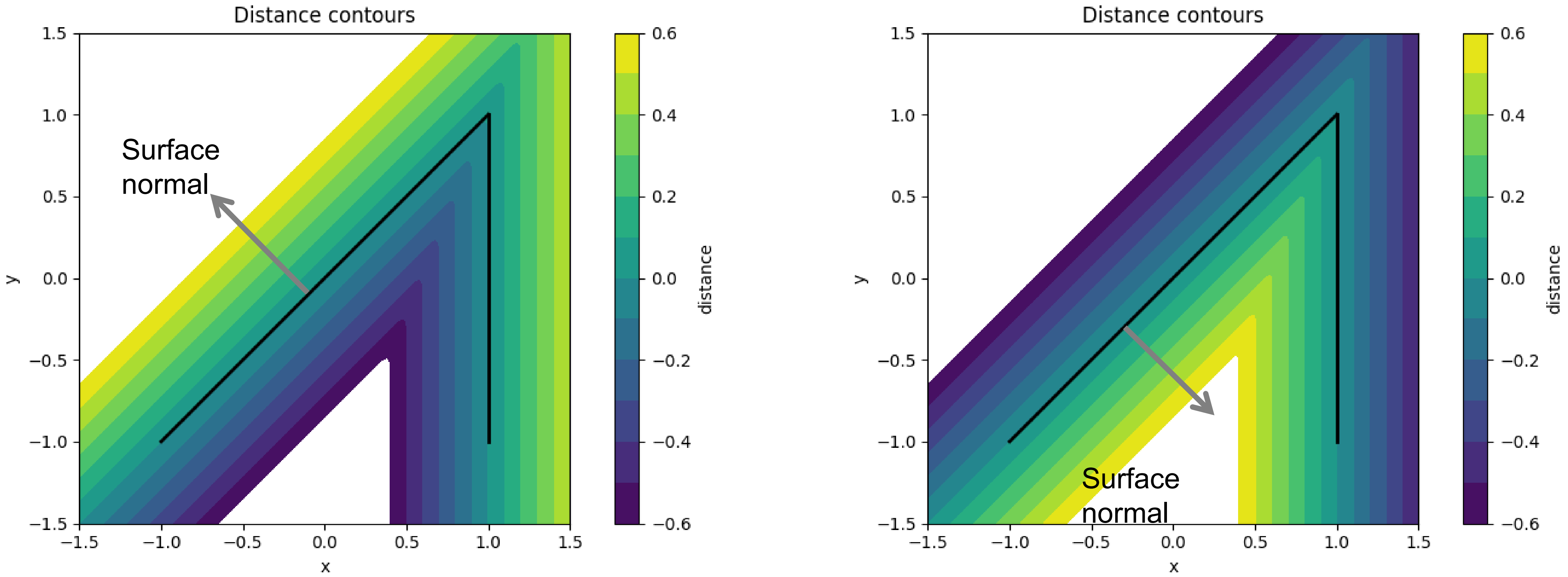
Project regrets / opportunities for improvement

1. Theoretical results for non-convex optimization solvers show disappointing provable bounds (very sub-linear solver convergence). Are there features of real materials which improve these bounds?
2. Unclear if the solver will be robust with asymmetric hypoelastic material models and following loads
3. Utilizing inexact solves. The CG trust-region solver is currently inexact, but the AI sub-problem could also be inexact to improve efficiency
4. We lack theoretical proofs that our second order update strategy, staggered with the nonlinear minimization solver is guaranteed to converge
5. Jax can exploit GPUs very well, but we have not
6. Continue effort on inverse problem
7. Mortar methods for more accurate and smooth contact enforcement
8. Still planning to open source our finite element code


Y
Z
X

Isn't the contact constraint non-smooth?

Yes, we use a smoothed signed distance function (levelset). This is a stronger smoothness than closest point projection to a smoothed surface.



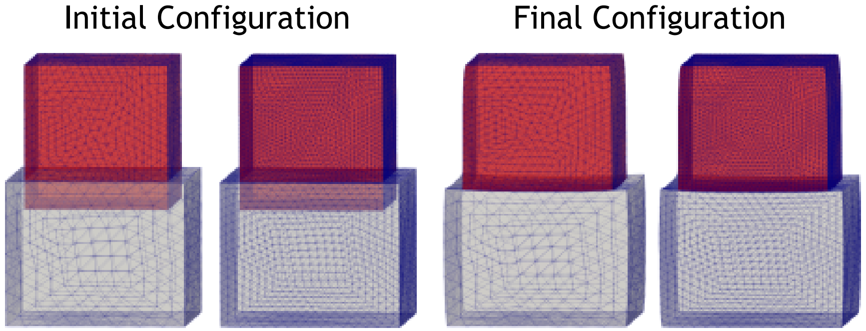
This part proved much harder than expected, but I'll skip the details!

Extra slides

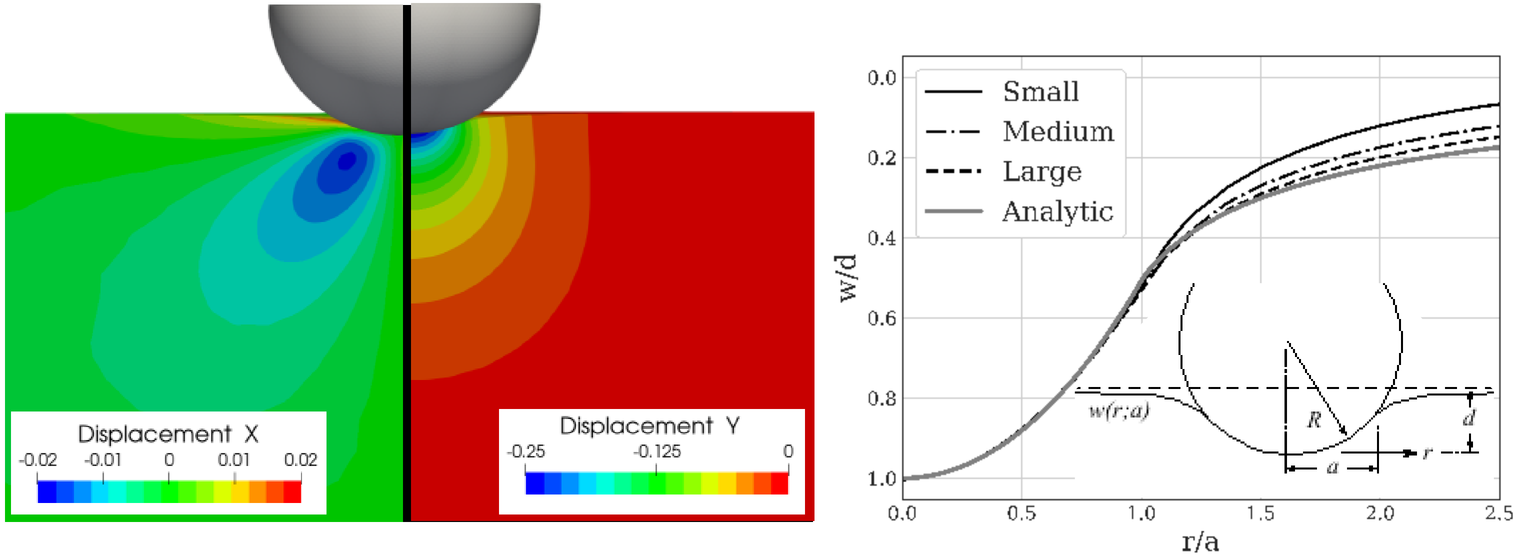
Contact Verification and Application to Deep Overlap Removal

- Smoothed level-set contact algorithm
- Demonstrated convergent under mesh refinement (1st order)
- Verified against analytic solution of Hertz contact problem
- Demonstrated robust contact enforcement, including large initial/evolving overlap
- Exercised on more complex/realistic geometry/assembly

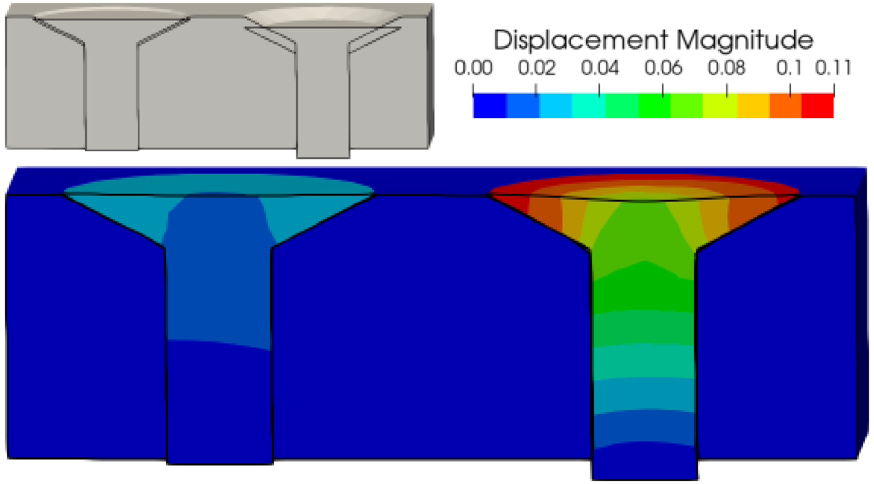
Deep Overlap Removal Enables Refinement of Imperfect CAD



Hertz Rigid Sphere Contacting Elastic Half-Space



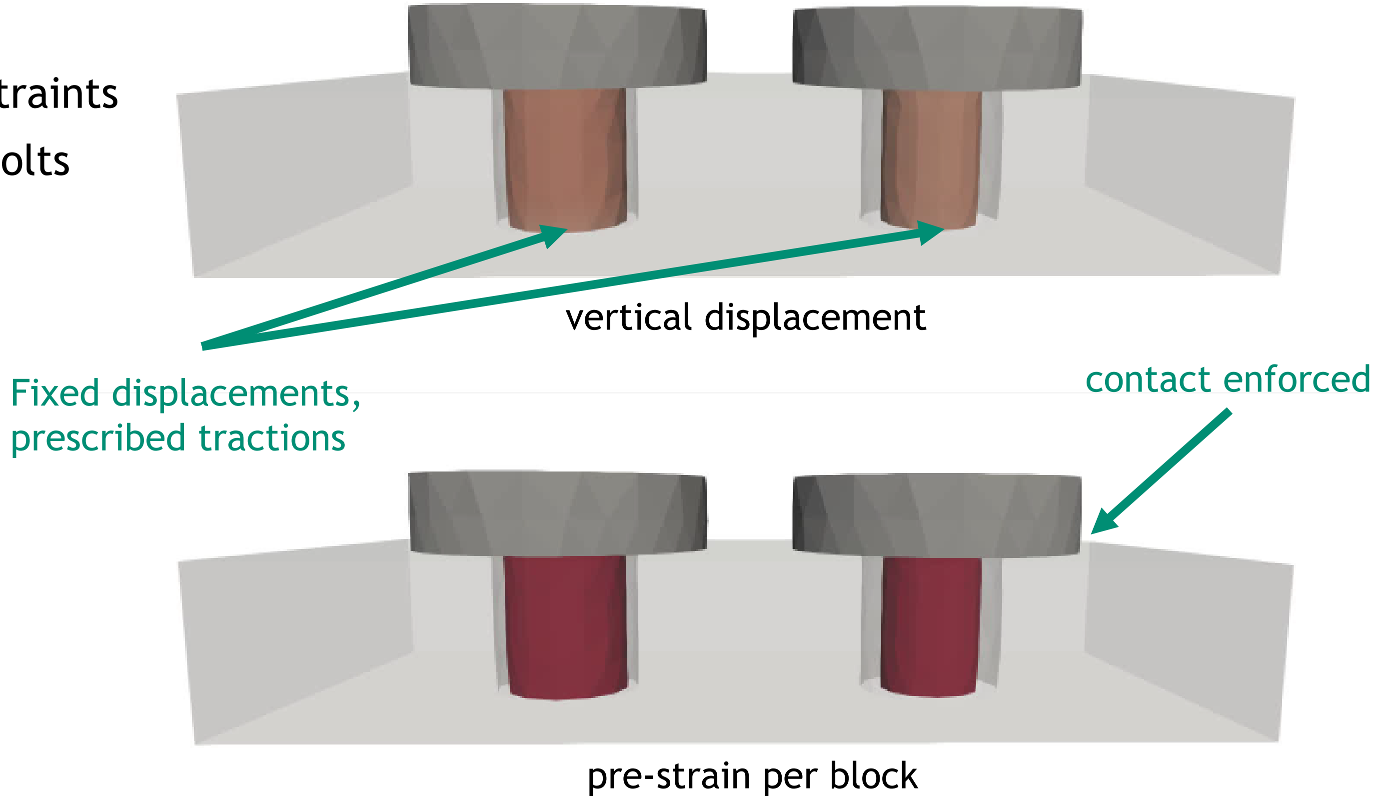
Pre-load with More Realistic Assembly



Contact inverse problem: bolt pre-loads

- Embedded sensitivities derived and implemented
- Inverse problem has multiple highly nonlinear constraints
- First exemplar: determine pre-strains in multiple bolts
- Can solve for pre-strain *per element* or *per block*

$$\min_{u,z,\lambda} \int_\Gamma \frac{1}{2}\|\bar{t} - t(u)\|^2$$
$$\text{s.t. } r(u,z) + c_{,u}(u)^T\lambda = 0$$
$$c(u) = 0$$



$$\begin{bmatrix} D & 0 & A^T & C^T \\ 0 & 0 & C & 0 \\ A & C^T & -\eta & 0 \\ C & 0 & 0 & -\epsilon \end{bmatrix} \begin{pmatrix} u \\ \lambda \\ \mu \\ l \end{pmatrix} = \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix}$$

Augmented linearized system

$$\min_{u,z,\lambda} \; d(u) + \left[r^T + \lambda^T c_{,u}\right]\mu^n + c^T l^n + \frac{1}{2\epsilon}\|c\|^2 + \frac{1}{2\eta}\|r + c_{,u}^T\lambda\|^2$$

Robustness and efficiency limited by

- Smoothness of contact constraint
- Optimization solver and parameters
- System scaling and pre-conditioning

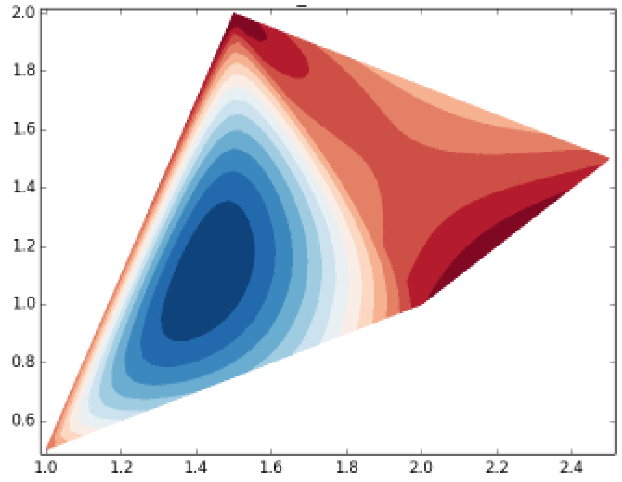
Novel contact algorithms

- Low degree-of-freedom C1 continuous “Gregory” tetrahedral
- DG contact: element stresses ‘feel’ contact
  - DG derivatives include contact surface jump

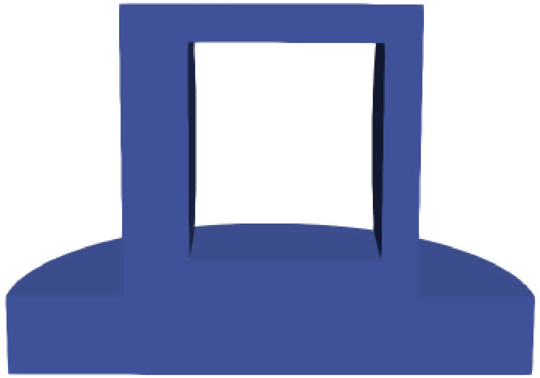
$$\nabla_{\mathrm{DG}} u_i = \nabla u_i - \int_{\Gamma^c} [\![u_i - \hat{u}_i]\!] \otimes \mathbf{N} \cdot \{\!\{ \mathbf{z} \}\!\} \, dS$$

- Smooth and search-less MPM-based contact algorithm
  - Uses background cubic b-spline
  - Enriched velocity field: angular momentum conservation*

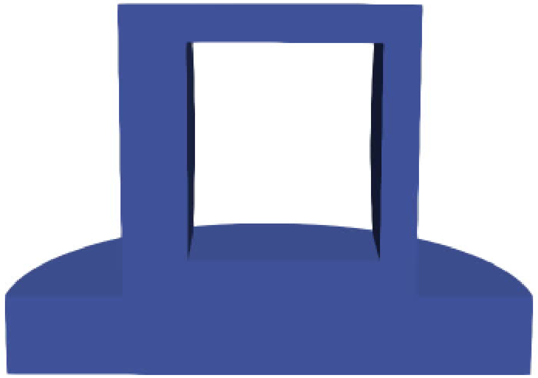
Continuous shape-function gradient



MPM contact on GPU: **400X** faster!



Serial legacy-contact



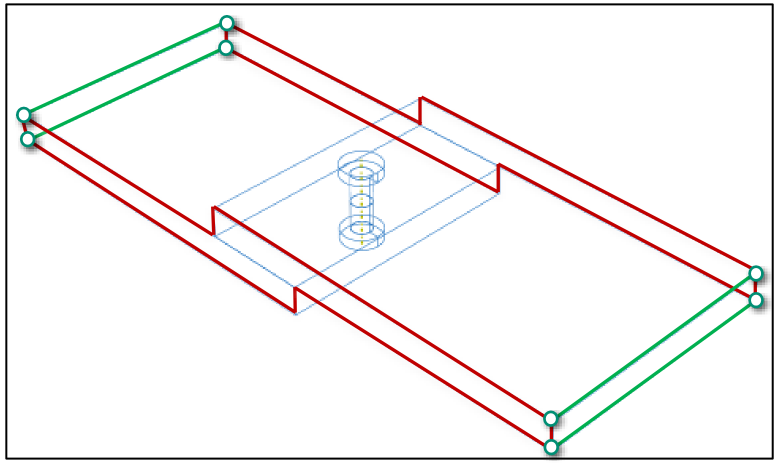
GPU MPM-contact **

*Leveraging ideas from Joey Teran, UCLA

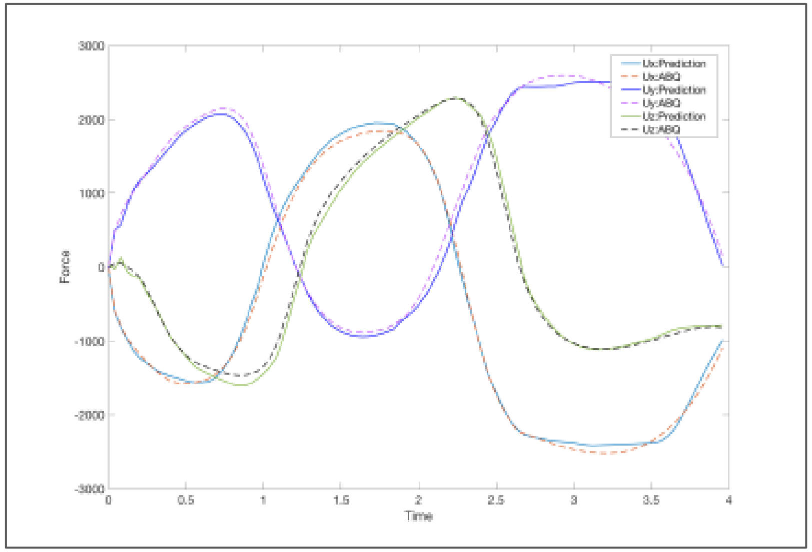
**Material calibration geometry from John Mersch

Academic Alliance: machine learned surrogates with contact

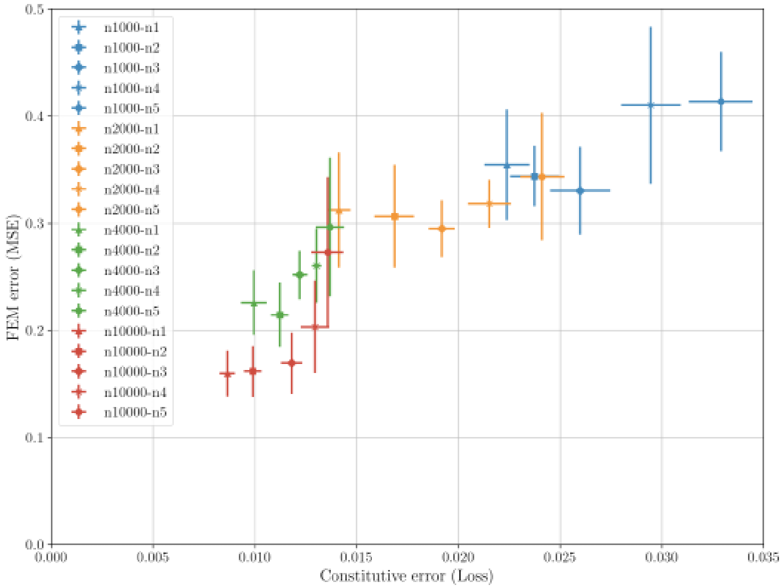
- Professor Julian Rimoli, Georgia Tech
- Train Long short-term memory Neural Net on lap joint
- Create effective "super-element"
- Insertion back into finite element models causes instabilities
- Requires consideration of proper physical constraints



Contact training problem



predicted forces



Model training error correlate with simulation error

Runtime: ~1 hour → ~10 seconds