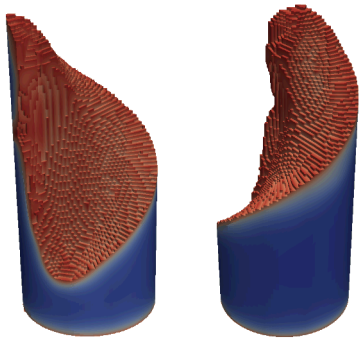
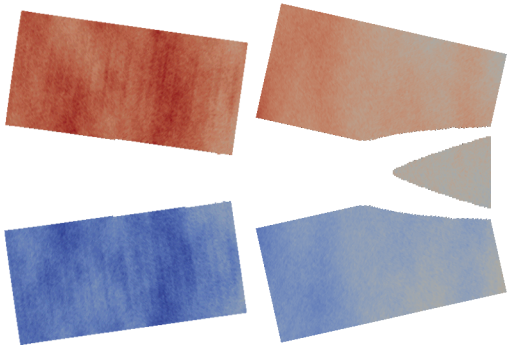
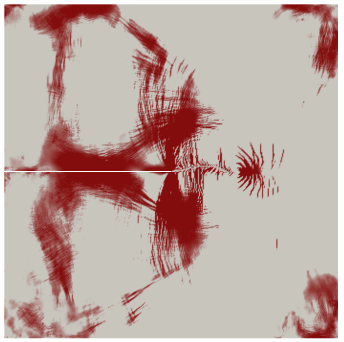
SAND2016-7022C

Exceptional service in the national interest

Sandia National Laboratories







Inverse methods for fracture based on a parabolic regularization of brittle cohesive laws

Michael Tupek

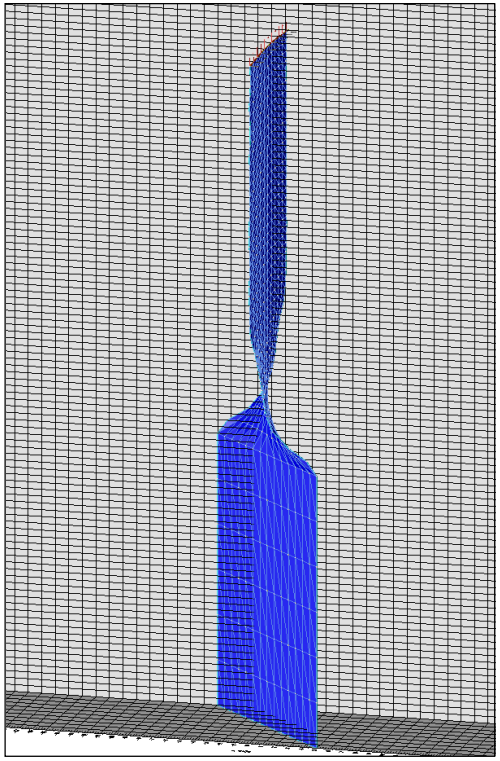
U.S. DEPARTMENT OF ENERGY

National Nuclear Security Administration

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. SAND NO. 2011-XXXXP

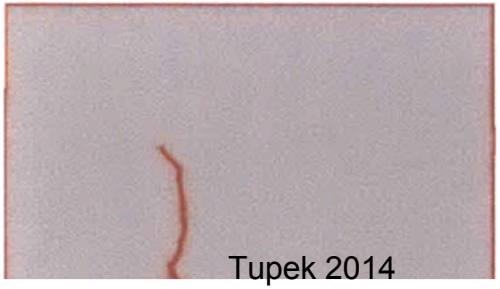
Fracture Renaissance

- Several promising fracture methods have emerged in recent years



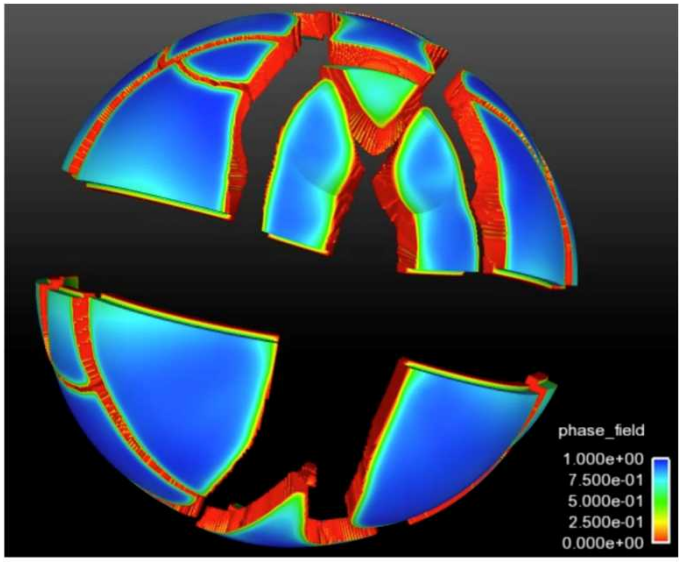
Duarte 2011

GFEM / XFEM



Tupek 2014

Peridynamics



Dolbow, Stevens 2015

Phase field

Inverse methods for fracture

Sandia National Laboratories

- Most fracture method focus on predictions
- This remains a significant challenge!
- However, predictions by themselves aren't that interesting

"Computers are useless, they can only give you answers."

-Pablo Picasso

- Predictive simulations should guide decision making

inverse = predictions → decisions

Sandia National Laboratories

Forward Problem:

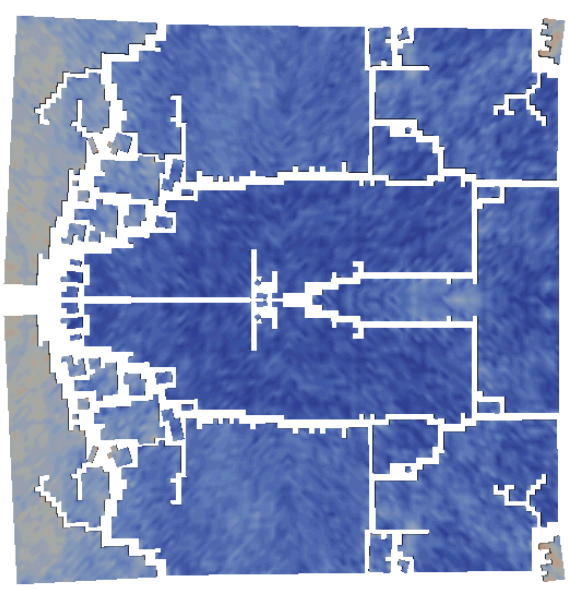
Phase Field Fracture

Classical damage models: ill-posed

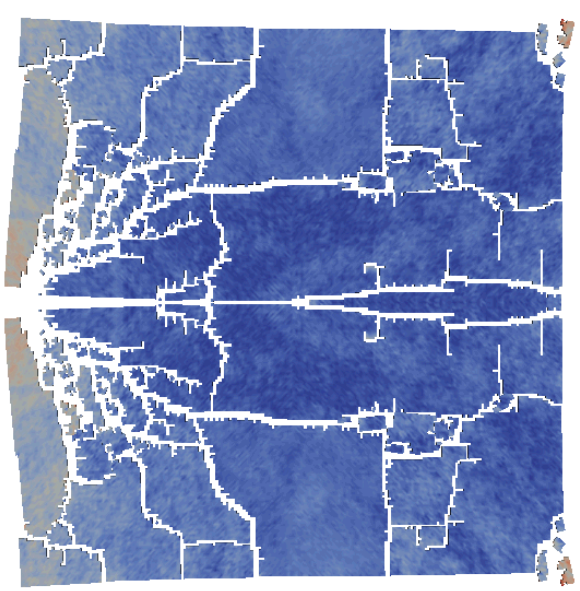
Sandia National Laboratories

- Fracture modeling is critical to predicting the performance and reliability of many Sandia components and systems
- Many fracture models used at Sandia (and elsewhere) are ill-posed and non-convergent

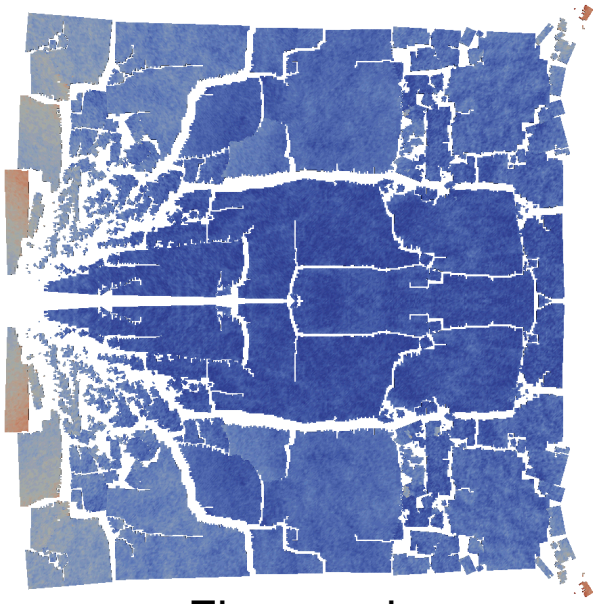
Max principal stress criterion with element death shows significant mesh dependence



Coarse mesh

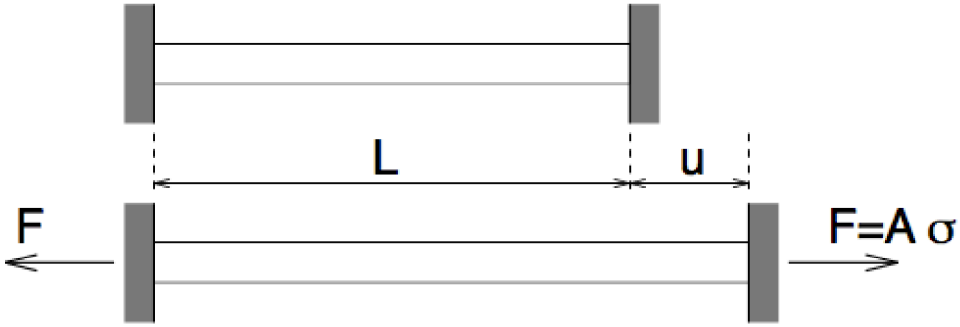


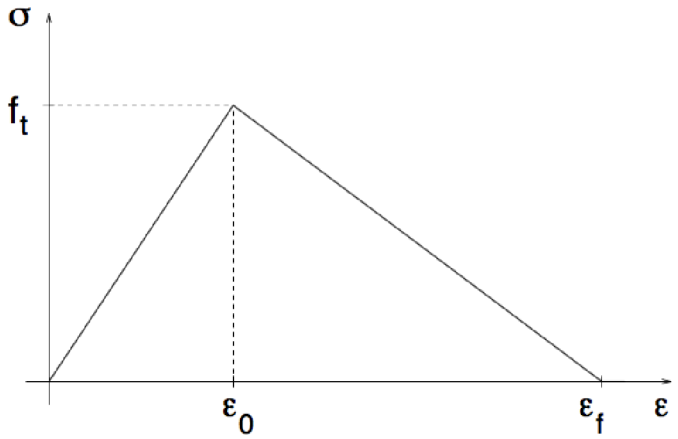
Medium mesh

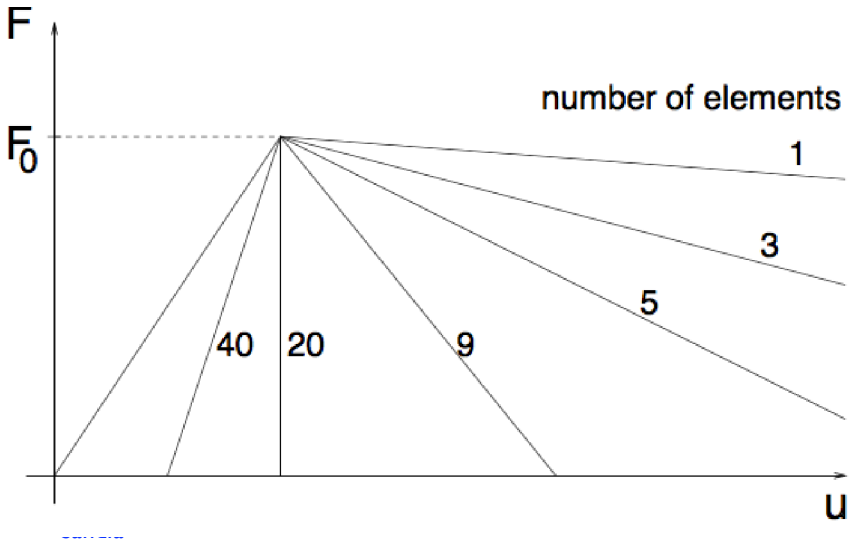


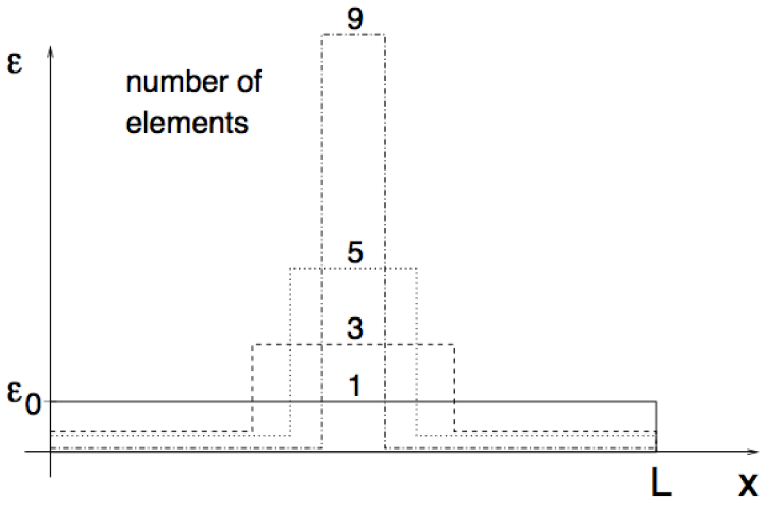
Fine mesh

Classical damage models: non-convergent









Phase field fracture

Standard potential energy:

$$_{pot}(\mathrm{u}, \Gamma) = \int_{\Omega} {}_{e}(r\ {}^{s}\mathrm{u})\, dv + \int_{\Gamma} G_c\, da$$

Approximate surface integral by a volume integral

Elliptically regularized potential energy:

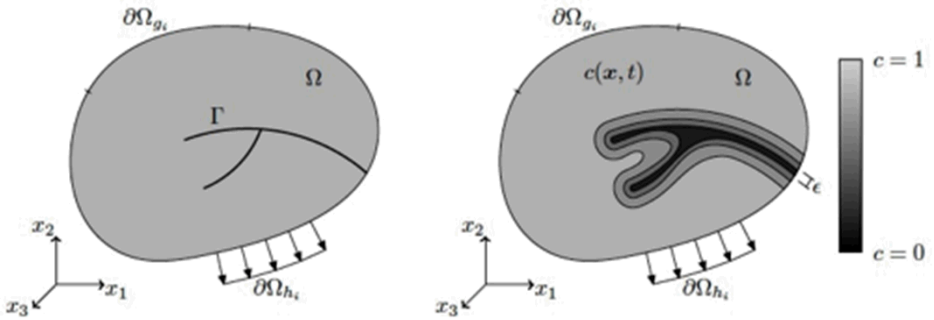
$$_{l}(\mathrm{u}, d) = \int_{\Omega} \left(g(d)\ {}^{+}_{e}(r\ {}^{s}\mathrm{u}) + {}^{-}_{e}(r\ {}^{s}\mathrm{u})\right) dv + \int_{\Omega} G_c \gamma_l\, da$$

Common crack density:

$$\gamma_l = \frac{1}{2l} d^2 + \frac{l}{2} |\nabla d|^2$$

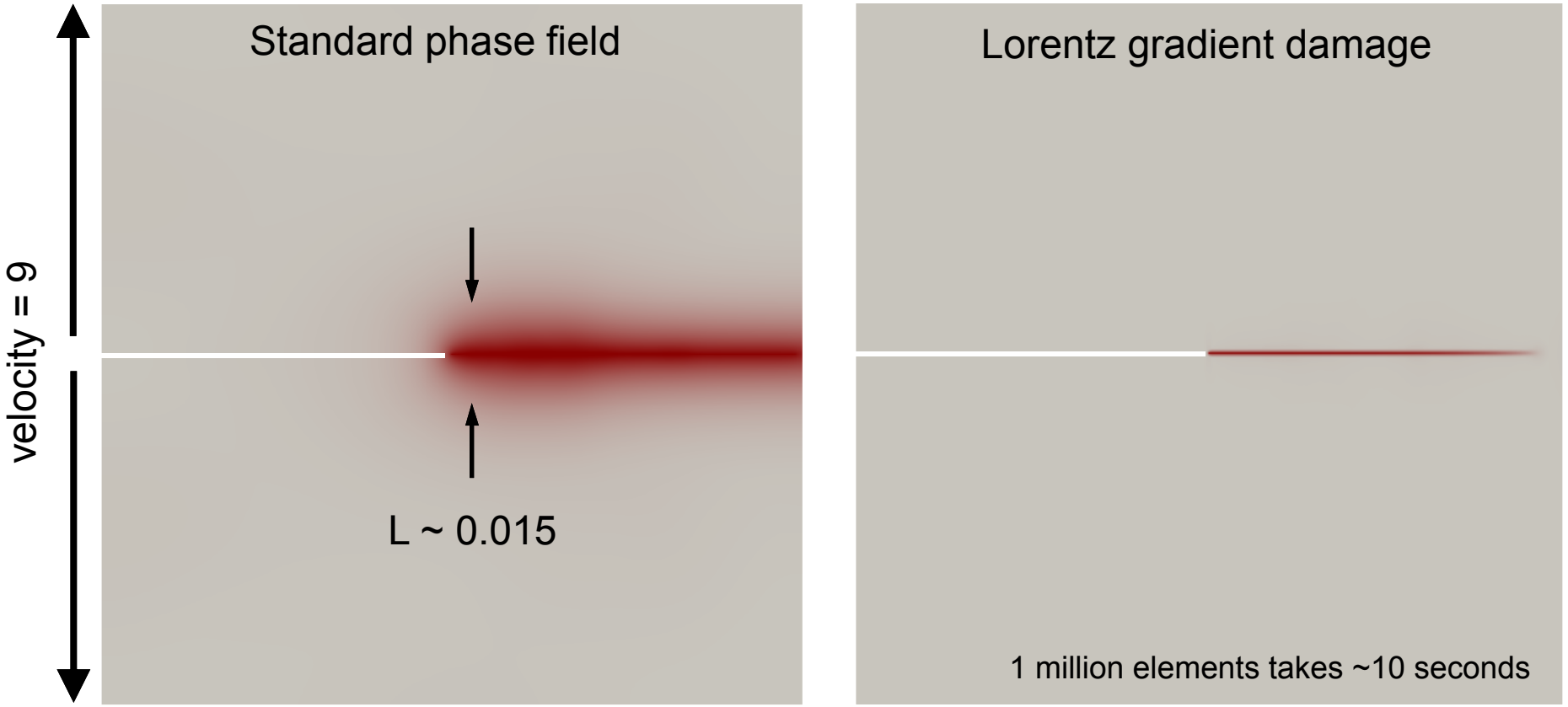
Stress degradation:

$$g = (1 - d)^2$$



Limitations to standard phase field

- No threshold, some damage for any $\varepsilon \neq 0$
- Regularization length scale tied directly to fracture properties $\sigma_{\max} = \frac{9}{16}\sqrt{\frac{EG_c}{6L}}$



$$\sigma_{\max} = 250\text{e}6 \quad G_c = 100e3 \quad E = 200\text{e}9$$

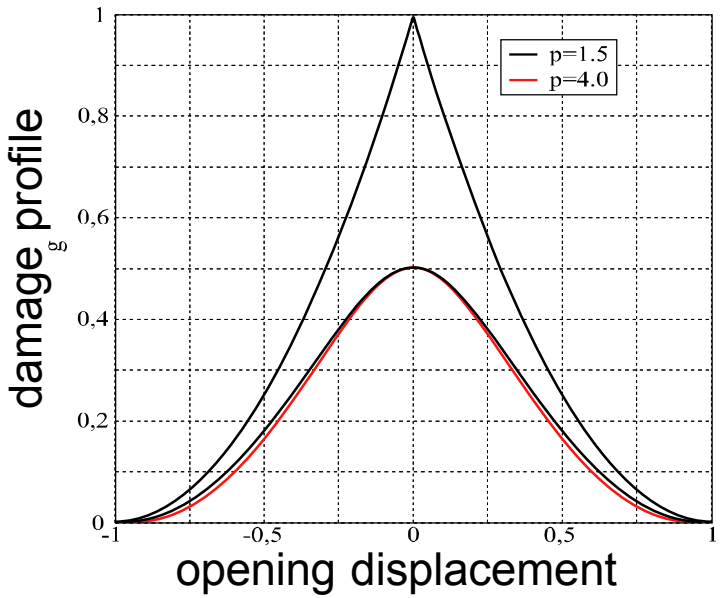
Cohesive gradient damage model

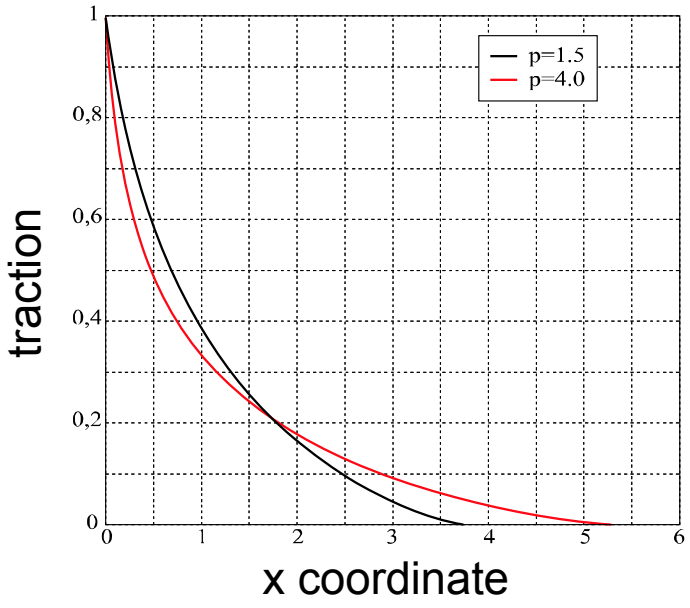
- Regularized gradient damage model by Lorentz, et al. 2011
- Shown to converge to cohesive zone model as $L \to 0$

$$\rho\ddot{\mathbf{u}} = \nabla \cdot \boldsymbol{\sigma}, \qquad \boldsymbol{\sigma} = g(\phi)\frac{\partial \psi_e^+}{\partial \boldsymbol{\varepsilon}} + \frac{\partial \psi_e^-}{\partial \boldsymbol{\varepsilon}}$$ Momentum balance

$$g'(\phi)\psi_e^+ + k = c\nabla^2\phi$$ Phase field equation

$$g(\phi) = \frac{(1-\phi)^2}{1 + (m-2)\phi + (1+pm)\phi^2}$$ Derivable from a phase potential





Parameters of damage model

Depends on cohesive fracture parameters:

$$k = \frac{3}{4}\frac{G_c}{L} \quad c = \frac{3}{8}LG_c \quad m = \frac{3}{2}\frac{EG_c}{\sigma_c^2 L}$$

Note: the fracture energy, G_c, is ignored in most damage models

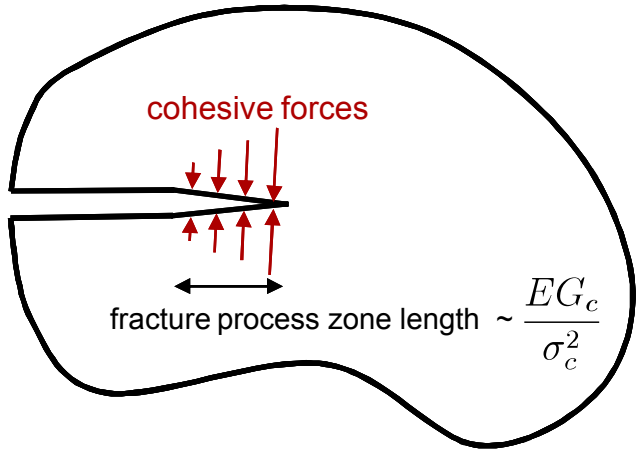
Physically justified constraints on the parameters ensures

- Regularization length scale resolves fracture process zone

$$L < \frac{3}{2(p+2)}\frac{EG_c}{\sigma_c^2}$$

- Monotonic stress decay

$$p \geq 1$$



Parabolic regularization

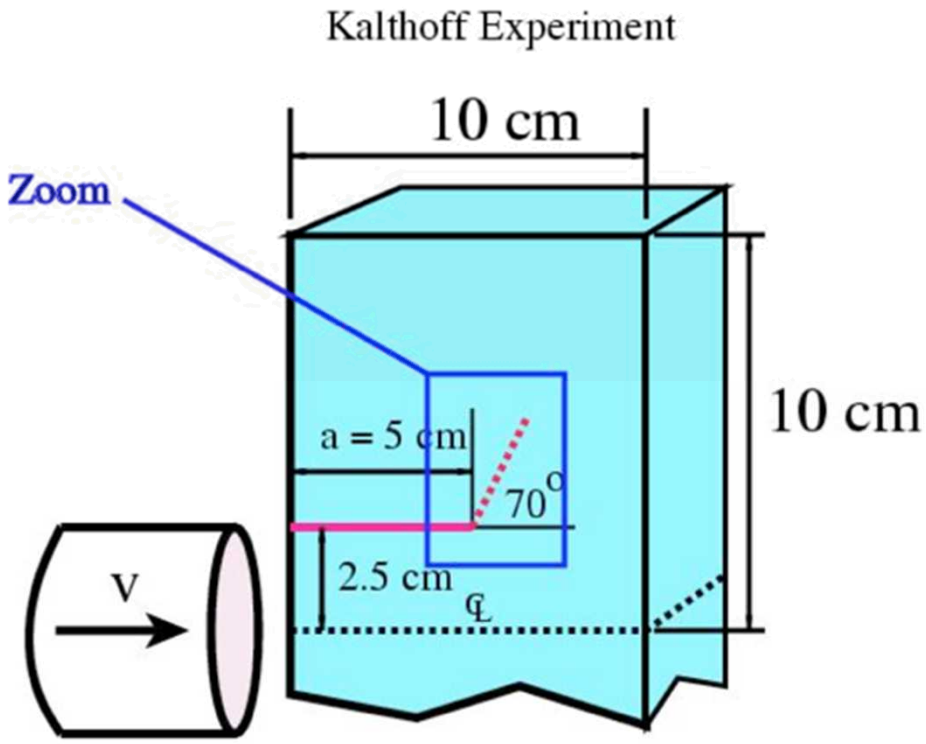
Sandia National Laboratories

- For explicitly integrated dynamic problems, solving a nonlinear phase field equation is VERY expensive
- Add a viscous regularization (e.g., Miehe 2010)

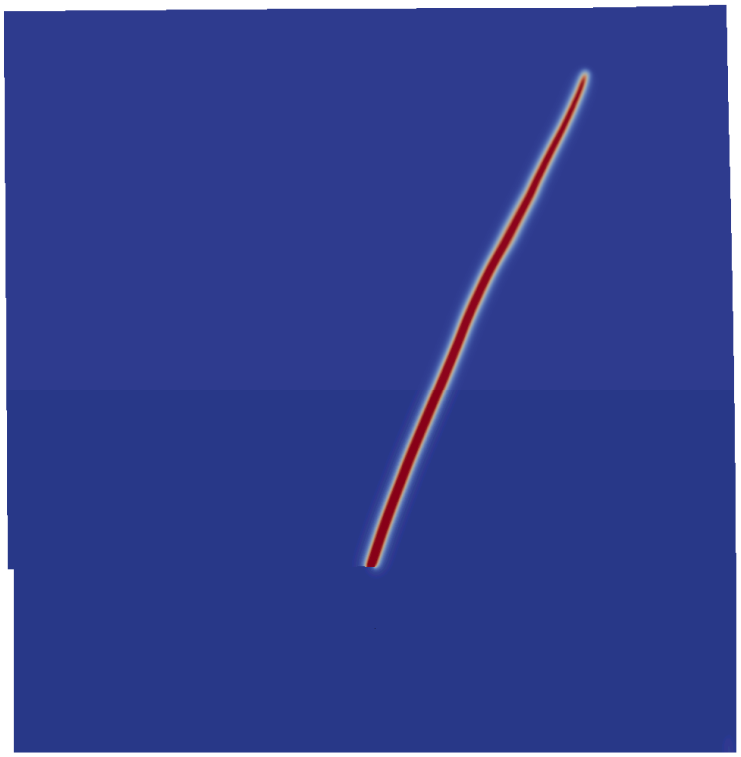
$$\eta\dot{\phi} = -g'(\phi)\psi_e^+ - k + c\nabla^2\phi, \quad \dot{\phi} \geq 0$$

- Integrate parabolic equation explicitly!
- Hyperbolic timestep constraint: $\Delta t \leq \frac{1}{s}\Delta x$
- Parabolic timestep constraint: $\Delta t \leq \frac{\eta}{c}(\Delta x)^2$
- Can use hyperbolic timestep if: $\eta \geq \frac{c}{s\Delta x} = \frac{3}{8}\frac{G_c}{s}\frac{L}{\Delta x}$

Kalthoff validation



Damage contour

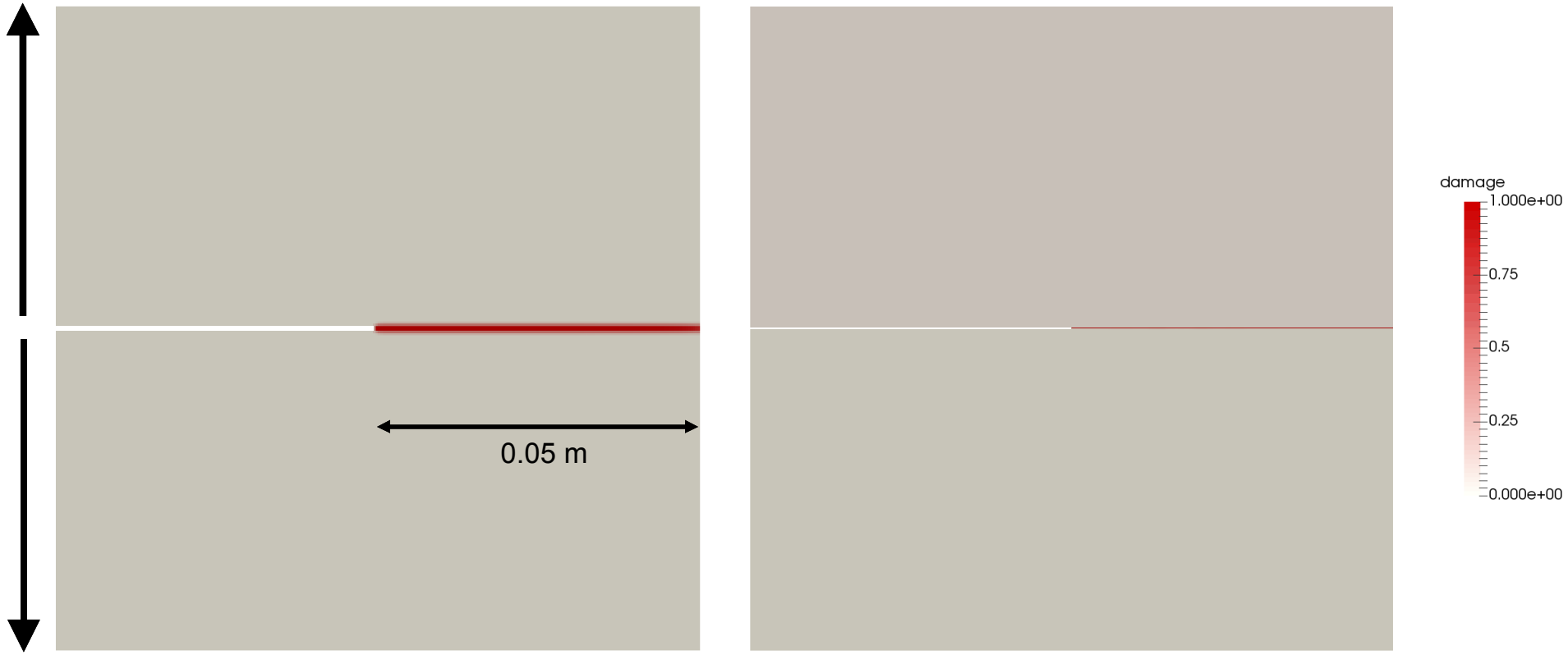


Achieves the expected crack propagation angle: ~70°

Mode-I: Length scale refinement

Length-scale decreases with Δx

v = 25 m/s



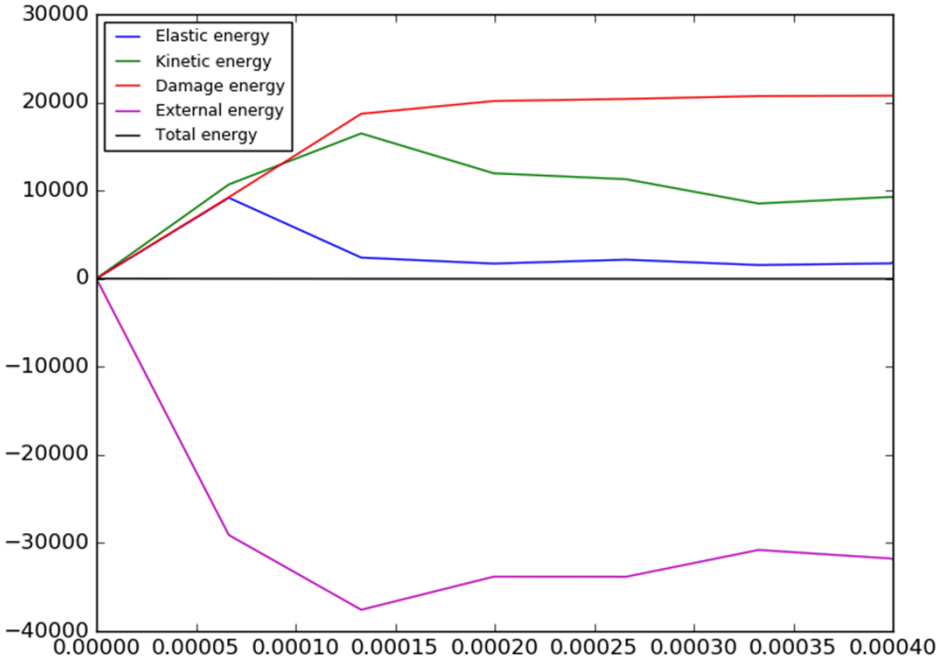
71.5 thousand elements

14.4 million elements

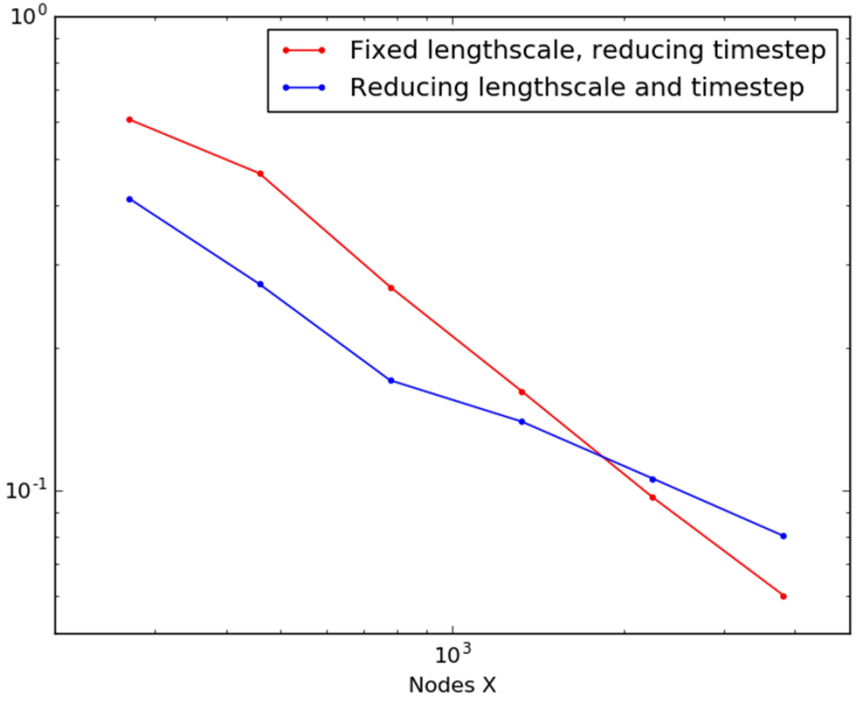
G_c = 4e5

Accuracy

Total numerical energy is preserved



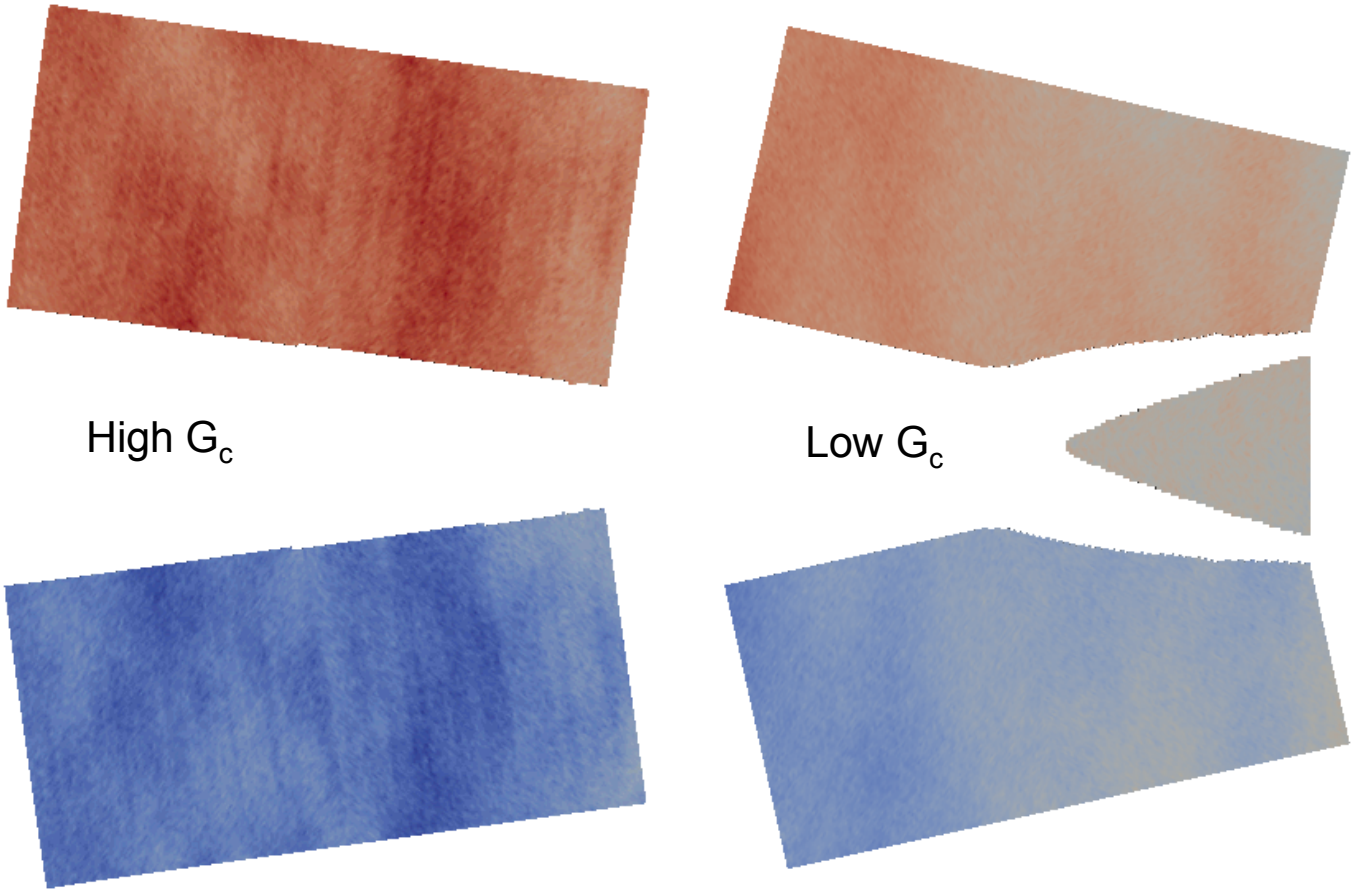
Dissipated fracture energy error



- Convergence rate of ~1 for fixed length scale
- Convergence rate of ~0.5 otherwise
- Dissipation due to viscous regularization is non-negligible

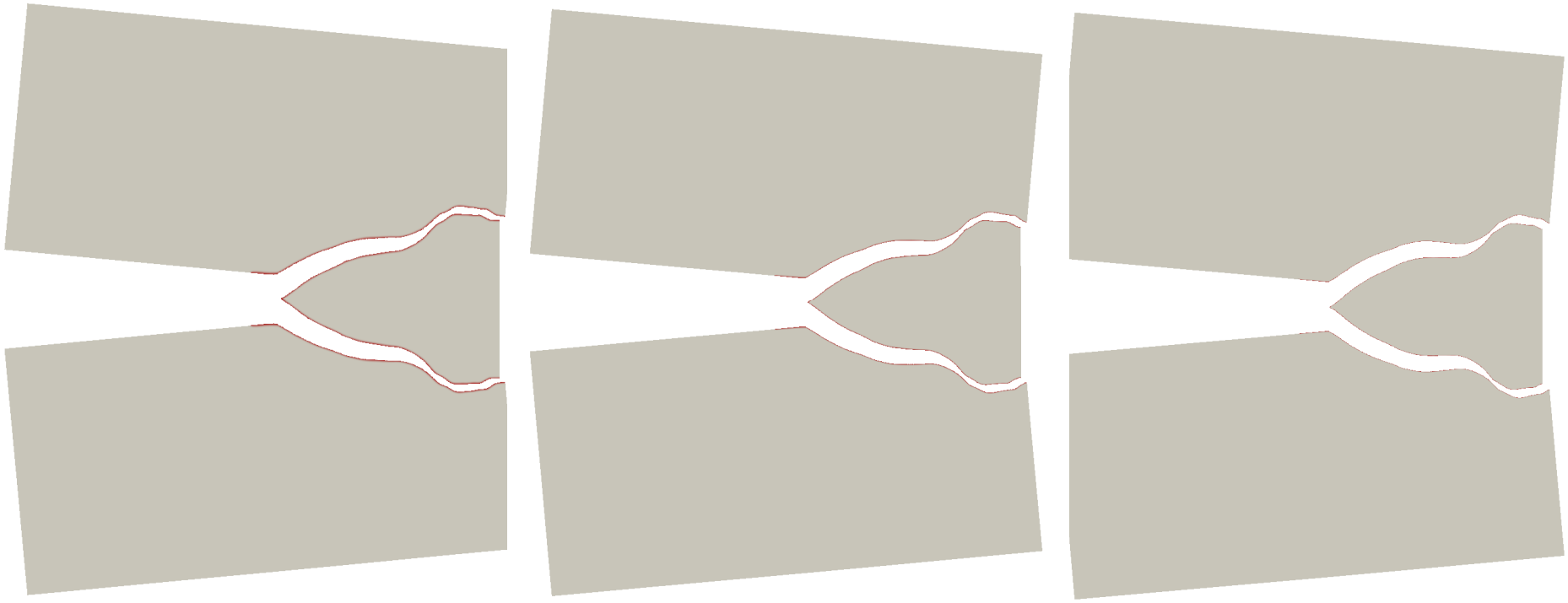
Importance of fracture energy

Mode-I crack transition to branching



Branching is emergent, not prescriptive

Branching: mesh insensitivity



1.4 million elements

4.1 million elements

11.8 million elements

Runs in ~5 hours on 1 GPU

Thermally loaded glass-metal interface

Experimentally observed crack path

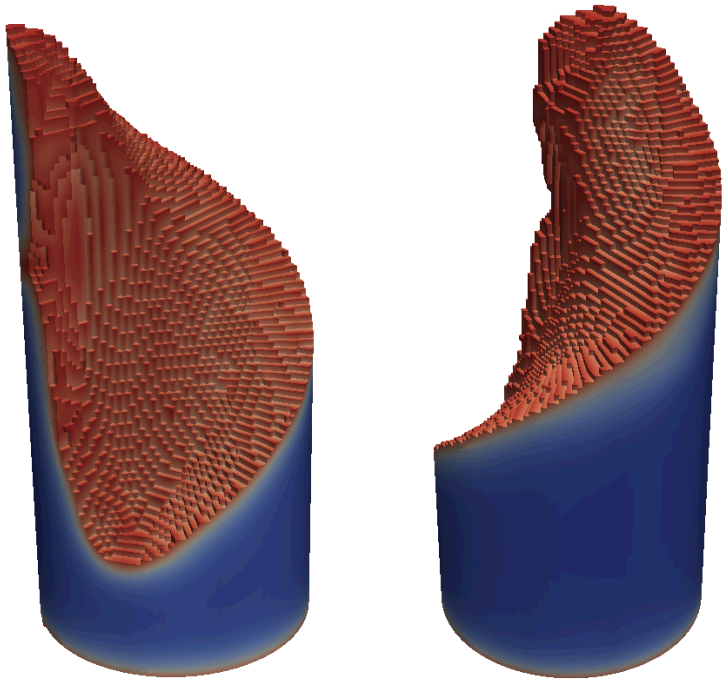
1.5 million element simulation runs in 6 hours on 1 GPU

- Loaded by temperature drop
- Metal on top contracts relative to glass below
- Initial vertical crack propagates upward
- Eventually turns to run parallel to the glass-metal interface

Demonstrations in 3D

Sandia National Laboratories

Ceramic impact

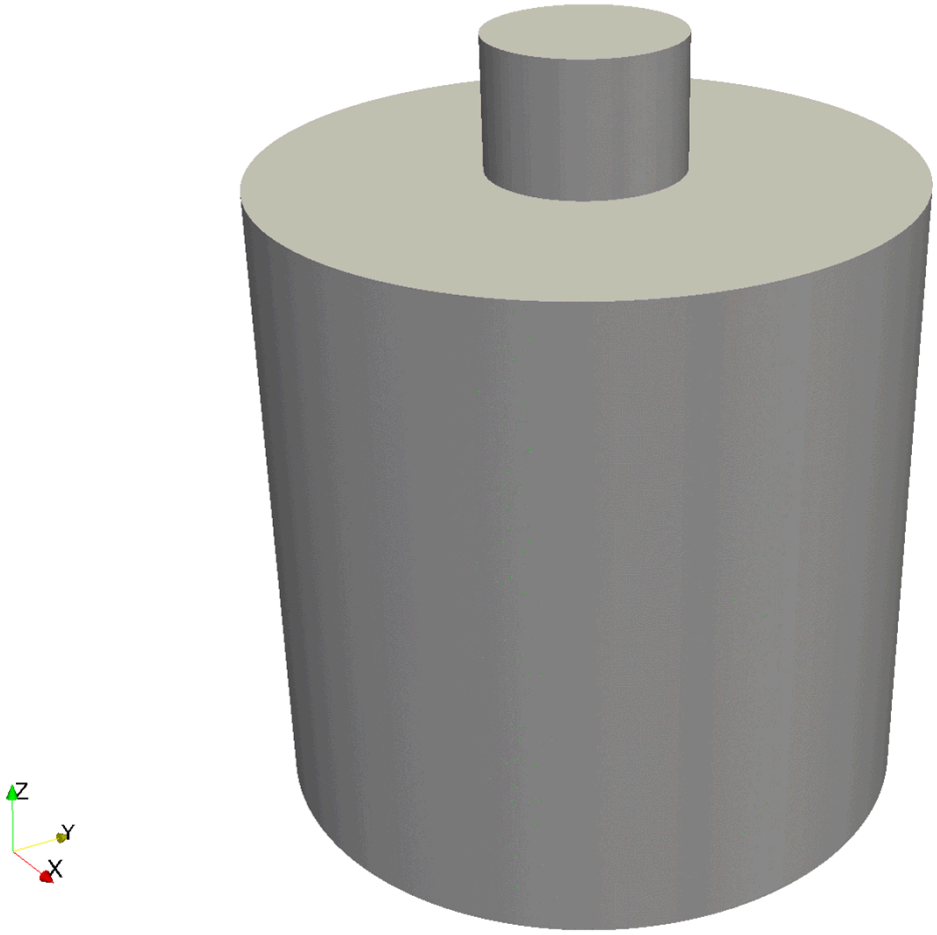


Brittle torsion fracture

- Captures complex 3D crack patterns
- No explicit geometry representation
- Naturally predicts initiation, branching and coalescence

Brittle impact demonstration

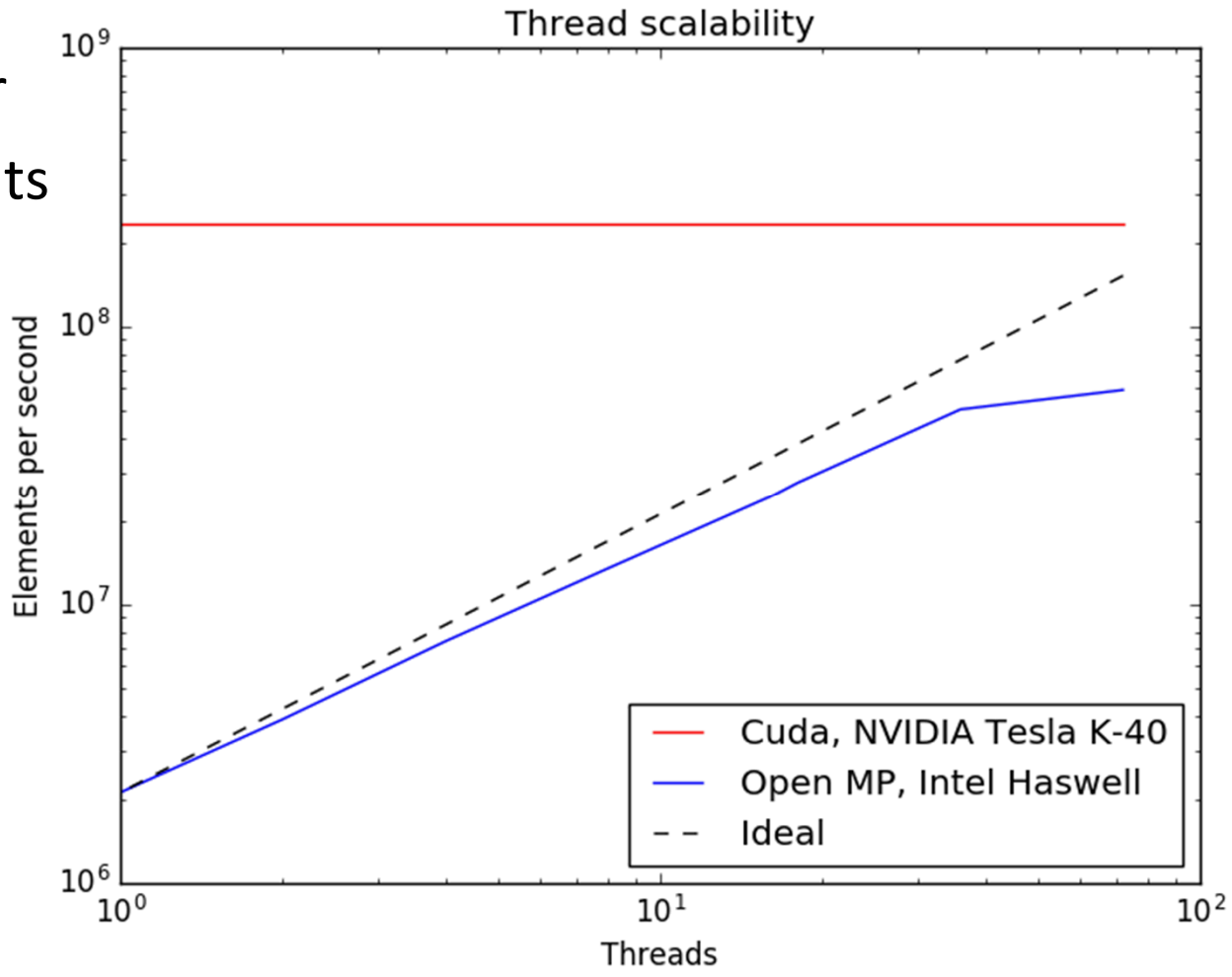
Sandia National Laboratories



Thread scalable, platform portable

Sandia National Laboratories

- Sandia's Kokkos library allows cross-platform, thread scalable implementations
- GPU: > 110 X faster
- 250 million elements per second
- OpenMP scales



Conclusions and future work

Sandia National Laboratories

- Efficient gradient damage models can be constructed which are cohesive
- Method is massively thread scalable (and MPI scalable)
- Explicit time integration is possible

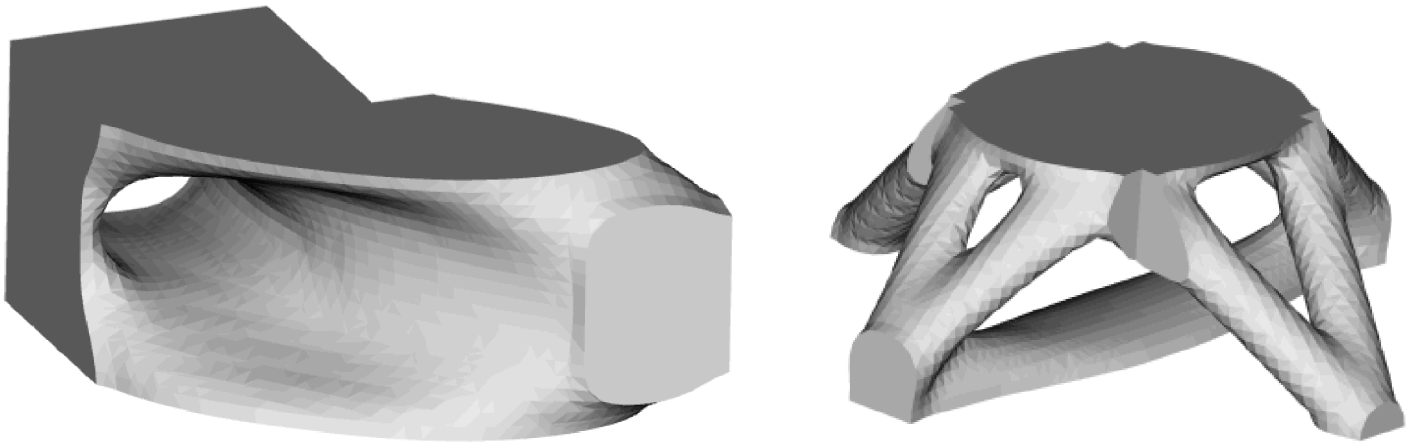
  (and more than 10 X faster than single linear phase solve)
- Ongoing work with John Dolbow to verify model in fragmentation scaling studies

Sandia National Laboratories

Inverse Problem

Inverse methods:

- Quantities of interest are differentiable functions
- Large # decision variables
- Often, # of decision variables is proportional to mesh resolution
- Require inexpensive gradient evaluations



Example: Topology optimization using

Sandia's Plato code

Inverse methods

Solve multiple “forward” prediction simulations in order to

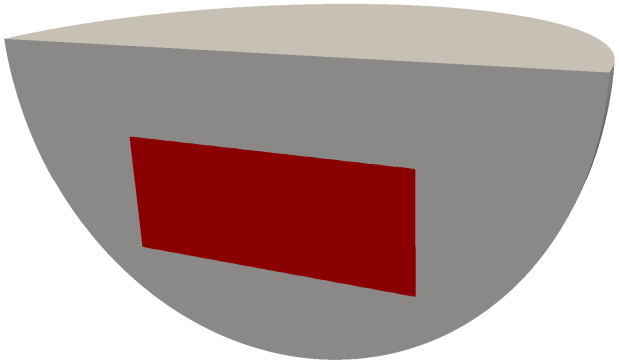
- Design
- Calibrate properties
- Match observations

$$\min_{\boldsymbol{\theta}} \; g(\mathbf{u}, \boldsymbol{\theta})$$

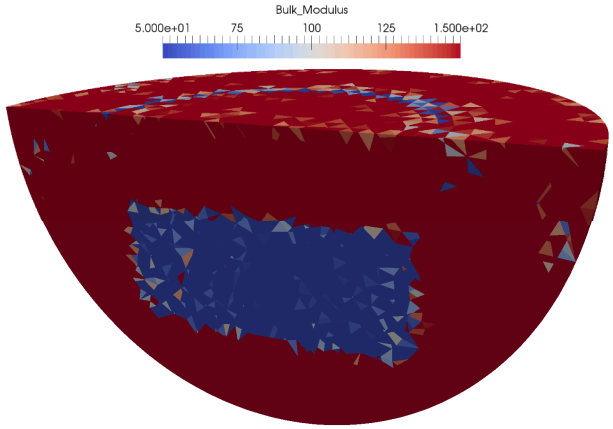
$$\text{s.t.} \;\; \mathcal{L}(\mathbf{u}, \boldsymbol{\theta}) = \mathbf{0}$$

Quantity of interest depends on
- physical solution $\mathbf{u}$
- design parameters $\boldsymbol{\theta}$

Constrained by the physics



Initially unknown hidden tunnel subjected to surface excitation



Reconstructed material properties due to observed surface displacements in Sierra-SD

Fracture Inverse Problems

Sandia National Laboratories

- **Non-destructive evaluation**: determine the damage at material points given observations of how waves propagate (unknowns ~ # elements)
- **Crack forensics**: determine what happened to a structure that caused it to damage in an observed way (unknowns ~ # number of BCs time # timesteps)
- **Heterogeneous material design**: how to design composite materials to maximize resistance to crack propagation (unknowns ~ # elements)

Fracture Inverse Problems

Sandia National Laboratories

- **Non-destructive evaluation**: determine the damage at material points given observations of how waves propagate (unknowns ~ # elements)
- **Crack forensics**: determine what happened to a structure that caused it to damage in an observed way (unknowns ~ # number of BCs time # timesteps)
- **Heterogeneous material design**: how to design composite materials to maximize resistance to crack propagation (unknowns ~ # elements)

Review of the adjoint method

Sandia National Laboratories

- Computing finite difference gradients requires many, many solves: O(#design parameter)
- Adjoint method requires just 1 additional solve!

Explicit update rule: $\mathbf{u}^n = \mathbf{h}^n(\mathbf{u}^{n-1}; \theta)$

Quantity of interest: $g(\mathbf{u}^N; \theta)$

Desired sensitivities: $\dfrac{dg(\mathbf{u}^N; \theta)}{d\theta}$

$$\mathcal{L}(\mathbf{u}, \boldsymbol{\lambda}; \theta) = g\left(\mathbf{u}^N; \theta\right) + \sum_n \langle \boldsymbol{\lambda}^n, \mathbf{h}^n(\mathbf{u}^{n-1}; \theta) - \mathbf{u}^n \rangle$$

$$\frac{d\mathcal{L}}{d\theta} = g_{,\boldsymbol{\theta}} + \langle g_{,\mathbf{u}}, \mathbf{u}^N_{,\theta} \rangle + \sum_n \langle \boldsymbol{\lambda}^n, \mathbf{h}^n_{,\theta} + \mathbf{h}^n_{,\mathbf{u}} \mathbf{u}^{n-1}_{,\theta} - \mathbf{u}^n_{,\theta} \rangle$$

Review of the adjoint method

Sandia National Laboratories

- Computing finite difference gradients requires many, many solves: O(#design parameter)
- Adjoint method requires just 1 additional solve!

Explicit update rule: $\mathbf{u}^n = \mathbf{h}^n(\mathbf{u}^{n-1}; \theta)$

Quantity of interest: $g(\mathbf{u}^N; \theta)$

Desired sensitivities: $\dfrac{dg(\mathbf{u}^N; \theta)}{d\theta}$

$$\mathcal{L}(\mathbf{u}, \boldsymbol{\lambda}; \theta) = g\left(\mathbf{u}^N; \theta\right) + \sum_n \langle \boldsymbol{\lambda}^n, \mathbf{h}^n(\mathbf{u}^{n-1}; \theta) - \mathbf{u}^n \rangle$$

$$\frac{d\mathcal{L}}{d\theta} = g_{,\boldsymbol{\theta}} + \langle g_{,\mathbf{u}}, \mathbf{u}^N_{,\theta} \rangle + \sum_n \langle \boldsymbol{\lambda}^n, \mathbf{h}^n_{,\theta} + \mathbf{h}^n_{,\mathbf{u}} \mathbf{u}^{n-1}_{,\theta} - \mathbf{u}^n_{,\theta} \rangle$$

Adjoint of a recursion relation

Final condition: $$\boldsymbol{\lambda}^N = g_{,\mathbf{u}}(\mathbf{u}^N;\theta)$$

Adjoint update: $$\boldsymbol{\lambda}^n = \left(\mathbf{h}^{n+1}_{,\mathbf{u}}\right)^T \boldsymbol{\lambda}^{n+1}$$

Sensitivity: $$\frac{d\mathcal{L}}{d\theta} = g_{,\theta} + \sum_n \langle \boldsymbol{\lambda}^n, \mathbf{h}^n_{,\theta} \rangle$$

Strang's Interpretation:

$$\mathbf{v}_{N\times 1} = (\mathbf{A}_{N\times N}\ \mathbf{B}_{N\times N})\ \mathbf{u}_{N\times 1}$$

vs

$$\mathbf{v}_{N\times 1} = \mathbf{A}_{N\times N}\ (\mathbf{B}_{N\times N}\ \mathbf{u}_{N\times 1})$$

Example with explicit Newmark

Explicit Newmark time integration

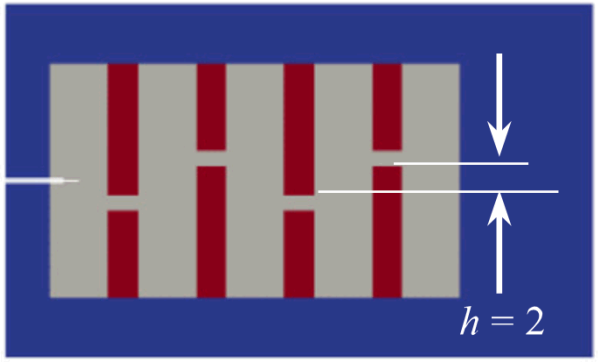
$$\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t\, \mathbf{v}^n + \frac{\Delta t^2}{2}\mathbf{a}^n$$
$$\mathbf{v}^{n+1/2} = \mathbf{v}^n + \frac{\Delta t}{2}\mathbf{a}^n$$
$$\mathbf{a}^{n+1} = \boldsymbol{\pi}^{n+1}(\mathbf{u}^{n+1})$$
$$\mathbf{v}^{n+1} = \mathbf{v}^{n+1/2} + \frac{\Delta t}{2}\mathbf{a}^{n+1}$$

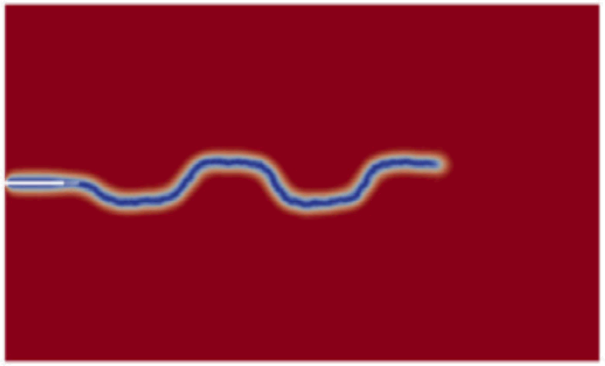
Explicit adjoint

$$\hat{\mathbf{a}}^{n+1/2} = \hat{\mathbf{a}}^{n+1} + \frac{\Delta t}{2}\hat{\mathbf{v}}^{n+1}$$
$$\hat{\mathbf{u}}^{n} = \hat{\mathbf{u}}^{n+1} + \hat{\mathbf{a}}^{n+1/2} \cdot \boldsymbol{\pi}^{n+1}_{,\mathbf{u}}$$
$$\hat{\mathbf{v}}^{n} = \hat{\mathbf{v}}^{n+1} + \Delta t\, \hat{\mathbf{u}}^{n}$$
$$\hat{\mathbf{a}}^{n} = \frac{\Delta t}{2}\hat{\mathbf{v}}^{n}$$

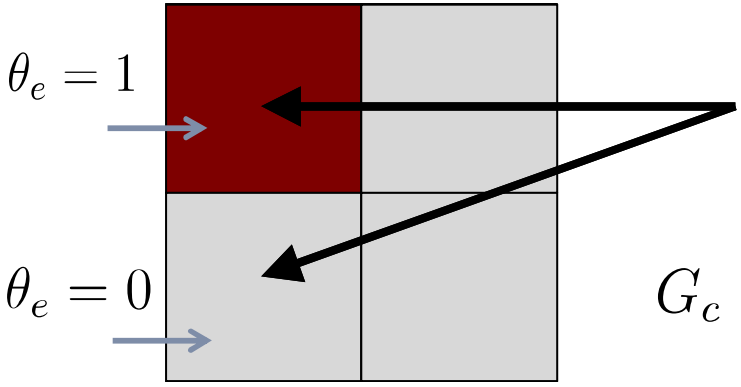
Heterogeneous material design

- Hossain, Hsueh, Bourdin, Bhattacharya 2014 show heterogeneous materials can be tougher than its constituents!





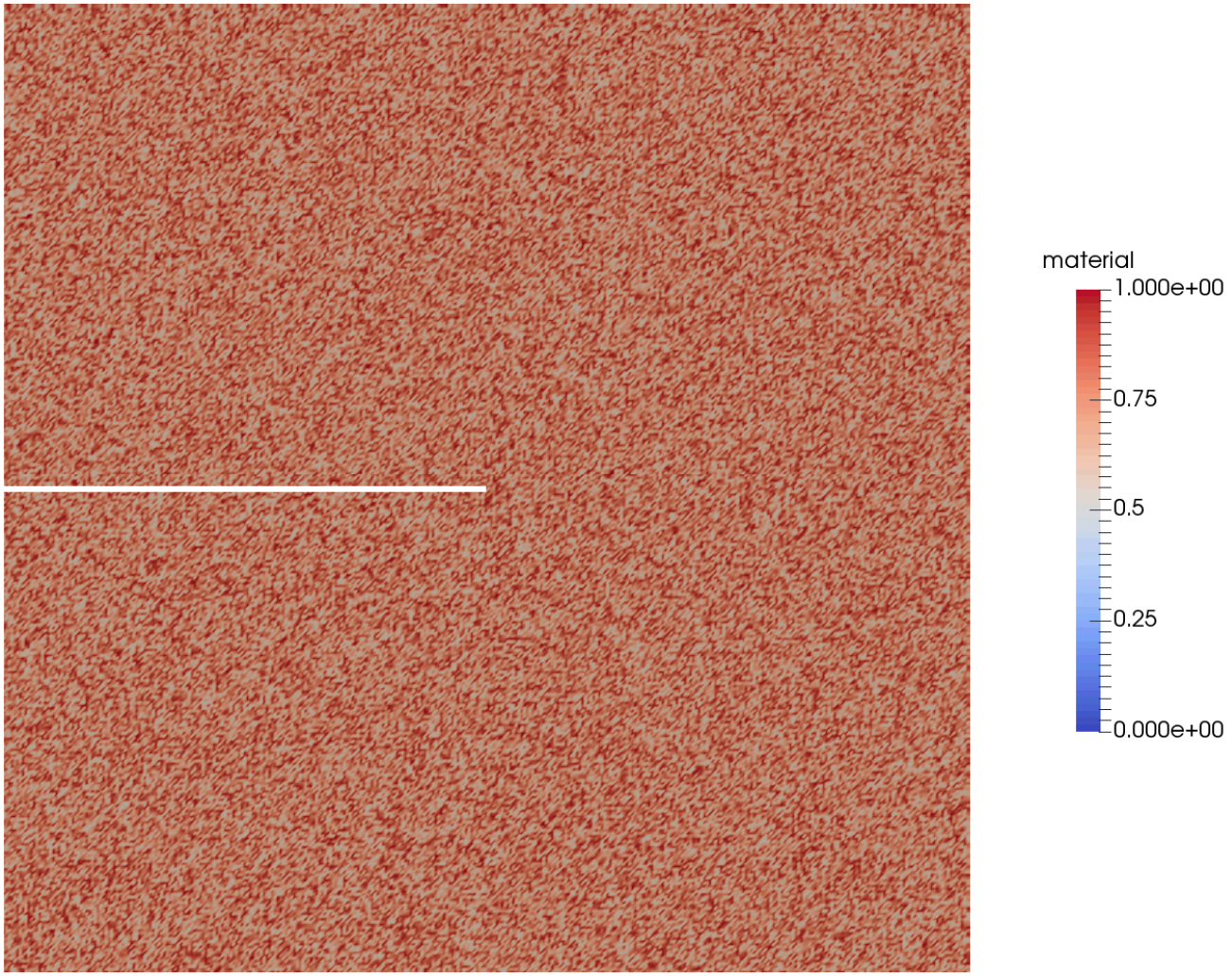
- Use adjoint framework and Sandia's *Rapid Optimization Library* to automatically design tougher material



Each element chooses to be compliant or stiff: $\theta_e \in [0, 1]$

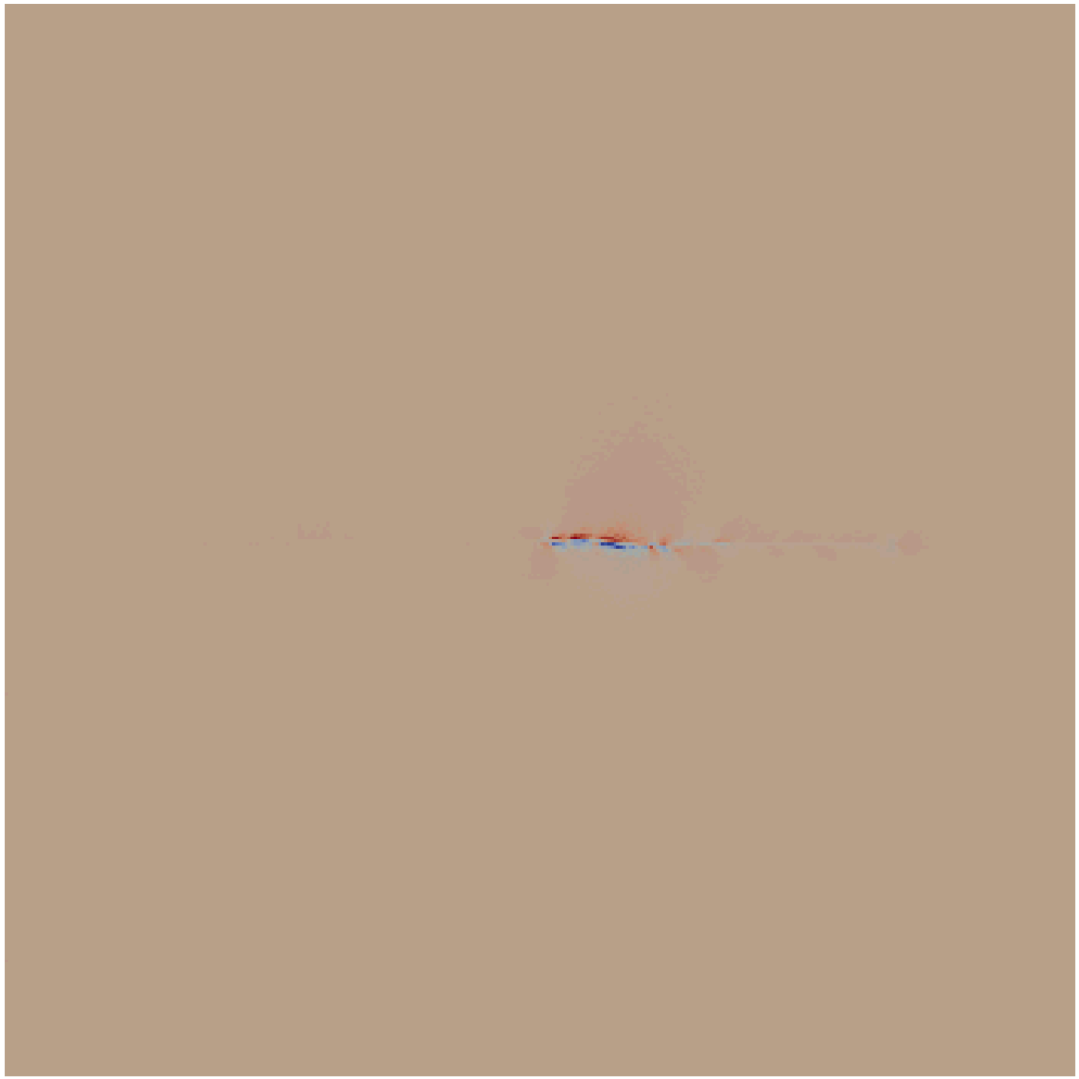
$$G_c = 4\text{e}5 \quad \rho = 8000 \quad E = 47.5\text{e}9 \text{ or } 190\text{e}9$$

Initial material property guess



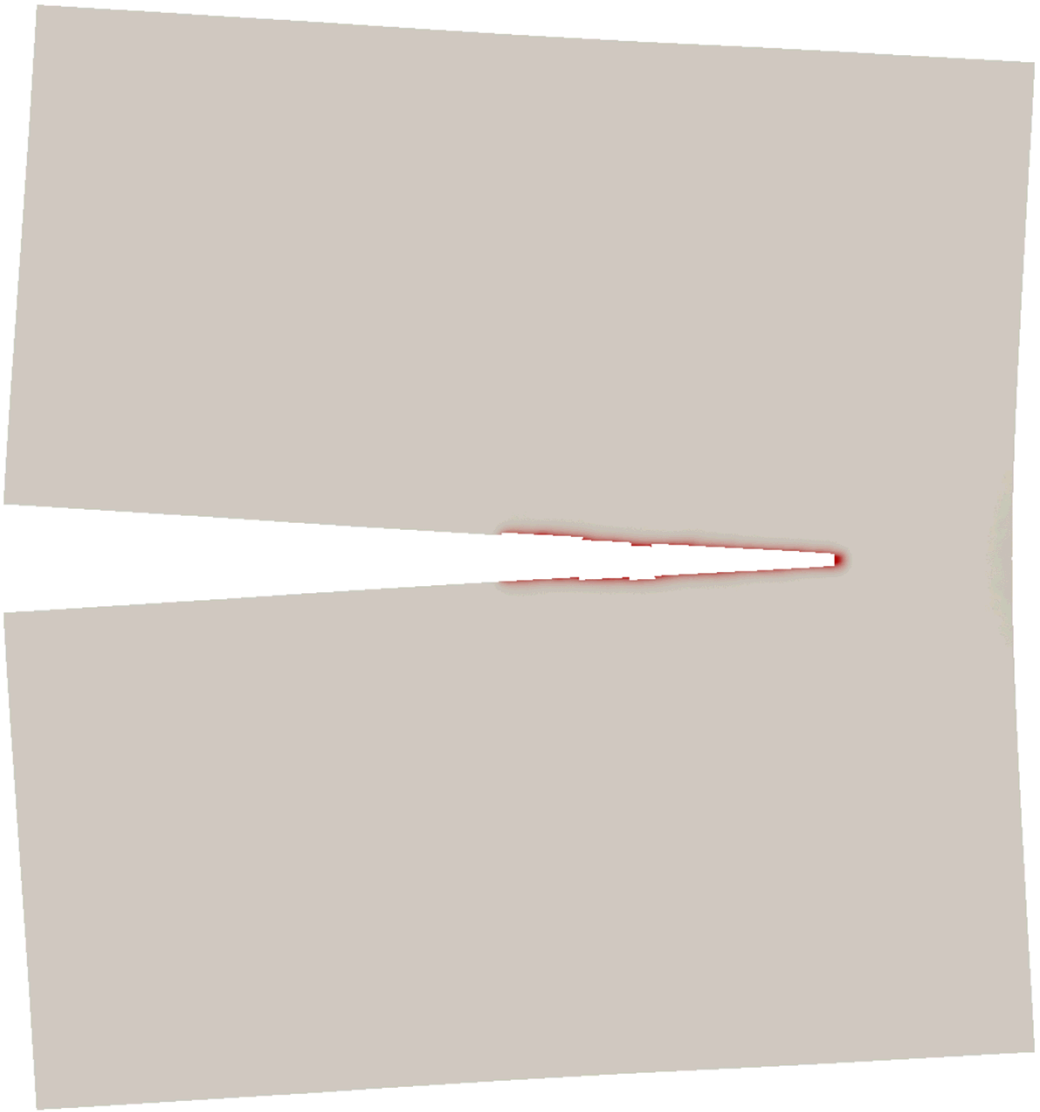
Damage evolution

Adjoint sensitivity to initial props

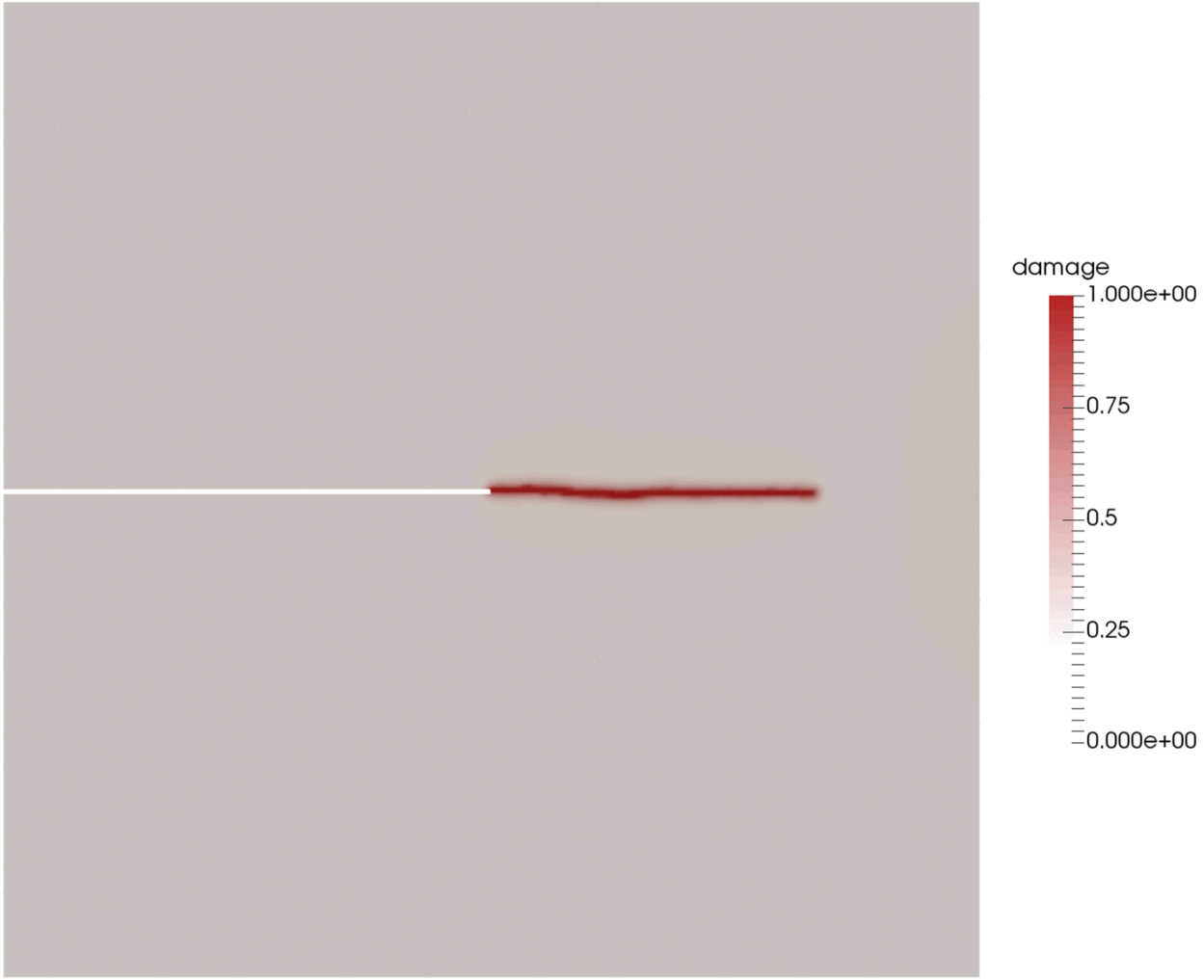


~6 times the cost of a forward simulation

Minimize crack propagaion

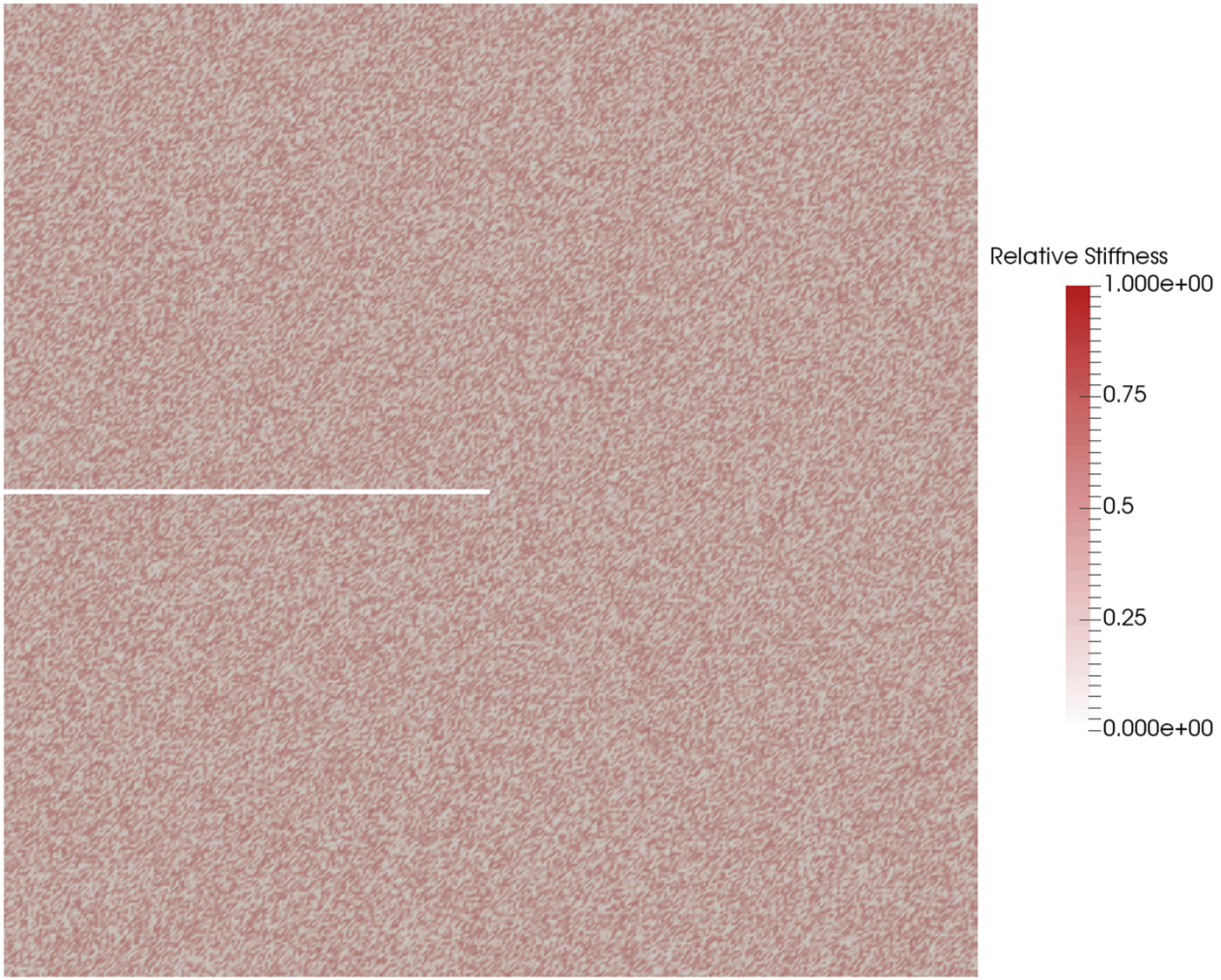


Minimize crack propagation



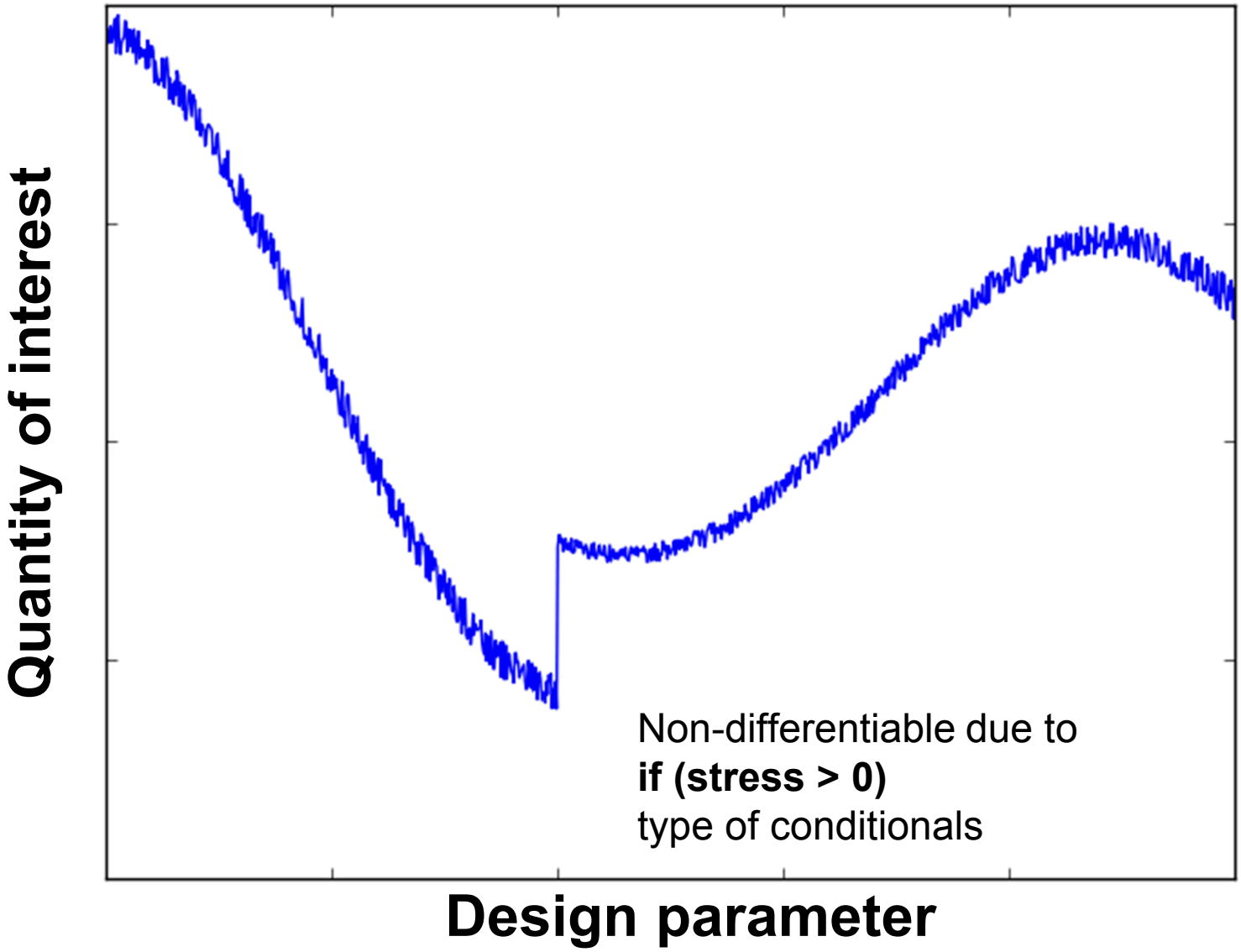
Un-deformed configuration

Find locally optimal design

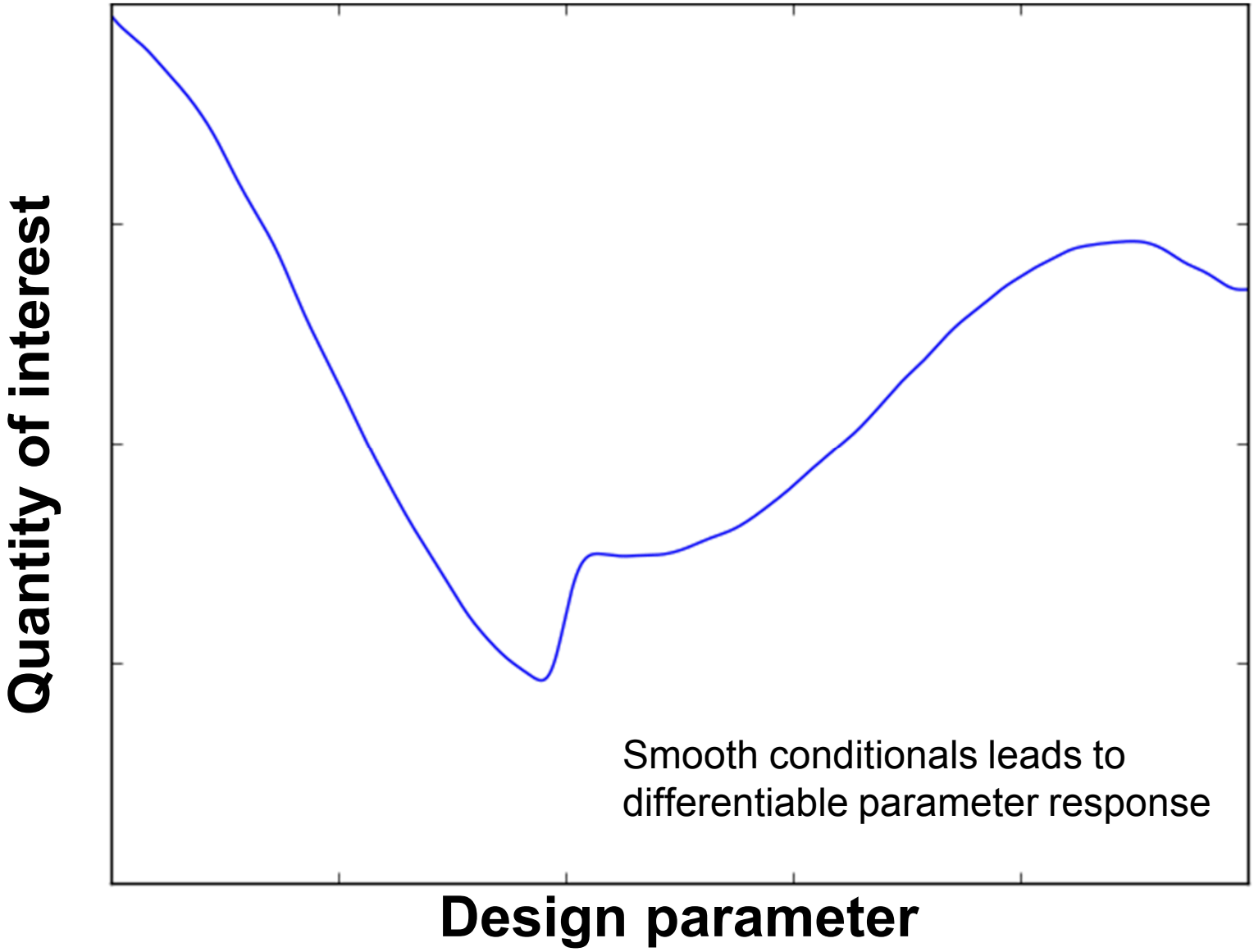


121801 design parameter gradients

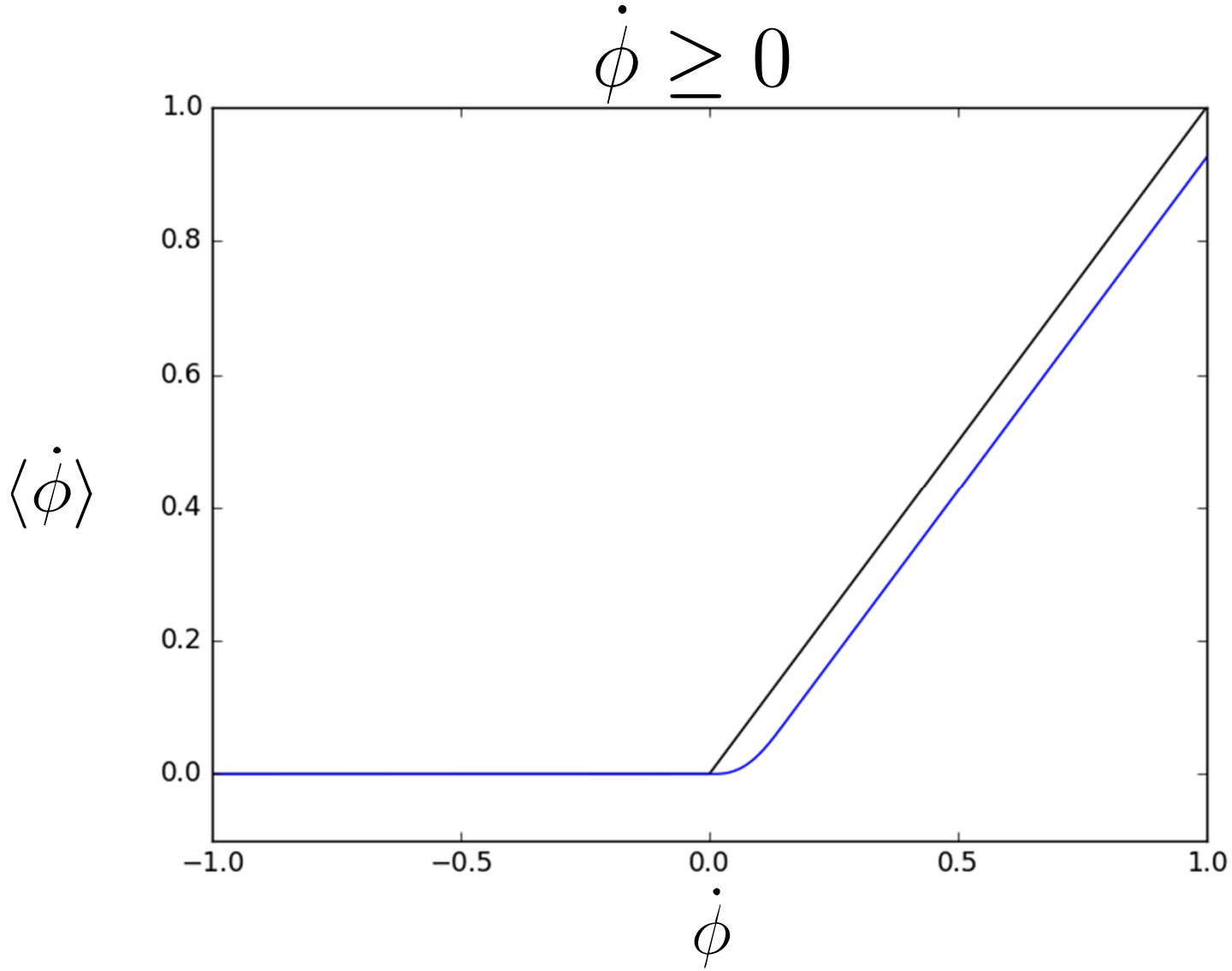
Non-smooth quantity of interest



Smoothed quantity of interest



Differentiable approximation to $\dot{\phi} \geq 0$



Library for efficient computation of parameter sensitivities

Sandia National Laboratories

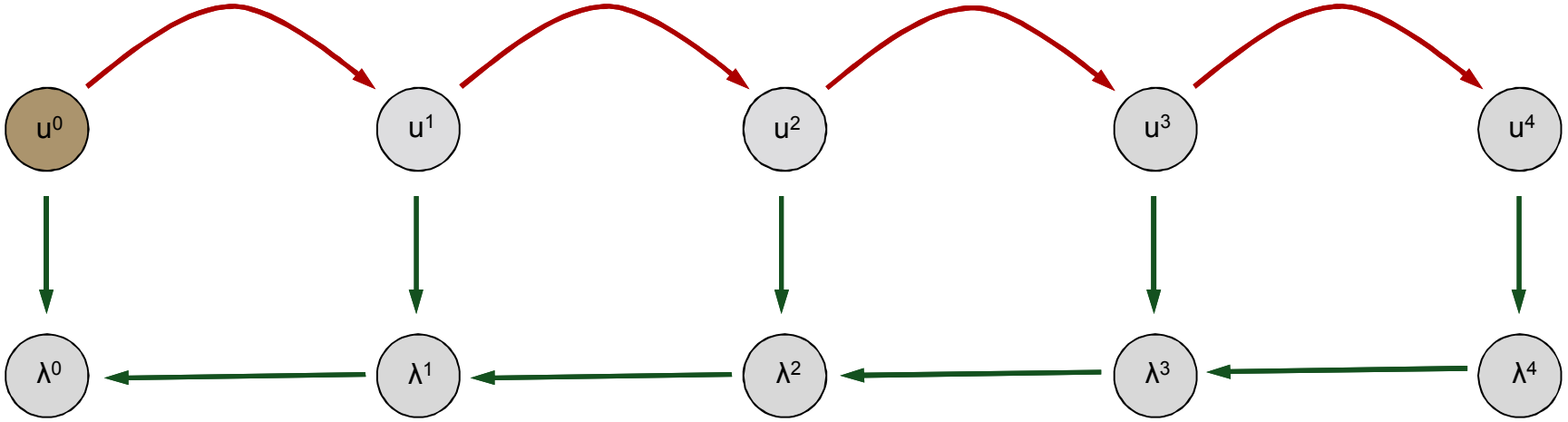
- Developed C++ library: simplifies the implementation and testing of *adjoint* sensitivities for explicit dynamic simulations
- Users define
  - State: **u** (e.g. displacements)
  - Design parameters: **θ** (e.g. initial material properties, BCs, etc.)
  - Operators:

$$\mathbf{u}^{n+1} = \mathbf{f}(\mathbf{u}^n, \boldsymbol{\theta}) \qquad \frac{\partial \mathbf{f}(\mathbf{u}^n, \boldsymbol{\theta}) \cdot \boldsymbol{\mu}}{\partial \mathbf{u}^n} \qquad \frac{\partial \mathbf{f}(\mathbf{u}^n, \boldsymbol{\theta}) \cdot \boldsymbol{\mu}}{\partial \boldsymbol{\theta}}$$

$$\text{q.o.i.} = g(\mathbf{u}^N, \boldsymbol{\theta}) \qquad \frac{\partial g(\mathbf{u}^N, \boldsymbol{\theta})}{\partial \mathbf{u}^N} \qquad \frac{\partial g(\mathbf{u}^N, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$$

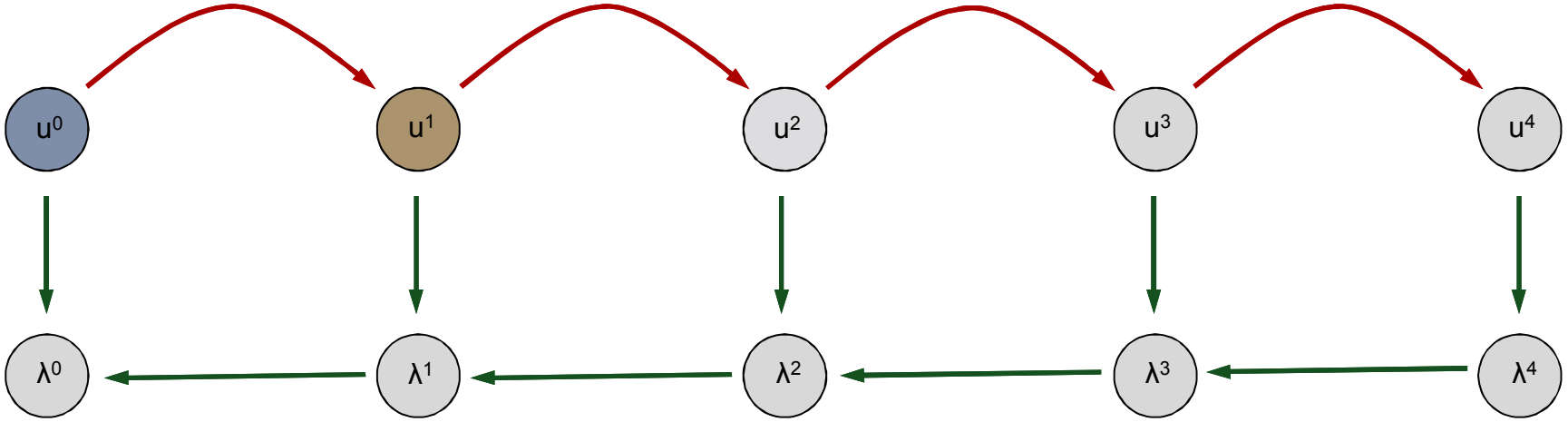
- Library automatically computes $\frac{dg(\mathbf{u}^N, \boldsymbol{\theta})}{d\boldsymbol{\theta}}$
- 1,000 – 1,000,000+ parameter sensitivities take time of ~10 forward solves

Checkpointing for backward integration



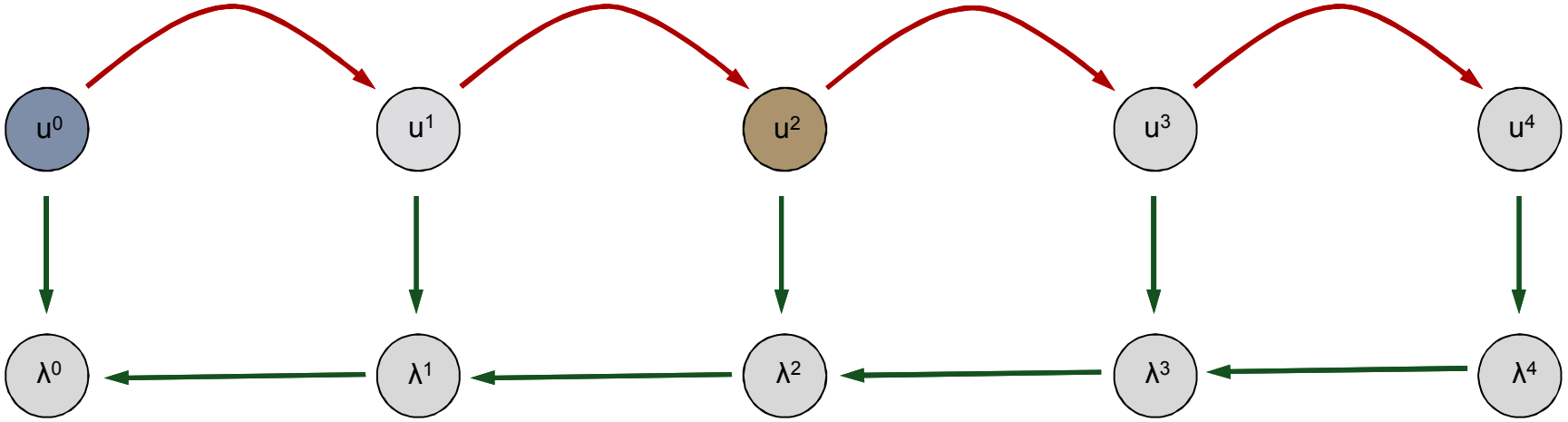
Using ***Minimal Repetition Checkpointing Algorithm*** by Q. Wang, et al.

Checkpointing for backward integration



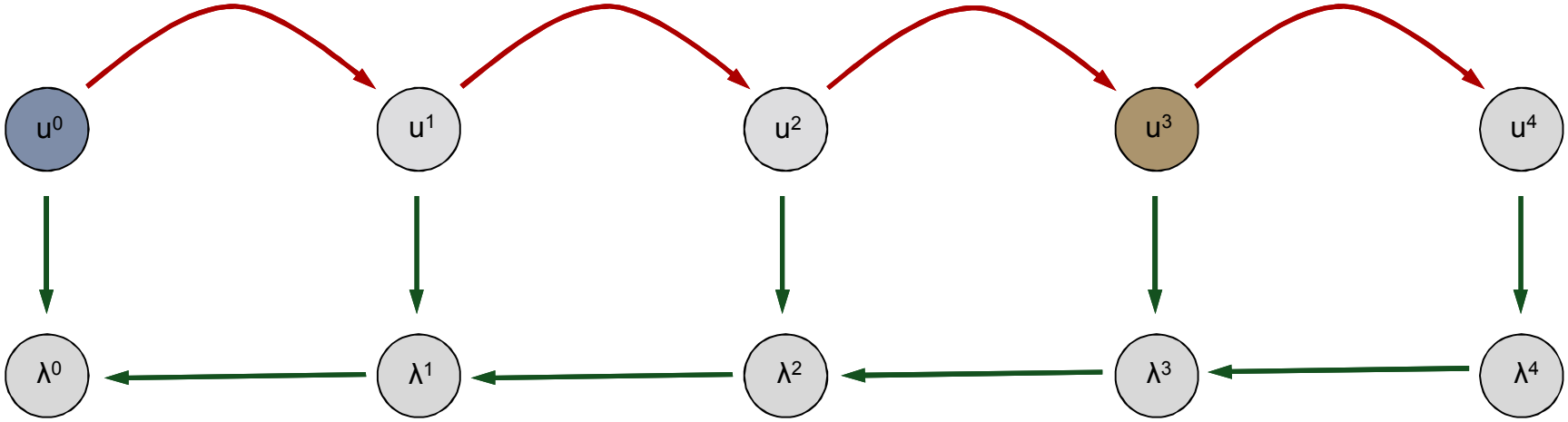
Using ***Minimal Repetition Checkpointing Algorithm*** by Q. Wang, et al.

Checkpointing for backward integration



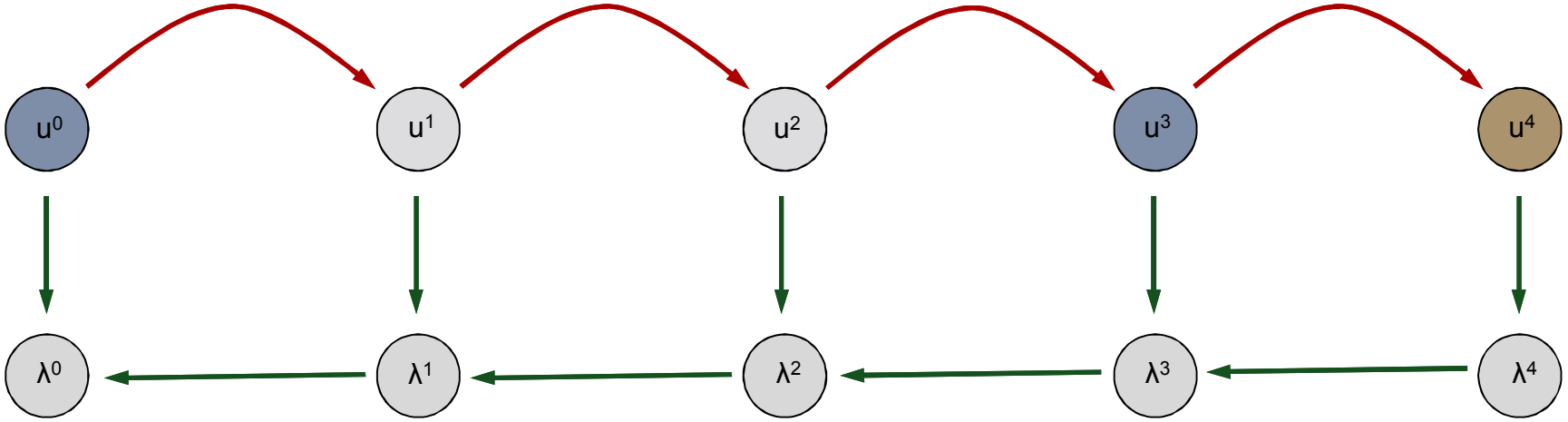
Using ***Minimal Repetition Checkpointing Algorithm*** by Q. Wang, et al.

Checkpointing for backward integration



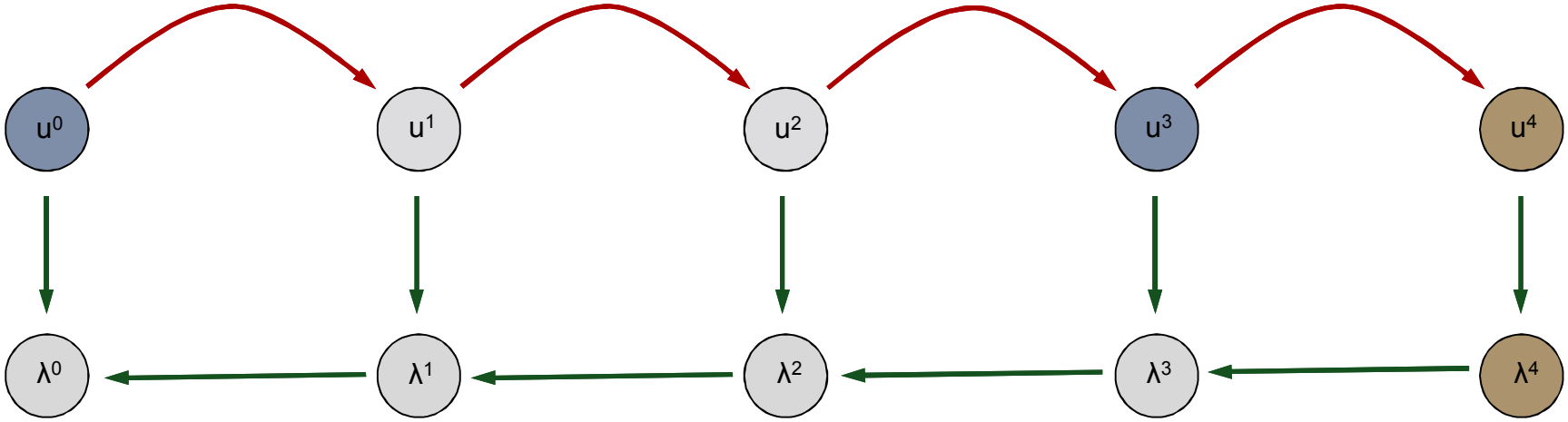
Using ***Minimal Repetition Checkpointing Algorithm*** by Q. Wang, et al.

Checkpointing for backward integration



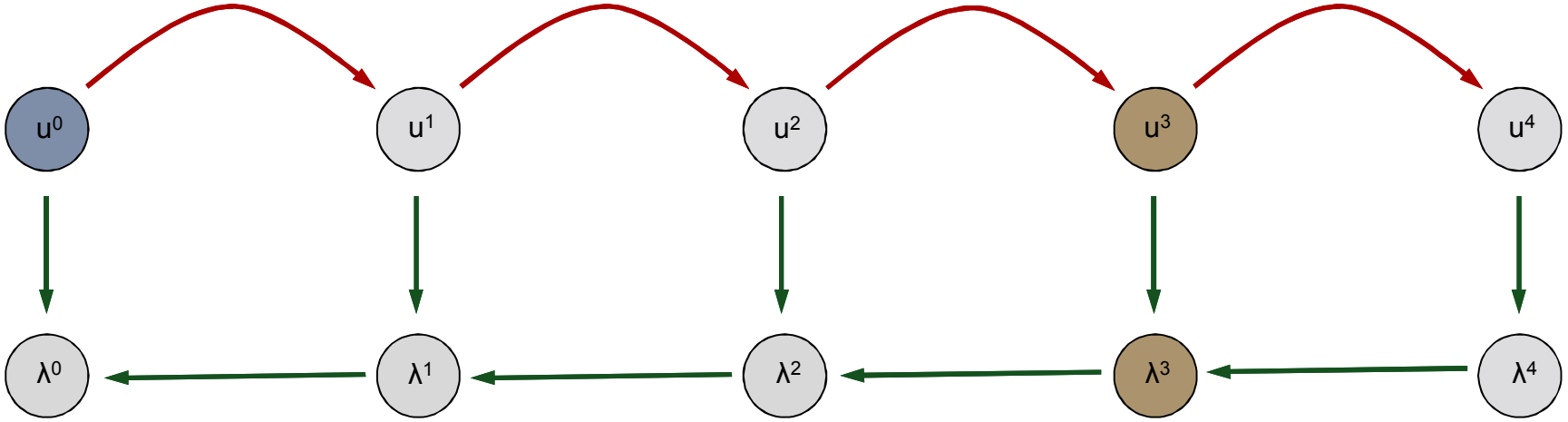
Using ***Minimal Repetition Checkpointing Algorithm*** by Q. Wang, et al.

Checkpointing for backward integration



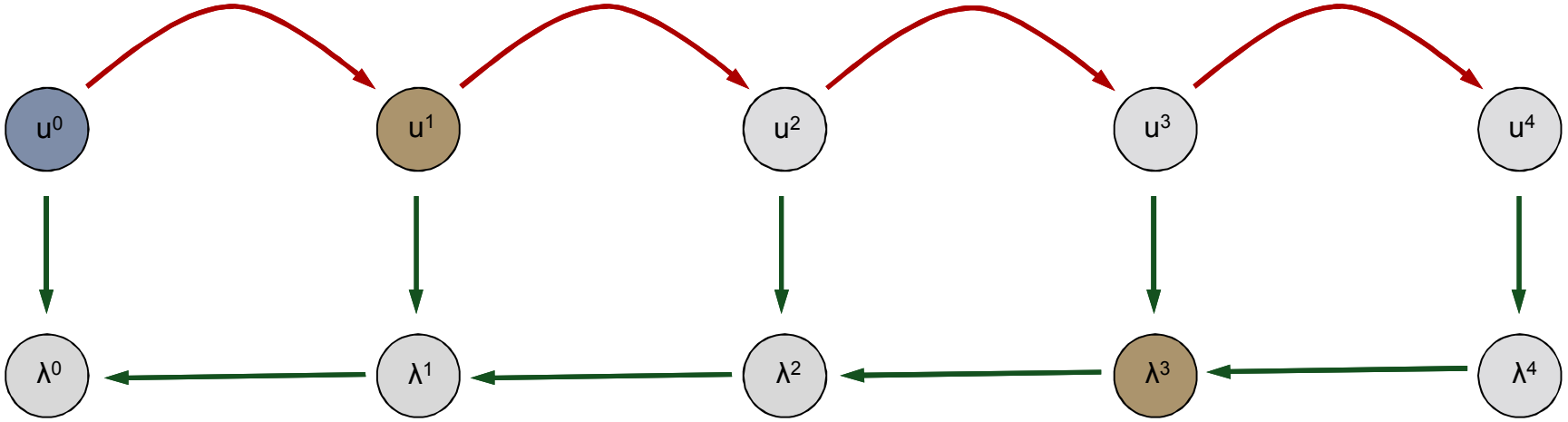
Using ***Minimal Repetition Checkpointing Algorithm*** by Q. Wang, et al.

Checkpointing for backward integration



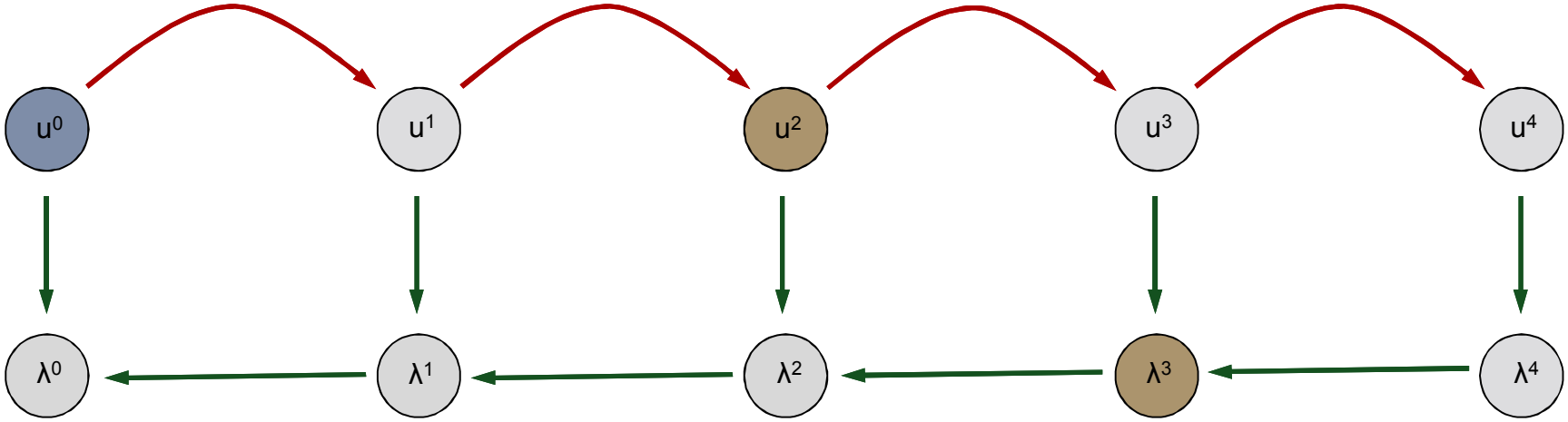
Using ***Minimal Repetition Checkpointing Algorithm*** by Q. Wang, et al.

Checkpointing for backward integration



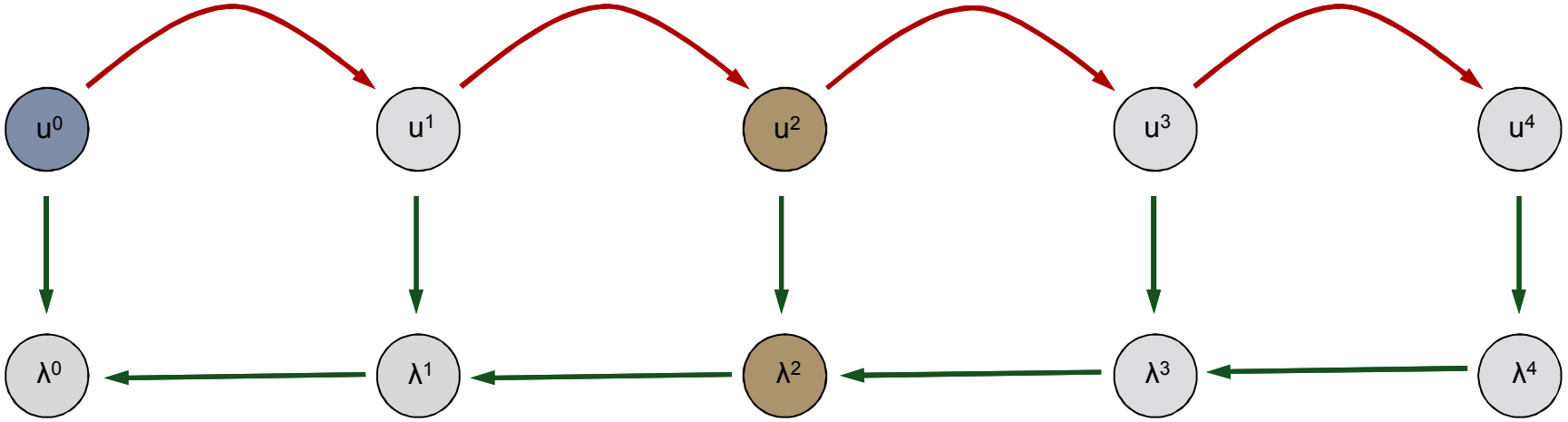
Using ***Minimal Repetition Checkpointing Algorithm*** by Q. Wang, et al.

Checkpointing for backward integration



Using ***Minimal Repetition Checkpointing Algorithm*** by Q. Wang, et al.

Checkpointing for backward integration



Using ***Minimal Repetition Checkpointing Algorithm*** by Q. Wang, et al.

Library for efficient computation of parameter sensitivities

Sandia National Laboratories

- Developed C++ library: simplifies the implementation and testing of *adjoint* sensitivities for explicit dynamic simulations
- Users define
  - State: **u** (e.g. displacements)
  - Design parameters: **θ** (e.g. initial material properties, BCs, etc.)
  - Operators:

$$\mathbf{u}^{n+1} = \mathbf{f}(\mathbf{u}^n, \boldsymbol{\theta}) \qquad \frac{\partial \mathbf{f}(\mathbf{u}^n, \boldsymbol{\theta}) \cdot \boldsymbol{\mu}}{\partial \mathbf{u}^n} \qquad \frac{\partial \mathbf{f}(\mathbf{u}^n, \boldsymbol{\theta}) \cdot \boldsymbol{\mu}}{\partial \boldsymbol{\theta}}$$

$$\text{q.o.i.} = g(\mathbf{u}^N, \boldsymbol{\theta}) \qquad \frac{\partial g(\mathbf{u}^N, \boldsymbol{\theta})}{\partial \mathbf{u}^N} \qquad \frac{\partial g(\mathbf{u}^N, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$$

- Library automatically computes $\frac{dg(\mathbf{u}^N, \boldsymbol{\theta})}{d\boldsymbol{\theta}}$
- 1,000 – 1,000,000+ parameter sensitivities take time of ~10 forward solves

Adding up the savings

Sandia National Laboratories

- Explicitly integrating damage: > 10 x
- Structured grid, linear kinematics: > 20 x
- GPU vs. CPU: > 100 x
- Adjoint sensitivities: > (#Elements / 10) x

- For 100,000 element simulation: > 2e8 times speedup
- From average human lifetime to ~10 seconds

Conclusions

Sandia National Laboratories

- Adjoint method is too important to be ignored
  - Non-trivial to implement and maintain
  - Requires differentiability
- Thread scalability is too important to be ignored
  - Data movement kills performance
  - May need to rethink the algorithms we use
  - Hardware trends are likely to continue
- Next steps:
  - Adjoint method for dynamic contact, plasticity, etc.
  - Improve usability and robustness of inverse framework

Thanks

Sandia National Laboratories

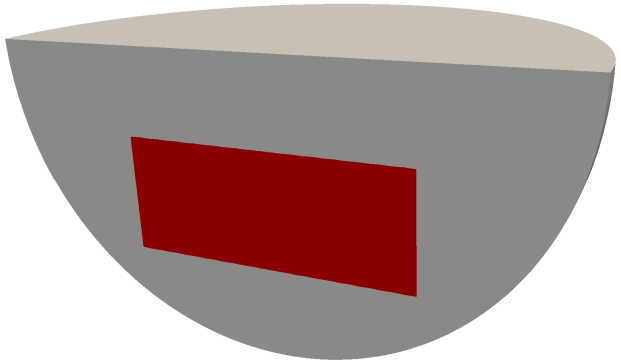
- Solid Mechanics Team
- John Dolbow
- Tim Walsh
- Kyle Starkey
- ROL team
- Kokkos team
- Sierra DevOps

Extra slides

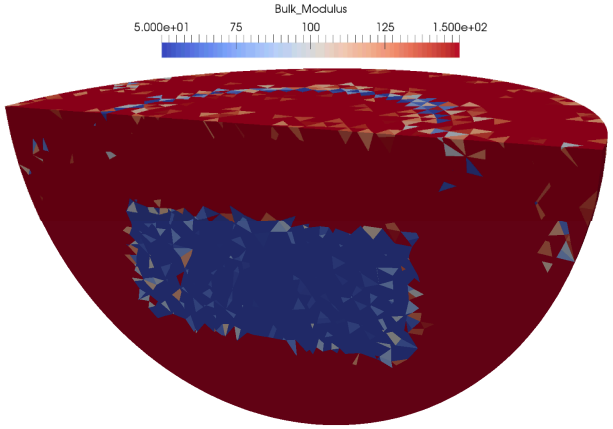
Sandia National Laboratories

Phase-field inversion examples

- Minimize discrepancy between observed displacements and simulated displacements by varying phase (volume fraction) element by element
- Sandia's ROL library used to perform the optimization

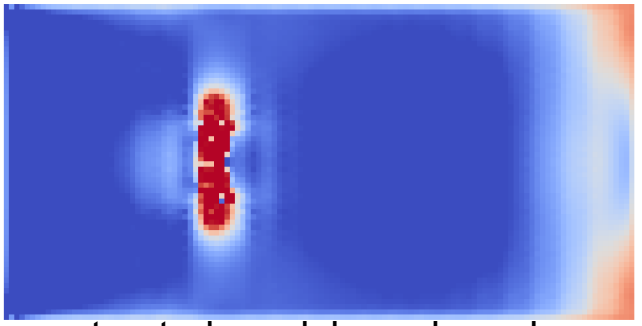


Initially unknown hidden tunnel subjected to surface excitation



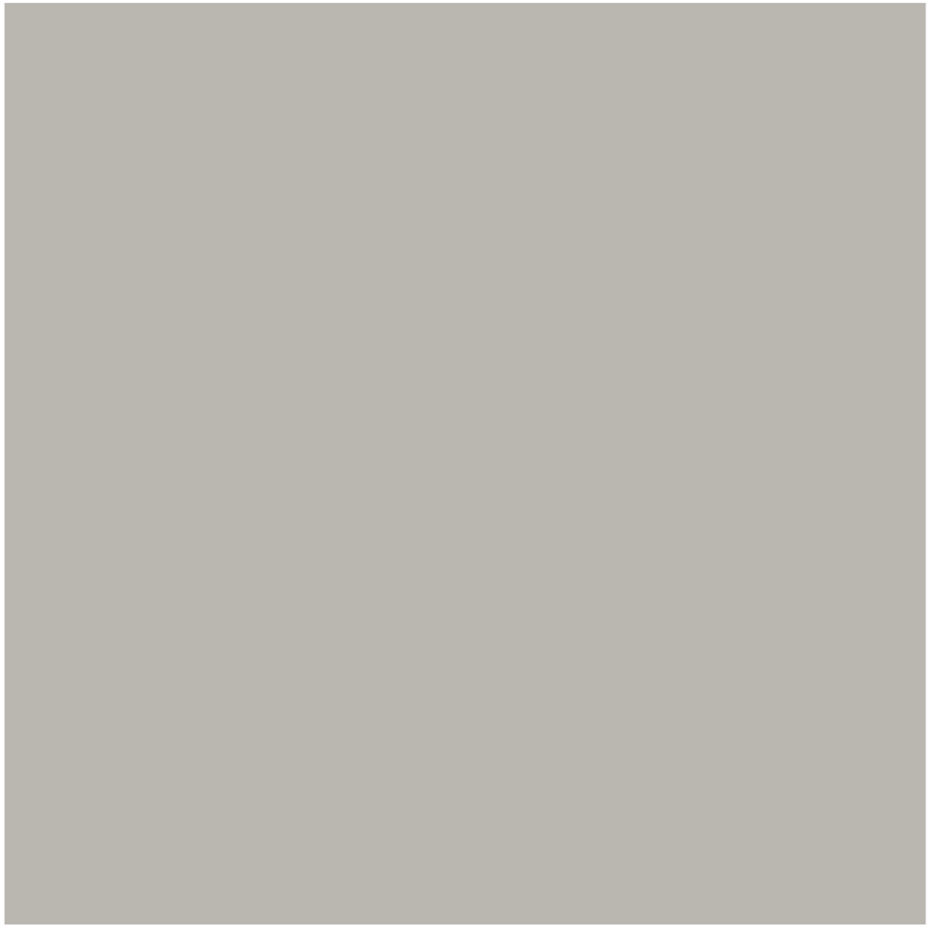
Reconstructed material properties due to observed surface displacements in Sierra-SD

Initial crack/damage in specimen

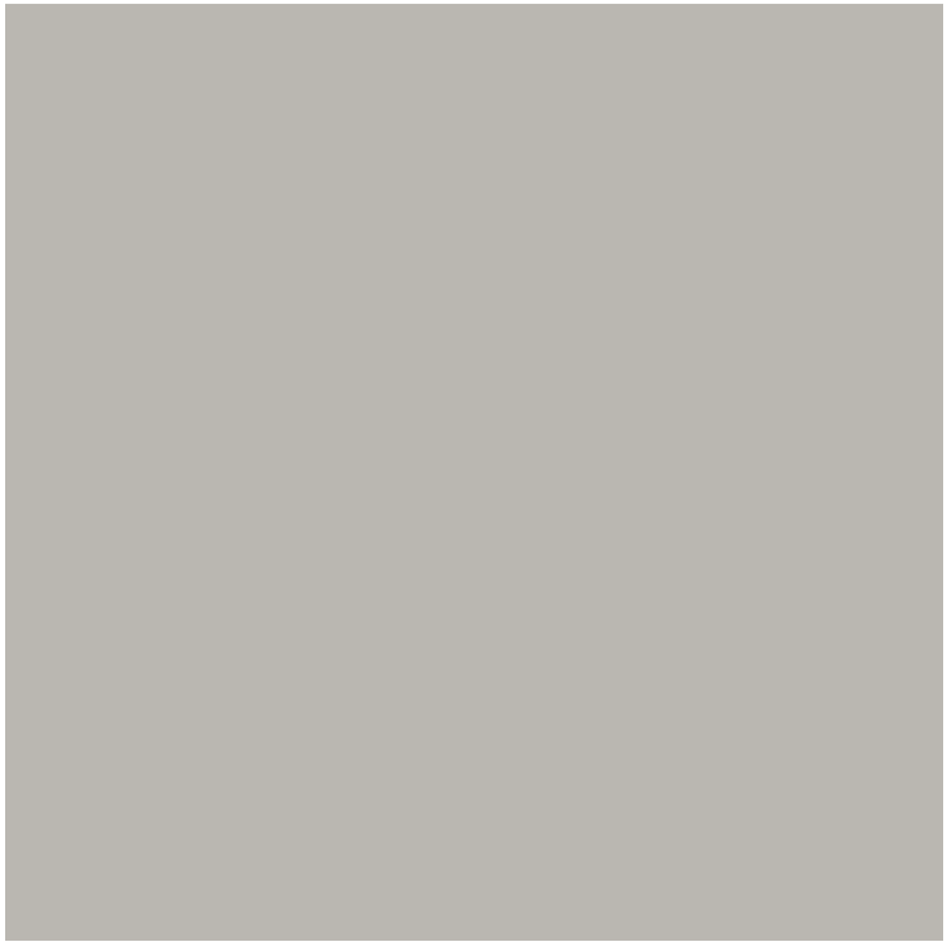


Reconstructed crack based on observed displacements due to acoustic excitation

High fracture energy release rate



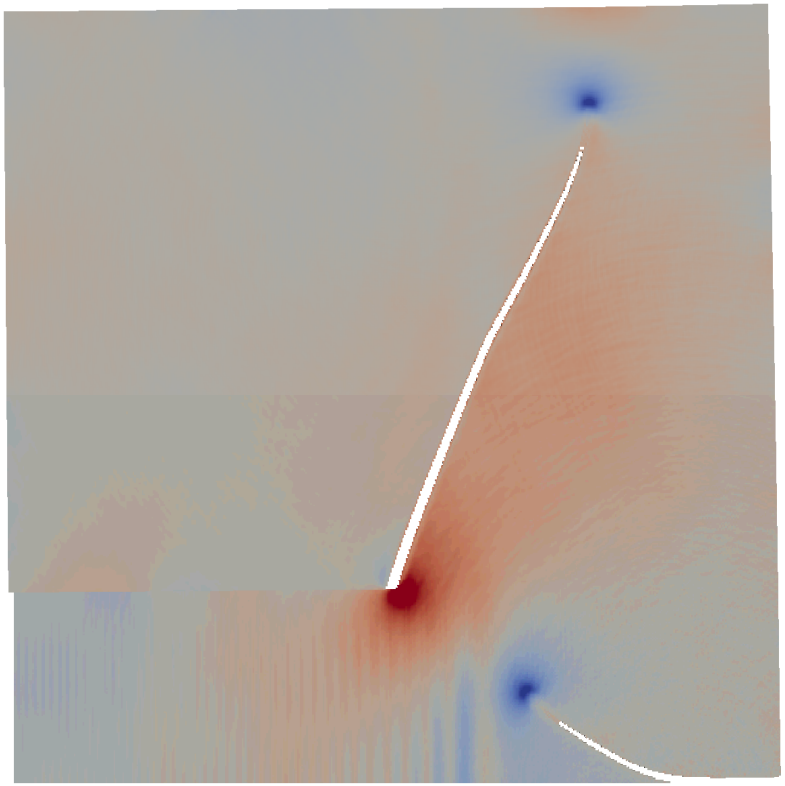
Low fracture energy release rate



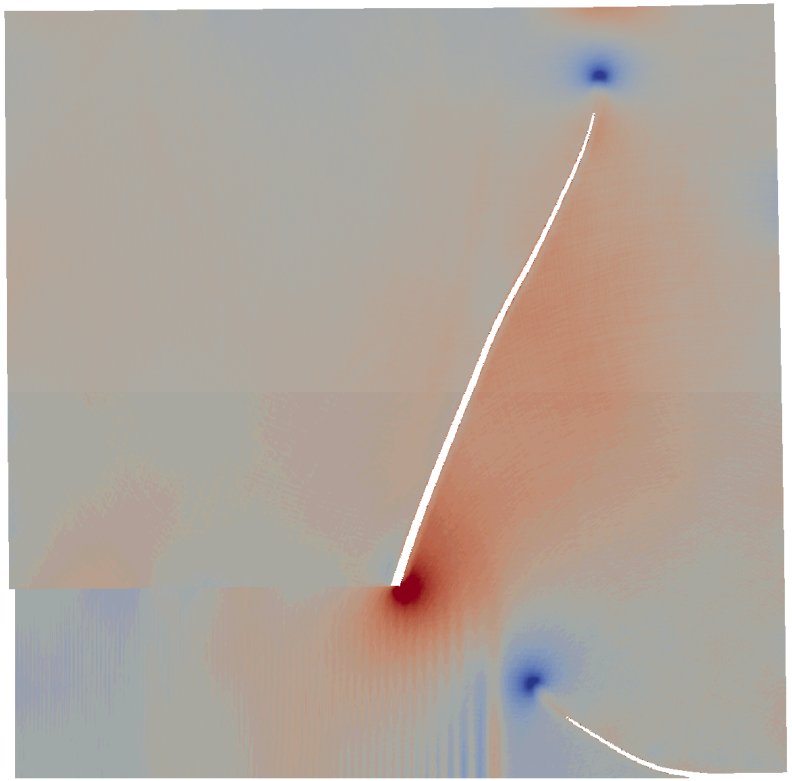
Mesh insensitivity

Sandia National Laboratories

Pressure contours for dynamic crack propagation

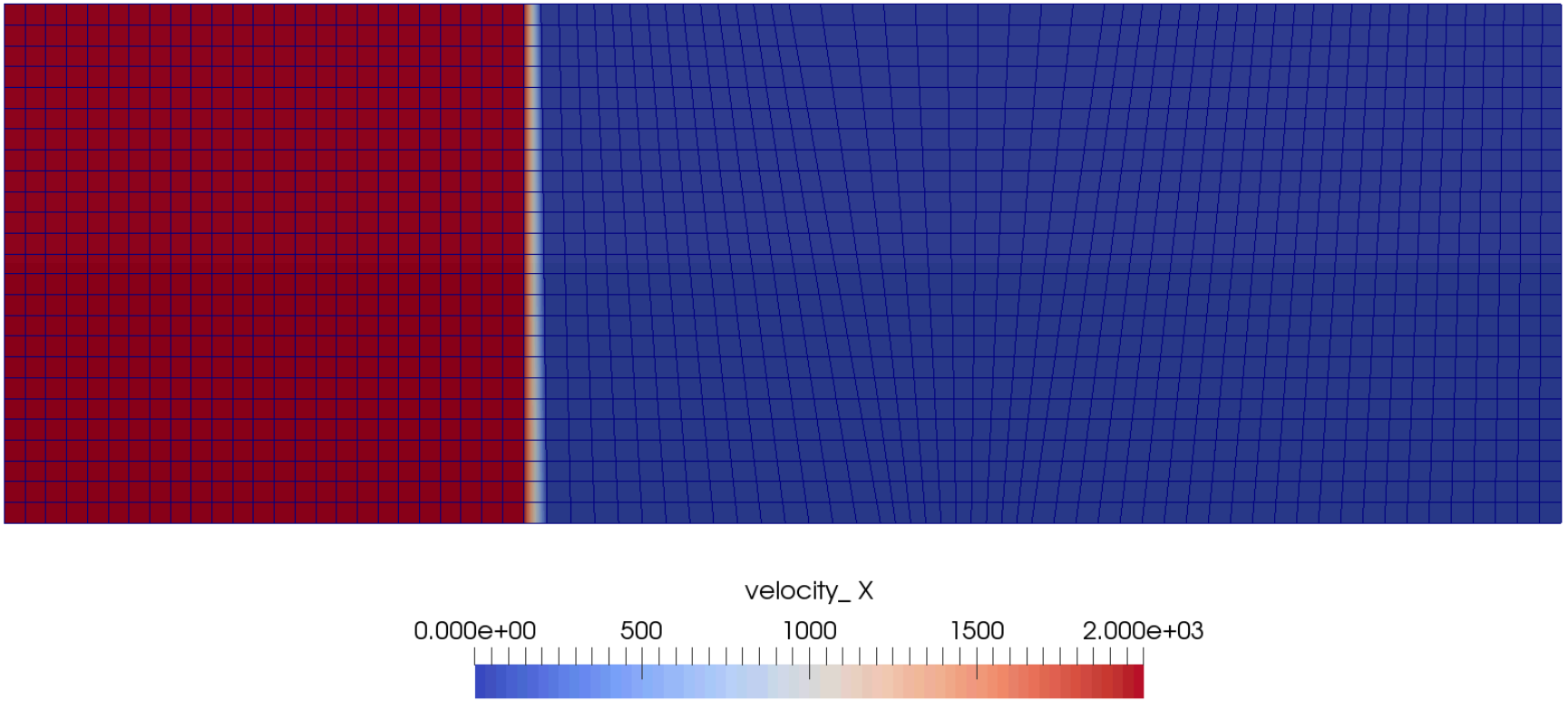


Medium mesh: 250k elements

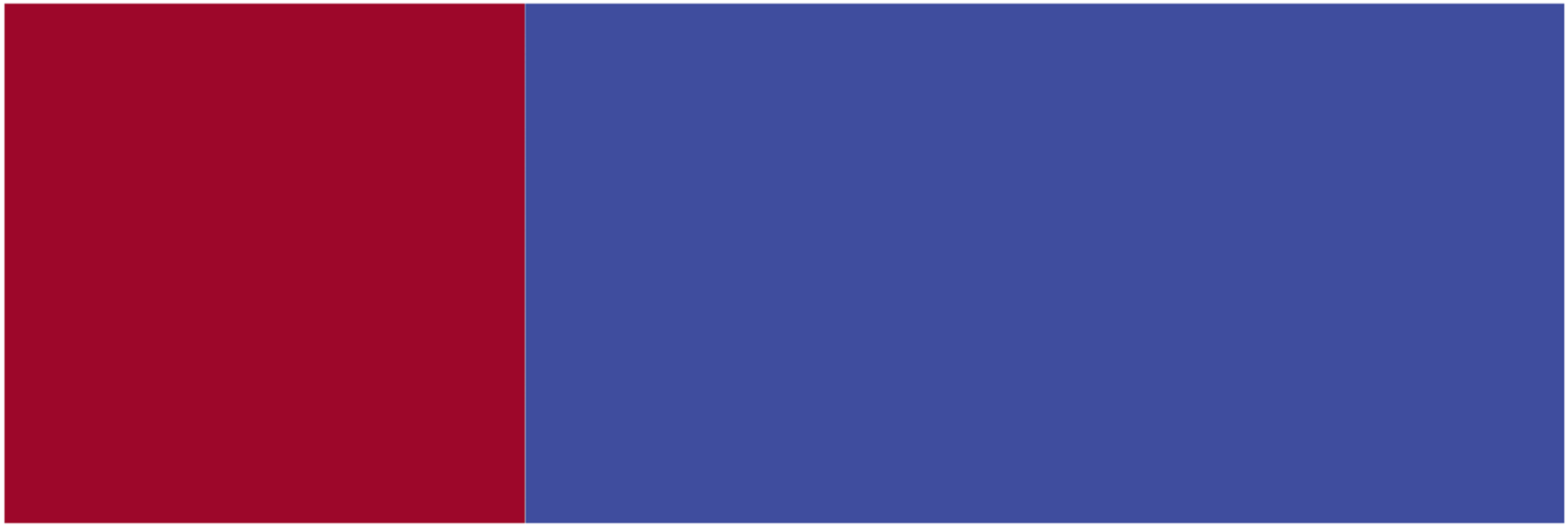


Fine mesh: 1,000k elements

Spall example: Initial velocity

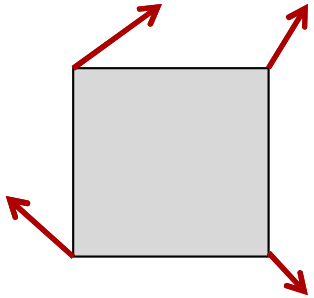


Spall example:

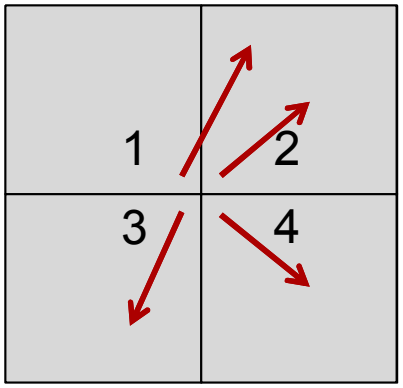


Determinacy in a threaded world

- Computing adjoint for explicit dynamics requires checkpointing and re-computing solutions
- This makes run-to-run determinacy necessary
- Use data duplication:



Loop over element:
store all element forces



Loop over nodes:
sum forces in predetermined order

Reintroducing the Lorentz model

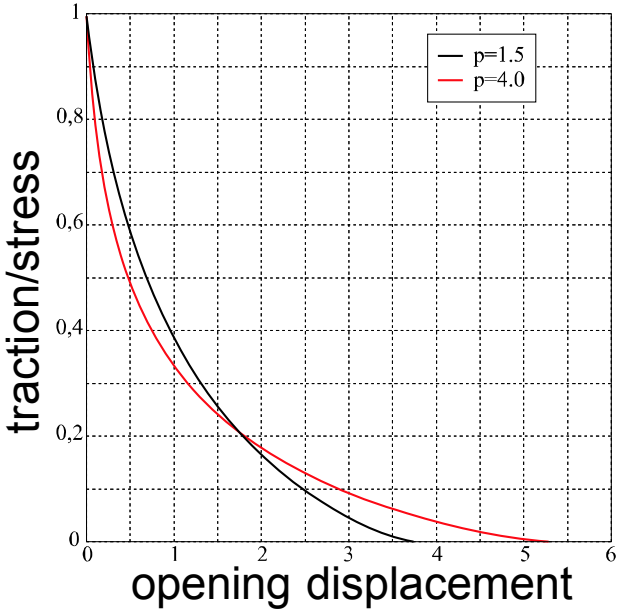
Sandia National Laboratories

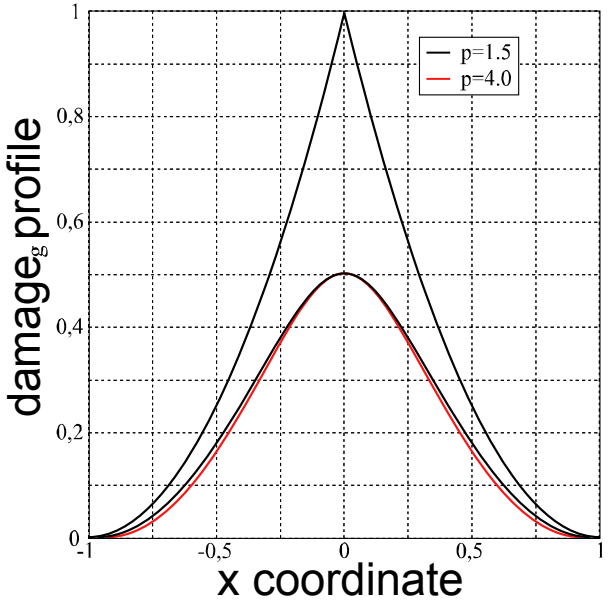
- An elliptic regularization of a cohesive zone model.
- Has a natural critical strain energy criterion.
- Decouples the regularization length scale from the process zone size (though Lorentz argues on physical grounds that the regularization length scale must be smaller than the process zone size).
- Added the now standard decomposition into positive and negative strain energies.
- Irreversibility introduces via $\dot{\phi} \geq 0$.
- However, VERY expensive to solve a nonlinear inequality constrained elliptic PDE, especially for explicitly integrated transient dynamics.

Improved forward modeling of crack propagation in brittle materials

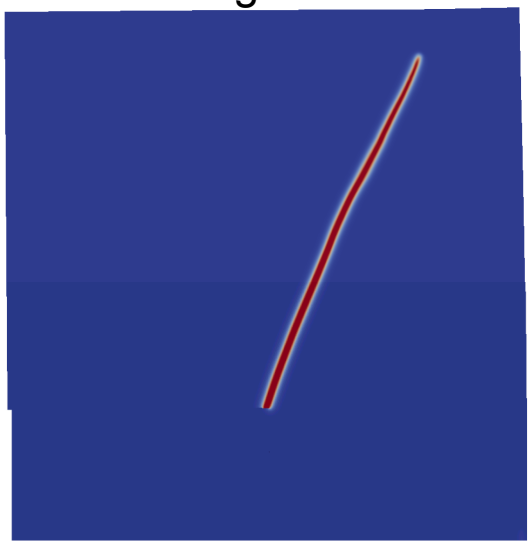
- Developed an explicitly integrated phase-field/gradient-damage model which avoids the usual nonlinear equation solve at each time step (>10 x better performance)
- Extension of the damage model by Lorentz, et al. 2011, which was shown to converge to a cohesive zone model as the length scale approaches 0

$$\eta\dot{d} = h(d)\psi(\boldsymbol{\sigma}) - \frac{3}{4}\frac{G_c}{l} + \frac{3}{8}G_c l\nabla\cdot\nabla d \qquad \rho\ddot{\mathbf{u}} = \nabla\cdot\boldsymbol{\sigma}(\boldsymbol{\epsilon}, d)$$





Damage contour



Cohesive phase-field traction separation law (left) and through 'crack' damage profile (right), p is a parameter of the model. Adapted from Lorentz, et al. 2011.

Kalthoff validation problem achieves the expected crack propagation angle: ~70°

Fracture inverse problems

Sandia National Laboratories

- Most analyses today are forward problem solves: simulate what happens in a given scenario
- However, what is often truly desired is the solution to the corresponding inverse problem
- Example fracture inverse problems:
  - Crack detection/non-destructive evaluation: determine the existence and locations of cracks in a structure
  - Crack forensics: find the loadings applied to a structure which result in the observed failure pattern
  - Material design: optimize the properties and geometry of a structure to resist crack initiation and propagation
- These last examples require accurate fracture prediction

General inverse problem statement as a PDE constrained optimization problem

$$\min_{\boldsymbol{\theta}} \; g(\mathbf{u}, \boldsymbol{\theta})$$

$$\text{s.t.} \;\; \mathcal{L}(\mathbf{u}, \boldsymbol{\theta}) = \mathbf{0}$$

$\mathbf{u}$: state variables
$\boldsymbol{\theta}$: design parameters
$\mathcal{L}$: differential operator
g: quantify of interest

Work in progress

- Collaborating with Sandia analysts to validate phase-field models for lab relevant 3D problems
- Applying the new inverse framework to the new phase-field crack propagation model in order to solve unprecedented crack inverse problems
- SANDIA reports and journal articles:
  - Brittle fracture phase-field modeling of a short-rod specimen (published SANDIA report)
  - A parabolic regularization of cohesive fracture for explicit dynamic crack propagation (to be submitted for peer review)
  - A C++ library for efficient adjoint sensitivities and automatic checkpointing in nonlinear explicit dynamic simulations (in progress SANDIA report)
- References:
  - Convergence of a gradient damage model toward a cohesive zone model, E. Lorentz, et al.
  - Minimal repetition dynamic checkpointing algorithm for unsteady adjoint calculation, Q. Wang, et al.
  - Crack identification by ‘arrival time’ using XFEM and a genetic algorithm, D. Rabinovich, et al.

- Consider a purely local model of damage
- Stress-strain relationship

$$\sigma(x,t) = (1 - d(x,t))C\varepsilon(x,t)$$

- Strain-displacement

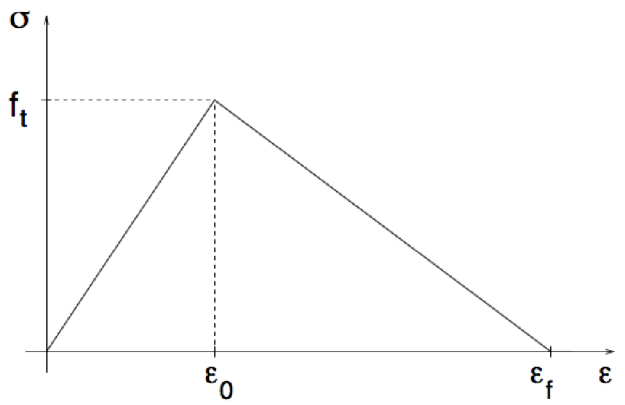
$$\varepsilon(x,t) = r^{s}u(x,t)$$

- Local state variable

$$Y(x,t) = Y(\varepsilon(x,t))$$

- Damage evolution

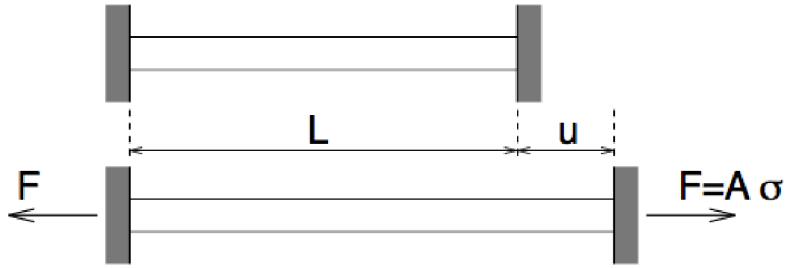
$$d(x,t)) = D\left(\max_{\tau \le t} Y(x,\tau)\right)$$

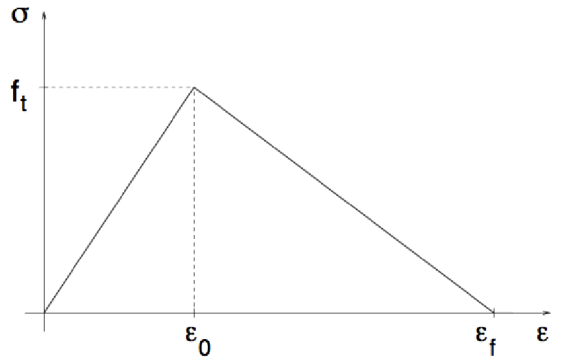


$$d(x,t)) = \frac{Y_f}{Y_f - Y_0}\left(1 - \frac{Y_0}{Y}\right)$$

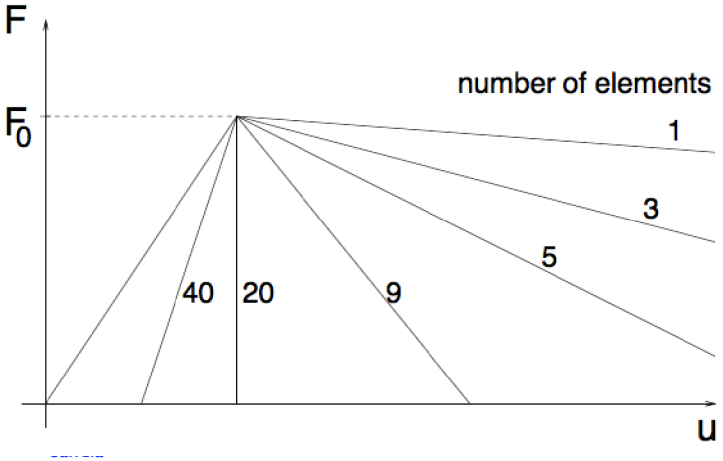
Fundamental Difficulty (cont'd)

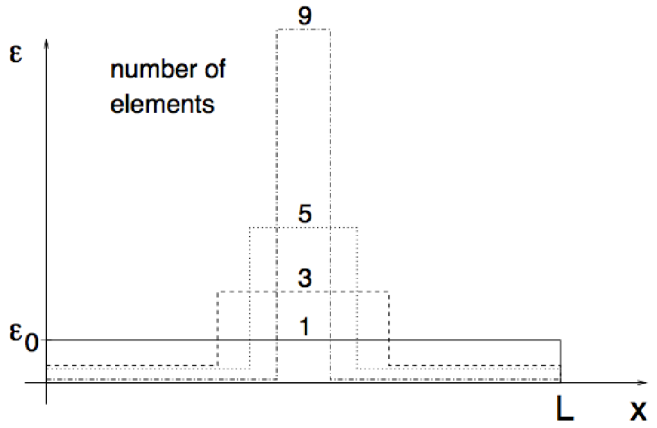
- Consider a 1D bar in tension (quasi-statics)





- Discretized with finite elements (from Jirasek, 2002):





- Basic idea: replace Y with a non-local version:

$$d(x,t)) = D \max_{\bar{x} \le t} \tilde{Y}(x,\bar{x})$$

- Integral-type models (Pijaudier-Cabot and Bazant, 1987) employ a non-local state variable, e.g.

$$\tilde{Y}(x,t) = \int_{V_x} \alpha(x,z)\, Y(z,t)\, dz$$

- Gradient-type models (de Borst et. al, 1995)

$$\tilde{Y}(x,t) - c\Delta \tilde{Y}(x,t) = Y(x,t) \quad \text{in } \Omega \qquad n \cdot \nabla \tilde{Y} = 0 \quad \text{on } \partial\Omega$$

- These two approaches are closely related (Huerta and Pijaudier-Cabot,

- Introduced by Rodriguez-Ferran et al. (2005)
- Stress-strain and strain-displacement:

$$\sigma(x,t) = (1 - d(x,t))C\varepsilon(x,t) \qquad \varepsilon(x,t) = r^s u(x,t)$$

- Non-local displacement:

$$\tilde{u}(x,t) - c\Delta\tilde{u}(x,t) = u(x,t) \quad \text{in } \Omega \qquad \tilde{u} = u \quad \text{on } \partial\Omega$$

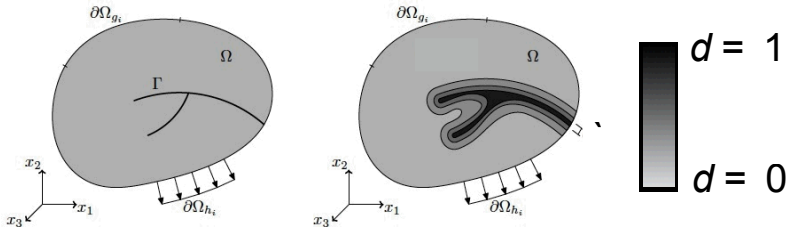
- Non-local strains and state variable:

$$\varepsilon_{NL}(x,t) = r^s \tilde{u}(x,t) \qquad Y_{NL}(x,t) = Y(\varepsilon_{NL}(x,t))$$

- Damage evolution:

$$d(x,t)) = D \max_{\tau \le t} Y_{NL}(x,\tau)$$

Phase-Field Methods for Fracture



- Motivated differently from gradient damage
- Basic idea: replace the sharp discontinuity by a small, but finite zone with sharp gradients in a mathematically consistent manner with a crack-density functional

$$\Gamma \approx \Gamma_l(d) = \int_\Omega \gamma_l(d, \nabla d)\, dV$$

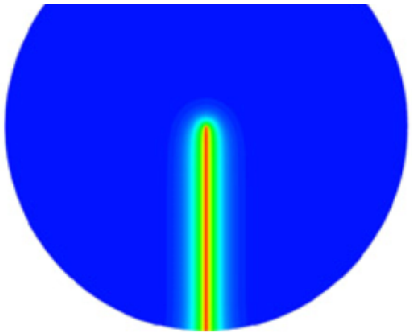
- Fracture energy (for brittle materials):

$$G_c\, dA \approx \int_\Omega G_c\, \gamma_l\, dV$$

Γ

- Common crack-density functional:

$$\gamma_l = \frac{1}{2l} d^2 + \frac{l}{2} |r\ d|^2$$



- Coincides with forms obtained in gamma-convergent regularizations of free-discontinuity problems (Ambrosio and Tortorelli, 1990).

- Minimization principle of diffusive crack topology

$$d(x) = \arg \left\{ \inf_{d \in W} \Gamma_l(d) \right\} \qquad W = \{ d | d(x) = 1 \text{ at } x \in \Gamma \}$$

- Gives rise to Euler-Lagrange equation:

$$d - l^2 \Delta d = 0 \quad \text{in } \Omega \qquad r\ d \cdot n = 0 \quad \text{on } \partial\Omega$$

- Standard total potential energy (elastostatics)

$$\Pi_{pot}(u, \Gamma) = \int_{\Omega} \psi_e(\varepsilon(u))\, dv + \int_{\Gamma} G_c\, da$$

- Phase field regularization

$$\Pi_l(u, d) = \int_{\Omega} \left(g(d)\, \psi_e^+(\varepsilon(u)) + \psi_e^-(\varepsilon(u)) \right) dv + \int_{\Gamma} G_c \gamma_l\, da$$

- Degradation function requirements

$$g : [0, 1] \to [0, 1]$$
$$g'(d) < 0 \quad d \in [0, 1]$$
$$g'(1) = 0$$

$$g = (1 - d)^2$$

- Stress-strain response

$$\sigma(x,t) = (1 - d(x,t))C\varepsilon(x,t)$$

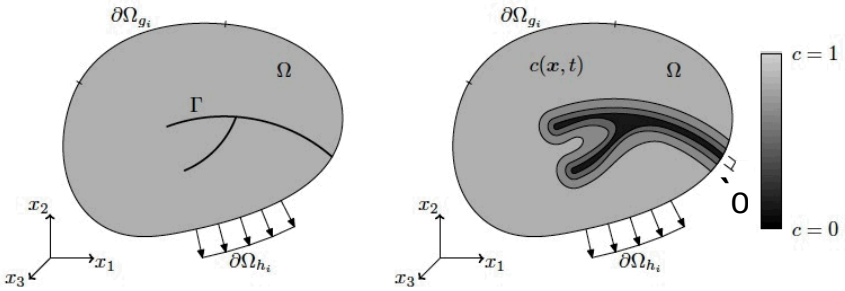
$$\sigma(x,t) = [(1 - d(x,t))^2 + k]C\varepsilon^+(x,t) + C\varepsilon^-(x,t)$$

- Non-local equations

$$\tilde{Y}(x,t) - c\Delta\tilde{Y}(x,t) = Y(x,t) \quad \text{in } \Omega$$

$$d\left[\frac{2lH(\varepsilon^+)}{G_c} + 1\right] - l^2\Delta d = \frac{2lH(\varepsilon^+)}{G_c}$$

- Total potential energy



$$\Psi_{pot}(u,\Gamma) = \int_{\Omega} \psi_e(\varepsilon^s u)\, dv + \int_{\Gamma} G_c\, da$$

- Approximate the fracture energy with a crack density functional (Miehe et al. 2010)

$$\int_{\Gamma} G_c\, da \approx \int_{\Omega} G_c \gamma_{c,n}\, dv \qquad \gamma_{c,n} = \frac{1}{4\ell_0}\left[(c-1)^2 + 4\ell_0^2 |\nabla c|^2\right]$$

- Implementation based on Borden et al. (2012)

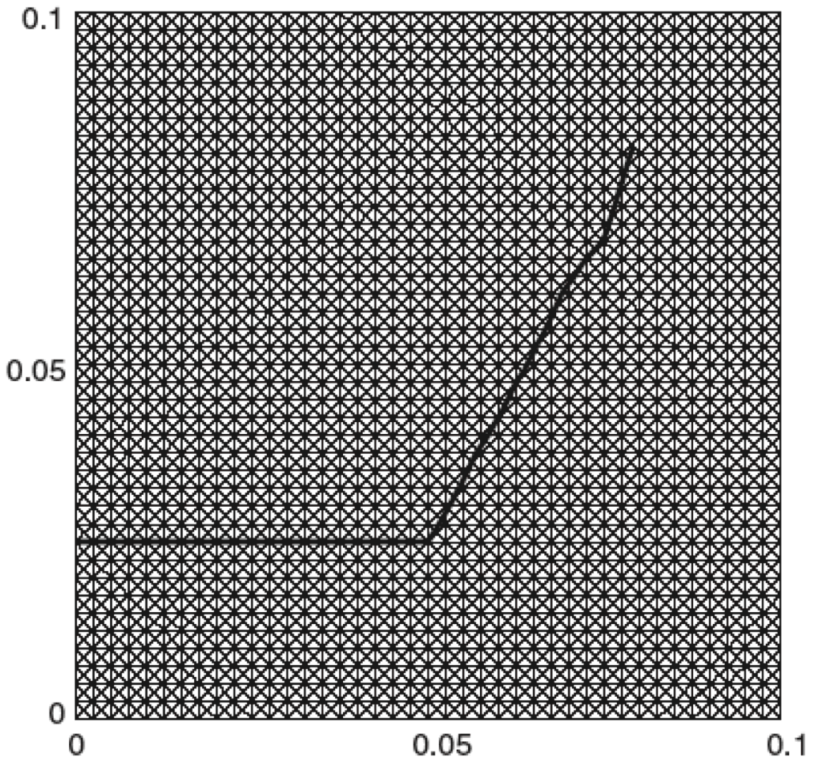
$$c = 1 - d$$

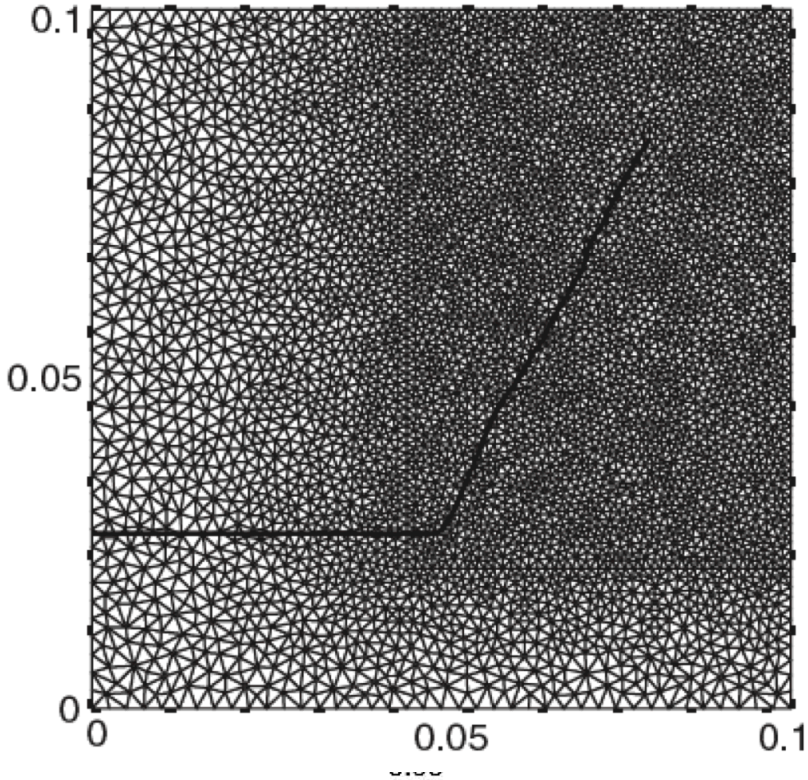
- Strong form for dynamics:

$$\frac{\partial \sigma_{ij}}{\partial x_j} = \rho \ddot{u}_i \quad \text{on } \Omega \times]0, T[$$

$$\left(\frac{4 \ell_0 (1-k) \mathcal{H}^+}{G_c} + 1 \right) c - 4 \ell_0^2 \frac{\partial^2 c}{\partial x_i^2} = 1 \quad \text{on } \Omega \times]0, T[$$

$$\sigma_{ij} = [(1-k)c^2 + k] \frac{\partial \psi_e^+}{\partial \varepsilon_{ij}} + \frac{\partial \psi_e^-}{\partial \varepsilon_{ij}}$$


0.1
0.05
0
0
0.05
0.1


0.1
0.05
0
0
0.05
0.1


0.04

Kalthoff Problem