Fracture Propagation

Matrix-Free Implementation

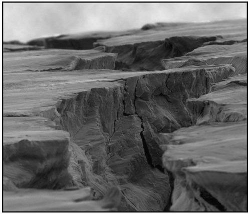
D. Jodlbauer

Johann Radon Institute for Computational and Applied Mathematics (RICAM)
Austrian Academy of Sciences (ÖAW)
Linz, Austria

August 7, 2018

Fractures

- Determine propagation of fractures
- Compute resulting stresses / forces / ...
- $\rightarrow$ require model & solver



Energy Minimization

- Systems tend to a state of minimal energy

  $\Rightarrow$ define fracture energy (Griffith[1])

  $\Rightarrow$ compute minimizer

- Solid:

  $$E_S(u) = \frac{1}{2}(\sigma(u), e(u))$$

- Fracture:

  $$E_C(\mathcal{C}) = G_c \mathcal{H}^{d-1}(\mathcal{C})$$

$\sigma, e \ldots$ stress / strain tensors, displacement u

$\mathcal{C} \ldots$ fracture, Hausdorff-measure $\mathcal{H}$, G_c = const, fracture energy release rate

[1] Griffith, "The phenomena of rupture and flow in solids", 1921.

Energy Minimization

- Solid:

  $E_S(u) = \frac{1}{2}(\sigma(u), e(u))$

- Fracture:

  $E_C(\mathcal{C}) = G_c \mathcal{H}^{d-1}(\mathcal{C})$

Some problems arise:

- $\mathcal{C}$ is a lower-dimensional object
- Handling $\mathcal{H}^{d-1}(\mathcal{C})$ is difficult in FEM

Solution:

- Extend $(d-1)$ fracture to d-dimensional "fracture"
- Approximate Hausdorff measure

Fracture Model - Phase-Field Approach

- Represent fracture via indicator function

$$\varphi = I_C = \begin{cases} 0 & \text{damaged material} \\ 1 & \text{undamaged material} \end{cases}$$

- Still discontinuous, $(d-1)$-dimensional function

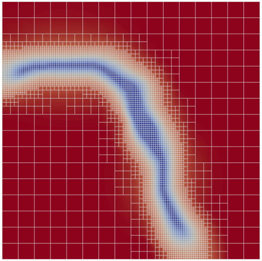
  $\rightarrow$ make continuous

- Extend $(d-1)$ fracture to d-dimensional object

Fracture Model - Phase-Field Approach

- Represent fracture via function

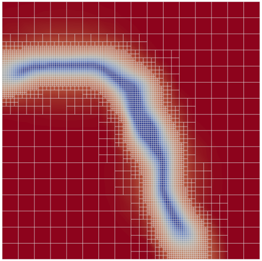
$$\varphi = \begin{cases} 0 & \text{damaged material} \\ ? & \text{partly/possibly damaged} \\ 1 & \text{undamaged material} \end{cases}$$



Fracture Model - Phase-Field Approach[2]

- Energy approximation

$$E(\varphi) = G_c \mathcal{H}^{d-1}(\varphi)$$
$$\rightsquigarrow G_c\left(\frac{1}{2\epsilon}\|1-\varphi\|^2 + \frac{\epsilon}{2}\|\nabla\varphi\|^2\right)$$



- Gamma-convergence as $\epsilon \to 0$

[2] Francfort and Marigo, "Revisiting brittle fracture as an energy minimization problem", 1998.

Energy Minimization

- Total energy

$$E(u,\varphi) = E_S(u,\varphi) + E_C(\varphi) \to \min$$

with

$$E_S(u,\varphi) = \frac{1}{2}(\varphi^2\sigma(u), e(u))$$

$$E_C(\varphi) = G_c\left(\frac{1}{2\epsilon}\|1-\varphi\|^2 + \frac{\epsilon}{2}\|\nabla\varphi\|^2\right)$$

Energy Minimization

Problem

Find displacement u *and fracture* φ *such that*

$$E(u,\varphi) = E_S(u,\varphi) + E_C(\varphi) \to min$$

subject to $\partial_t \varphi \leq 0$ *(crack irreversibility).*

Some "features":

- Non-linear, non-convex (indefinite) optimization problem
- Variational inequality

Considerations

- Convex wrt. u and φ separately
  $\rightarrow$ partitioned solution methods
- Non-convexity (caused by φ^2-terms) leads to problems with convergence (e.g. Active-Set)
- Avoid by extrapolation in time: replace $\varphi^2 \rightsquigarrow \tilde{\varphi}^2$ in ∇E

Extrapolation

- Predict φ using previous values
- Caveat: time-lagging, difficult to obtain fracture velocity
- "Hessian" no-longer symmetric (since $\partial_\varphi \tilde{\varphi}^2 = 0$)
- However: modifies problem & no regularity in time (yet)
- But: convex, (better) convergence for non-linear solvers

(Modified) Active Set Method[3]

- Assemble residual $R(U_h^k)$
- Determine active set: $A^n := \{\text{dof } i : (M^{-1}R)_i + c(\varphi_h^k - \varphi_h^{n-1})_i > 0\}$
- Enforce constraints: $\varphi^n = \varphi^{n-1}, \ \delta\varphi_h^k = 0$ on A^n
- Compute (reduced) Newton update $\nabla^2 E \cdot \delta U = -\nabla E$
- Repeat until convergence ($A^n = A^{n-1}$ and Newton criterion)

[3] Heister, Wheeler, and Wick, "A primal-dual active set method and predictor-corrector mesh adaptivity for computing fracture propagation using a phase-field approach", 2015.

Solving the Linear Systems

- Assembling & solving take most of the time
- Have to assemble/rebuild AMG/LU quite frequently, i.e. when
  - Mesh refinement
  - Active set changes
  - Linearization point changes
- Can we avoid it?

Matrix-Free Formulation

Motivation:

- Iterative solvers need only matrix-vector products
- Instead of assembling A and compute $A \cdot v$, assemble Av directly (ie. cell-wise matrix-vector product)

Matrix-Free Formulation

$$\Rightarrow A \cdot v = \sum_C P_C^T A_C P_C \cdot v = \sum_C P_C^T A_C \cdot v_C$$

$$= \sum_C P_C^T B_C^T D_C B_C v_C = \sum_C P_C^T B_R^T J_C^{-1} D_C J_C^{-T} B_R v_C$$

P_C . . . local to global mapping

B_R . . . gradient matrix on reference cell

J_C . . . cell transformation (block-diagonal)

D_C . . . diagonal coefficient matrix

Matrix-Free Formulation

Benefits:

- No need to store matrix A
- Automatic "rebuild" on every "multiplication"

Not so good:

- Matrix-free
- Restrictions on solver (GMG and ?)
- Speed? Surely it must be slower . . .

Speed

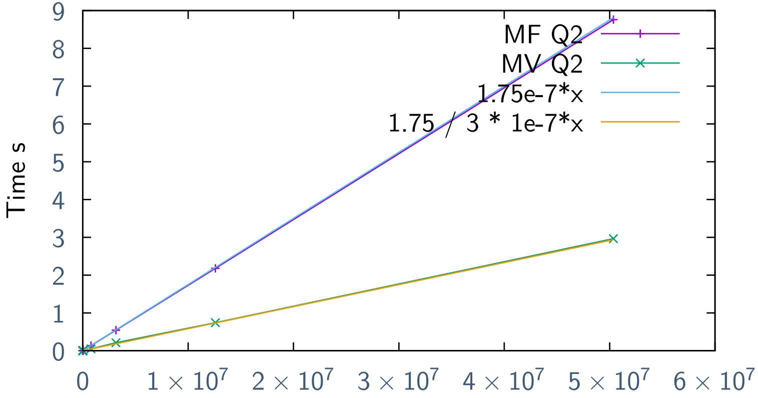
- Computational work per dof for vmult

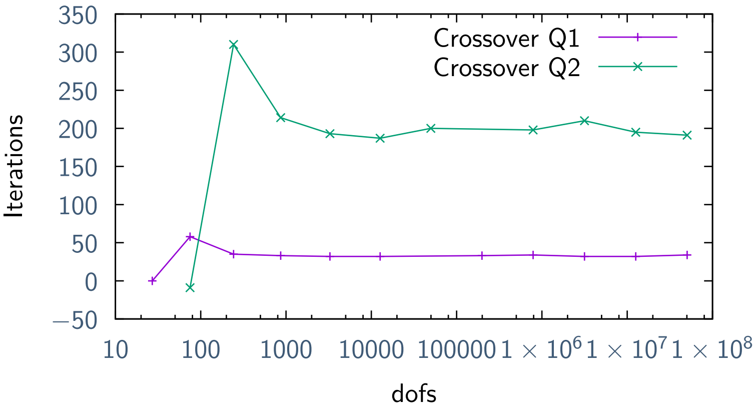
  $4d^2p$ for MF vs $2p^d$ for MV (without assembling!)

- For higher p, MF is even faster!
- Some claim that $p > 1$ not beneficial for PFF ...
- For large problems with $p = 1$, MV runs out of L3

  $\rightarrow$ MF can be faster as well (depends on hardware)

- Otherwise: we still save time from not assembling $\rightarrow$ crossover

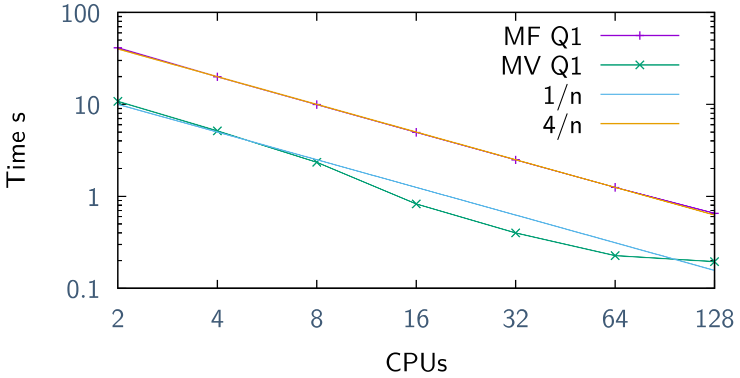

Time s
0
1
2
3
4
5
6
7
8
9
0
1 × 10^7
2 × 10^7
3 × 10^7
4 × 10^7
5 × 10^7
6 × 10^7
MF Q2
MV Q2
1.75e-7*x
1.75 / 3 * 1e-7*x



So far, everything was 2d. In 3d: crossover = 20k for 1m dofs using Q2.

Parallelization - Perfect Scalability

$\approx 200m$ dofs in $2d$ using $Q1$ elements



Matrix-Free Geometric Multigrid

- Matrix-free operator on each level
- Several options:
  - block-diagonal MINRES(GMG) preconditioner (for GMRES)
  - monolithic GMG (preconditioner for GMRES)
  - non-linear GMG (future work)
- Smoother: block-wise Chebyshev
- $4 - 10$ iterations (no timings yet . . .)

Further Considerations

- Require current linearization point on each level
- Using restricted linearization points does not seem to work (so far)
- Require active set on each level
  - all coarse dofs close enought to active set
  - other options

Summary

- Matrix-free implementation with (block-wise) GMG & Active-Set solver
- Less memory requirements
- Potentially faster for high order
- Better parallel scalability

Future Work

- Test / Time current solvers
- Adaptivity & parallelization
- Higher-order, dG
- Non-linear GMG or
- Avoid/improve coarse active set/linearization points
- Convergence studies, choice of ϵ, ...

Thank you.

Francfort, G. and J.-J. Marigo. "Revisiting brittle fracture as an energy minimization problem". In: *J. Mech. Phys. Solids* 46.8 (1998), pp. 1319–1342.

Griffith, A. "The phenomena of rupture and flow in solids". In: *Philos. Trans. R. Soc. London* 221 (1921), pp. 163–198.

Heister, T., M. F. Wheeler, and T. Wick. "A primal-dual active set method and predictor-corrector mesh adaptivity for computing fracture propagation using a phase-field approach". In: *Computer methods in applied mechanics and engineering* 290 (2015), pp. 466–495.