Poroelastic fracturing

T. Kvamsdal[1,2] E. Fonn[2] A. M. Kvarving[2] K. M. Okstad[2]
K. Johannessen[2]

[1]Department of Mathematical Sciences, NTNU

[2]Applied Mathematics and Cybernetics, SINTEF Digital

NTNU SINTEF

Physical model

Poroelasticity

- Fundamental conservation of momentum

$$\nabla \cdot \boldsymbol{\sigma}^{\mathrm{t}} + \rho \boldsymbol{f}_{\mathrm{b}} = \boldsymbol{0}$$

- Darcy flow: contribution to stress

$$\boldsymbol{\sigma}^{\mathrm{t}} = \boldsymbol{\sigma}^{\mathrm{e}} + \alpha p \boldsymbol{I}$$

- Darcy flow: mass balance equation

$$\alpha \nabla \cdot \dot{\boldsymbol{u}} + \frac{1}{M}\dot{p} - \nabla \cdot [\boldsymbol{\kappa} \cdot (\nabla p - \rho_{\mathrm{f}} \boldsymbol{f}_{\mathrm{b}})] = q_{\mathrm{b}}$$

Poroelasticity

- Fundamental conservation of momentum

$$\nabla \cdot \boldsymbol{\sigma}^{\mathrm{t}} + \rho \boldsymbol{f}_{\mathrm{b}} = \mathbf{0}$$

- Darcy flow: contribution to stress

$$\boldsymbol{\sigma}^{\mathrm{t}} = \boldsymbol{\sigma}^{\mathrm{e}} + \alpha p \boldsymbol{I}$$

- Darcy flow: mass balance equation

$$\alpha \nabla \cdot \dot{\boldsymbol{u}} + \frac{1}{M}\dot{p} - \nabla \cdot [\boldsymbol{\kappa} \cdot (\nabla p - \rho_{\mathrm{f}} \boldsymbol{f}_{\mathrm{b}})] = q_{\mathrm{b}}$$

$\boldsymbol{\sigma}^{\mathrm{t}}$: total stress

Poroelasticity

- Fundamental conservation of momentum

$$\nabla \cdot \boldsymbol{\sigma}^{\mathrm{t}} + \rho \boldsymbol{f}_{\mathrm{b}} = \mathbf{0}$$

- Darcy flow: contribution to stress

$$\boldsymbol{\sigma}^{\mathrm{t}} = \boldsymbol{\sigma}^{\mathrm{e}} + \alpha p \boldsymbol{I}$$

- Darcy flow: mass balance equation

$$\alpha \nabla \cdot \dot{\boldsymbol{u}} + \frac{1}{M} \dot{p} - \nabla \cdot [\boldsymbol{\kappa} \cdot (\nabla p - \rho_{\mathrm{f}} \boldsymbol{f}_{\mathrm{b}})] = q_{\mathrm{b}}$$

ρ: total mass density

Poroelasticity

- Fundamental conservation of momentum

$$\nabla \cdot \boldsymbol{\sigma}^{\mathrm{t}} + \rho \boldsymbol{f}_{\mathrm{b}} = \mathbf{0}$$

- Darcy flow: contribution to stress

$$\boldsymbol{\sigma}^{\mathrm{t}} = \boldsymbol{\sigma}^{\mathrm{e}} + \alpha p \boldsymbol{I}$$

- Darcy flow: mass balance equation

$$\alpha \nabla \cdot \dot{\boldsymbol{u}} + \frac{1}{M}\dot{p} - \nabla \cdot [\boldsymbol{\kappa} \cdot (\nabla p - \rho_{\mathrm{f}} \boldsymbol{f}_{\mathrm{b}})] = q_{\mathrm{b}}$$

$\boldsymbol{f}_{\mathrm{b}}$: body forces

Poroelasticity

- Fundamental conservation of momentum

$$\nabla \cdot \boldsymbol{\sigma}^{\mathrm{t}} + \rho \boldsymbol{f}_{\mathrm{b}} = \mathbf{0}$$

- Darcy flow: contribution to stress

$$\boldsymbol{\sigma}^{\mathrm{t}} = \boldsymbol{\sigma}^{\mathrm{e}} + \alpha p \boldsymbol{I}$$

- Darcy flow: mass balance equation

$$\alpha \nabla \cdot \dot{\boldsymbol{u}} + \frac{1}{M}\dot{p} - \nabla \cdot [\boldsymbol{\kappa} \cdot (\nabla p - \rho_{\mathrm{f}} \boldsymbol{f}_{\mathrm{b}})] = q_{\mathrm{b}}$$

Poroelasticity

- Fundamental conservation of momentum

$$\nabla \cdot \boldsymbol{\sigma}^{\mathrm{t}} + \rho \boldsymbol{f}_{\mathrm{b}} = \mathbf{0}$$

- Darcy flow: contribution to stress

$$\boldsymbol{\sigma}^{\mathrm{t}} = \boldsymbol{\sigma}^{\mathrm{e}} + \alpha p \boldsymbol{I}$$

- Darcy flow: mass balance equation

$$\alpha \nabla \cdot \dot{\boldsymbol{u}} + \frac{1}{M} \dot{p} - \nabla \cdot [\boldsymbol{\kappa} \cdot (\nabla p - \rho_{\mathrm{f}} \boldsymbol{f}_{\mathrm{b}})] = q_{\mathrm{b}}$$

$\boldsymbol{\sigma}^{\mathrm{e}} = \boldsymbol{\sigma}(\boldsymbol{\varepsilon})$: effective stress

Poroelasticity

- Fundamental conservation of momentum

$$\nabla \cdot \boldsymbol{\sigma}^{\mathrm{t}} + \rho \boldsymbol{f}_{\mathrm{b}} = \mathbf{0}$$

- Darcy flow: contribution to stress

$$\boldsymbol{\sigma}^{\mathrm{t}} = \boldsymbol{\sigma}^{\mathrm{e}} + \alpha p \boldsymbol{I}$$

- Darcy flow: mass balance equation

$$\alpha \nabla \cdot \dot{\boldsymbol{u}} + \frac{1}{M}\dot{p} - \nabla \cdot [\boldsymbol{\kappa} \cdot (\nabla p - \rho_{\mathrm{f}} \boldsymbol{f}_{\mathrm{b}})] = q_{\mathrm{b}}$$

α: Biot's coefficient (typically $\alpha = 1$)

Poroelasticity

- Fundamental conservation of momentum

$$\nabla \cdot \boldsymbol{\sigma}^{\mathrm{t}} + \rho \boldsymbol{f}_{\mathrm{b}} = \mathbf{0}$$

- Darcy flow: contribution to stress

$$\boldsymbol{\sigma}^{\mathrm{t}} = \boldsymbol{\sigma}^{\mathrm{e}} + \alpha p \boldsymbol{I}$$

- Darcy flow: mass balance equation

$$\alpha \nabla \cdot \dot{\boldsymbol{u}} + \frac{1}{M} \dot{p} - \nabla \cdot [\boldsymbol{\kappa} \cdot (\nabla p - \rho_{\mathrm{f}} \boldsymbol{f}_{\mathrm{b}})] = q_{\mathrm{b}}$$

p: pressure of fluid in porous material — a primary unknown

Poroelasticity

- Fundamental conservation of momentum

$$\nabla \cdot \boldsymbol{\sigma}^{\mathrm{t}} + \rho \boldsymbol{f}_{\mathrm{b}} = \mathbf{0}$$

- Darcy flow: contribution to stress

$$\boldsymbol{\sigma}^{\mathrm{t}} = \boldsymbol{\sigma}^{\mathrm{e}} + \alpha p \boldsymbol{I}$$

- Darcy flow: mass balance equation

$$\alpha \nabla \cdot \dot{\boldsymbol{u}} + \frac{1}{M}\dot{p} - \nabla \cdot [\boldsymbol{\kappa} \cdot (\nabla p - \rho_{\mathrm{f}} \boldsymbol{f}_{\mathrm{b}})] = q_{\mathrm{b}}$$

Poroelasticity

- Fundamental conservation of momentum

$$\nabla \cdot \boldsymbol{\sigma}^{\mathrm{t}} + \rho \boldsymbol{f}_{\mathrm{b}} = \boldsymbol{0}$$

- Darcy flow: contribution to stress

$$\boldsymbol{\sigma}^{\mathrm{t}} = \boldsymbol{\sigma}^{\mathrm{e}} + \alpha p \boldsymbol{I}$$

- Darcy flow: mass balance equation

$$\alpha \nabla \cdot \dot{\boldsymbol{u}} + \frac{1}{M} \dot{p} - \nabla \cdot [\boldsymbol{\kappa} \cdot (\nabla p - \rho_{\mathrm{f}} \boldsymbol{f}_{\mathrm{b}})] = q_{\mathrm{b}}$$

$\boldsymbol{u}$: displacement vector field — a primary unknown

Poroelasticity

- Fundamental conservation of momentum

$$\nabla \cdot \boldsymbol{\sigma}^{\mathrm{t}} + \rho \boldsymbol{f}_{\mathrm{b}} = \mathbf{0}$$

- Darcy flow: contribution to stress

$$\boldsymbol{\sigma}^{\mathrm{t}} = \boldsymbol{\sigma}^{\mathrm{e}} + \alpha p \boldsymbol{I}$$

- Darcy flow: mass balance equation

$$\alpha \nabla \cdot \dot{\boldsymbol{u}} + \frac{1}{M} \dot{p} - \nabla \cdot [\boldsymbol{\kappa} \cdot (\nabla p - \rho_{\mathrm{f}} \boldsymbol{f}_{\mathrm{b}})] = q_{\mathrm{b}}$$

$1/M$: specific storage coefficient, a measure of compressibility of fluid

Poroelasticity

- Fundamental conservation of momentum

$$\nabla \cdot \boldsymbol{\sigma}^{\mathrm{t}} + \rho \boldsymbol{f}_{\mathrm{b}} = \boldsymbol{0}$$

- Darcy flow: contribution to stress

$$\boldsymbol{\sigma}^{\mathrm{t}} = \boldsymbol{\sigma}^{\mathrm{e}} + \alpha p \boldsymbol{I}$$

- Darcy flow: mass balance equation

$$\alpha \nabla \cdot \dot{\boldsymbol{u}} + \frac{1}{M} \dot{p} - \nabla \cdot [\boldsymbol{\kappa} \cdot (\nabla p - \rho_{\mathrm{f}} \boldsymbol{f}_{\mathrm{b}})] = q_{\mathrm{b}}$$

$$\frac{1}{M} = \frac{\alpha - n}{K_{\mathrm{s}}} + \frac{n}{K_{\mathrm{f}}}$$

Poroelasticity

- Fundamental conservation of momentum

$$\nabla \cdot \boldsymbol{\sigma}^{\mathrm{t}} + \rho \boldsymbol{f}_{\mathrm{b}} = \boldsymbol{0}$$

- Darcy flow: contribution to stress

$$\boldsymbol{\sigma}^{\mathrm{t}} = \boldsymbol{\sigma}^{\mathrm{e}} + \alpha p \boldsymbol{I}$$

- Darcy flow: mass balance equation

$$\alpha \nabla \cdot \dot{\boldsymbol{u}} + \frac{1}{M} \dot{p} - \nabla \cdot [\boldsymbol{\kappa} \cdot (\nabla p - \rho_{\mathrm{f}} \boldsymbol{f}_{\mathrm{b}})] = q_{\mathrm{b}}$$

$\boldsymbol{\kappa}$: permeability tensor field

Poroelasticity

- Fundamental conservation of momentum

$$\nabla \cdot \boldsymbol{\sigma}^{\mathrm{t}} + \rho \boldsymbol{f}_{\mathrm{b}} = \mathbf{0}$$

- Darcy flow: contribution to stress

$$\boldsymbol{\sigma}^{\mathrm{t}} = \boldsymbol{\sigma}^{\mathrm{e}} + \alpha p \boldsymbol{I}$$

- Darcy flow: mass balance equation

$$\alpha \nabla \cdot \dot{\boldsymbol{u}} + \frac{1}{M} \dot{p} - \nabla \cdot [\boldsymbol{\kappa} \cdot (\nabla p - \rho_{\mathrm{f}} \boldsymbol{f}_{\mathrm{b}})] = q_{\mathrm{b}}$$

ρ_{f}: fluid density

Poroelasticity

- Fundamental conservation of momentum

$$\nabla \cdot \boldsymbol{\sigma}^{\mathrm{t}} + \rho \boldsymbol{f}_{\mathrm{b}} = \mathbf{0}$$

- Darcy flow: contribution to stress

$$\boldsymbol{\sigma}^{\mathrm{t}} = \boldsymbol{\sigma}^{\mathrm{e}} + \alpha p \boldsymbol{I}$$

- Darcy flow: mass balance equation

$$\alpha \nabla \cdot \dot{\boldsymbol{u}} + \frac{1}{M}\dot{p} - \nabla \cdot [\boldsymbol{\kappa} \cdot (\nabla p - \rho_{\mathrm{f}} \boldsymbol{f}_{\mathrm{b}})] = q_{\mathrm{b}}$$

q_{b}: fluid sources and sinks

Poroelasticity

System of equations resulting from variational formulation

$$\begin{aligned} \boldsymbol{u}_i, \delta\boldsymbol{u}_i &\in H^1(\Omega)^d \\ p_i, \delta p_i &\in \left\{ p \in L^2(\Omega) \;\middle|\; \int_\Omega p = 0 \right\} \end{aligned}$$

$$\begin{pmatrix} & \\ \boldsymbol{Q}^\intercal & \boldsymbol{S} \end{pmatrix} \begin{pmatrix} \dot{\boldsymbol{u}} \\ \dot{\boldsymbol{p}} \end{pmatrix} + \begin{pmatrix} \boldsymbol{K} & -\boldsymbol{Q} \\ & \boldsymbol{P} \end{pmatrix} \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{p} \end{pmatrix} = \begin{pmatrix} \boldsymbol{f}_u \\ \boldsymbol{f}_p \end{pmatrix}$$

Poroelasticity

System of equations resulting from variational formulation

$$\boldsymbol{u}_i, \delta\boldsymbol{u}_i \in H^1(\Omega)^d$$
$$p_i, \delta p_i \in \left\{ p \in L^2(\Omega) \;\middle|\; \int_\Omega p = 0 \right\}$$

$$\begin{pmatrix} & \\ \boldsymbol{Q}^\intercal & \boldsymbol{S} \end{pmatrix} \begin{pmatrix} \dot{\boldsymbol{u}} \\ \dot{\boldsymbol{p}} \end{pmatrix} + \begin{pmatrix} \boldsymbol{K} & -\boldsymbol{Q} \\ & \boldsymbol{P} \end{pmatrix} \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{p} \end{pmatrix} = \begin{pmatrix} \boldsymbol{f}_u \\ \boldsymbol{f}_p \end{pmatrix}$$

The coupling matrix

$$\boldsymbol{Q}_{ij} = \alpha \int_\Omega \nabla\delta\boldsymbol{u}_j : p_i \boldsymbol{I}$$

Poroelasticity

System of equations resulting from variational formulation

$$\boldsymbol{u}_i, \delta\boldsymbol{u}_i \in H^1(\Omega)^d$$

$$p_i, \delta p_i \in \left\{ p \in L^2(\Omega) \;\middle|\; \int_\Omega p = 0 \right\}$$

$$\begin{pmatrix} & \\ \boldsymbol{Q}^\intercal & \boldsymbol{S} \end{pmatrix} \begin{pmatrix} \dot{\boldsymbol{u}} \\ \dot{\boldsymbol{p}} \end{pmatrix} + \begin{pmatrix} \boldsymbol{K} & -\boldsymbol{Q} \\ & \boldsymbol{P} \end{pmatrix} \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{p} \end{pmatrix} = \begin{pmatrix} \boldsymbol{f}_u \\ \boldsymbol{f}_p \end{pmatrix}$$

The storativity matrix

$$\boldsymbol{S}_{ij} = \int_\Omega c\,\delta p_i p_j$$

Poroelasticity

System of equations resulting from variational formulation

$$\boldsymbol{u}_i, \delta\boldsymbol{u}_i \in H^1(\Omega)^d$$

$$p_i, \delta p_i \in \left\{ p \in L^2(\Omega) \;\middle|\; \int_\Omega p = 0 \right\}$$

$$\begin{pmatrix} & \\ \boldsymbol{Q}^\intercal & \boldsymbol{S} \end{pmatrix} \begin{pmatrix} \dot{\boldsymbol{u}} \\ \dot{\boldsymbol{p}} \end{pmatrix} + \begin{pmatrix} \boldsymbol{K} & -\boldsymbol{Q} \\ & \boldsymbol{P} \end{pmatrix} \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{p} \end{pmatrix} = \begin{pmatrix} \boldsymbol{f}_u \\ \boldsymbol{f}_p \end{pmatrix}$$

The permeability matrix

$$\boldsymbol{P}_{ij} = \int_\Omega \nabla \delta p_i^\intercal \boldsymbol{\kappa} \nabla p_j$$

Poroelasticity

System of equations resulting from variational formulation

$$\boldsymbol{u}_i, \delta\boldsymbol{u}_i \in H^1(\Omega)^d$$

$$p_i, \delta p_i \in \left\{ p \in L^2(\Omega) \;\middle|\; \int_\Omega p = 0 \right\}$$

$$\begin{pmatrix} & \\ \boldsymbol{Q}^\intercal & \boldsymbol{S} \end{pmatrix} \begin{pmatrix} \dot{\boldsymbol{u}} \\ \dot{\boldsymbol{p}} \end{pmatrix} + \begin{pmatrix} \boldsymbol{K} & -\boldsymbol{Q} \\ & \boldsymbol{P} \end{pmatrix} \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{p} \end{pmatrix} = \begin{pmatrix} \boldsymbol{f}_u \\ \boldsymbol{f}_p \end{pmatrix}$$

The stiffness matrix

$$\boldsymbol{K}_{ij} = \int_\Omega \boldsymbol{\varepsilon}(\delta\boldsymbol{u}_i) : \boldsymbol{D}\boldsymbol{\varepsilon}(\boldsymbol{u}_j)$$

Poroelasticity

System of equations resulting from variational formulation

$$\boldsymbol{u}_i, \delta \boldsymbol{u}_i \in H^1(\Omega)^d$$

$$p_i, \delta p_i \in \left\{ p \in L^2(\Omega) \;\middle|\; \int_\Omega p = 0 \right\}$$

$$\begin{pmatrix} & \\ \boldsymbol{Q}^\top & \boldsymbol{S} \end{pmatrix} \begin{pmatrix} \dot{\boldsymbol{u}} \\ \dot{\boldsymbol{p}} \end{pmatrix} + \begin{pmatrix} \boldsymbol{K} & -\boldsymbol{Q} \\ & \boldsymbol{P} \end{pmatrix} \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{p} \end{pmatrix} = \begin{pmatrix} \boldsymbol{f}_u \\ \boldsymbol{f}_p \end{pmatrix}$$

Momentum load

$$(\boldsymbol{f}_u)_i = \int_\Omega \delta \boldsymbol{u}_i \cdot \rho \boldsymbol{f}_{\mathrm{b}} + \int_{\Gamma_{\mathrm{n}}} \delta \boldsymbol{u}_i \cdot \overline{\boldsymbol{t}}$$

Poroelasticity

System of equations resulting from variational formulation

$$\boldsymbol{u}_i, \delta\boldsymbol{u}_i \in H^1(\Omega)^d$$

$$p_i, \delta p_i \in \left\{ p \in L^2(\Omega) \;\middle|\; \int_\Omega p = 0 \right\}$$

$$\begin{pmatrix} & \\ \boldsymbol{Q}^\top & \boldsymbol{S} \end{pmatrix} \begin{pmatrix} \dot{\boldsymbol{u}} \\ \dot{\boldsymbol{p}} \end{pmatrix} + \begin{pmatrix} \boldsymbol{K} & -\boldsymbol{Q} \\ & \boldsymbol{P} \end{pmatrix} \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{p} \end{pmatrix} = \begin{pmatrix} \boldsymbol{f}_u \\ \boldsymbol{f}_p \end{pmatrix}$$

Flux load

$$(\boldsymbol{f}_p)_i = \int_\Omega \delta\boldsymbol{p}_i q_\mathrm{b}$$

Poroelasticity

- Note that $\boldsymbol{K}$ and $\boldsymbol{f}_u$ are “standard” elasticity system matrices
- This serves as a convenient “plugging point” for substituting different elasticity models in the same Darcy flow interpretation
- E.g. dynamic elasticity

$$\begin{pmatrix} \boldsymbol{M} & \\ & \end{pmatrix} \begin{pmatrix} \ddot{\boldsymbol{u}} \\ \ddot{\boldsymbol{p}} \end{pmatrix} + \begin{pmatrix} \boldsymbol{C} & \\ \boldsymbol{Q}^\top & \boldsymbol{S} \end{pmatrix} \begin{pmatrix} \dot{\boldsymbol{u}} \\ \dot{\boldsymbol{p}} \end{pmatrix} + \begin{pmatrix} \boldsymbol{K} & -\boldsymbol{Q} \\ & \boldsymbol{P} \end{pmatrix} \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{p} \end{pmatrix} = \begin{pmatrix} \boldsymbol{f}_u \\ \boldsymbol{f}_p \end{pmatrix}$$

Poroelasticity

- Note that $\boldsymbol{K}$ and $\boldsymbol{f}_u$ are "standard" elasticity system matrices
- This serves as a convenient "plugging point" for substituting different elasticity models in the same Darcy flow interpretation
- E.g. dynamic elasticity

$$\begin{pmatrix} \boldsymbol{M} & \\ & \end{pmatrix} \begin{pmatrix} \ddot{\boldsymbol{u}} \\ \ddot{\boldsymbol{p}} \end{pmatrix} + \begin{pmatrix} \boldsymbol{C} & \\ \boldsymbol{Q}^\top & \boldsymbol{S} \end{pmatrix} \begin{pmatrix} \dot{\boldsymbol{u}} \\ \dot{\boldsymbol{p}} \end{pmatrix} + \begin{pmatrix} \boldsymbol{K} & -\boldsymbol{Q} \\ & \boldsymbol{P} \end{pmatrix} \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{p} \end{pmatrix} = \begin{pmatrix} \boldsymbol{f}_u \\ \boldsymbol{f}_p \end{pmatrix}$$

Poroelasticity

- Note that $\boldsymbol{K}$ and $\boldsymbol{f}_u$ are "standard" elasticity system matrices
- This serves as a convenient "plugging point" for substituting different elasticity models in the same Darcy flow interpretation
- E.g. dynamic elasticity

$$\begin{pmatrix} \boldsymbol{M} & \\ & \end{pmatrix} \begin{pmatrix} \ddot{\boldsymbol{u}} \\ \ddot{\boldsymbol{p}} \end{pmatrix} + \begin{pmatrix} \boldsymbol{C} & \\ \boldsymbol{Q}^\mathsf{T} & \boldsymbol{S} \end{pmatrix} \begin{pmatrix} \dot{\boldsymbol{u}} \\ \dot{\boldsymbol{p}} \end{pmatrix} + \begin{pmatrix} \boldsymbol{K} & -\boldsymbol{Q} \\ & \boldsymbol{P} \end{pmatrix} \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{p} \end{pmatrix} = \begin{pmatrix} \boldsymbol{f}_u \\ \boldsymbol{f}_p \end{pmatrix}$$

Poroelasticity

- Note that $\boldsymbol{K}$ and $\boldsymbol{f}_u$ are "standard" elasticity system matrices
- This serves as a convenient "plugging point" for substituting different elasticity models in the same Darcy flow interpretation
- E.g. dynamic elasticity

$$\begin{pmatrix} \boldsymbol{M} & \\ & \end{pmatrix} \begin{pmatrix} \ddot{\boldsymbol{u}} \\ \ddot{\boldsymbol{p}} \end{pmatrix} + \begin{pmatrix} \boldsymbol{C} & \\ \boldsymbol{Q}^\intercal & \boldsymbol{S} \end{pmatrix} \begin{pmatrix} \dot{\boldsymbol{u}} \\ \dot{\boldsymbol{p}} \end{pmatrix} + \begin{pmatrix} \boldsymbol{K} & -\boldsymbol{Q} \\ & \boldsymbol{P} \end{pmatrix} \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{p} \end{pmatrix} = \begin{pmatrix} \boldsymbol{f}_u \\ \boldsymbol{f}_p \end{pmatrix}$$

The mass matrix

$$M_{ij} = \int_\Omega \delta \boldsymbol{u}_i \cdot \boldsymbol{u}_j$$

Poroelasticity

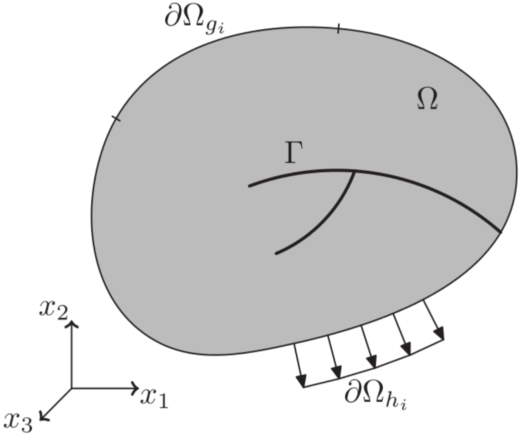
- Note that $\boldsymbol{K}$ and $\boldsymbol{f}_u$ are "standard" elasticity system matrices
- This serves as a convenient "plugging point" for substituting different elasticity models in the same Darcy flow interpretation
- E.g. dynamic elasticity

$$\begin{pmatrix} \boldsymbol{M} & \\ & \end{pmatrix} \begin{pmatrix} \ddot{\boldsymbol{u}} \\ \ddot{\boldsymbol{p}} \end{pmatrix} + \begin{pmatrix} \boldsymbol{C} & \\ \boldsymbol{Q}^\intercal & \boldsymbol{S} \end{pmatrix} \begin{pmatrix} \dot{\boldsymbol{u}} \\ \dot{\boldsymbol{p}} \end{pmatrix} + \begin{pmatrix} \boldsymbol{K} & -\boldsymbol{Q} \\ & \boldsymbol{P} \end{pmatrix} \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{p} \end{pmatrix} = \begin{pmatrix} \boldsymbol{f}_u \\ \boldsymbol{f}_p \end{pmatrix}$$

The damping matrix

$$\boldsymbol{C} = a\boldsymbol{M} + b\boldsymbol{K}$$

Domain with internal discontinuity



Energy functional for dynamic brittle fracture

$$\Psi(\boldsymbol{u}, \dot{\boldsymbol{u}}, \Gamma) = \int_\Omega \left(\frac{\rho}{2} \dot{\boldsymbol{u}} \cdot \dot{\boldsymbol{u}} - \psi_e(\boldsymbol{u}) \right) - \int_\Gamma \mathcal{G}_c$$

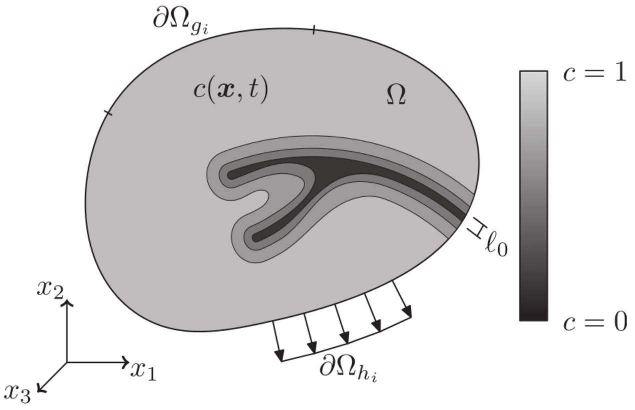
where

Γ is the unknown crack path

$\psi_e = \frac{1}{2}\lambda(\mathrm{tr}\,\varepsilon)^2 + \mu\,\mathrm{tr}(\varepsilon : \varepsilon)$ is the strain energy density function,
λ and μ are the Lamè material parameters,
and $\varepsilon(\boldsymbol{u})$ is the 2nd-order strain tensor

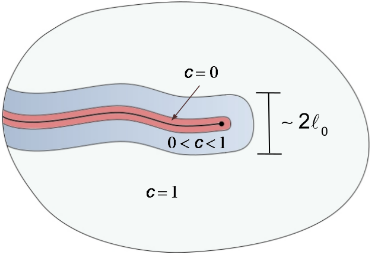
$\mathcal{G}_c$ is the fracture energy density

Phase-field approximation of the discontinuity



Phase-field model

- Resolves individual cracks down to some length scale
- Diffuse interface $\Rightarrow$ no interface tracking



$$\begin{cases} c = 1 & \Rightarrow \quad \text{undamaged material} \\ 0 < c < 1 & \Rightarrow \quad \text{damaged material} \\ c = 0 & \Rightarrow \quad \text{cracked material} \end{cases}$$

Energy functional for dynamic brittle fracture

- Approximation of the fracture energy:

$$\int_\Gamma \mathcal{G}_c \approx \begin{cases} \int_\Omega \mathcal{G}_c \left(\frac{(1-c)^2}{4\ell_0} + \ell_0 |\nabla c|^2 \right) & \text{2nd-order} \\ \int_\Omega \mathcal{G}_c \left(\frac{(1-c)^2}{4\ell_0} + \frac{\ell_0}{2} |\nabla c|^2 + \frac{\ell_0^3}{4} (\nabla^2 c)^2 \right) & \text{4th-order} \end{cases}$$

  where $c \in [0, 1]$ is the phase field parameter, and ℓ_0 is a chosen length scale defining the "thickness" of the damaged material (crack) zone.

- Split of elastic strain energy density into tensile and compressive parts

$$\psi_e(\boldsymbol{u}) = g(c)\psi^+(\boldsymbol{\varepsilon}) + \psi^-(\boldsymbol{\varepsilon})$$

  where $g(c)$ is a degradation function (typically chosen as c^2), and ψ^+ and ψ^- are tensile and compressive contributions, respectively

Small strain brittle fracture

- Strain tensor:

$$\varepsilon(\boldsymbol{u}) = \frac{1}{2}\left(\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T\right)$$

- Stress tensor:

$$\boldsymbol{\sigma}(\boldsymbol{u}) = \frac{\partial}{\partial \boldsymbol{\varepsilon}}\psi_e(\varepsilon) = g(c)\frac{\partial}{\partial \boldsymbol{\varepsilon}}\psi^+(\varepsilon) + \frac{\partial}{\partial \boldsymbol{\varepsilon}}\psi^-(\varepsilon)$$

- Minimizing $\Psi(\boldsymbol{u}, \dot{\boldsymbol{u}}, \Gamma) \approx \Psi(\boldsymbol{u}, \dot{\boldsymbol{u}}, c)$ with respect to $\boldsymbol{u}$ and c yields the strong form of the brittle crack problem:

$$\nabla \cdot \boldsymbol{\sigma}(\boldsymbol{u}) = \rho \ddot{\boldsymbol{u}} \qquad \text{linear momentum}$$

$$\frac{2\ell_0}{\mathcal{G}_c} g'(c)\psi^+ + c - 4\ell_0^2 \nabla^2 c = 1 \qquad \text{phase-field (2nd-order)}$$

$$\frac{2\ell_0}{\mathcal{G}_c} g'(c)\psi^+ + c - 2\ell_0^2 \nabla^2 c + \ell_0^4 \nabla^4 c = 1 \qquad \text{Phase-field (4th-order)}$$

on $\Omega \times]0, T]$.

Strain history field

To ensure that the developed crack does not close again, i.e., $\Gamma(t) \subset \Gamma(t+\Delta t) \ \ \forall \ \Delta t > 0$, the tensile energy density ψ^+ in the phase-field equation is replaced by a history field $\mathcal{H}(\boldsymbol{x}, t)$, satisfying $\mathcal{H} \geq \psi^+$, $\dot{\mathcal{H}} \geq 0$ and $\dot{\mathcal{H}}(\mathcal{H} - \psi^+) = 0$. Thus

$$\frac{2\ell_0}{\mathcal{G}_c} g'(c)\mathcal{H} + c - 4\ell_0^2 \nabla^2 c = 1 \qquad \text{(2nd-order)}$$

$$\frac{2\ell_0}{\mathcal{G}_c} g'(c)\mathcal{H} + c - 2\ell_0^2 \nabla^2 c + \ell_0^4 \nabla^4 c = 1 \qquad \text{(4th-order)}$$

Boundary and initial conditions

$$\begin{array}{rclll} u_\alpha & = & g_\alpha & \text{on} \quad \partial\Omega_{g_\alpha} \times [0,T] & : \text{Dirichlet condition on } u_\alpha \\ \sigma_{\alpha\beta} n_\beta & = & h_\alpha & \text{on} \quad \partial\Omega_{h_\alpha} \times [0,T] & : \text{Neumann condition on the} \\ & & & & \quad \alpha\text{th traction component} \\ \nabla c \cdot \boldsymbol{n} & = & 0 & \text{on} \quad \Omega \times [0,T] & : \text{Neumann condition on } c \end{array}$$

$$\begin{array}{rclll} \boldsymbol{u}(\boldsymbol{x},0) & = & \boldsymbol{u}_0(\boldsymbol{x}) & \forall \quad \boldsymbol{x} \in \Omega & : \text{Initial condition on displacement} \\ \dot{\boldsymbol{u}}(\boldsymbol{x},0) & = & \boldsymbol{v}_0(\boldsymbol{x}) & \forall \quad \boldsymbol{x} \in \Omega & : \text{Initial condition on velocity} \\ \mathcal{H}(\boldsymbol{x},0) & = & \mathcal{H}_0(\boldsymbol{x}) & \forall \quad \boldsymbol{x} \in \Omega & : \text{Initial strain-history field} \end{array}$$

A non-zero $\mathcal{H}_0$ can be used to model pre-existing cracks or other geometric features to be captured by the mesh topology, e.g.

$$\mathcal{H}_0(\boldsymbol{x}) = \left(\frac{1}{c_0} - 1\right) \frac{\mathcal{G}_c}{4\ell_0} \left(1 - \min\left\{\frac{d(\boldsymbol{x},l)}{\ell_0}, 1\right\}\right)$$

where $d(\boldsymbol{x},l)$ denotes the shortest distance from $\boldsymbol{x}$ to the curve l describing the initial crack geometry, and c_0 is phase-field value in the initial crack.

Spectral decomposision of strains

To establish the tensile (ψ^+) and compressive (ψ^-) contributions of the elastic strain energy, the eigenvalues, λ_α, and associated egenvectors, $\boldsymbol{n}_\alpha$, of the strain tensor $\boldsymbol{\varepsilon}$, are computed such that

$$\boldsymbol{\varepsilon} = \sum_\alpha \lambda_\alpha \, \boldsymbol{n}_\alpha \otimes \boldsymbol{n}_\alpha = \boldsymbol{\varepsilon}^+ + \boldsymbol{\varepsilon}^-$$

$$= \sum_\alpha \langle \lambda_\alpha \rangle \, \boldsymbol{n}_\alpha \otimes \boldsymbol{n}_\alpha + \sum_\alpha (\lambda_\alpha - \langle \lambda_\alpha \rangle) \, \boldsymbol{n}_\alpha \otimes \boldsymbol{n}_\alpha$$

Then

$$\psi^+ = \frac{\lambda}{2} \langle \mathrm{tr}\, \varepsilon \rangle^2 + \mu \, \mathrm{tr}(\varepsilon^+ : \varepsilon^+)$$

$$\psi^- = \frac{\lambda}{2} (\mathrm{tr}\, \varepsilon - \langle \mathrm{tr}\, \varepsilon \rangle)^2 + \mu \, \mathrm{tr}(\varepsilon^- : \varepsilon^-)$$

Spectral decomposision of strains

To establish the tensile (ψ^+) and compressive (ψ^-) contributions of the elastic strain energy, the eigenvalues, λ_α, and associated egenvectors, $\boldsymbol{n}_\alpha$, of the strain tensor $\boldsymbol{\varepsilon}$, are computed such that

$$\boldsymbol{\varepsilon} = \sum_\alpha \lambda_\alpha \, \boldsymbol{n}_\alpha \otimes \boldsymbol{n}_\alpha = \boldsymbol{\varepsilon}^+ + \boldsymbol{\varepsilon}^-$$
$$= \sum_\alpha \langle \lambda_\alpha \rangle \, \boldsymbol{n}_\alpha \otimes \boldsymbol{n}_\alpha + \sum_\alpha (\lambda_\alpha - \langle \lambda_\alpha \rangle) \, \boldsymbol{n}_\alpha \otimes \boldsymbol{n}_\alpha$$

Then

$$\psi^+ = \frac{\lambda}{2} \langle \operatorname{tr} \varepsilon \rangle^2 + \mu \operatorname{tr}(\varepsilon^+ : \varepsilon^+)$$
$$\psi^- = \frac{\lambda}{2} (\operatorname{tr} \varepsilon - \langle \operatorname{tr} \varepsilon \rangle)^2 + \mu \operatorname{tr}(\varepsilon^- : \varepsilon^-)$$

$\langle x \rangle = 1/2(x + |x|)$, the positive part of x

Spectral decomposision of strains

To establish the tensile (ψ^+) and compressive (ψ^-) contributions of the elastic strain energy, the eigenvalues, λ_α, and associated egenvectors, $\boldsymbol{n}_\alpha$, of the strain tensor $\boldsymbol{\varepsilon}$, are computed such that

$$\boldsymbol{\varepsilon} = \sum_\alpha \lambda_\alpha \, \boldsymbol{n}_\alpha \otimes \boldsymbol{n}_\alpha = \boldsymbol{\varepsilon}^+ + \boldsymbol{\varepsilon}^-$$

$$= \sum_\alpha \langle \lambda_\alpha \rangle \, \boldsymbol{n}_\alpha \otimes \boldsymbol{n}_\alpha + \sum_\alpha (\lambda_\alpha - \langle \lambda_\alpha \rangle) \, \boldsymbol{n}_\alpha \otimes \boldsymbol{n}_\alpha$$

Then

$$\psi^+ = \frac{\lambda}{2} \langle \operatorname{tr} \varepsilon \rangle^2 + \mu \operatorname{tr}(\varepsilon^+ : \varepsilon^+)$$

$$\psi^- = \frac{\lambda}{2} (\operatorname{tr} \varepsilon - \langle \operatorname{tr} \varepsilon \rangle)^2 + \mu \operatorname{tr}(\varepsilon^- : \varepsilon^-)$$

$\boldsymbol{\varepsilon}^+$: the tensile strain tensor

Spectral decomposision of strains

To establish the tensile (ψ^+) and compressive (ψ^-) contributions of the elastic strain energy, the eigenvalues, λ_α, and associated egenvectors, $\boldsymbol{n}_\alpha$, of the strain tensor $\boldsymbol{\varepsilon}$, are computed such that

$$\boldsymbol{\varepsilon} = \sum_\alpha \lambda_\alpha \, \boldsymbol{n}_\alpha \otimes \boldsymbol{n}_\alpha = \boldsymbol{\varepsilon}^+ + \boldsymbol{\varepsilon}^-$$

$$= \sum_\alpha \langle \lambda_\alpha \rangle \, \boldsymbol{n}_\alpha \otimes \boldsymbol{n}_\alpha + \sum_\alpha (\lambda_\alpha - \langle \lambda_\alpha \rangle) \, \boldsymbol{n}_\alpha \otimes \boldsymbol{n}_\alpha$$

Then

$$\psi^+ = \frac{\lambda}{2} \langle \operatorname{tr} \varepsilon \rangle^2 + \mu \operatorname{tr}(\varepsilon^+ : \varepsilon^+)$$

$$\psi^- = \frac{\lambda}{2} (\operatorname{tr} \varepsilon - \langle \operatorname{tr} \varepsilon \rangle)^2 + \mu \operatorname{tr}(\varepsilon^- : \varepsilon^-)$$

$\boldsymbol{\varepsilon}^-$: the compressive strain tensor

Spectral decomposision of strains

To establish the tensile (ψ^+) and compressive (ψ^-) contributions of the elastic strain energy, the eigenvalues, λ_α, and associated egenvectors, $\boldsymbol{n}_\alpha$, of the strain tensor $\boldsymbol{\varepsilon}$, are computed such that

$$\begin{aligned}\boldsymbol{\varepsilon} &= \sum_\alpha \lambda_\alpha \, \boldsymbol{n}_\alpha \otimes \boldsymbol{n}_\alpha = \boldsymbol{\varepsilon}^+ + \boldsymbol{\varepsilon}^- \\ &= \sum_\alpha \langle \lambda_\alpha \rangle \, \boldsymbol{n}_\alpha \otimes \boldsymbol{n}_\alpha + \sum_\alpha (\lambda_\alpha - \langle \lambda_\alpha \rangle) \, \boldsymbol{n}_\alpha \otimes \boldsymbol{n}_\alpha\end{aligned}$$

Then

$$\begin{aligned}\psi^+ &= \frac{\lambda}{2} \langle \mathrm{tr}\, \boldsymbol{\varepsilon} \rangle^2 + \mu \, \mathrm{tr}(\boldsymbol{\varepsilon}^+ : \boldsymbol{\varepsilon}^+) \\ \psi^- &= \frac{\lambda}{2} (\mathrm{tr}\, \boldsymbol{\varepsilon} - \langle \mathrm{tr}\, \boldsymbol{\varepsilon} \rangle)^2 + \mu \, \mathrm{tr}(\boldsymbol{\varepsilon}^- : \boldsymbol{\varepsilon}^-)\end{aligned}$$

Flow in fractures

- The effect of c on the Darcy flow is realized by artificially inflating permeability in open fractures, to model Poiseuille flow.

$$\boldsymbol{\kappa}_{\mathrm{m}} = \kappa \boldsymbol{I} + (1-c)^b \left(\frac{w^2}{12} - \kappa \right) \left(\boldsymbol{I} - \boldsymbol{n}\boldsymbol{n}^{\mathsf{T}} \right)$$

- w is the regularized crack width, $w^2 = (\lambda_\perp - 1)^2 L_\perp^2 \chi_{c<c_{\mathrm{crit}}}$

Flow in fractures

- The effect of c on the Darcy flow is realized by artificially inflating permeability in open fractures, to model Poiseuille flow.

$$\boldsymbol{\kappa}_{\mathrm{m}} = \kappa \boldsymbol{I} + (1-c)^b \left(\frac{w^2}{12} - \kappa \right) \left(\boldsymbol{I} - \boldsymbol{n}\boldsymbol{n}^{\mathsf{T}} \right)$$

- w is the regularized crack width, $w^2 = (\lambda_\perp - 1)^2 L_\perp^2 \chi_{c<c_{\mathrm{crit}}}$

Flow in fractures

- The effect of c on the Darcy flow is realized by artificially inflating permeability in open fractures, to model Poiseuille flow.

$$\boldsymbol{\kappa}_{\mathrm{m}} = \kappa \boldsymbol{I} + (1-c)^b \left(\frac{w^2}{12} - \kappa \right) \left(\boldsymbol{I} - \boldsymbol{n}\boldsymbol{n}^{\mathsf{T}} \right)$$

- w is the regularized crack width, $w^2 = (\lambda_\perp - 1)^2 L_\perp^2 \chi_{c<c_{\mathrm{crit}}}$

The crack normal vector in physical coordinates

$$\boldsymbol{n} = \frac{(\nabla \boldsymbol{u})^{-\mathsf{T}} \nabla c}{|(\nabla \boldsymbol{u})^{-\mathsf{T}} \nabla c|}$$

Flow in fractures

- The effect of c on the Darcy flow is realized by artificially inflating permeability in open fractures, to model Poiseuille flow.

$$\boldsymbol{\kappa}_{\mathrm{m}} = \kappa \boldsymbol{I} + (1-c)^b \left(\frac{w^2}{12} - \kappa \right) \left(\boldsymbol{I} - \boldsymbol{n}\boldsymbol{n}^{\mathsf{T}} \right)$$

- w is the regularized crack width, $w^2 = (\lambda_{\perp} - 1)^2 L_{\perp}^2 \chi_{c<c_{\text{crit}}}$

The local perpendicular stretch

$$\lambda_{\perp} = (\nabla \boldsymbol{u}) \frac{\nabla c}{|\nabla c|} \cdot \boldsymbol{n}$$

Flow in fractures

- The effect of c on the Darcy flow is realized by artificially inflating permeability in open fractures, to model Poiseuille flow.

$$\boldsymbol{\kappa}_{\mathrm{m}} = \kappa \boldsymbol{I} + (1-c)^b \left(\frac{w^2}{12} - \kappa \right) (\boldsymbol{I} - \boldsymbol{n}\boldsymbol{n}^{\mathsf{T}})$$

- w is the regularized crack width, $w^2 = (\lambda_\perp - 1)^2 L_\perp^2 \chi_{c<c_{\mathrm{crit}}}$

$L_\perp$: length scale roughly tracing ℓ and meshwidth

Flow in fractures

- The effect of c on the Darcy flow is realized by artificially inflating permeability in open fractures, to model Poiseuille flow.

$$\boldsymbol{\kappa}_{\mathrm{m}} = \kappa \boldsymbol{I} + (1-c)^b \left(\frac{w^2}{12} - \kappa \right) \left(\boldsymbol{I} - \boldsymbol{n}\boldsymbol{n}^{\mathsf{T}} \right)$$

- w is the regularized crack width, $w^2 = (\lambda_{\perp} - 1)^2 L_{\perp}^2 \chi_{c<c_{\text{crit}}}$

κ: isotropic un-fractured permeability

Coupling and nonlinearities

Superiterations

- Two primary solvers
  - A Joint poroelastic solver for displacement and pressure
  - B Separate solver for integrity
- Solution for each timestep obtained in an interlaced manner (standard coupling technique)
  1. solve A for $(\boldsymbol{u}_{n+1}^{(1)}, p_{n+1}^{(1)})$
  2. solve B for $c_{n+1}^{(1)}$
  3. solve A for $(\boldsymbol{u}_{n+1}^{(2)}, p_{n+1}^{(2)})$
  4. solve B for $c_{n+1}^{(2)}$
  5. etc.
- Fully coupled all-in-one solvers for all three unknowns have also been successful

Superiterations

- Two primary solvers
  - A Joint poroelastic solver for displacement and pressure
  - B Separate solver for integrity
- Solution for each timestep obtained in an interlaced manner (standard coupling technique)
  1. solve A for $(\boldsymbol{u}_{n+1}^{(1)}, p_{n+1}^{(1)})$
  2. solve B for $c_{n+1}^{(1)}$
  3. solve A for $(\boldsymbol{u}_{n+1}^{(2)}, p_{n+1}^{(2)})$
  4. solve B for $c_{n+1}^{(2)}$
  5. etc.
- Fully coupled all-in-one solvers for all three unknowns have also been successful

Superiterations

- Two primary solvers
  - A Joint poroelastic solver for displacement and pressure
  - B Separate solver for integrity
- Solution for each timestep obtained in an interlaced manner (standard coupling technique)
  1. solve A for $(\boldsymbol{u}_{n+1}^{(1)}, p_{n+1}^{(1)})$
  2. solve B for $c_{n+1}^{(1)}$
  3. solve A for $(\boldsymbol{u}_{n+1}^{(2)}, p_{n+1}^{(2)})$
  4. solve B for $c_{n+1}^{(2)}$
  5. etc.
- Fully coupled all-in-one solvers for all three unknowns have also been successful

Subiterations

- The poroelastic solver A must itself also be iterative
- Multiple sources of nonlinearity:
  - due to the tensile/compressive energy splitting
  - due to inherently nonlinear elasticity models
  - due to iterative time solvers for dynamic problems (e.g. Newmark)

Backward Euler for quasistatic problems

$$\begin{pmatrix} \boldsymbol{K}(c) & -\boldsymbol{Q} \\ \boldsymbol{Q}^\intercal/\delta_t & \boldsymbol{P}(c)+\boldsymbol{S}/\delta_t \end{pmatrix} \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{p} \end{pmatrix}_{n+1} = \begin{pmatrix} \boldsymbol{f}_u \\ \boldsymbol{f}_p \end{pmatrix}_{n+1} + \begin{pmatrix} \boldsymbol{Q}^\intercal & \boldsymbol{S} \end{pmatrix} \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{p} \end{pmatrix}_{n+1}$$

Subiterations

- The poroelastic solver A must itself also be iterative
- Multiple sources of nonlinearity:
  - due to the tensile/compressive energy splitting
  - due to inherently nonlinear elasticity models
  - due to iterative time solvers for dynamic problems (e.g. Newmark)

Backward Euler for quasistatic problems

$$\begin{pmatrix} \boldsymbol{K}(c) & -\boldsymbol{Q} \\ \boldsymbol{Q}^\intercal/\delta_t & \boldsymbol{P}(c) + \boldsymbol{S}/\delta_t \end{pmatrix} \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{p} \end{pmatrix}_{n+1} = \begin{pmatrix} \boldsymbol{f}_u \\ \boldsymbol{f}_p \end{pmatrix}_{n+1} + \begin{pmatrix} \boldsymbol{Q}^\intercal & \boldsymbol{S} \end{pmatrix} \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{p} \end{pmatrix}_{n+1}$$

Subiterations

- The poroelastic solver A must itself also be iterative
- Multiple sources of nonlinearity:
  - due to the tensile/compressive energy splitting
  - due to inherently nonlinear elasticity models
  - due to iterative time solvers for dynamic problems (e.g. Newmark)

Backward Euler for quasistatic problems

$$\begin{pmatrix} \boldsymbol{K}(c) & -\boldsymbol{Q} \\ \boldsymbol{Q}^\intercal/\delta_t & \boldsymbol{P}(c) + \boldsymbol{S}/\delta_t \end{pmatrix} \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{p} \end{pmatrix}_{n+1} = \begin{pmatrix} \boldsymbol{f}_u \\ \boldsymbol{f}_p \end{pmatrix}_{n+1} + \begin{pmatrix} \boldsymbol{Q}^\intercal & \boldsymbol{S} \end{pmatrix} \begin{pmatrix} \boldsymbol{u} \\ \boldsymbol{p} \end{pmatrix}_{n+1}$$

Newmark for dynamic problems

Given $\boldsymbol{a}_{n+1}^i = (\ddot{\boldsymbol{u}}, \ddot{\boldsymbol{p}})_{n+1}^i$, $\boldsymbol{v}_{n+1}^i = (\dot{\boldsymbol{u}}, \dot{\boldsymbol{p}})_{n+1}^i$, $\boldsymbol{d}_{n+1}^i = (\boldsymbol{u}, \boldsymbol{p})_{n+1}^i$

Solve

$$\boldsymbol{M}^* \Delta \boldsymbol{a} = \begin{pmatrix} \boldsymbol{f}_u \\ \boldsymbol{f}_p \end{pmatrix}_{n+1} - \tilde{\boldsymbol{M}} \boldsymbol{a}_{n+1}^i - \tilde{\boldsymbol{C}} \boldsymbol{v}_{n+1}^i - \tilde{\boldsymbol{K}} \boldsymbol{d}_{n+1}^i$$

Correct

$$\begin{aligned} \boldsymbol{a}_{n+1}^{i+1} &= \boldsymbol{a}_{n+1}^i + \Delta \boldsymbol{a} \\ \boldsymbol{v}_{n+1}^{i+1} &= \boldsymbol{v}_{n+1}^i + \gamma \delta_t \Delta \boldsymbol{a} \\ \boldsymbol{d}_{n+1}^{i+1} &= \boldsymbol{d}_{n+1}^i + \beta \delta_t^2 \Delta \boldsymbol{a} \end{aligned}$$

Newmark for dynamic problems

Given $\boldsymbol{a}^i_{n+1} = (\ddot{\boldsymbol{u}}, \ddot{\boldsymbol{p}})^i_{n+1}$, $\boldsymbol{v}^i_{n+1} = (\dot{\boldsymbol{u}}, \dot{\boldsymbol{p}})^i_{n+1}$, $\boldsymbol{d}^i_{n+1} = (\boldsymbol{u}, \boldsymbol{p})^i_{n+1}$

Solve

$$\boldsymbol{M}^* \Delta \boldsymbol{a} = \begin{pmatrix} \boldsymbol{f}_u \\ \boldsymbol{f}_p \end{pmatrix}_{n+1} - \tilde{\boldsymbol{M}} \boldsymbol{a}^i_{n+1} - \tilde{\boldsymbol{C}} \boldsymbol{v}^i_{n+1} - \tilde{\boldsymbol{K}} \boldsymbol{d}^i_{n+1}$$

Correct

$$\begin{aligned} \boldsymbol{a}^{i+1}_{n+1} &= \boldsymbol{a}^i_{n+1} + \Delta \boldsymbol{a} \\ \boldsymbol{v}^{i+1}_{n+1} &= \boldsymbol{v}^i_{n+1} + \gamma \delta_t \Delta \boldsymbol{a} \\ \boldsymbol{d}^{i+1}_{n+1} &= \boldsymbol{d}^i_{n+1} + \beta \delta_t^2 \Delta \boldsymbol{a} \end{aligned}$$

Predicted values for acceleration, velocity and solution

Newmark for dynamic problems

Given $\boldsymbol{a}^i_{n+1} = (\ddot{\boldsymbol{u}}, \ddot{\boldsymbol{p}})^i_{n+1}$, $\boldsymbol{v}^i_{n+1} = (\dot{\boldsymbol{u}}, \dot{\boldsymbol{p}})^i_{n+1}$, $\boldsymbol{d}^i_{n+1} = (\boldsymbol{u}, \boldsymbol{p})^i_{n+1}$
Solve

$$\boldsymbol{M}^* \Delta \boldsymbol{a} = \begin{pmatrix} \boldsymbol{f}_u \\ \boldsymbol{f}_p \end{pmatrix}_{n+1} - \tilde{\boldsymbol{M}} \boldsymbol{a}^i_{n+1} - \tilde{\boldsymbol{C}} \boldsymbol{v}^i_{n+1} - \tilde{\boldsymbol{K}} \boldsymbol{d}^i_{n+1}$$

Correct

$$\boldsymbol{a}^{i+1}_{n+1} = \boldsymbol{a}^i_{n+1} + \Delta \boldsymbol{a}$$
$$\boldsymbol{v}^{i+1}_{n+1} = \boldsymbol{v}^i_{n+1} + \gamma \delta_t \Delta \boldsymbol{a}$$
$$\boldsymbol{d}^{i+1}_{n+1} = \boldsymbol{d}^i_{n+1} + \beta \delta_t^2 \Delta \boldsymbol{a}$$

Newmark for dynamic problems

Given $\boldsymbol{a}^i_{n+1} = (\ddot{\boldsymbol{u}}, \ddot{\boldsymbol{p}})^i_{n+1}$, $\boldsymbol{v}^i_{n+1} = (\dot{\boldsymbol{u}}, \dot{\boldsymbol{p}})^i_{n+1}$, $\boldsymbol{d}^i_{n+1} = (\boldsymbol{u}, \boldsymbol{p})^i_{n+1}$
Solve

$$\boldsymbol{M}^*\Delta\boldsymbol{a} = \begin{pmatrix} \boldsymbol{f}_u \\ \boldsymbol{f}_p \end{pmatrix}_{n+1} - \tilde{\boldsymbol{M}}\boldsymbol{a}^i_{n+1} - \tilde{\boldsymbol{C}}\boldsymbol{v}^i_{n+1} - \tilde{\boldsymbol{K}}\boldsymbol{d}^i_{n+1}$$

Correct

$$\begin{aligned} \boldsymbol{a}^{i+1}_{n+1} &= \boldsymbol{a}^i_{n+1} + \Delta\boldsymbol{a} \\ \boldsymbol{v}^{i+1}_{n+1} &= \boldsymbol{v}^i_{n+1} + \gamma\delta_t\Delta\boldsymbol{a} \\ \boldsymbol{d}^{i+1}_{n+1} &= \boldsymbol{d}^i_{n+1} + \beta\delta_t^2\Delta\boldsymbol{a} \end{aligned}$$

$$\tilde{\boldsymbol{M}} = \begin{pmatrix} \boldsymbol{M} & \\ & \end{pmatrix}$$

Newmark for dynamic problems

Given $\boldsymbol{a}^i_{n+1} = (\ddot{\boldsymbol{u}}, \ddot{\boldsymbol{p}})^i_{n+1}$, $\boldsymbol{v}^i_{n+1} = (\dot{\boldsymbol{u}}, \dot{\boldsymbol{p}})^i_{n+1}$, $\boldsymbol{d}^i_{n+1} = (\boldsymbol{u}, \boldsymbol{p})^i_{n+1}$
Solve

$$\boldsymbol{M}^* \Delta \boldsymbol{a} = \begin{pmatrix} \boldsymbol{f}_u \\ \boldsymbol{f}_p \end{pmatrix}_{n+1} - \tilde{\boldsymbol{M}} \boldsymbol{a}^i_{n+1} - \tilde{\boldsymbol{C}} \boldsymbol{v}^i_{n+1} - \tilde{\boldsymbol{K}} \boldsymbol{d}^i_{n+1}$$

Correct

$$\begin{aligned} \boldsymbol{a}^{i+1}_{n+1} &= \boldsymbol{a}^i_{n+1} + \Delta \boldsymbol{a} \\ \boldsymbol{v}^{i+1}_{n+1} &= \boldsymbol{v}^i_{n+1} + \gamma \delta_t \Delta \boldsymbol{a} \\ \boldsymbol{d}^{i+1}_{n+1} &= \boldsymbol{d}^i_{n+1} + \beta \delta_t^2 \Delta \boldsymbol{a} \end{aligned}$$

$$\tilde{\boldsymbol{C}} = \begin{pmatrix} \boldsymbol{C} & \\ \boldsymbol{Q}^\intercal & \boldsymbol{S} \end{pmatrix}$$

Newmark for dynamic problems

Given $\boldsymbol{a}^i_{n+1} = (\ddot{\boldsymbol{u}}, \ddot{\boldsymbol{p}})^i_{n+1}$, $\boldsymbol{v}^i_{n+1} = (\dot{\boldsymbol{u}}, \dot{\boldsymbol{p}})^i_{n+1}$, $\boldsymbol{d}^i_{n+1} = (\boldsymbol{u}, \boldsymbol{p})^i_{n+1}$
Solve

$$\boldsymbol{M}^* \Delta \boldsymbol{a} = \begin{pmatrix} \boldsymbol{f}_u \\ \boldsymbol{f}_p \end{pmatrix}_{n+1} - \tilde{\boldsymbol{M}} \boldsymbol{a}^i_{n+1} - \tilde{\boldsymbol{C}} \boldsymbol{v}^i_{n+1} - \tilde{\boldsymbol{K}} \boldsymbol{d}^i_{n+1}$$

Correct

$$\boldsymbol{a}^{i+1}_{n+1} = \boldsymbol{a}^i_{n+1} + \Delta \boldsymbol{a}$$
$$\boldsymbol{v}^{i+1}_{n+1} = \boldsymbol{v}^i_{n+1} + \gamma \delta_t \Delta \boldsymbol{a}$$
$$\boldsymbol{d}^{i+1}_{n+1} = \boldsymbol{d}^i_{n+1} + \beta \delta_t^2 \Delta \boldsymbol{a}$$

$$\tilde{\boldsymbol{K}} = \begin{pmatrix} \boldsymbol{K} & -\boldsymbol{Q} \\ & \boldsymbol{P} \end{pmatrix}$$

Newmark for dynamic problems

Given $\boldsymbol{a}^i_{n+1} = (\ddot{\boldsymbol{u}}, \ddot{\boldsymbol{p}})^i_{n+1}$, $\boldsymbol{v}^i_{n+1} = (\dot{\boldsymbol{u}}, \dot{\boldsymbol{p}})^i_{n+1}$, $\boldsymbol{d}^i_{n+1} = (\boldsymbol{u}, \boldsymbol{p})^i_{n+1}$
Solve

$$\boldsymbol{M}^* \Delta \boldsymbol{a} = \begin{pmatrix} \boldsymbol{f}_u \\ \boldsymbol{f}_p \end{pmatrix}_{n+1} - \tilde{\boldsymbol{M}} \boldsymbol{a}^i_{n+1} - \tilde{\boldsymbol{C}} \boldsymbol{v}^i_{n+1} - \tilde{\boldsymbol{K}} \boldsymbol{d}^i_{n+1}$$

Correct

$$\begin{aligned} \boldsymbol{a}^{i+1}_{n+1} &= \boldsymbol{a}^i_{n+1} + \Delta \boldsymbol{a} \\ \boldsymbol{v}^{i+1}_{n+1} &= \boldsymbol{v}^i_{n+1} + \gamma \delta_t \Delta \boldsymbol{a} \\ \boldsymbol{d}^{i+1}_{n+1} &= \boldsymbol{d}^i_{n+1} + \beta \delta_t^2 \Delta \boldsymbol{a} \end{aligned}$$

$$\boldsymbol{M}^* = \tilde{\boldsymbol{M}} + \gamma \delta_t \tilde{\boldsymbol{C}} + \beta \delta_t^2 \tilde{\boldsymbol{K}}$$

Newmark for dynamic problems

Given $\boldsymbol{a}^i_{n+1} = (\ddot{\boldsymbol{u}}, \ddot{\boldsymbol{p}})^i_{n+1}$, $\boldsymbol{v}^i_{n+1} = (\dot{\boldsymbol{u}}, \dot{\boldsymbol{p}})^i_{n+1}$, $\boldsymbol{d}^i_{n+1} = (\boldsymbol{u}, \boldsymbol{p})^i_{n+1}$
Solve

$$\boldsymbol{M}^* \Delta \boldsymbol{a} = \begin{pmatrix} \boldsymbol{f}_u \\ \boldsymbol{f}_p \end{pmatrix}_{n+1} - \tilde{\boldsymbol{M}} \boldsymbol{a}^i_{n+1} - \tilde{\boldsymbol{C}} \boldsymbol{v}^i_{n+1} - \tilde{\boldsymbol{K}} \boldsymbol{d}^i_{n+1}$$

Correct

$$\begin{aligned} \boldsymbol{a}^{i+1}_{n+1} &= \boldsymbol{a}^i_{n+1} + \Delta \boldsymbol{a} \\ \boldsymbol{v}^{i+1}_{n+1} &= \boldsymbol{v}^i_{n+1} + \gamma \delta_t \Delta \boldsymbol{a} \\ \boldsymbol{d}^{i+1}_{n+1} &= \boldsymbol{d}^i_{n+1} + \beta \delta_t^2 \Delta \boldsymbol{a} \end{aligned}$$

Newmark for dynamic problems

Given $\boldsymbol{a}^i_{n+1} = (\ddot{\boldsymbol{u}}, \ddot{\boldsymbol{p}})^i_{n+1}$, $\boldsymbol{v}^i_{n+1} = (\dot{\boldsymbol{u}}, \dot{\boldsymbol{p}})^i_{n+1}$, $\boldsymbol{d}^i_{n+1} = (\boldsymbol{u}, \boldsymbol{p})^i_{n+1}$
Solve

$$\boldsymbol{M}^* \Delta \boldsymbol{a} = \begin{pmatrix} \boldsymbol{f}_u \\ \boldsymbol{f}_p \end{pmatrix}_{n+1} - \tilde{\boldsymbol{M}} \boldsymbol{a}^i_{n+1} - \tilde{\boldsymbol{C}} \boldsymbol{v}^i_{n+1} - \tilde{\boldsymbol{K}} \boldsymbol{d}^i_{n+1}$$

Correct

$$\begin{aligned} \boldsymbol{a}^{i+1}_{n+1} &= \boldsymbol{a}^i_{n+1} + \Delta \boldsymbol{a} \\ \boldsymbol{v}^{i+1}_{n+1} &= \boldsymbol{v}^i_{n+1} + \gamma \delta_t \Delta \boldsymbol{a} \\ \boldsymbol{d}^{i+1}_{n+1} &= \boldsymbol{d}^i_{n+1} + \beta \delta_t^2 \Delta \boldsymbol{a} \end{aligned}$$

Stability: $2\beta \geq \gamma \geq 1/2$, accuracy: $\gamma = 1/2$

Adaptivity

- Adaptive refinement is almost mandatory
  - Fractures require high spatial resolution to resolve well (see: ℓ)
  - ... but only locally
- Refining elements with small c *a posteriori* is dubious: fractures propagate slower through coarse meshes
- Thus a third layer of iterations: whenever refinement is needed, re-run the last handful of timesteps on the finer mesh.

Adaptivity

- Adaptive refinement is almost mandatory
  - Fractures require high spatial resolution to resolve well (see: ℓ)
  - ... but only locally
- Refining elements with small c *a posteriori* is dubious: fractures propagate slower through coarse meshes
- Thus a third layer of iterations: whenever refinement is needed, re-run the last handful of timesteps on the finer mesh.

Adaptivity

- Adaptive refinement is almost mandatory
  - Fractures require high spatial resolution to resolve well (see: ℓ)
  - . . . but only locally
- Refining elements with small c *a posteriori* is dubious: fractures propagate slower through coarse meshes
- Thus a third layer of iterations: whenever refinement is needed, re-run the last handful of timesteps on the finer mesh.

Adaptive mesh refinement of the crack path

Adaptive mesh refinement

- A fine mesh resolution is required to correctly capture the crack development.
- Using a fine uniform mesh is easiest and safest, but too costly.
- An adaptive strategy that refines the mesh only where the crack is propagating is needed.
- We use a multi-pass procedure, using the phase-field value as refinement criterium.
- A linear or quadratic LR B-Spline discretization is used, allowing for local refinement.
- When an initial crack is present, the mesh is refined based on the distance d_e^{c0} from the element center to the initial crack path, before the simulation is started.

Adaptive mesh refinement, initial state

- Load the initial, uniform, background mesh
- d^{tol} = min. distance to initial crack path for non-refined elements
  $\approx h^0$ (characteristic element size of the initial mesh)

FOR $i = 1$ TO number of initial refinement cycles DO

- Refine all elements e, where $d_e^{c0} < d^{\mathrm{tol}}$
- $d^{\mathrm{tol}} = d^{\mathrm{tol}}/2$

END DO

Adaptive mesh refinement, multi-pass algorithm

- n_{step} = total number of time steps
- n^c_{step} = number of time steps in each refinement cycle
- n_{cycle} = max. number of refinement cycles before continuing

FOR $i = 1$ TO n_{step} DO ! Time step loop

FOR $j = 1$ TO n_{cycle} TO

- Restore solution state for time t_{i-1}

FOR $k = 0$ TO $n^c_{\text{step}} - 1$ TO

- Compute elasticity and phase-field solutions at time t_{i+k}

END DO

- Refine all elements e, for which $|c|_e < c_{\text{tol}}$

IF no elements were refined THEN exit DO-loop

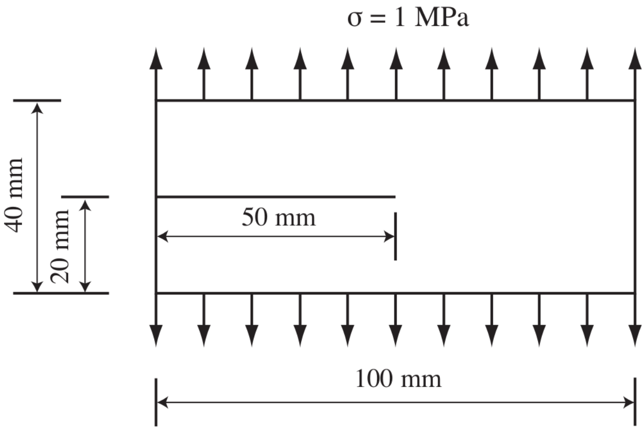
END DO

- $i = i + n^c_{\text{step}}$

END DO

Pre-notched Rectangular Plate

Pre-notched rectangular plate

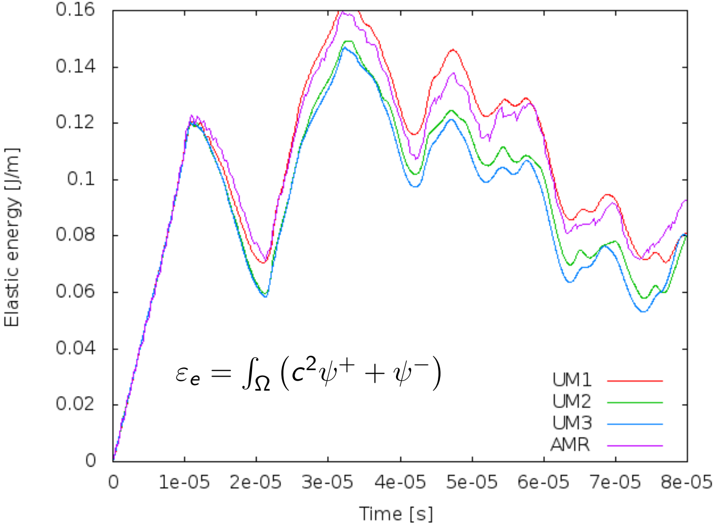


$$\rho = 2450\,\mathrm{kg/m^3}$$
$$E = 32\,\mathrm{GPa}$$
$$\nu = 0.2$$
$$\mathcal{G}_c = 3\,\mathrm{J/m}$$

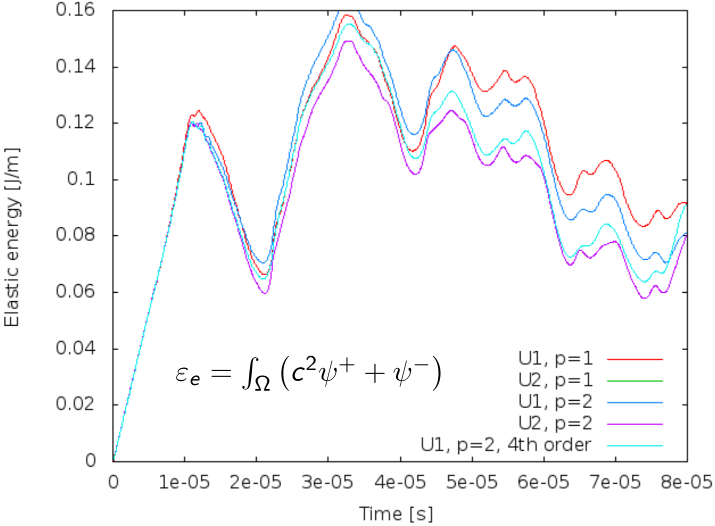
Mesh and time step size

Mesh	p	n_{el}	n_{dof}	δ_t [s]
U1	2	400×160	133650	1.0×10^{-7}
U2	2	800×320	523250	5.0×10^{-8}
U3	2	1600×640	2066580	2.5×10^{-8}
A0	2	4054	7676	1.0×10^{-7}
$\vdots$		$\vdots$	$\vdots$	
An	2	8686	15648	1.0×10^{-7}

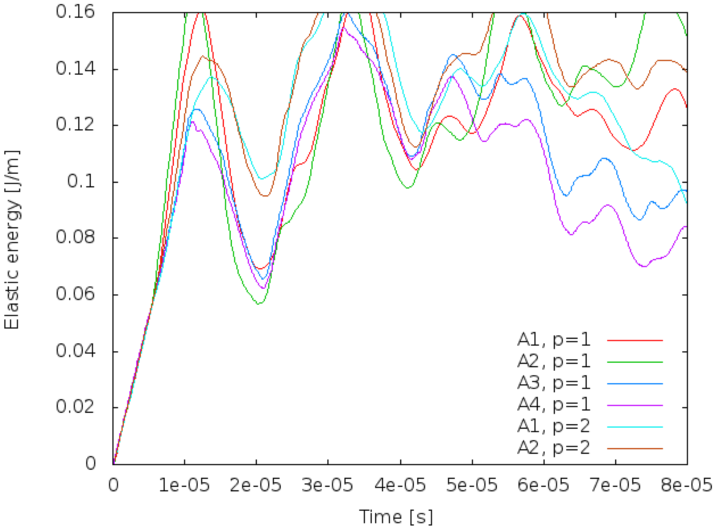
Elastic energy



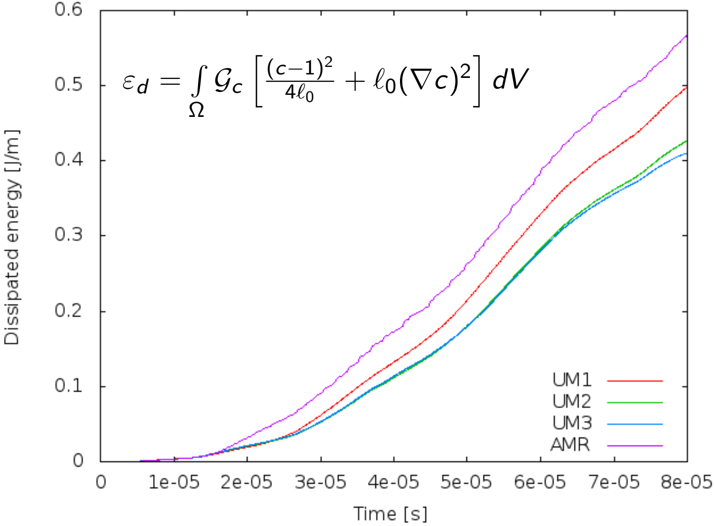
Elastic energy, uniform meshes



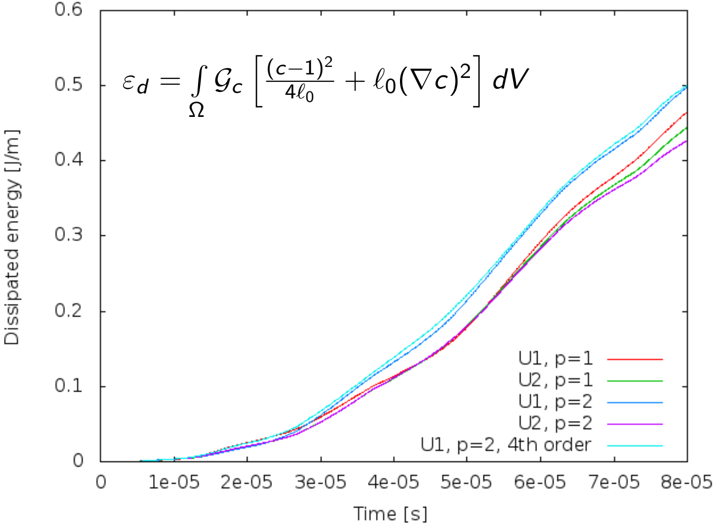
Elastic energy, adapted meshes



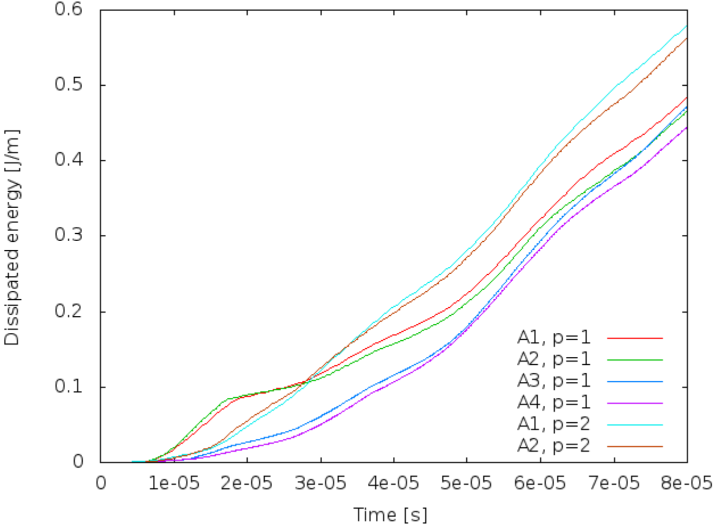
Dissipated energy



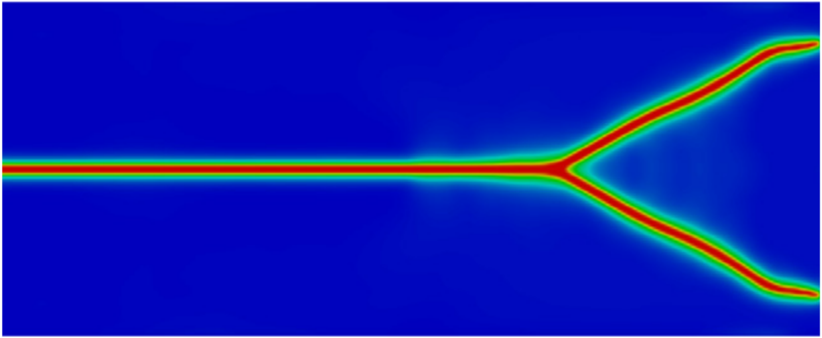
Dissipated energy, uniform meshes



Dissipated energy, adapted meshes

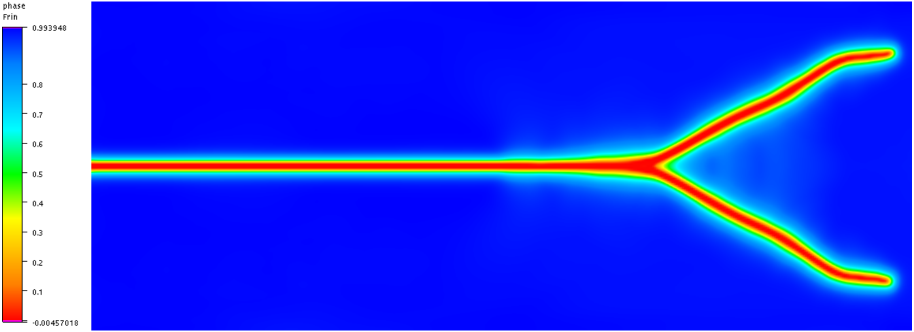


Phase-field for Mesh U1, p=2



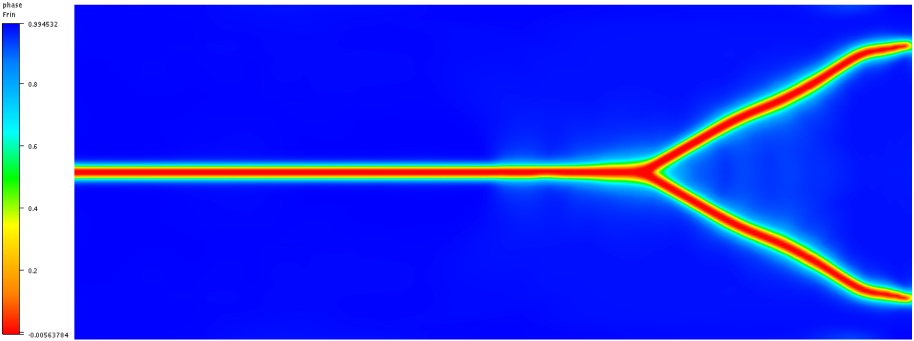
from M. J. Borden *et al.*, “A phase-field description of dynamic brittle fracture”, Comput. Methods Appl. Mech. Engrg. 217–220 (2012) 77–95.

Phase-field for Mesh U1, p=2



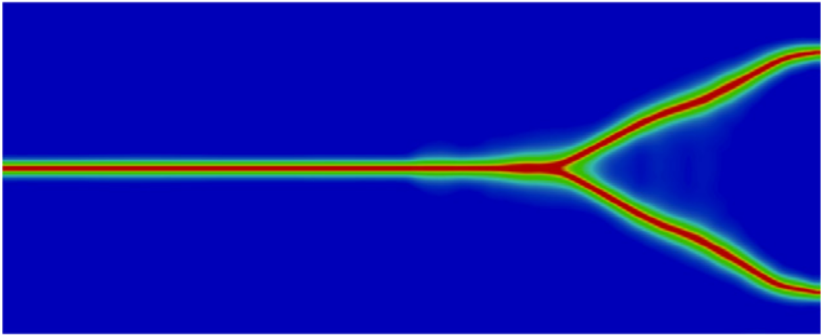
with IFEM

Phase-field for Mesh U1, p=2



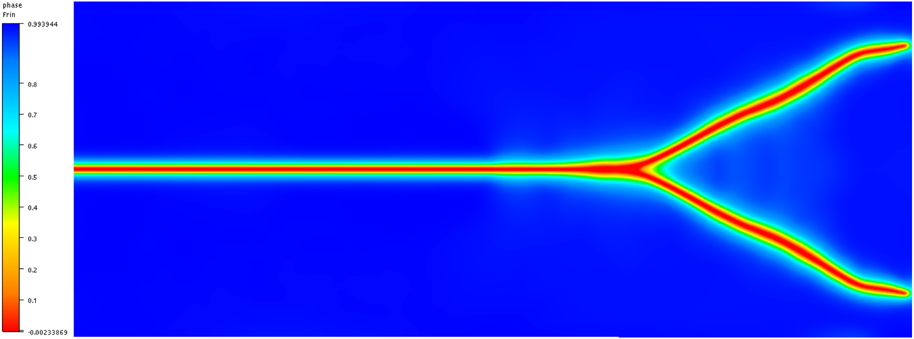
with IFEM (4th order phase field)

Phase-field for Mesh U2, p=2



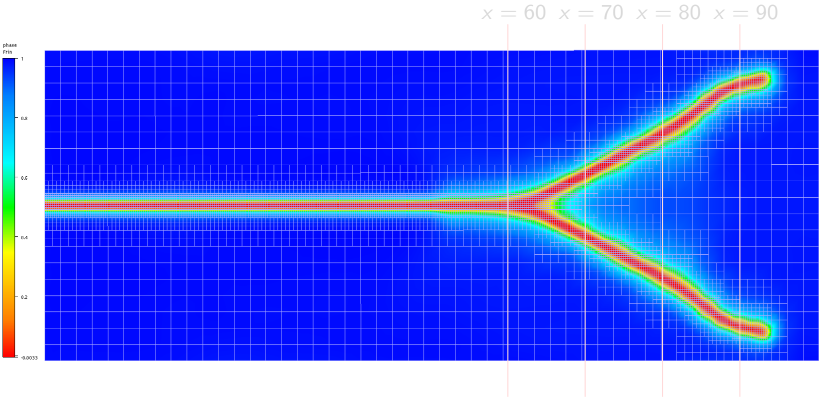
from M. J. Borden *et al.*, "A phase-field description of dynamic brittle fracture", Comput. Methods Appl. Mech. Engrg. 217–220 (2012) 77–95.

Phase-field for Mesh U2, p=2

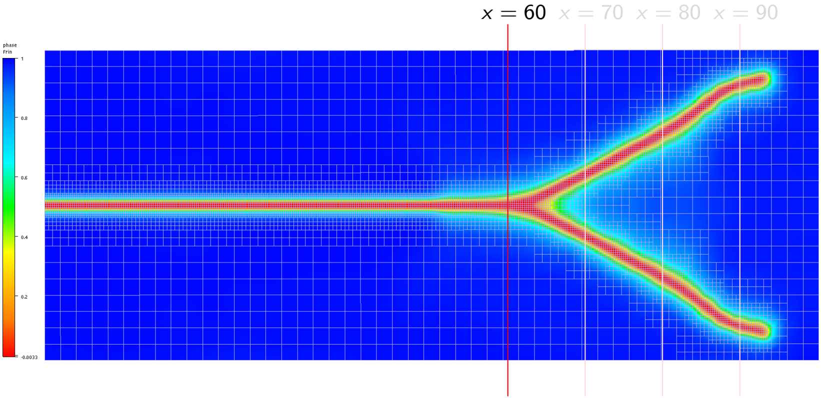


with IFEM

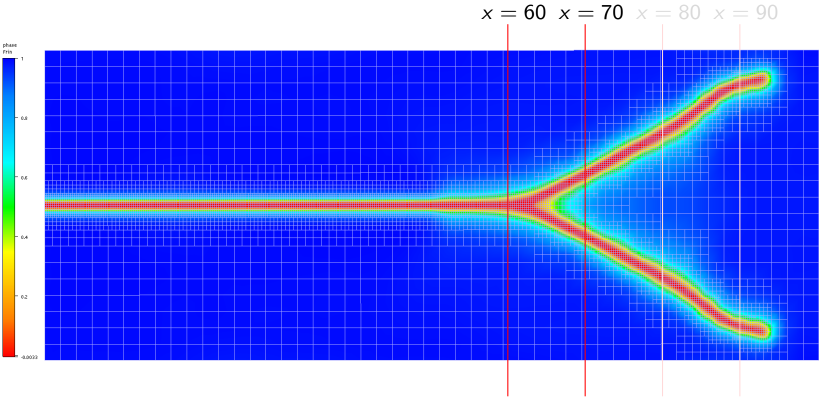
Phase-field on adapted mesh, p=2



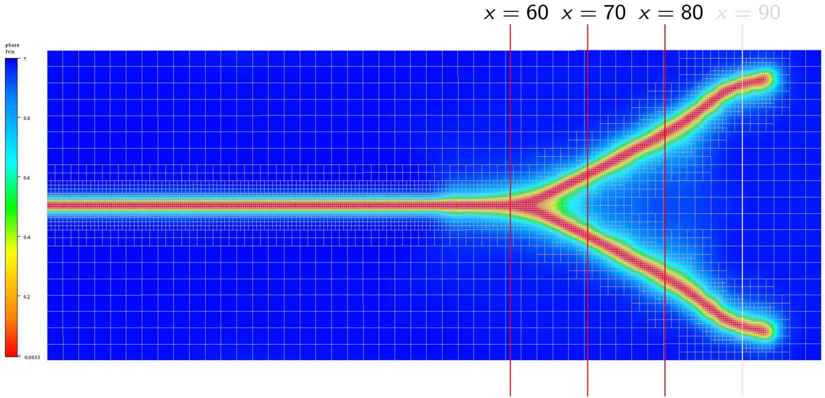
Phase-field on adapted mesh, p=2



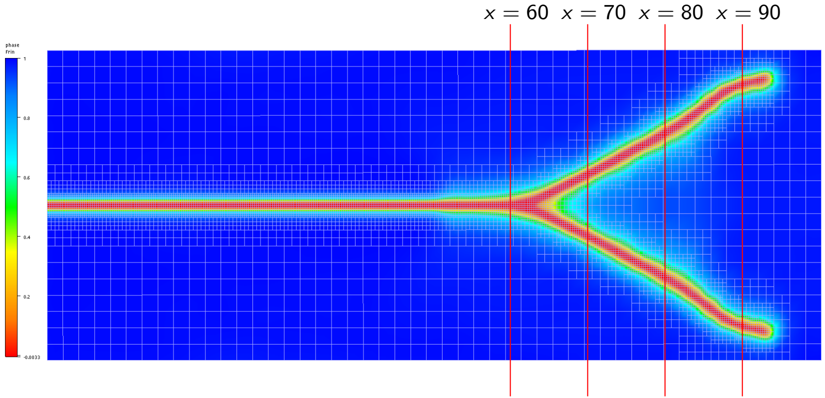
Phase-field on adapted mesh, p=2



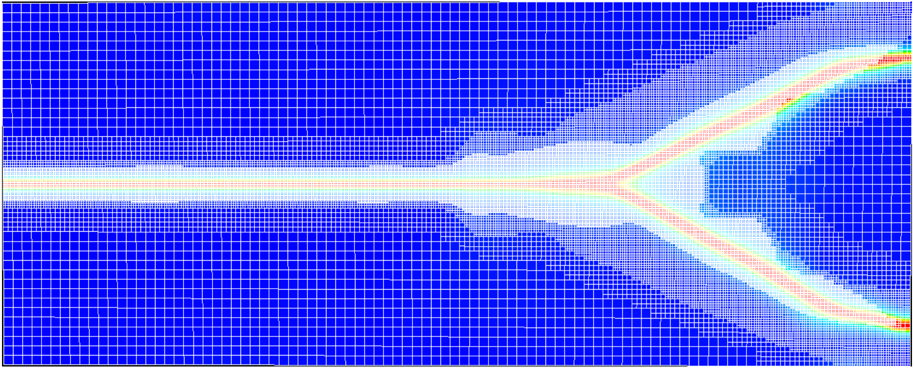
Phase-field on adapted mesh, p=2



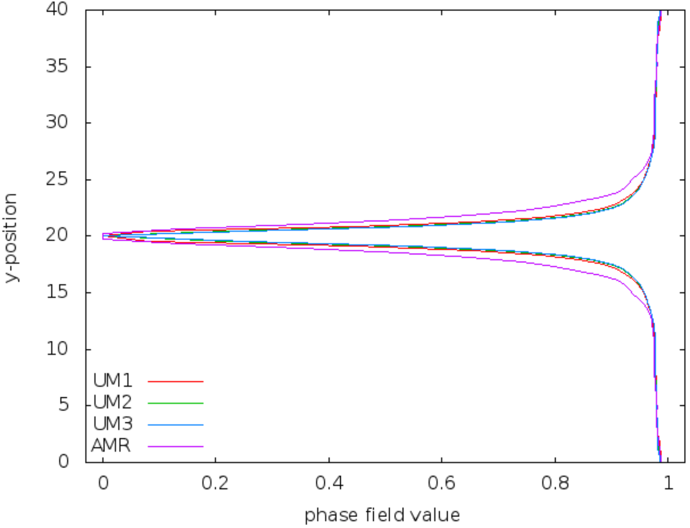
Phase-field on adapted mesh, p=2



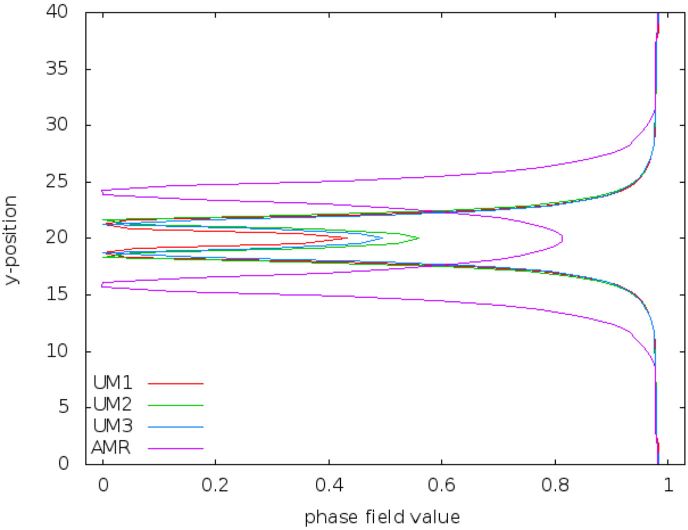
Phase-field on 4th adapted mesh, p=1



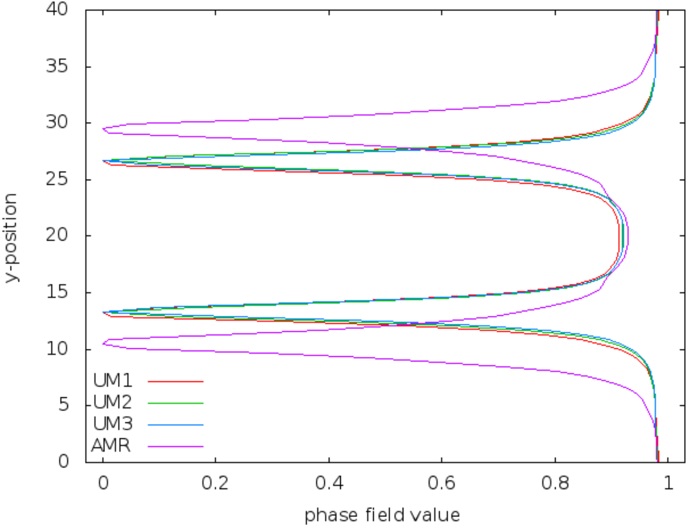
Phase field along the vertical line $x = 60$ at $t = 0.079\,\mathrm{ms}$



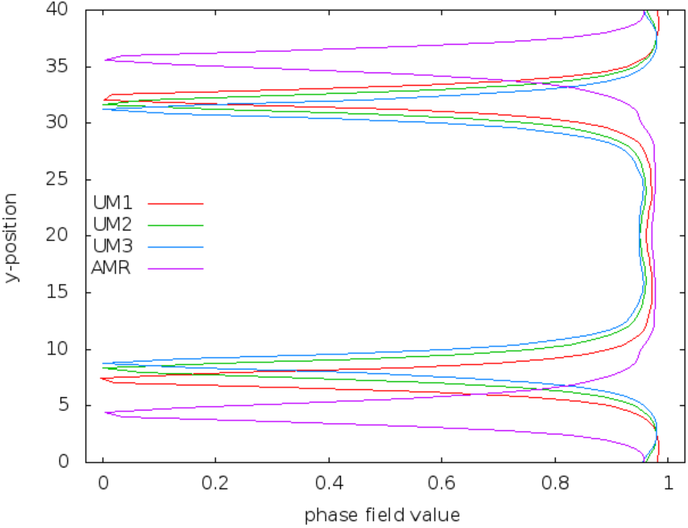
Phase field along the vertical line $x = 70$ at $t = 0.079\,\mathrm{ms}$



Phase field along the vertical line $x = 80$ at $t = 0.079\,\text{ms}$

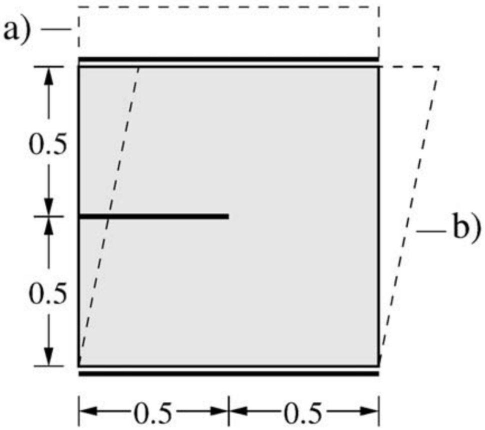


Phase field along the vertical line $x = 90$ at $t = 0.079\,\mathrm{ms}$



Pre-notched square plate

Pre-notched square plate



$$E = 210\,\text{kN}/\text{mm}^2$$
$$\nu = 0.3$$
$$\mathcal{G}_c = 2.7\,\text{mN}/\text{mm}$$
$$\ell_0 = 0.0075\,\text{mm}$$

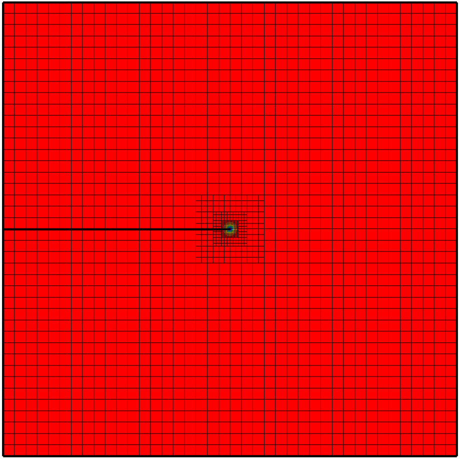
Adaptive with 3, 4 and 5 refinement levels

$p = 1, 2$ (LR B-splines)

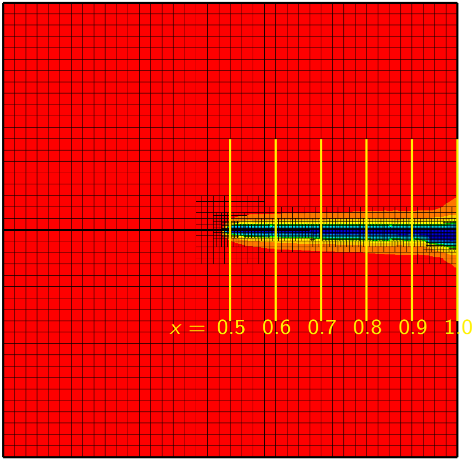
a) Tension test
b) Pure shear test

(Figure from C. Miehe, M. Hofacker, F. Welschinger, A phase field model for rate-independent crack propagation: Robust algorithmic implementation based on operator splits. Computer Methods in Applied Mechanics and Engineering, vol. 199 (2010), pp. 2765–2778.)

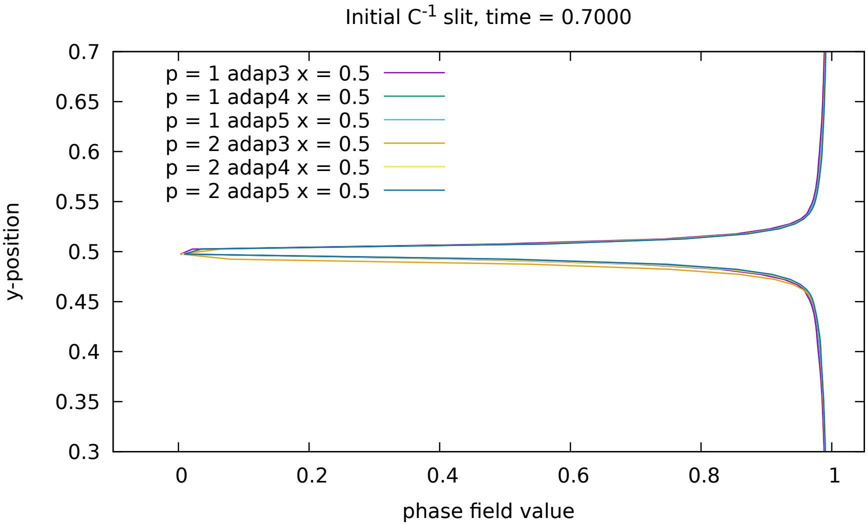
Tension test, initial crack by C^{-1}-continuity



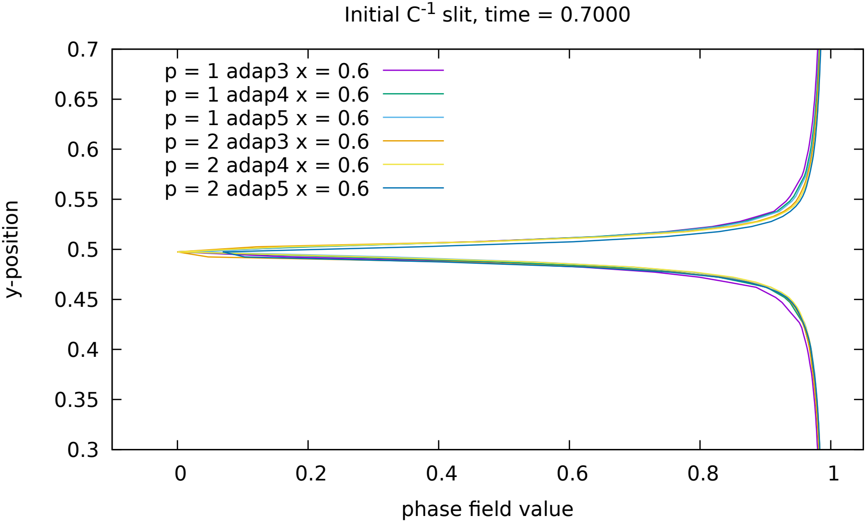
Tension test, initial crack by C^{-1}-continuity



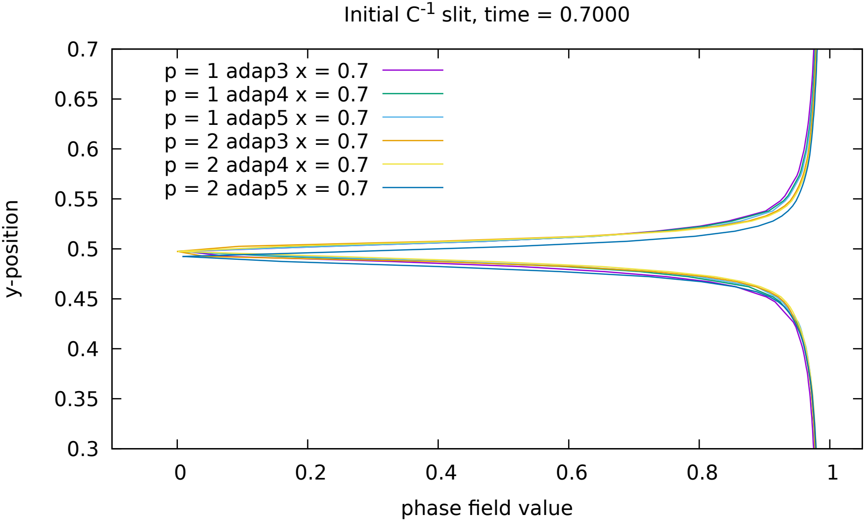
Tension test, phase field along $x = 0.5$ at $t = 0.7$



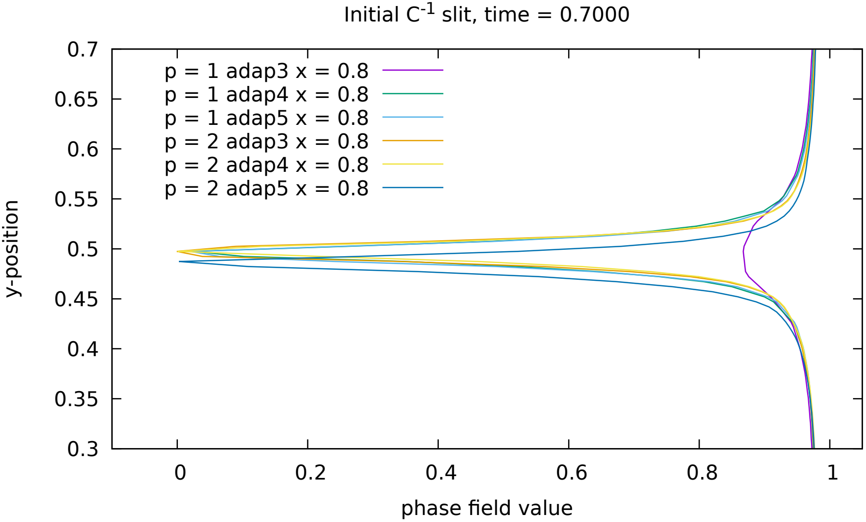
Tension test, phase field along $x = 0.6$ at $t = 0.7$



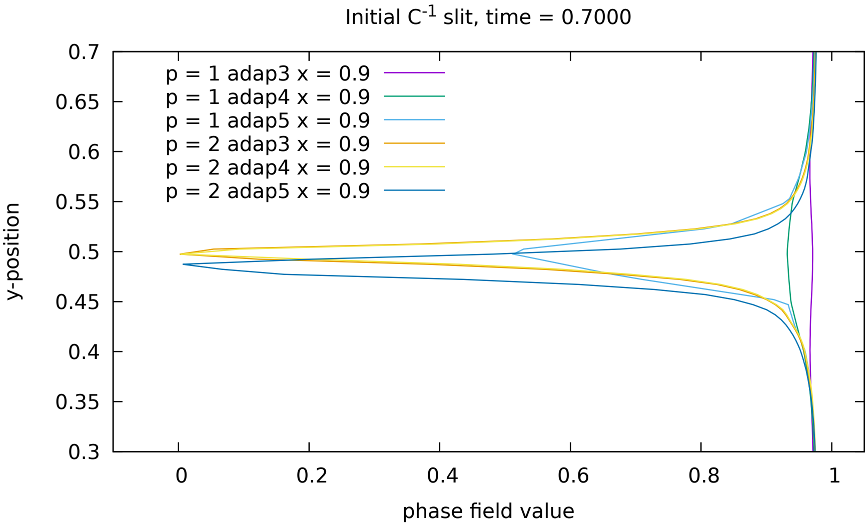
Tension test, phase field along $x = 0.7$ at $t = 0.7$



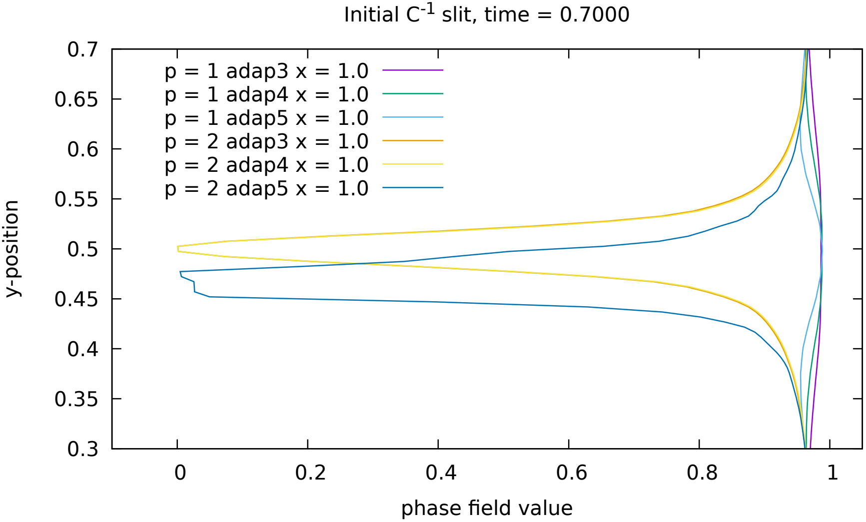
Tension test, phase field along $x = 0.8$ at $t = 0.7$



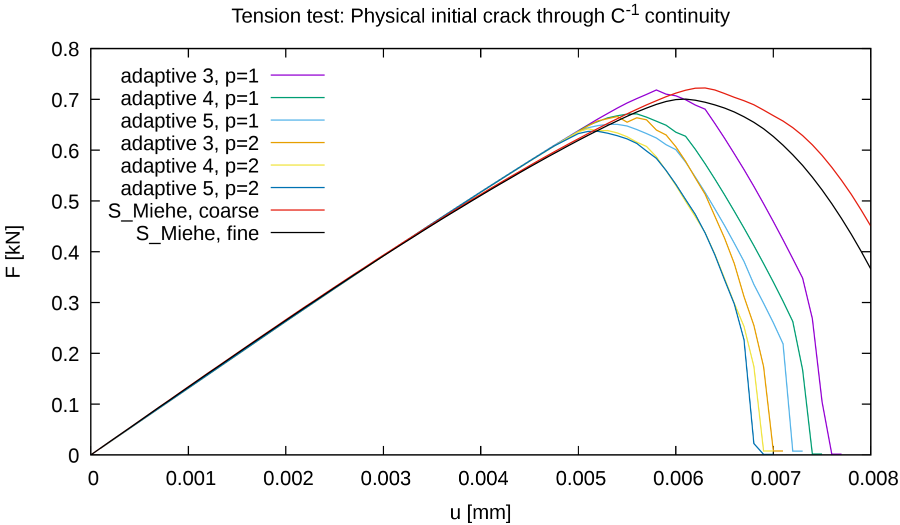
Tension test, phase field along $x = 0.9$ at $t = 0.7$



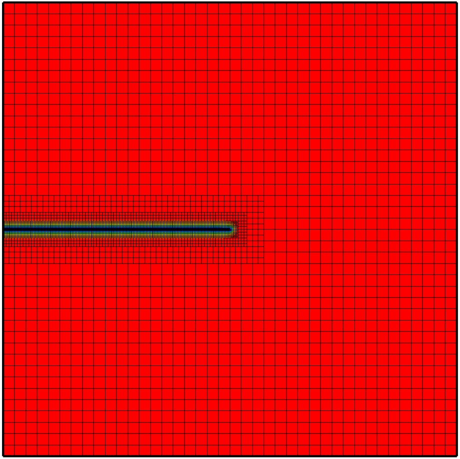
Tension test, phase field along $x = 1.0$ at $t = 0.7$



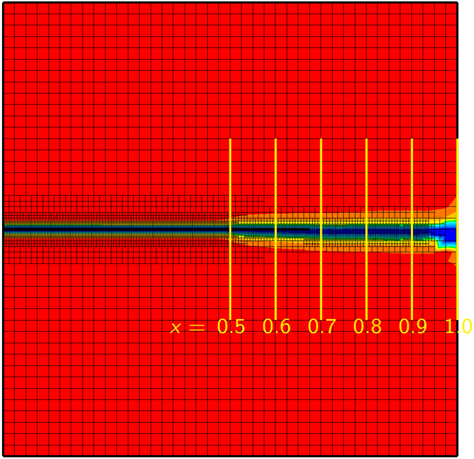
Tension test, reaction force vs. displacement



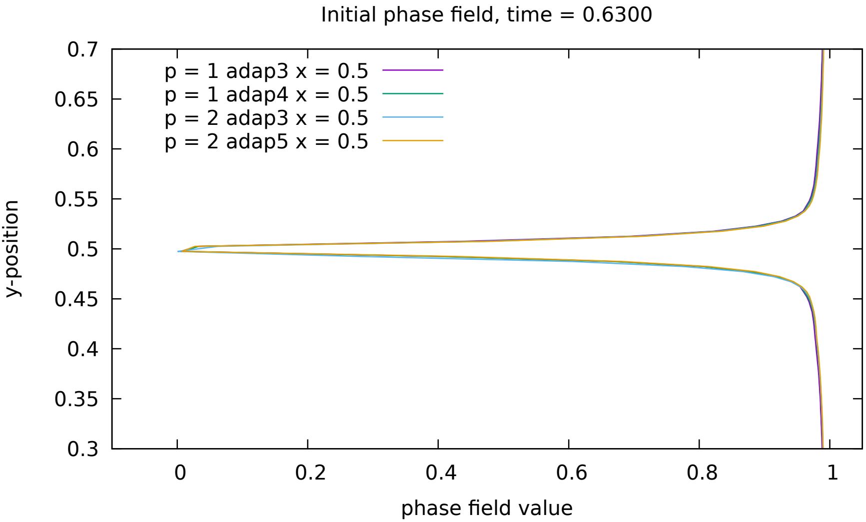
Tension test, initial crack by phase field



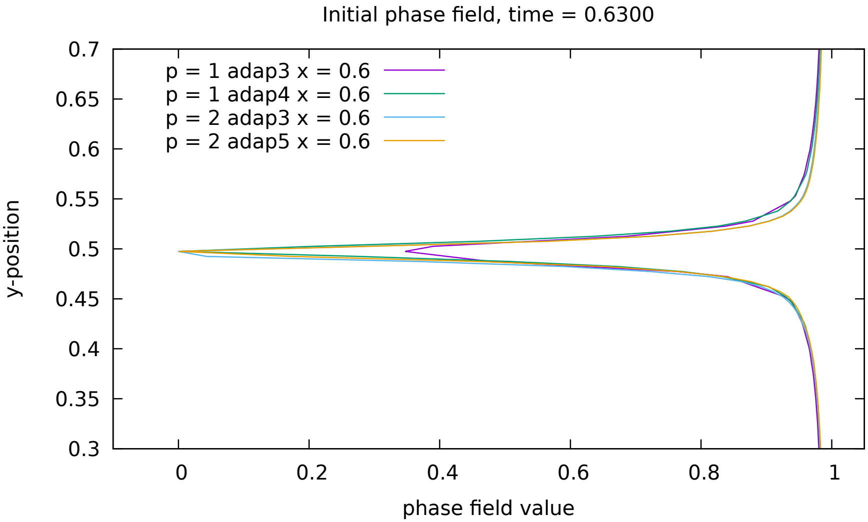
Tension test, initial crack by phase field



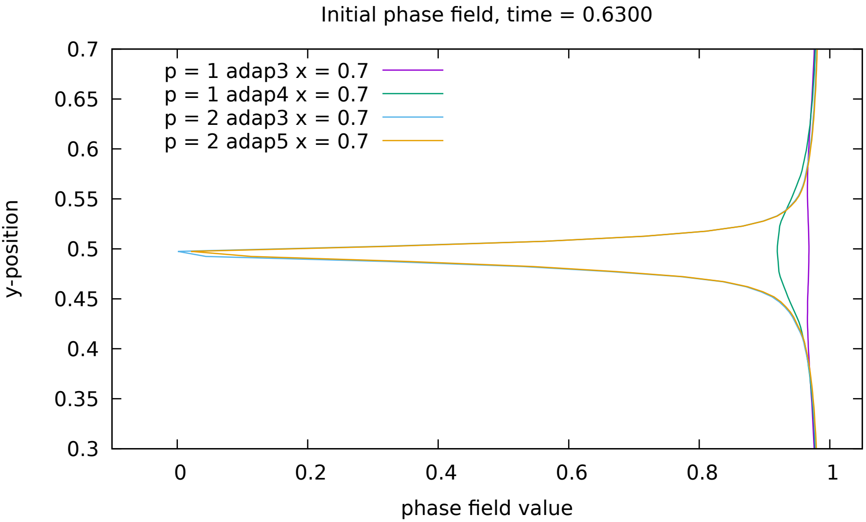
Tension test, phase field along $x = 0.5$ at $t = 0.63$



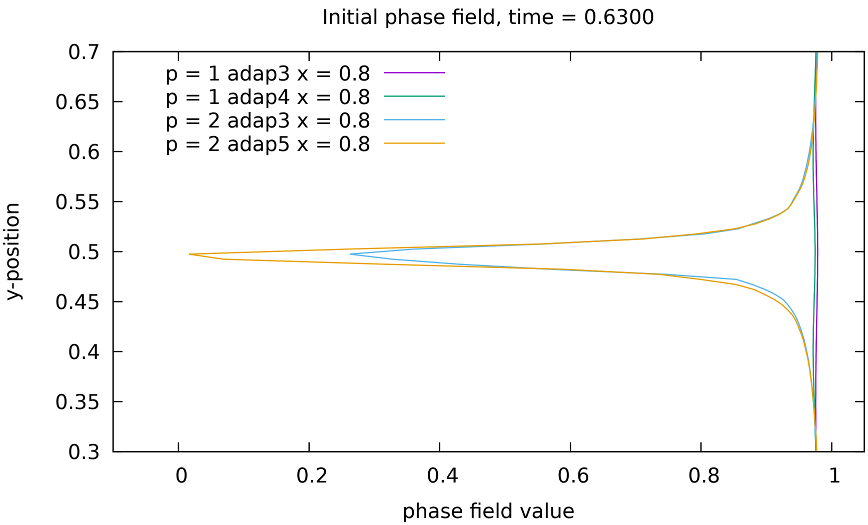
Tension test, phase field along $x = 0.6$ at $t = 0.63$



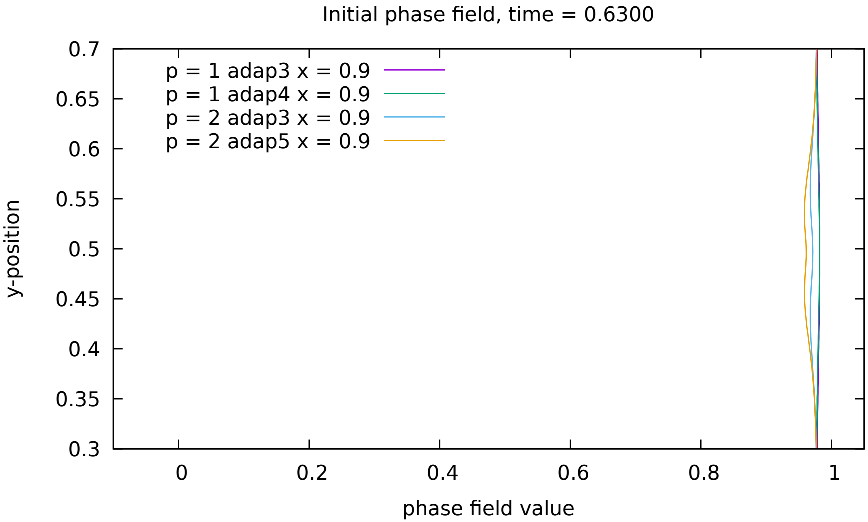
Tension test, phase field along $x = 0.7$ at $t = 0.63$



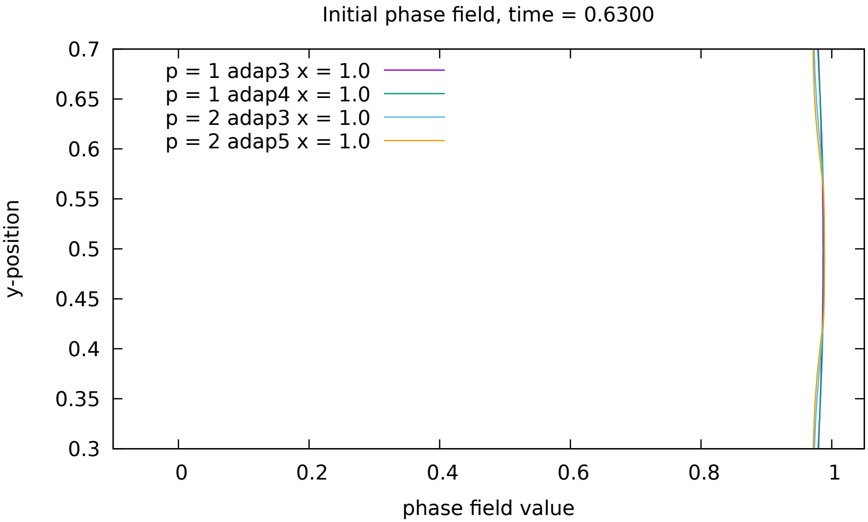
Tension test, phase field along $x = 0.8$ at $t = 0.63$



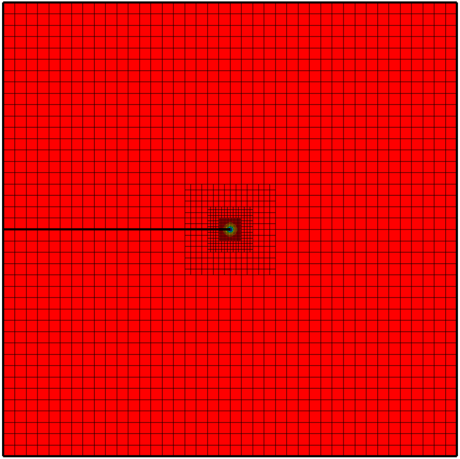
Tension test, phase field along $x = 0.9$ at $t = 0.63$



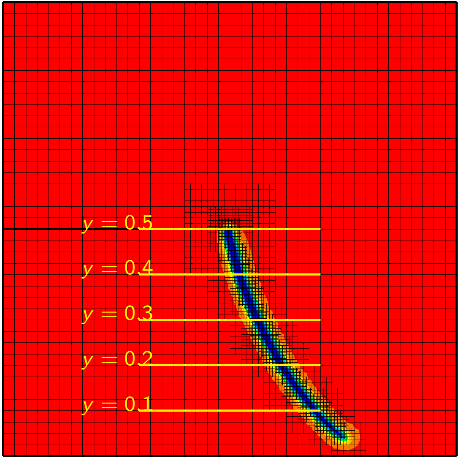
Tension test, phase field along $x = 1.0$ at $t = 0.63$



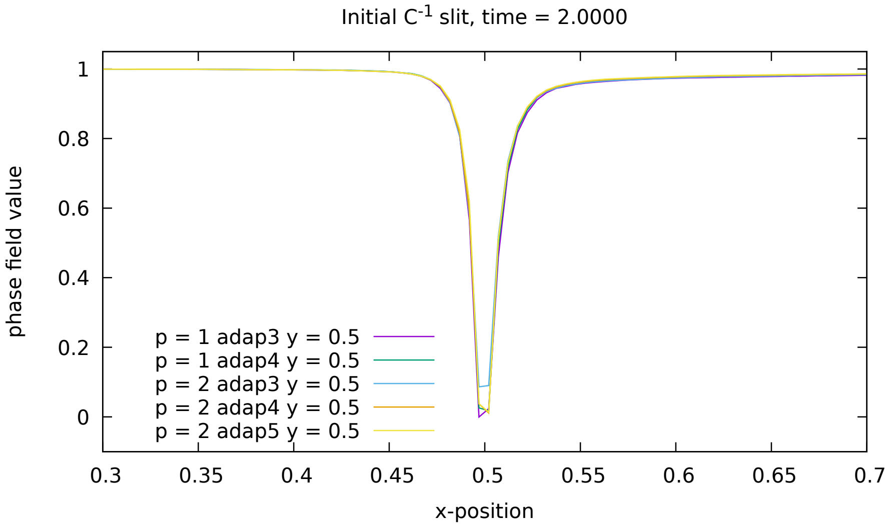
Shear test, initial crack by C^{-1}-continuity



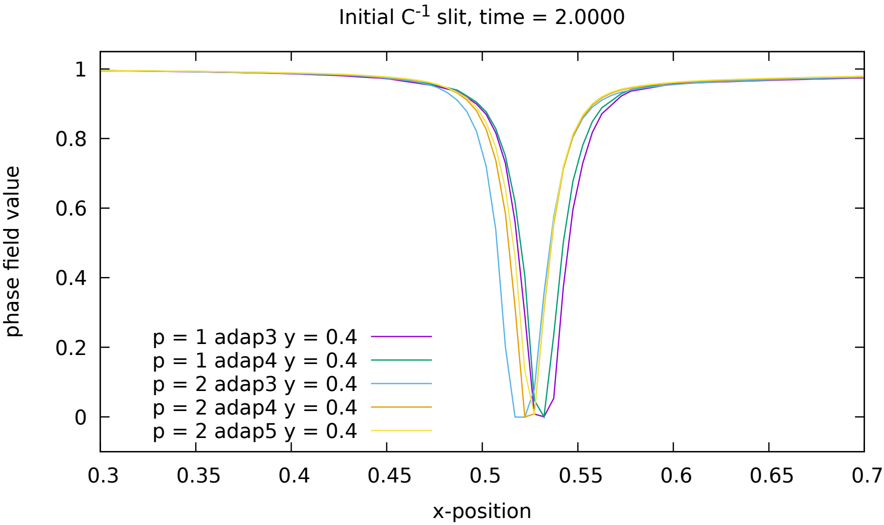
Shear test, initial crack by C^{-1}-continuity



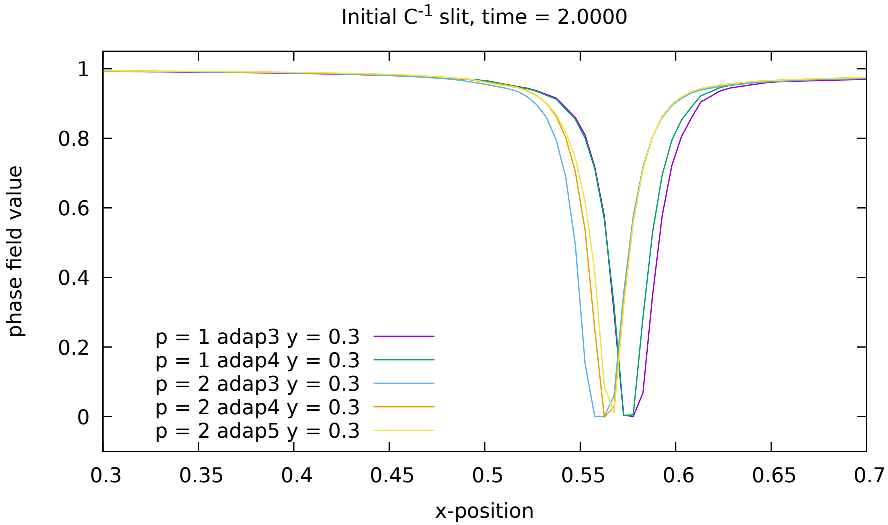
Shear test, phase field along $y = 0.5$ at $t = 2$



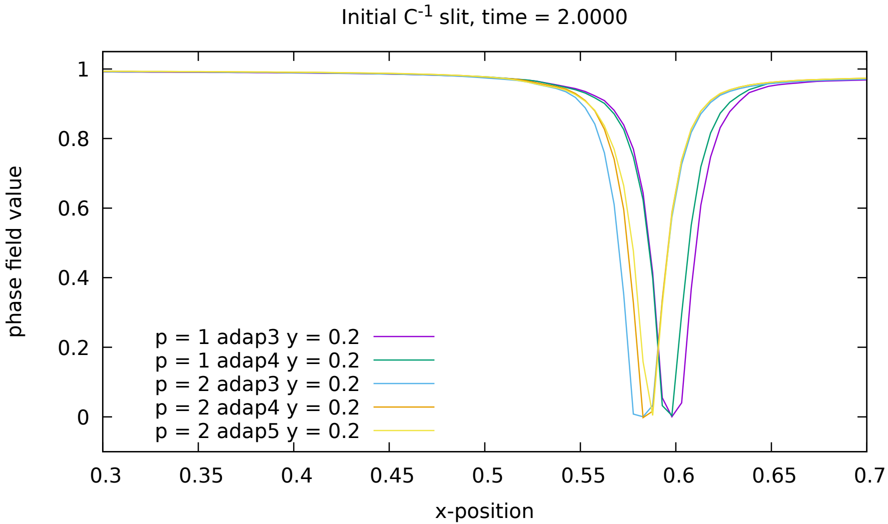
Shear test, phase field along $y = 0.4$ at $t = 2$



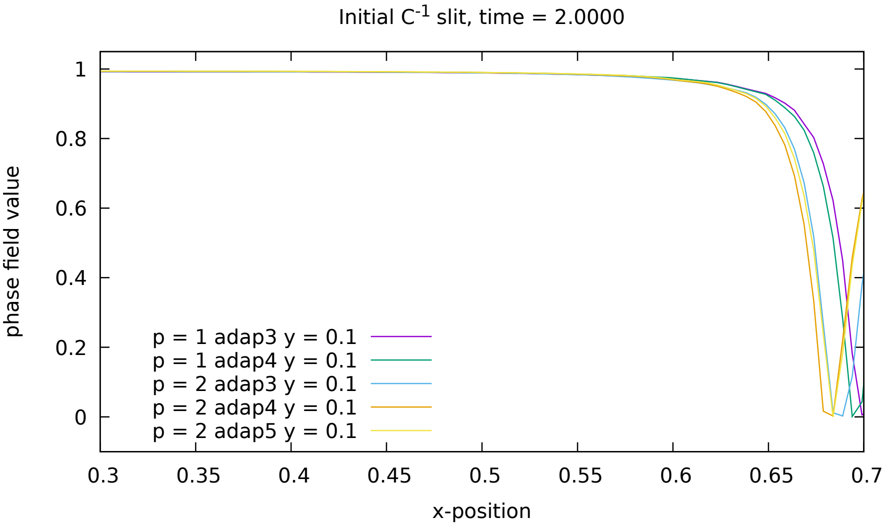
Shear test, phase field along $y = 0.3$ at $t = 2$



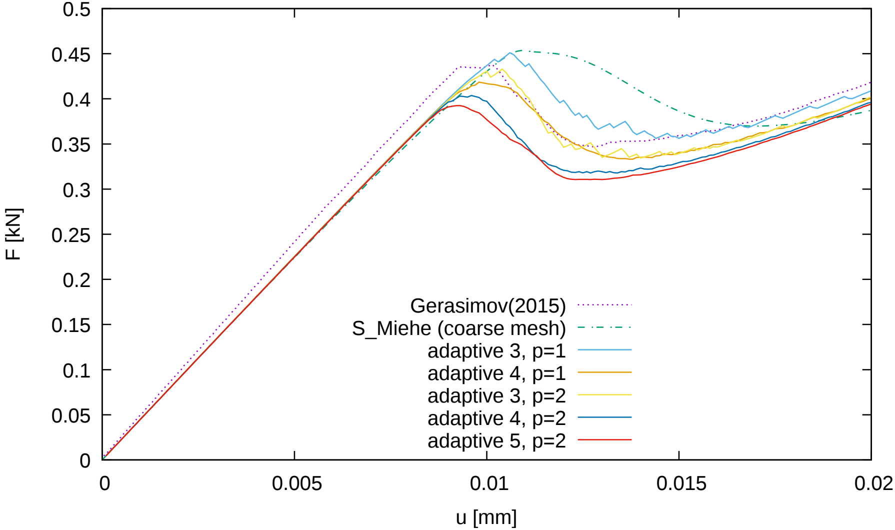
Shear test, phase field along $y = 0.2$ at $t = 2$



Shear test, phase field along $y = 0.1$ at $t = 2$

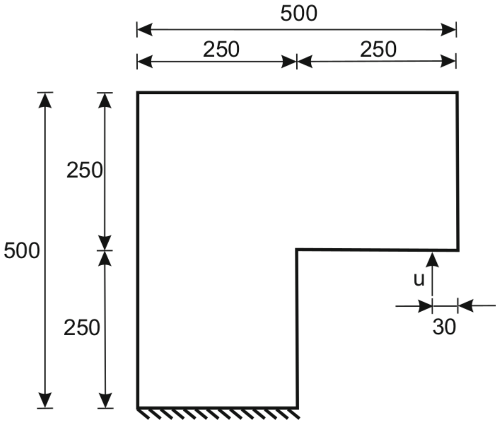


Shear test, reaction force vs. displacement



L-shaped domain

L-shaped domain

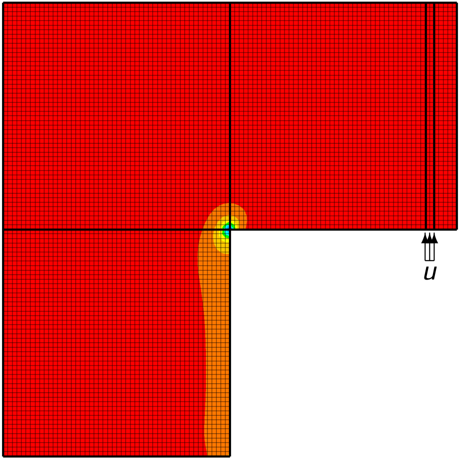


$$E = 25.85\,\mathrm{MN/mm}^2$$
$$\nu = 0.18$$
$$\mathcal{G}_c = 0.09\,\mathrm{kN/mm}$$
$$\ell_0 = 1.875\,\mathrm{mm}$$

The displacement u is applied over a 10 mm wide segment

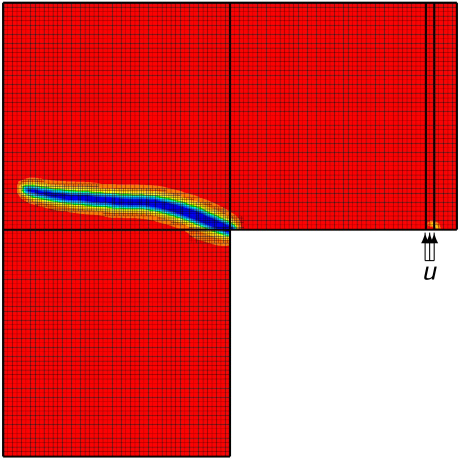
(Figure from M. Ambati, T. Gerasimov, L. De Lorenzis, A review on phase-field models of brittle fracture and a new fast hybrid formulation. Computational Mechanics, vol. 55 (2014), pp. 383–405.)

L-shaped domain, 5 patches, $3 \times 50 \times 50$, $p = 1$



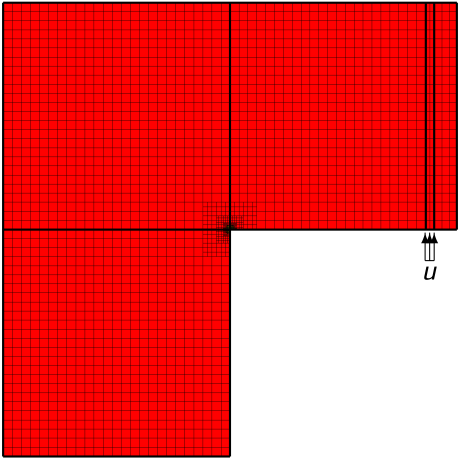
Initial grid

L-shaped domain, 5 patches, $3 \times 50 \times 50$, $p = 1$



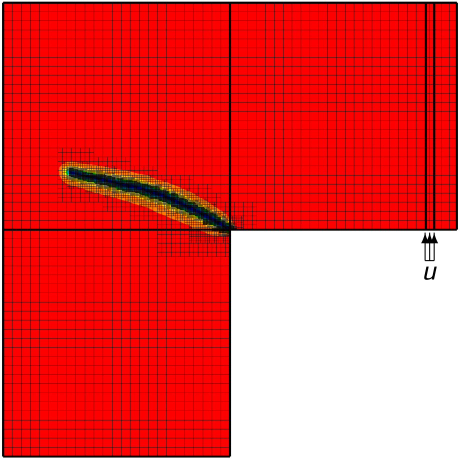
Final grid

L-shaped domain, 5 patches, $3 \times 25 \times 25$, $p = 1$



Initial grid

L-shaped domain, 5 patches, $3 \times 25 \times 25$, $p = 1$

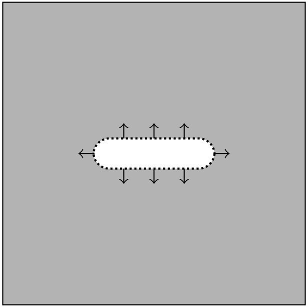


Final grid

The Sneddon case

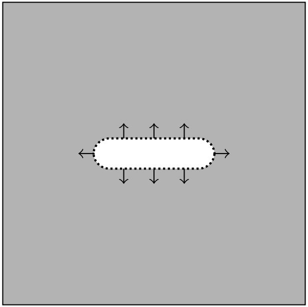
The Sneddon Case

- Square domain with a pre-formed horizontal crack in the middle, with some prescribed half-length ℓ.
- The crack is pressurized with a flux, causing widening and potentially crack propagation.



The Sneddon Case

- Square domain with a pre-formed horizontal crack in the middle, with some prescribed half-length ℓ.
- The crack is pressurized with a flux, causing widening and potentially crack propagation.



The Sneddon case

- Has some theoretical results associated with a fracture of the "interface" type.
- Literature contains a rich variety of vaguely dissimilar parameters, modeling choices and assumptions.
- Makes direct comparisons quite challenging.

Author	ℓ	L	h	ℓ_0	E	ν	$\mathcal{G}_c$
Bourdin (A)	0.2	4		0.01	1 Pa		1 N/m
Bourdin (B)	0.2	8	$0.02\overline{2}$		1 Pa		1 N/m
Lee	0.2	4	$0.02\overline{2}$	0.045	1 Pa	0.2	1 N/m
Singh	0.2		$0.0\overline{2}$		10 GPa	0.3	100 N/m
Us (A)	0.2	4	0.025	0.025	1 Pa	0.2	1 N/m
Us (B)	0.2	4	0.05	0.05	10 GPa	0.0	1 N/m

The Sneddon case

- Has some theoretical results associated with a fracture of the "interface" type.
- Literature contains a rich variety of vaguely dissimilar parameters, modeling choices and assumptions.
- Makes direct comparisons quite challenging.

Author	ℓ	L	h	ℓ_0	E	ν	$\mathcal{G}_c$
Bourdin (A)	0.2	4		0.01	1 Pa		1 N/m
Bourdin (B)	0.2	8	$0.02\overline{2}$		1 Pa		1 N/m
Lee	0.2	4	$0.02\overline{2}$	0.045	1 Pa	0.2	1 N/m
Singh	0.2		$0.0\overline{2}$		10 GPa	0.3	100 N/m
Us (A)	0.2	4	0.025	0.025	1 Pa	0.2	1 N/m
Us (B)	0.2	4	0.05	0.05	10 GPa	0.0	1 N/m

The Sneddon case

- Has some theoretical results associated with a fracture of the "interface" type.
- Literature contains a rich variety of vaguely dissimilar parameters, modeling choices and assumptions.
- Makes direct comparisons quite challenging.

Author	ℓ	L	h	ℓ_0	E	ν	$\mathcal{G}_c$
Bourdin (A)	0.2	4		0.01	1 Pa		1 N/m
Bourdin (B)	0.2	8	$0.02\overline{2}$		1 Pa		1 N/m
Lee	0.2	4	$0.02\overline{2}$	0.045	1 Pa	0.2	1 N/m
Singh	0.2		$0.0\overline{2}$		10 GPa	0.3	100 N/m
Us (A)	0.2	4	0.025	0.025	1 Pa	0.2	1 N/m
Us (B)	0.2	4	0.05	0.05	10 GPa	0.0	1 N/m

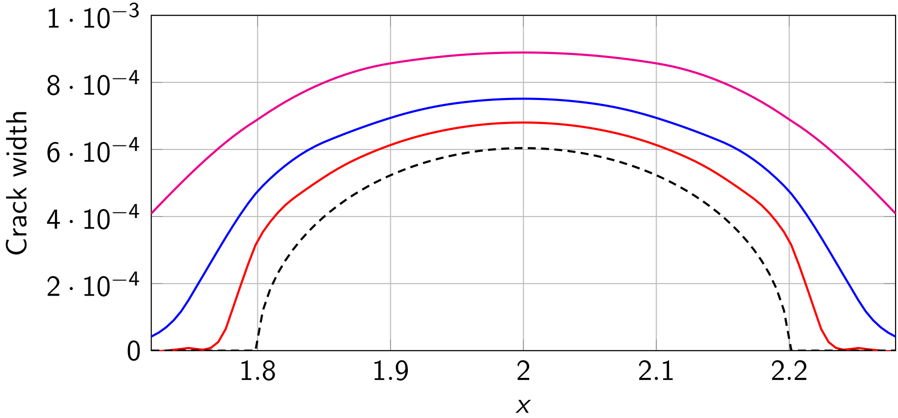
The Sneddon case

- Has some theoretical results associated with a fracture of the "interface" type.
- Literature contains a rich variety of vaguely dissimilar parameters, modeling choices and assumptions.
- Makes direct comparisons quite challenging.

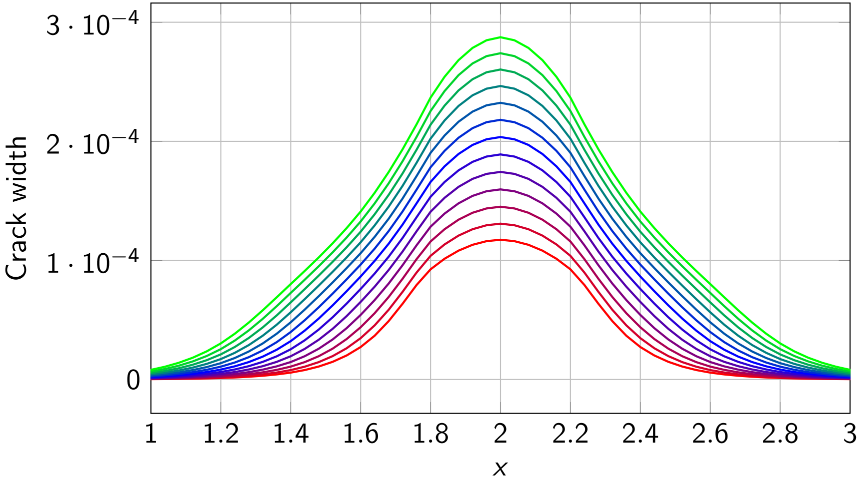
Author	ℓ	L	h	ℓ_0	E	ν	$\mathcal{G}_c$
Bourdin (A)	0.2	4		0.01	1 Pa		1 N/m
Bourdin (B)	0.2	8	$0.02\overline{2}$		1 Pa		1 N/m
Lee	0.2	4	$0.02\overline{2}$	0.045	1 Pa	0.2	1 N/m
Singh	0.2		$0.0\overline{2}$		10 GPa	0.3	100 N/m
Us (A)	0.2	4	0.025	0.025	1 Pa	0.2	1 N/m
Us (B)	0.2	4	0.05	0.05	10 GPa	0.0	1 N/m

Crack width for various meshes ($E = 1\,\mathrm{Pa}$, $\mathcal{G}_c = 1\,\mathrm{N/m}$)

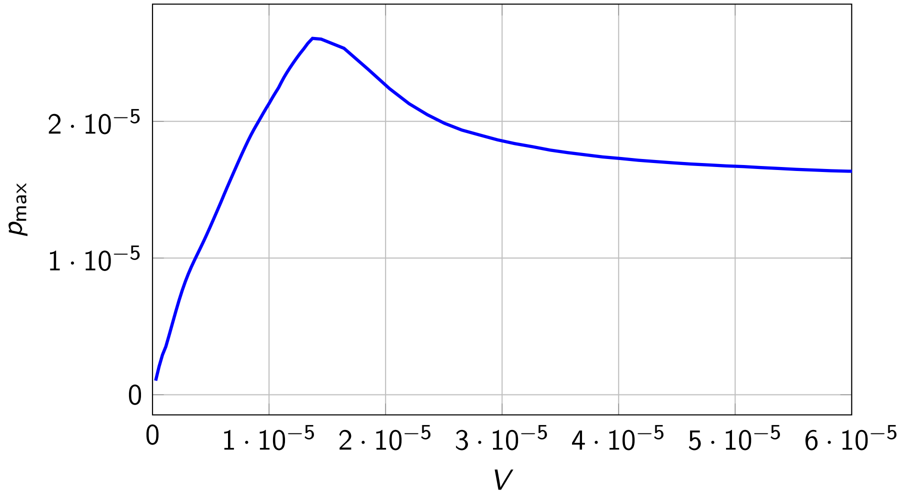
Crack width for $N = 40, 80, 160$ elements compared to theoretical



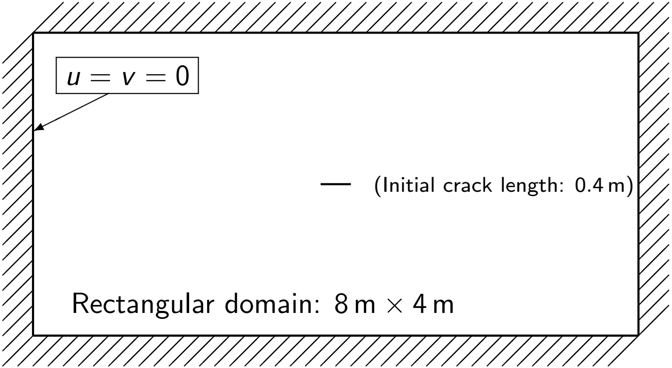
Crack width for various times ($E = 10\,\text{GPa}$, $\mathcal{G}_c = 1\,\text{N/m}$)



Pressure vs. crack volume



The Sneddon case: Internal crack with injected fluid



M.F. Wheeler, T. Wick, W. Wollner. An augmented-Lagrangian method for the phase-field approach for pressurized fractures. Comput. Methods Appl. Mech. Engrg. 271 (2014) 69–85.

The Sneddon case: Internal crack with injected fluid

Material parameters

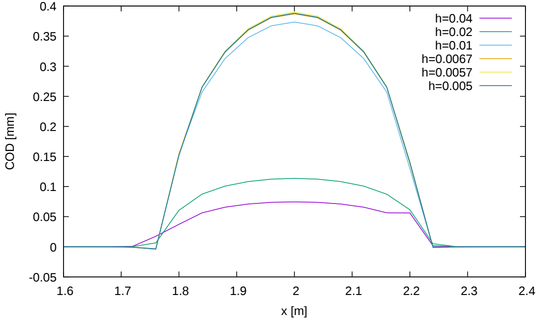
$$E = 1\,\text{Pa}$$
$$\nu = 0.2$$
$$\mathcal{G}_c = 1\,\text{N/m}$$
$$\ell_0 = 5\,\text{mm}$$

Injected fluid pressure

$$p(t) \rightarrow \mathbf{f}_p = \int_\Omega p(t) \nabla c \mathbf{N}$$

- $p = 0.001$ (constant)
- $p = t$ (linearly increasing)

Sneddon: Constant pressure case



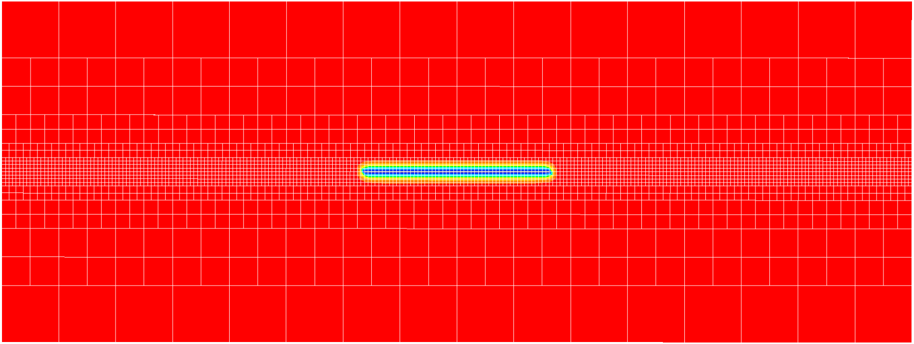
Calculated crack opening displacement:

$$\text{COD}(x) = \int_y \mathbf{u}(x, y) \cdot \nabla c(x, y)$$

Sneddon: Constant pressure case

- In this test we used a square domain (4×4)
- Uniform meshes:
  $h = 0.04 \Rightarrow 100 \times 100$ elements,
  $h = 0.005 \Rightarrow 800 \times 800$ elements.
- Seems to converge for $h < 0.0067$ $(600 \times 600$ elements).

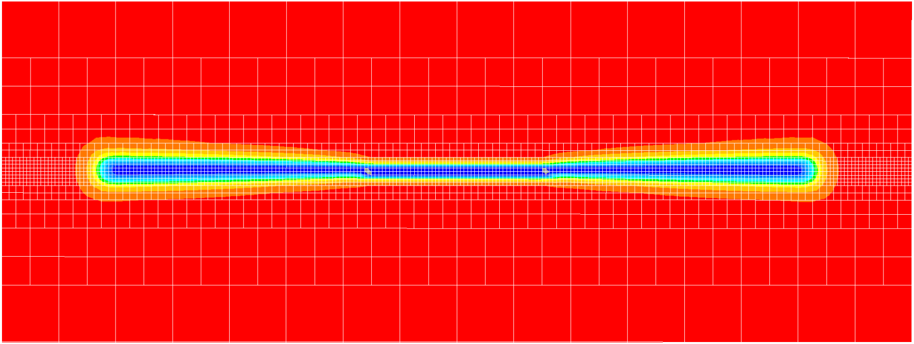
Sneddon: linearly increasing pressure



Initial phase field.

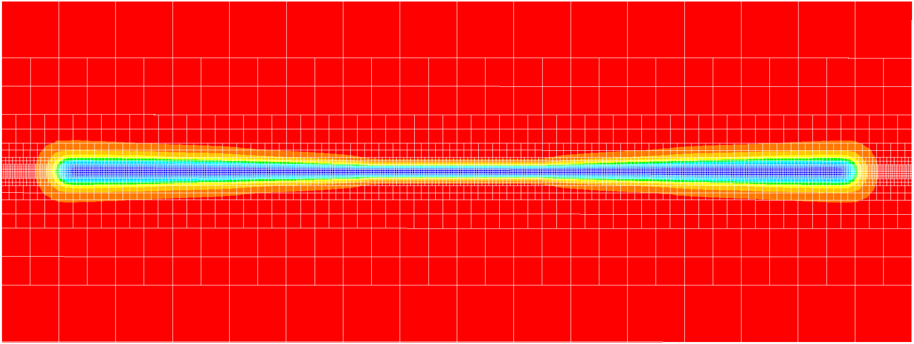
- Uniform background mesh, 32×16 elements $\rightarrow h = 0.25$
- 5 levels pre-refinement along center line $\rightarrow h_{\text{min}} = 00078125$

Sneddon: linearly increasing pressure



Phase field at $p = t = 2.12$

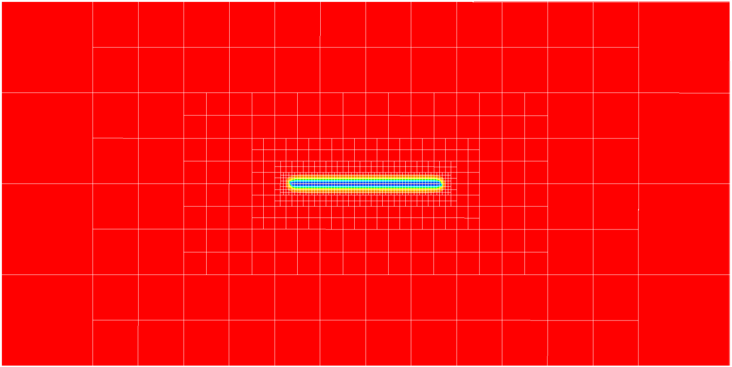
Sneddon: linearly increasing pressure



Phase field at $p = t = 2.05$.

- Uniform background mesh, 32×16 elements $\rightarrow h = 0.25$
- 6 levels pre-refinement along center line $\rightarrow h_{\text{min}} = 0.00390625$

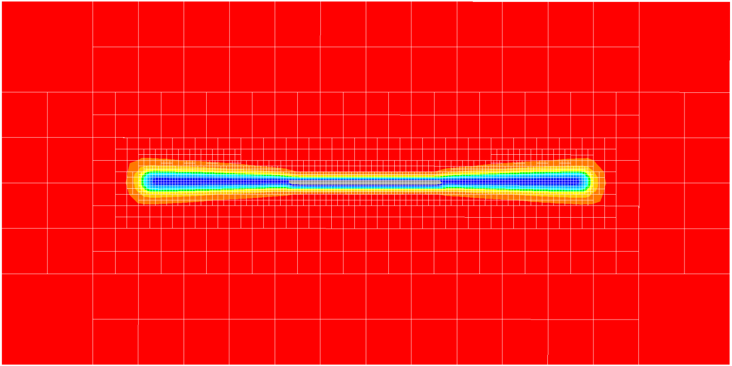
Sneddon: linearly increasing pressure



Phase field at $p = t = 0$ on initial mesh.

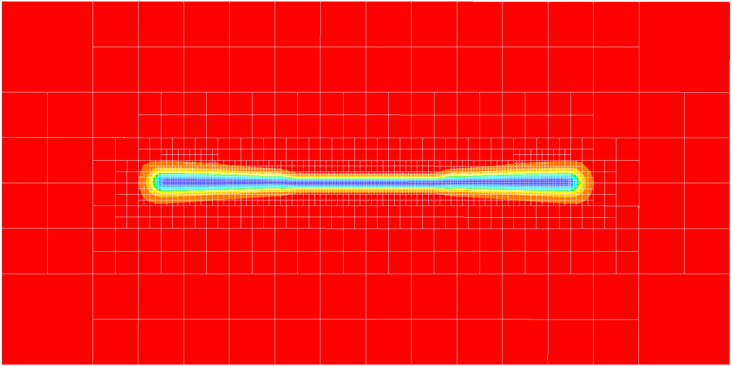
- Uniform background mesh, 32×16 elements $\rightarrow h = 0.25$
- 5 levels adaptive refinement $\rightarrow h_{\text{min}} = 00078125$

Sneddon: linearly increasing pressure



Phase field at $t = 2.12$ on adapted mesh.

Sneddon: linearly increasing pressure

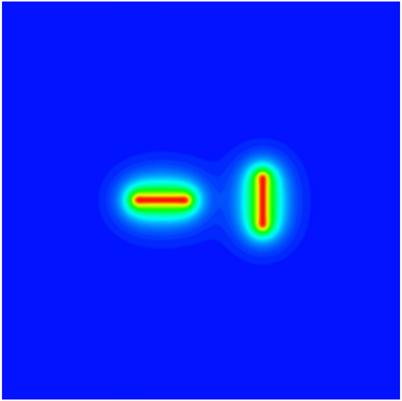


Phase field at $t = 2.05$ on adapted mesh.

- Uniform background mesh, 32×16 elements $\rightarrow h = 0.25$
- 6 levels adaptive refinement $\rightarrow h_{\min} = 0.00390625$

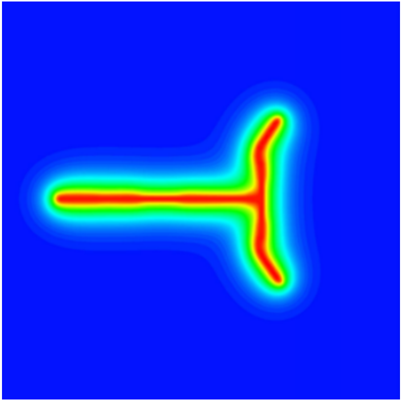
Perpendicular crack case

Perpendicular crack case



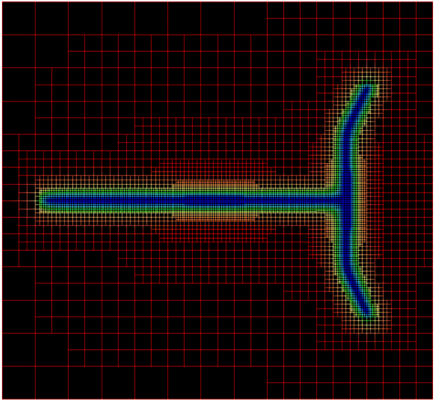
Figures reproduced from Miehe and Mauthe (2015)

Perpendicular crack case



Figures reproduced from Miehe and Mauthe (2015)

Perpendicular crack case



IFEM

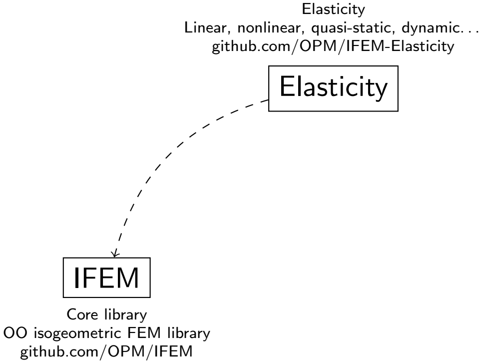
Implementation

IFEM

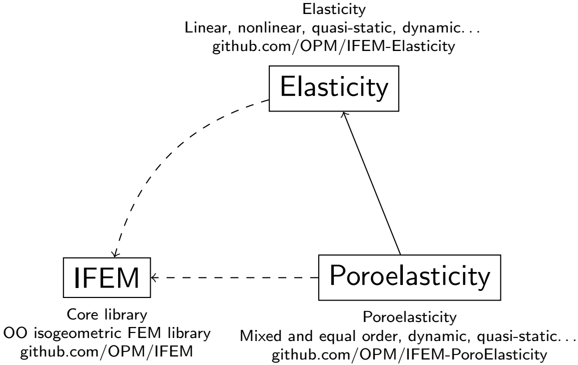
IFEM

Core library
OO isogeometric FEM library
github.com/OPM/IFEM

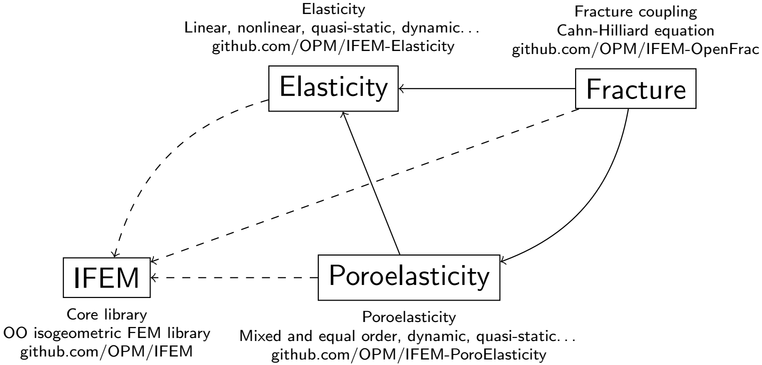
IFEM


Elasticity
Linear, nonlinear, quasi-static, dynamic. . .
github.com/OPM/IFEM-Elasticity
Elasticity
IFEM
Core library
OO isogeometric FEM library
github.com/OPM/IFEM

IFEM


Elasticity
Linear, nonlinear, quasi-static, dynamic. . .
github.com/OPM/IFEM-Elasticity
Elasticity
IFEM
Poroelasticity
Core library
OO isogeometric FEM library
github.com/OPM/IFEM
Poroelasticity
Mixed and equal order, dynamic, quasi-static. . .
github.com/OPM/IFEM-PoroElasticity

IFEM


Elasticity
Linear, nonlinear, quasi-static, dynamic. . .
github.com/OPM/IFEM-Elasticity
Fracture coupling
Cahn-Hilliard equation
github.com/OPM/IFEM-OpenFrac
Elasticity
Fracture
IFEM
Poroelasticity
Core library
OO isogeometric FEM library
github.com/OPM/IFEM
Poroelasticity
Mixed and equal order, dynamic, quasi-static. . .
github.com/OPM/IFEM-PoroElasticity

Basic usage

```
LinEl inputfile.xinp [OPTIONS...]
PoroElasticity inputfile.xinp [OPTIONS...]
FractureDynamics inputfile.xinp [OPTIONS...]
```

Basic usage

```
LinEl inputfile.xinp [OPTIONS...]
PoroElasticity inputfile.xinp [OPTIONS...]
FractureDynamics inputfile.xinp [OPTIONS...]
```

Input file in XML format specifying geometry, material parameters, boundary conditions, etc.

Basic usage

```
LinEl inputfile.xinp [OPTIONS...]
PoroElasticity inputfile.xinp [OPTIONS...]
FractureDynamics inputfile.xinp [OPTIONS...]
```

More significant simulation options are often given on the command line.

General command-line options

- `-hdf5`: Turn on output in HDF5 format. Use IFEM-to-VT[1] to convert to e.g. VTK.
- `-vtf (0|1)`: Output ASCII/binary VTF files.
- `-(dense|spr|superlu|samg|petsc)`: Choose linear solver backend. (PETSc recommended.)
- `-LR`: Use locally refined spline basis functions instead of tensorial splines. (Needed for adaptivity.)

[1] https://github.com/TheBB/IFEM-to-VT

General command-line options

- `-hdf5`: Turn on output in HDF5 format. Use IFEM-to-VT[1] to convert to e.g. VTK.
- `-vtf (0|1)`: Output ASCII/binary VTF files.
- `-(dense|spr|superlu|samg|petsc)`: Choose linear solver backend. (PETSc recommended.)
- `-LR`: Use locally refined spline basis functions instead of tensorial splines. (Needed for adaptivity.)

[1] https://github.com/TheBB/IFEM-to-VT

General command-line options

- `-hdf5`: Turn on output in HDF5 format. Use IFEM-to-VT[1] to convert to e.g. VTK.
- `-vtf (0|1)`: Output ASCII/binary VTF files.
- `-(dense|spr|superlu|samg|petsc)`: Choose linear solver backend. (PETSc recommended.)
- `-LR`: Use locally refined spline basis functions instead of tensorial splines. (Needed for adaptivity.)

[1] https://github.com/TheBB/IFEM-to-VT

General command-line options

- `-hdf5`: Turn on output in HDF5 format. Use IFEM-to-VT[1] to convert to e.g. VTK.
- `-vtf (0|1)`: Output ASCII/binary VTF files.
- `-(dense|spr|superlu|samg|petsc)`: Choose linear solver backend. (PETSc recommended.)
- `-LR`: Use locally refined spline basis functions instead of tensorial splines. (Needed for adaptivity.)

[1] https://github.com/TheBB/IFEM-to-VT

PoroElasticity command-line options

- `-2D`: Enable elastic plane strain.
- `-dyn`: Enable dynamic Newmark time-based solver.
- `-halfstatic`: Quasi-static elastic solver coupled with dynamic flow solver.
- `-fullstatic`: Fully quasi-static formulation.
- `-mixed`: Reduced continuity mixed order formulation.
- `-mixed-full`: Full continuity mixed order formulation.

PoroElasticity command-line options

- `-2D`: Enable elastic plane strain.
- `-dyn`: Enable dynamic Newmark time-based solver.
- `-halfstatic`: Quasi-static elastic solver coupled with dynamic flow solver.
- `-fullstatic`: Fully quasi-static formulation.
- `-mixed`: Reduced continuity mixed order formulation.
- `-mixed-full`: Full continuity mixed order formulation.

PoroElasticity command-line options

- `-2D`: Enable elastic plane strain.
- `-dyn`: Enable dynamic Newmark time-based solver.
- `-halfstatic`: Quasi-static elastic solver coupled with dynamic flow solver.
- `-fullstatic`: Fully quasi-static formulation.
- `-mixed`: Reduced continuity mixed order formulation.
- `-mixed-full`: Full continuity mixed order formulation.

PoroElasticity command-line options

- `-2D`: Enable elastic plane strain.
- `-dyn`: Enable dynamic Newmark time-based solver.
- `-halfstatic`: Quasi-static elastic solver coupled with dynamic flow solver.
- `-fullstatic`: Fully quasi-static formulation.
- `-mixed`: Reduced continuity mixed order formulation.
- `-mixed-full`: Full continuity mixed order formulation.

PoroElasticity command-line options

- `-2D`: Enable elastic plane strain.
- `-dyn`: Enable dynamic Newmark time-based solver.
- `-halfstatic`: Quasi-static elastic solver coupled with dynamic flow solver.
- `-fullstatic`: Fully quasi-static formulation.
- `-mixed`: Reduced continuity mixed order formulation.
- `-mixed-full`: Full continuity mixed order formulation.

PoroElasticity command-line options

- `-2D`: Enable elastic plane strain.
- `-dyn`: Enable dynamic Newmark time-based solver.
- `-halfstatic`: Quasi-static elastic solver coupled with dynamic flow solver.
- `-fullstatic`: Fully quasi-static formulation.
- `-mixed`: Reduced continuity mixed order formulation.
- `-mixed-full`: Full continuity mixed order formulation.

FractureDynamics command-line options

- `-adap`: Enable adaptivity
- `-poro`: Enable poroelastic backend (otherwise, pure elasticity is used)
- `-nocrack`: Disable fracture phase-field coupling.
- `-explcrack`: Enable explicit crack formulation.
- `-semiimplicit`: Enable semi-implicit coupling.

FractureDynamics command-line options

- `-adap`: Enable adaptivity
- `-poro`: Enable poroelastic backend (otherwise, pure elasticity is used)
- `-nocrack`: Disable fracture phase-field coupling.
- `-explcrack`: Enable explicit crack formulation.
- `-semiimplicit`: Enable semi-implicit coupling.

FractureDynamics command-line options

- `-adap`: Enable adaptivity
- `-poro`: Enable poroelastic backend (otherwise, pure elasticity is used)
- `-nocrack`: Disable fracture phase-field coupling.
- `-explcrack`: Enable explicit crack formulation.
- `-semiimplicit`: Enable semi-implicit coupling.

FractureDynamics command-line options

- `-adap`: Enable adaptivity
- `-poro`: Enable poroelastic backend (otherwise, pure elasticity is used)
- `-nocrack`: Disable fracture phase-field coupling.
- `-explcrack`: Enable explicit crack formulation.
- `-semiimplicit`: Enable semi-implicit coupling.

FractureDynamics command-line options

- `-adap`: Enable adaptivity
- `-poro`: Enable poroelastic backend (otherwise, pure elasticity is used)
- `-nocrack`: Disable fracture phase-field coupling.
- `-explcrack`: Enable explicit crack formulation.
- `-semiimplicit`: Enable semi-implicit coupling.

Input file format

A basic overview follows. More information can be found on the OPM IFEM wiki.[2]. The file is constituted of several *contexts*.

```
<simulation>
  <context1>
    settings...
  </context1>
  <context2>
    settings...
  </context2>
  more contexts...
</simulation>
```

[2] https://github.com/opm/ifem/wiki

Geometry context

Example: a geometry of five patches stored in `foo.g2`,[3] split over two processes, where the first three patches should be *hp*-refined.

```
<geometry>
  <partitioning procs="2">
    <part proc="0" lower="1" upper="3"/>
    <part proc="1" lower="4" upper="5"/>
  </partitioning>
  <patchfile>foo.g2</patchfile>
  <refine lowerpatch="1" upperpatch="3" u="1" v="2" w="3"/>
  <raiseorder lowerpatch="1" upperpatch="3" u="1" v="2" w="3"/>
</geometry>
```

[3] Splipy can be used to generate G2 files: https://github.com/sintefmath/Splipy

Topology sets

Use topology sets to bundle boundary components into named units for easier application of boundary conditions.

```
<geometry>
  <topologysets>
    <set name="myset" type="face">
      <item patch="1">1 2 3</item>
    </set>
    <set name="yourset" type="edge">
      <item patch="1">4</item>
    </set>
    <set name="theirset" type="vertex">
      <item patch="2">5</item>
    </set>
  </topologysets>
</geometry>
```

Patch topology

For multipatch geometries, patch-to-patch topology must be specified.

```
<geometry>
  <topology>
    <connection master="1" midx="4" slave="2" sidx="3" />
    ...
  </topology>
</geometry>
```

Boundary conditions

Use the <boundaryconditions> context to apply boundary conditions.

```
<boundaryconditions>
  <dirichlet set="myset" basis="1" comp="2"/>
  <neumann set="yourset" comp="12" direction="0">-500</neumann>
  <neumann set="theirset" type="expression">x * y * z</neumann>
</boundaryconditions>
```

Timestepping

Use the <timestepping> context. It's quite simple.

```
<timestepping>
  <step start="0.0" end="0.5" dt="0.05"/>
  <step start="0.5" end="2.0" dt="0.1"/>
</timestepping>
```

Other contexts of interest

- <adaptive>: Fine-tune parameters for adaptive mesh refinement.
- <initialconditions>: Set initial conditions for time-stepper.
- <linearsolver>: Fine-tune parameters for the linear solver (preconditioners, tolerances, number of iterations, multigrid. . .)
- <newmarksolver>: Parameters for the dynamic Newmark time-stepping algorithm.
- <postprocessing>: Can be used to specify extra output options, such as sampling resolution, projections used for recovery of secondary solutions and adaptive simulations, or debug output of the LHS and RHS.
- <restart>: Used to restart a simulation from the last state of a previous run.

Other contexts of interest

- <adaptive>: Fine-tune parameters for adaptive mesh refinement.
- <initialconditions>: Set initial conditions for time-stepper.
- <linearsolver>: Fine-tune parameters for the linear solver (preconditioners, tolerances, number of iterations, multigrid. . .)
- <newmarksolver>: Parameters for the dynamic Newmark time-stepping algorithm.
- <postprocessing>: Can be used to specify extra output options, such as sampling resolution, projections used for recovery of secondary solutions and adaptive simulations, or debug output of the LHS and RHS.
- <restart>: Used to restart a simulation from the last state of a previous run.

Other contexts of interest

- <adaptive>: Fine-tune parameters for adaptive mesh refinement.
- <initialconditions>: Set initial conditions for time-stepper.
- <linearsolver>: Fine-tune parameters for the linear solver (preconditioners, tolerances, number of iterations, multigrid. . .)
- <newmarksolver>: Parameters for the dynamic Newmark time-stepping algorithm.
- <postprocessing>: Can be used to specify extra output options, such as sampling resolution, projections used for recovery of secondary solutions and adaptive simulations, or debug output of the LHS and RHS.
- <restart>: Used to restart a simulation from the last state of a previous run.

Other contexts of interest

- <adaptive>: Fine-tune parameters for adaptive mesh refinement.
- <initialconditions>: Set initial conditions for time-stepper.
- <linearsolver>: Fine-tune parameters for the linear solver (preconditioners, tolerances, number of iterations, multigrid. . .)
- <newmarksolver>: Parameters for the dynamic Newmark time-stepping algorithm.
- <postprocessing>: Can be used to specify extra output options, such as sampling resolution, projections used for recovery of secondary solutions and adaptive simulations, or debug output of the LHS and RHS.
- <restart>: Used to restart a simulation from the last state of a previous run.

Other contexts of interest

- <adaptive>: Fine-tune parameters for adaptive mesh refinement.
- <initialconditions>: Set initial conditions for time-stepper.
- <linearsolver>: Fine-tune parameters for the linear solver (preconditioners, tolerances, number of iterations, multigrid...)
- <newmarksolver>: Parameters for the dynamic Newmark time-stepping algorithm.
- <postprocessing>: Can be used to specify extra output options, such as sampling resolution, projections used for recovery of secondary solutions and adaptive simulations, or debug output of the LHS and RHS.
- <restart>: Used to restart a simulation from the last state of a previous run.

Other contexts of interest

- `<adaptive>`: Fine-tune parameters for adaptive mesh refinement.
- `<initialconditions>`: Set initial conditions for time-stepper.
- `<linearsolver>`: Fine-tune parameters for the linear solver (preconditioners, tolerances, number of iterations, multigrid. . .)
- `<newmarksolver>`: Parameters for the dynamic Newmark time-stepping algorithm.
- `<postprocessing>`: Can be used to specify extra output options, such as sampling resolution, projections used for recovery of secondary solutions and adaptive simulations, or debug output of the LHS and RHS.
- `<restart>`: Used to restart a simulation from the last state of a previous run.