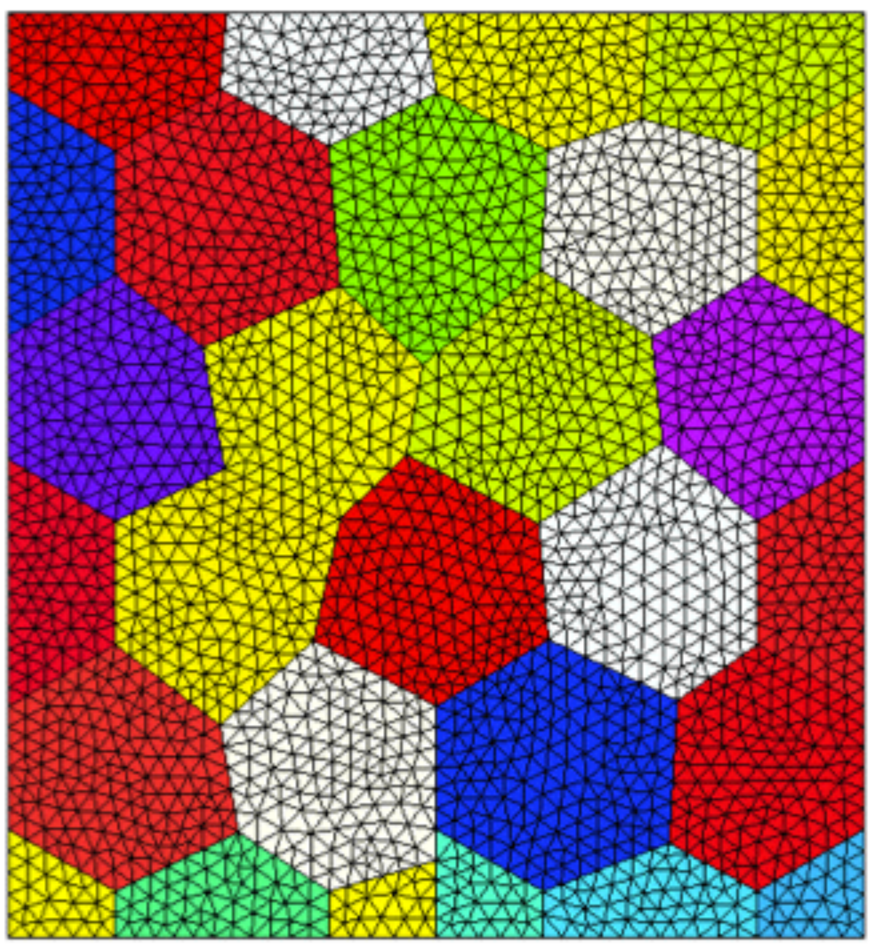
Crack modelling using zero-thickness interface elements

Nguyen Vinh Phu

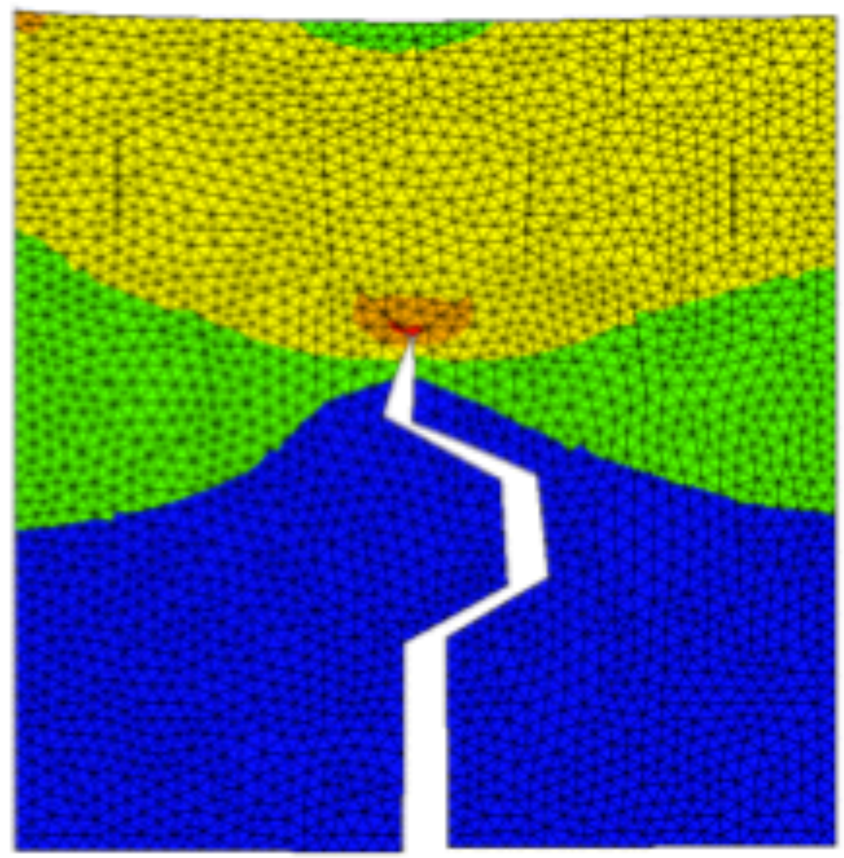
The University of Adelaide

Introduction

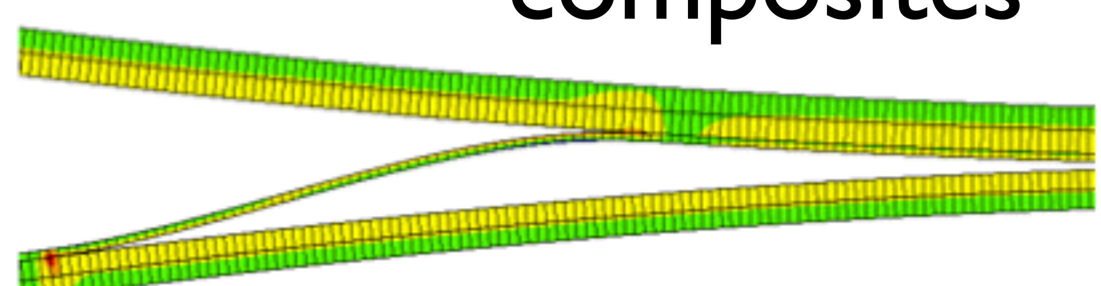
There exists problems in which the crack path is known in advance...



inter-granular cracking in polycrystals



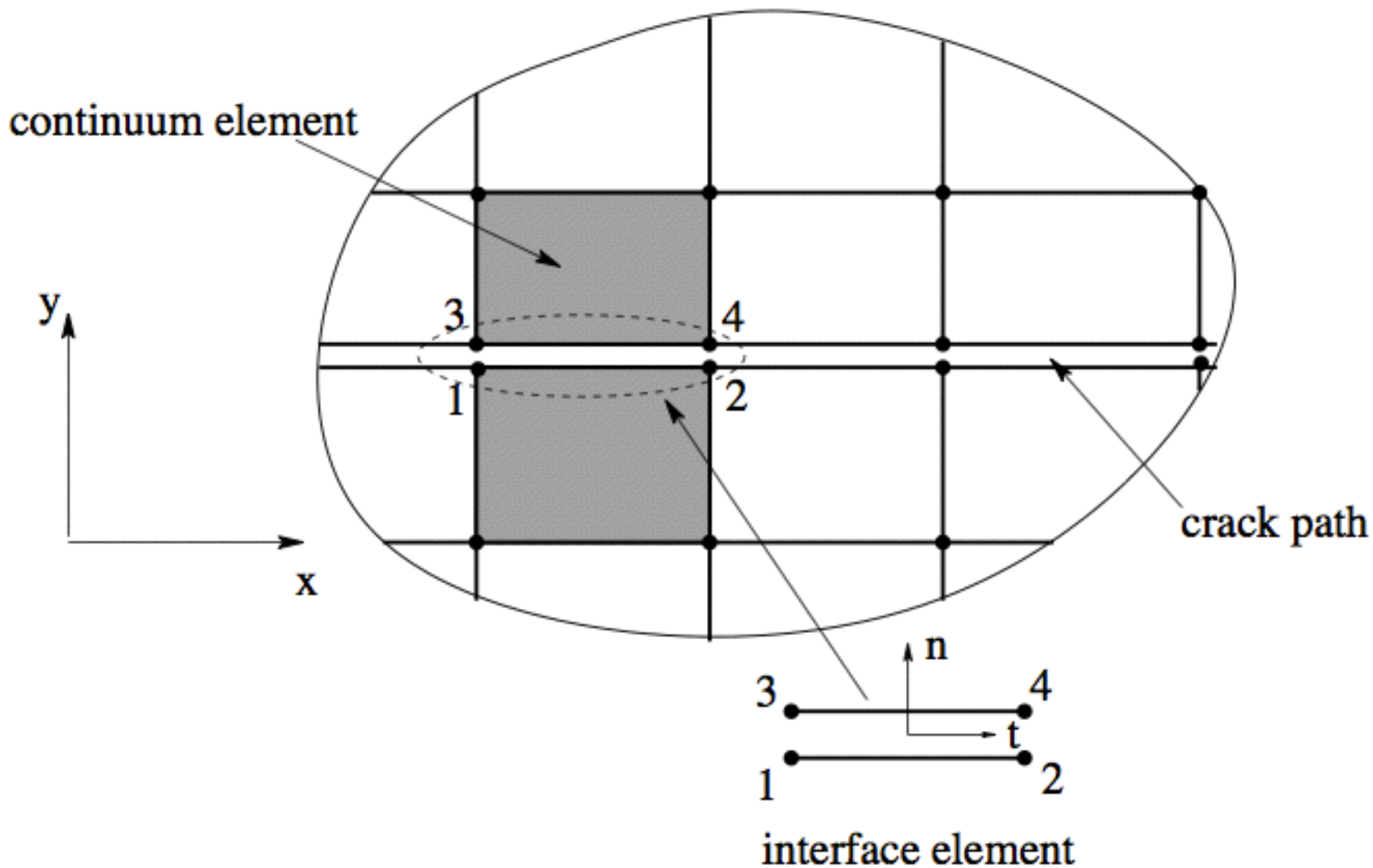
delaminated composites



Introduction (cont.)

...then the use of interface elements is recommended

- easy to implement (2D, 3D)
- available in major FE packages: ABAQUS, LS-DYNA

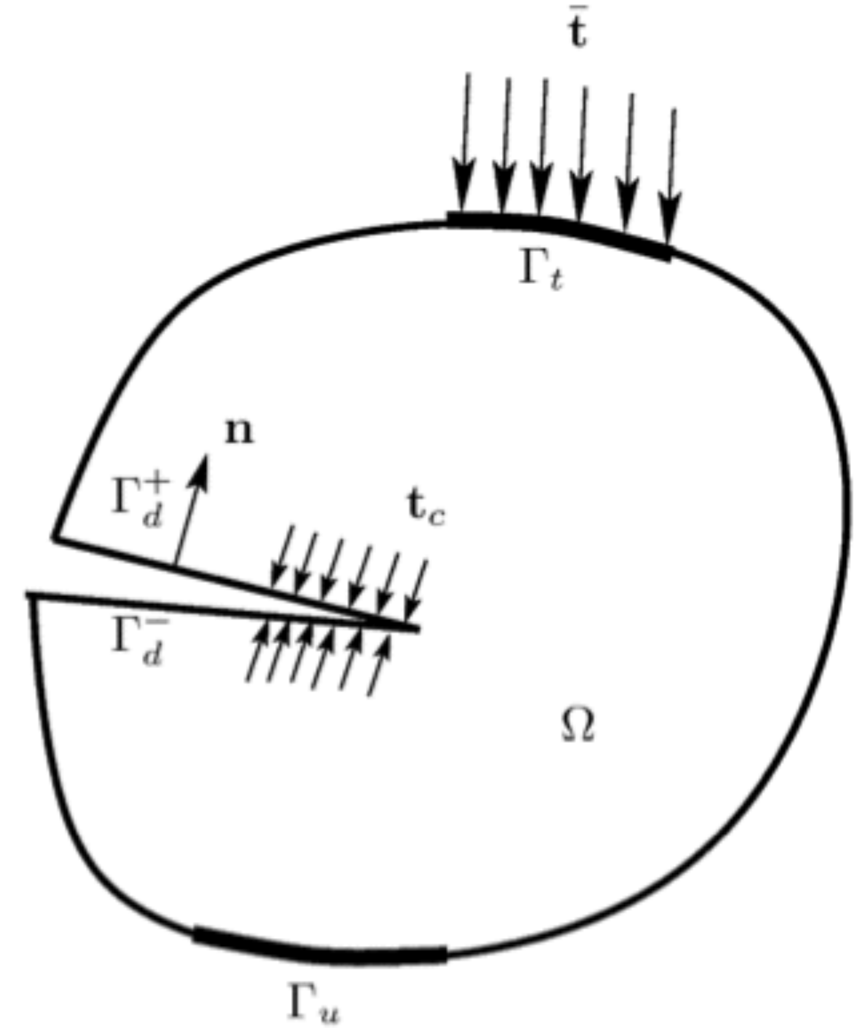


Alternatives:
XFEM/GFEM

Cohesive crack model

Governing equations (strong form)

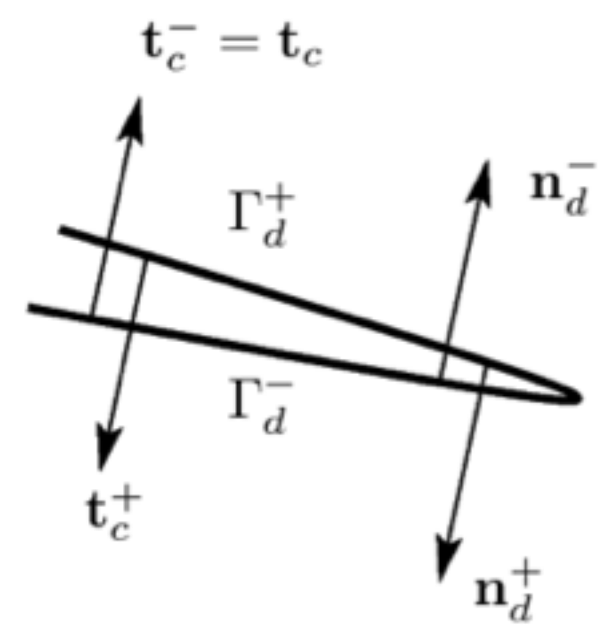
$$
\begin{aligned}
\nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{b} - \rho \ddot{\mathbf{u}} &= 0 \quad \mathbf{x} \in \Omega \\
\mathbf{n} \cdot \boldsymbol{\sigma} &= \bar{\mathbf{t}} \quad \mathbf{x} \in \Gamma_t \\
\mathbf{u} &= \bar{\mathbf{u}} \quad \mathbf{x} \in \Gamma_u \\
\mathbf{n}_d^+ \cdot \boldsymbol{\sigma} = \mathbf{t}_c^+; \quad \mathbf{n}_d^- \cdot \boldsymbol{\sigma} &= \mathbf{t}_c^-; \quad \mathbf{t}_c^+ = -\mathbf{t}_c = -\mathbf{t}_c^- \quad \mathbf{x} \in \Gamma_d
\end{aligned}
$$



Constitutive equations

$\dot{\boldsymbol{\sigma}} = \mathbf{D}\dot{\boldsymbol{\epsilon}} \longrightarrow$ deformation

$\dot{\mathbf{t}}^c = \mathbf{T}[\![\dot{\mathbf{u}}]\!] \longrightarrow$ separation



Cohesive crack model

Weak form

$$\delta W^{\mathrm{kin}} = \delta W^{\mathrm{ext}} - \delta W^{\mathrm{int}} - \delta W^{\mathrm{coh}}$$

new term

where

$$\delta W^{\mathrm{kin}} = \int_{\Omega} \delta \mathbf{u} \cdot \rho \ddot{\mathbf{u}} \mathrm{d}\Omega$$

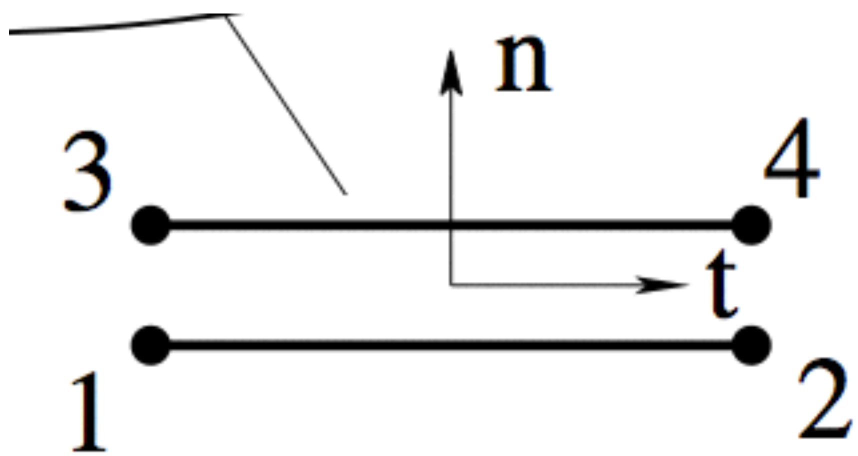
(skipped for static problems)

$$\delta W^{\mathrm{int}} = \int_{\Omega} \nabla^s \delta \mathbf{u} : \boldsymbol{\sigma} \mathrm{d}\Omega$$

$$\delta W^{\mathrm{ext}} = \int_{\Omega} \delta \mathbf{u} \cdot \rho \mathbf{b} \mathrm{d}\Omega + \int_{\Gamma_t} \delta \mathbf{u} \cdot \bar{\mathbf{t}} \mathrm{d}\Gamma_t$$

$$\delta W^{\mathrm{coh}} = \int_{\Gamma_d} \delta [\![\mathbf{u}]\!] \cdot \mathbf{t}^{\mathrm{c}} \mathrm{d}\Gamma_d$$

Discrete equations



interface element

$$\mathbf{u}^+ = \mathbf{N}^{\text{int}}\mathbf{u}^+ \quad \text{upper face}$$

$$\mathbf{u}^- = \mathbf{N}^{\text{int}}\mathbf{u}^- \quad \text{lower face}$$

$$\mathbf{N}^{\text{int}} = \begin{bmatrix} N_1 & 0 & N_2 & 0 \\ 0 & N_1 & 0 & N_2 \end{bmatrix}$$

$$N_1 = 0.5(1-\xi)$$
$$N_2 = 0.5(1+\xi)$$

$$[\![\mathbf{u}(\mathbf{x})]\!] = \mathbf{u}^+ - \mathbf{u}^- = \mathbf{N}^{\text{int}}(\mathbf{u}^+ - \mathbf{u}^-)$$

Solid elements

$$\mathbf{u} = N_I \mathbf{u}_I, \quad \delta\mathbf{u} = N_I \delta\mathbf{u}_I$$

Discrete equation

$$\mathbf{Ma} = \mathbf{f}^{\text{ext}} - \mathbf{f}^{\text{int}} - \mathbf{f}^{\text{coh}}$$

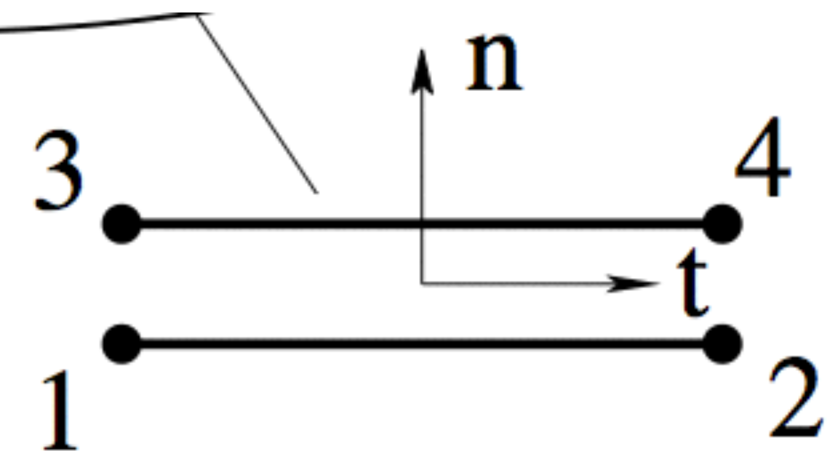
$$\boxed{\begin{aligned} \mathbf{M}_e &= \int_{\Omega_e} \rho \mathbf{N}^{\mathrm{T}}\mathbf{N}\, d\Omega_e \\ \mathbf{f}_e^{\text{int}} &= \int_{\Omega_e} \mathbf{B}^{\mathrm{T}}\boldsymbol{\sigma}\, d\Omega_e \\ \mathbf{f}_e^{\text{ext}} &= \int_{\Omega_e} \rho \mathbf{N}^{\mathrm{T}}\mathbf{b}\, d\Omega_e + \int_{\Gamma_t^e} \mathbf{N}^{\mathrm{T}}\bar{\mathbf{t}}\, d\Gamma_t^e \end{aligned}}$$

Discrete equations (cont.)

assembled

$$\mathbf{f}^{\mathrm{coh}}_{ie,+} = \int_\Gamma \mathbf{N}^{\mathrm{T}}_{\mathrm{int}} \mathbf{t}^{c} \mathrm{d}\Gamma \longrightarrow$$

$$\mathbf{f}^{\mathrm{coh}}_{ie,-} = -\int_\Gamma \mathbf{N}^{\mathrm{T}}_{\mathrm{int}} \mathbf{t}^{c} \mathrm{d}\Gamma \longrightarrow$$



interface element

Static problems

$$\mathbf{f}^{\mathrm{ext}} = \mathbf{f}^{\mathrm{int}} + \mathbf{f}^{\mathrm{coh}}$$

Linearization (Newton-Raphson)

$$\dot{\mathbf{t}}^{c} = \mathbf{T}[\![\dot{\mathbf{u}}]\!]$$

$$\dot{\mathbf{t}}^{c} = \mathbf{T}\mathbf{N}^{\mathrm{int}}(\dot{\mathbf{u}}^{+} - \dot{\mathbf{u}}^{-})$$

$$\dot{\mathbf{t}}^{c} = \mathbf{Q}\mathbf{T}\mathbf{Q}^{\mathrm{T}}\mathbf{N}^{\mathrm{int}}(\dot{\mathbf{u}}^{+} - \dot{\mathbf{u}}^{-})$$

transformation matrix

$$\mathbf{Q} = \begin{bmatrix} \cos(\mathbf{i}_1, \mathbf{n}) & \cos(\mathbf{i}_1, \mathbf{s}) \\ \cos(\mathbf{i}_2, \mathbf{n}) & \cos(\mathbf{i}_2, \mathbf{s}) \end{bmatrix}$$

$$\mathbf{f}^{\mathrm{coh}}_{ie,+} = \int_{\Gamma} \mathbf{N}^{\mathrm{T}}_{\mathrm{int}} \mathbf{t}^{\mathrm{c}} \mathrm{d}\Gamma$$

$$\mathbf{f}^{\mathrm{coh}}_{ie,-} = -\int_{\Gamma} \mathbf{N}^{\mathrm{T}}_{\mathrm{int}} \mathbf{t}^{\mathrm{c}} \mathrm{d}\Gamma$$

$$\dot{\mathbf{t}}^{\mathrm{c}} = \mathbf{Q}\mathbf{T}\mathbf{Q}^{\mathrm{T}}\mathbf{N}^{\mathrm{int}}(\dot{\mathbf{u}}^{+} - \dot{\mathbf{u}}^{-})$$

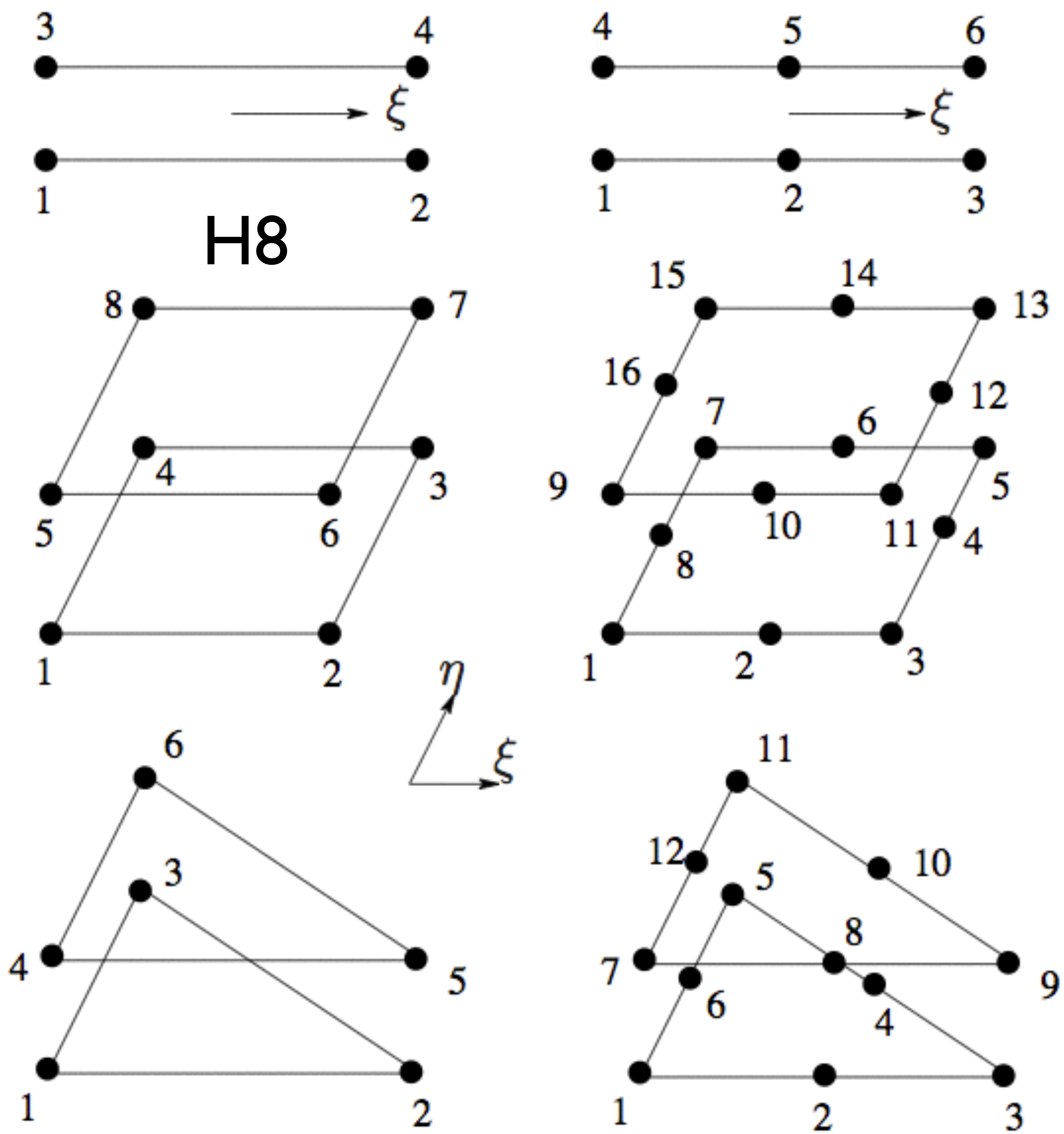
$$\begin{bmatrix} \dfrac{\partial \mathbf{f}^{\mathrm{coh}}_{ie,+}}{\partial \mathbf{u}} \\[2ex] \dfrac{\partial \mathbf{f}^{\mathrm{coh}}_{ie,-}}{\partial \mathbf{u}} \end{bmatrix} = \begin{bmatrix} \dfrac{\partial \mathbf{f}^{\mathrm{coh}}_{ie,+}}{\partial \mathbf{u}^{+}} & \dfrac{\partial \mathbf{f}^{\mathrm{coh}}_{ie,+}}{\partial \mathbf{u}^{-}} \\[2ex] \dfrac{\partial \mathbf{f}^{\mathrm{coh}}_{ie,-}}{\partial \mathbf{u}^{+}} & \dfrac{\partial \mathbf{f}^{\mathrm{coh}}_{ie,-}}{\partial \mathbf{u}^{-}} \end{bmatrix} \begin{bmatrix} \delta \mathbf{u}^{+} \\ \delta \mathbf{u}^{-} \end{bmatrix}$$

```
// assembly of stiffness due to cohesive tractio

mbuilder.addBlock ( idofsA, idofsA, elemMat );
mbuilder.addBlock ( idofsB, idofsB, elemMat );

elemMat = -elemMat;

mbuilder.addBlock ( idofsA, idofsB, elemMat );
mbuilder.addBlock ( idofsB, idofsA, elemMat );
```

$$\mathbf{K}^{int}_{e} = \begin{bmatrix} \boxed{\int_{\Gamma} \mathbf{N}^{\mathrm{T}}\mathbf{Q}^{\mathrm{T}}\mathbf{T}\mathbf{Q}\mathbf{N} d\Gamma} & -\int_{\Gamma} \mathbf{N}^{\mathrm{T}}\mathbf{Q}^{\mathrm{T}}\mathbf{T}\mathbf{Q}\mathbf{N} d\Gamma \\ -\int_{\Gamma} \mathbf{N}^{\mathrm{T}}\mathbf{Q}^{\mathrm{T}}\mathbf{T}\mathbf{Q}\mathbf{N} d\Gamma & \int_{\Gamma} \mathbf{N}^{\mathrm{T}}\mathbf{Q}^{\mathrm{T}}\mathbf{T}\mathbf{Q}\mathbf{N} d\Gamma \end{bmatrix}$$

Common interface elements

Solid elements Q4/T3 Q8/T6



2D

3D

Numerical integration

It has been observed numerically that integrating the internal force and stiffness matrix of interface elements using the standard Gauss rule led to oscillatory response [de Borst, IJNME, 1993].

$\longrightarrow$ Newton-Cotes integration scheme
for interface elements

Cohesive laws

Mode I Bilinear cohesive law (traction-separation law)

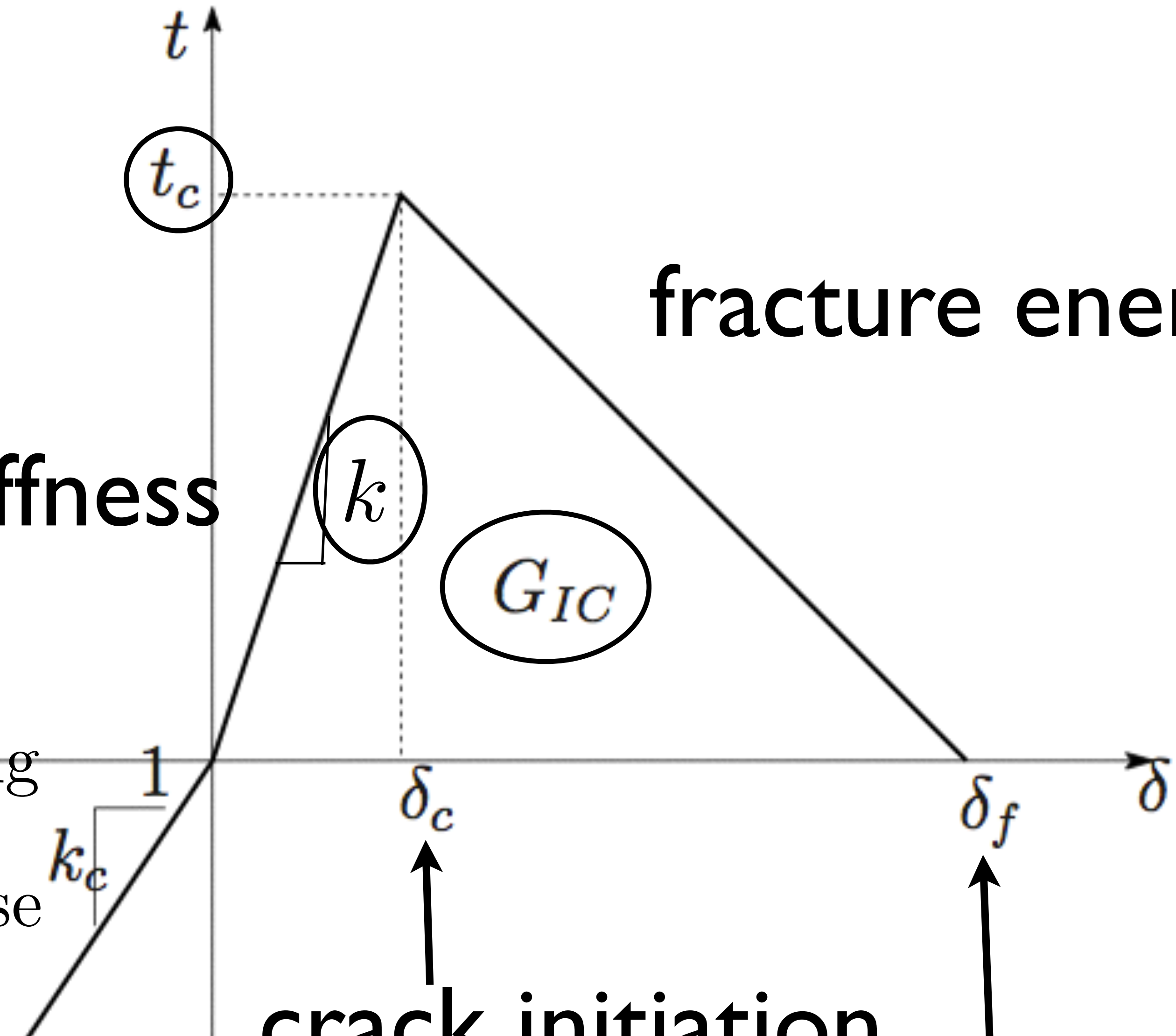
$$G_{Ic} = \frac{1}{2} t_c \delta_f = \frac{1}{2} k \delta_c \delta_f$$

$$\boxed{\delta_f = \frac{2G_{Ic}}{k\delta_c}} \qquad \boxed{\delta_c = \frac{t_c}{k}}$$

tensile strength

fracture energy

elastic stiffness



$$t = (1-d)k\delta$$

$$d = \begin{cases} \frac{\delta_f}{\delta}\kappa & \text{if loading} \\ \frac{\kappa_{max}\delta_f}{\kappa_{max}(\delta_f - \delta_c) + \delta_c} & \text{otherwise} \end{cases}$$

crack initiation

$$\kappa = \frac{\delta - \delta_c}{\delta_f - \delta_c}$$

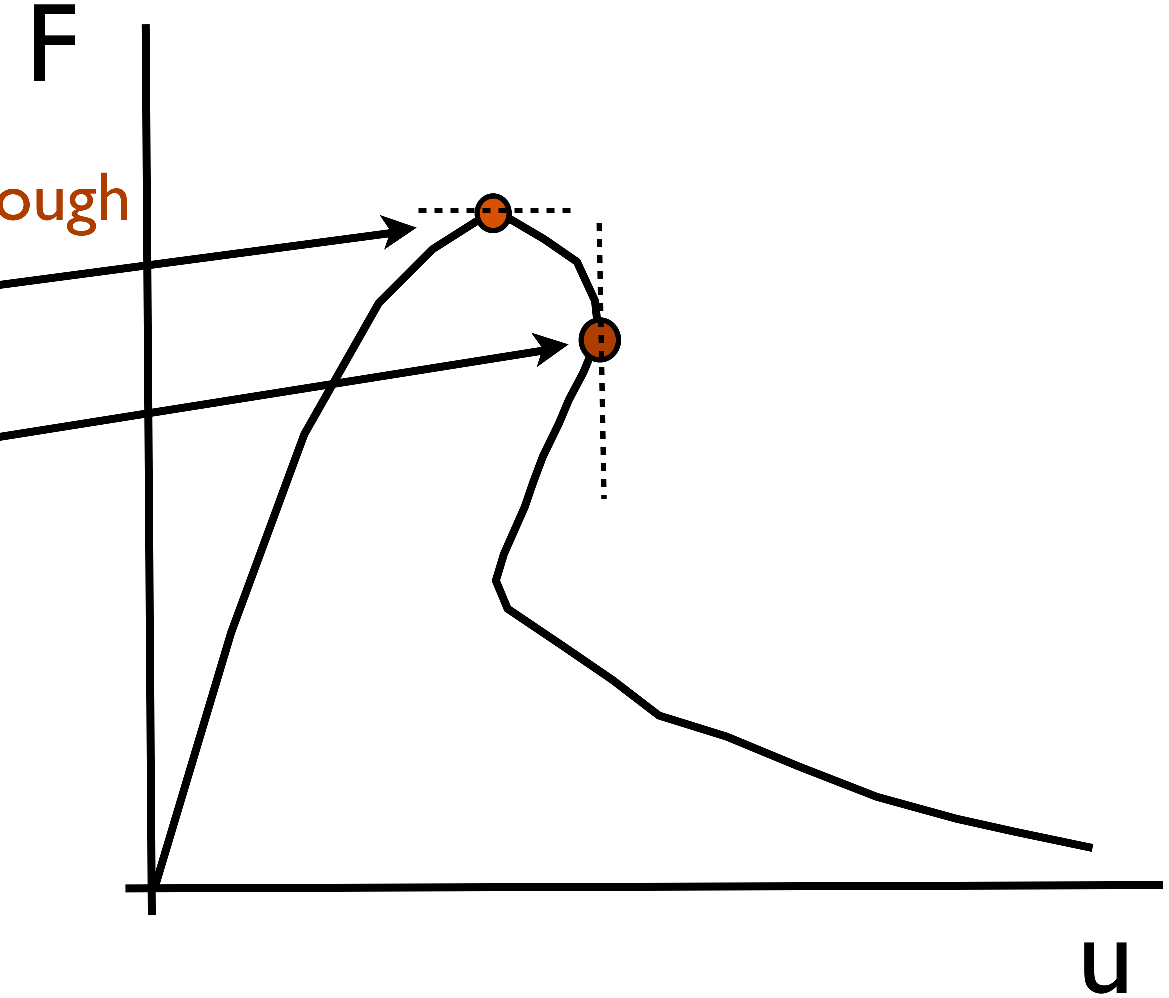
crack propagation

Implementation aspects

- How to generate interface element meshes?
- Solution control

cannot pass the snap-through

  - load control: NO
  - displacement control
  - path following control (arc-length methods)

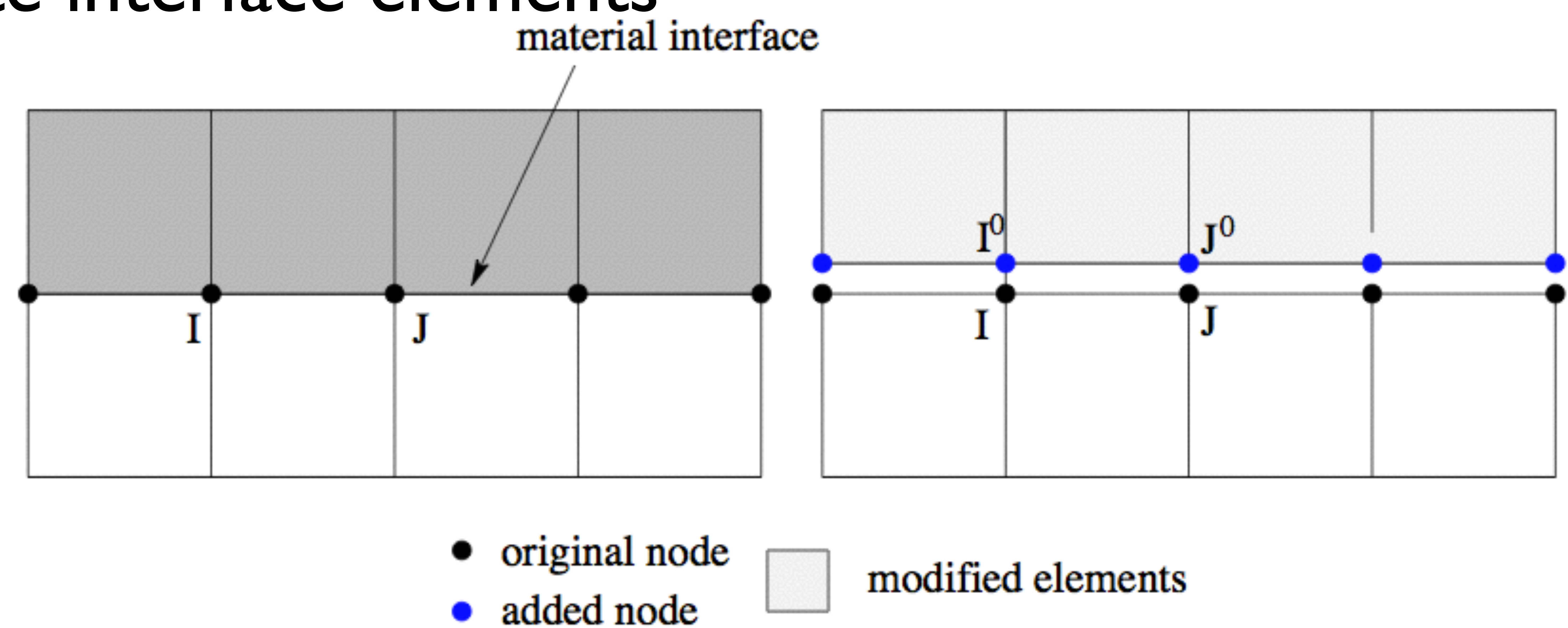


cannot pass the snap-back

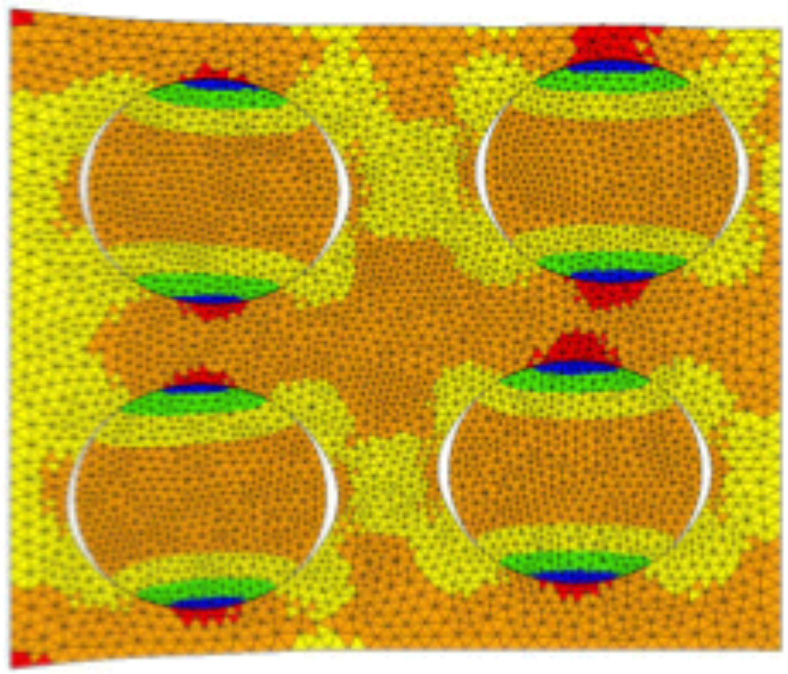
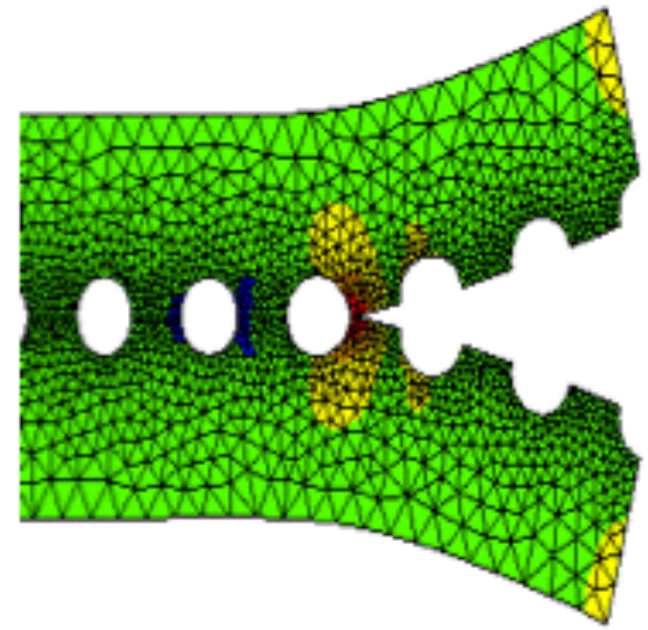
Mesh generation

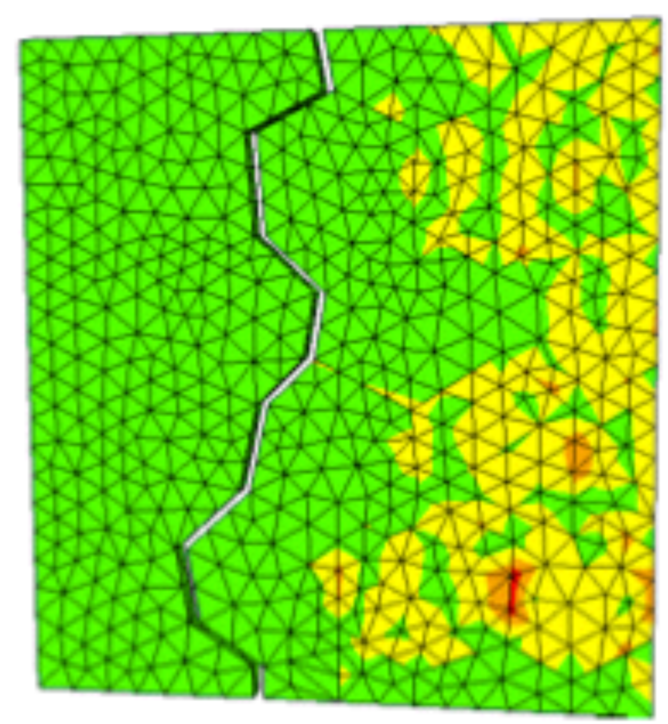
A C++ code was written to

- read a Gmsh mesh file
- double nodes along a path defined by the user
- modify the solid elements involved and
- generate interface elements



Some application examples



Path following method

Riks 1972

λ load factor

$$\begin{bmatrix} \mathbf{f}^{\text{int}}(\mathbf{u}) - \lambda \mathbf{g} \\ \phi(\mathbf{u}, \lambda) \end{bmatrix} = 0 \quad \mathbf{f}^{\text{ext}} = \lambda \mathbf{g}$$

reference load vector

Newton-Raphson $\phi(\mathbf{u}, \lambda)$ arc-length/constraint function

$$\begin{bmatrix} \mathbf{f}^{\text{int}}(\mathbf{u}_{(k)}) - \lambda_{(k)} \mathbf{g} \\ \phi(\mathbf{u}_{(k)}, \lambda_{(k)}) \end{bmatrix} + \begin{bmatrix} \mathbf{K} & -\mathbf{g} \\ \mathbf{v}^{\text{T}} & w \end{bmatrix}^{(k)} \cdot \begin{bmatrix} \Delta \mathbf{u} \\ \Delta \lambda \end{bmatrix} = 0$$

where

$$\mathbf{K} = \frac{\partial \mathbf{f}^{\text{int}}}{\partial \mathbf{u}}, \quad \mathbf{v} = \frac{\partial \phi}{\partial \mathbf{u}}, \quad w = \frac{\partial \phi}{\partial \lambda}$$

$$\longrightarrow \quad \begin{bmatrix} \Delta \mathbf{u} \\ \Delta \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{u}_I \\ 0 \end{bmatrix} - \frac{\mathbf{v}^{\text{T}} \mathbf{u}_I + \phi}{\mathbf{v}^{\text{T}} \mathbf{u}_{II} + w} \begin{bmatrix} \mathbf{u}_{II} \\ 1 \end{bmatrix}$$

$$\mathbf{u}_I = \mathbf{K}^{-1} \mathbf{r}, \quad \mathbf{u}_{II} = \mathbf{K}^{-1} \mathbf{g}$$

correction $\begin{bmatrix} \mathbf{u} \\ \lambda \end{bmatrix}^{(k+1)} = \begin{bmatrix} \mathbf{u} \\ \lambda \end{bmatrix}^{(k)} + \begin{bmatrix} \Delta \mathbf{u} \\ \Delta \lambda \end{bmatrix}$

Energy control

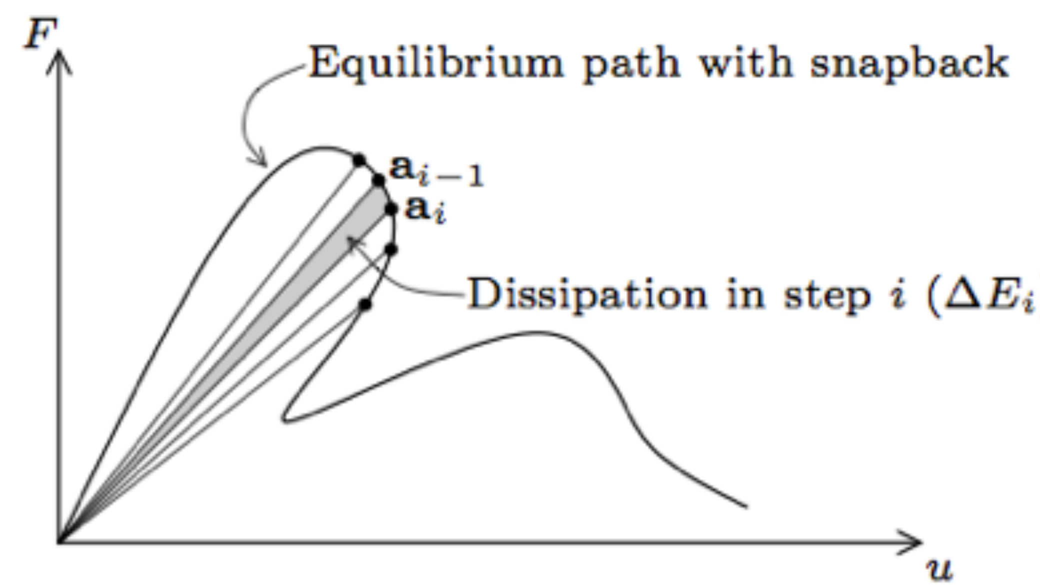
$$\boldsymbol{\epsilon} = \mathbf{B}\mathbf{a} \qquad \mathbf{f}^{\text{int}} = \int_\Omega \mathbf{B}^{\mathrm{T}}\boldsymbol{\sigma}$$

Gutierrez 2004

equilibrium

$$V = \frac{1}{2}\int_\Omega \boldsymbol{\epsilon}^{\mathrm{T}}\boldsymbol{\sigma} = \frac{1}{2}\int_\Omega \mathbf{a}^{\mathrm{T}}\mathbf{B}^{\mathrm{T}}\boldsymbol{\sigma} = \frac{1}{2}\mathbf{a}^{\mathrm{T}}\mathbf{f}^{\text{int}} = \frac{1}{2}\lambda\mathbf{a}^{\mathrm{T}}\mathbf{g}$$

$$\dot{V} = \frac{1}{2}\lambda\dot{\mathbf{a}}^{\mathrm{T}}\mathbf{g} + \frac{1}{2}\dot{\lambda}\mathbf{a}^{\mathrm{T}}\mathbf{g}$$



Energy release rate $\quad \dot{V} - \lambda\dot{\mathbf{a}}^{\mathrm{T}}\mathbf{g}$

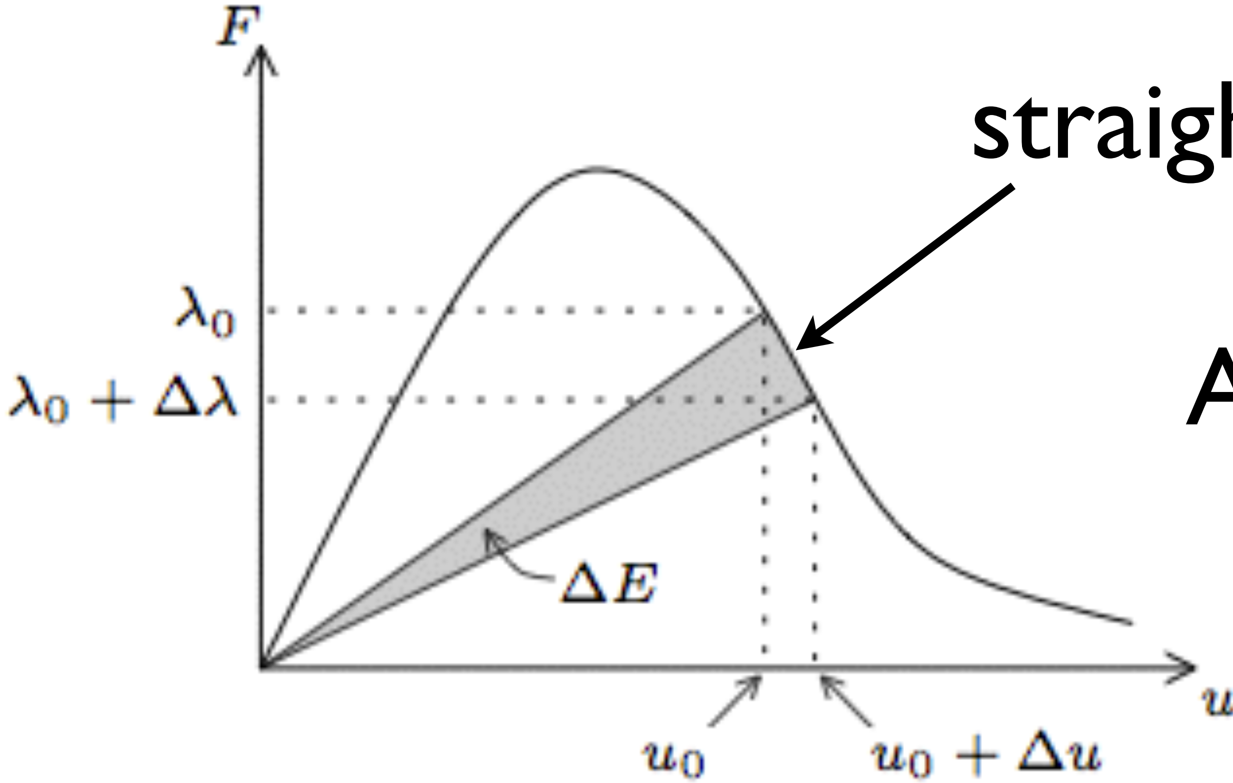
$$G = \frac{1}{2}\lambda\dot{\mathbf{a}}^{\mathrm{T}}\mathbf{g} - \frac{1}{2}\dot{\lambda}\mathbf{a}^{\mathrm{T}}\mathbf{g} \qquad \boxed{G > 0}$$

Arc-length function

predefined amount of energy to be released [Nm]

$$\phi = \frac{1}{2}\left[\lambda^{(n)}(\mathbf{a}^{\mathrm{T}}_{(n+1)} - \mathbf{a}^{\mathrm{T}}_{(n)}) - \Delta\lambda^{(n)}\mathbf{a}^{\mathrm{T}}_{(n)}\right]\mathbf{g} - \Delta\tau = 0$$

forward Euler



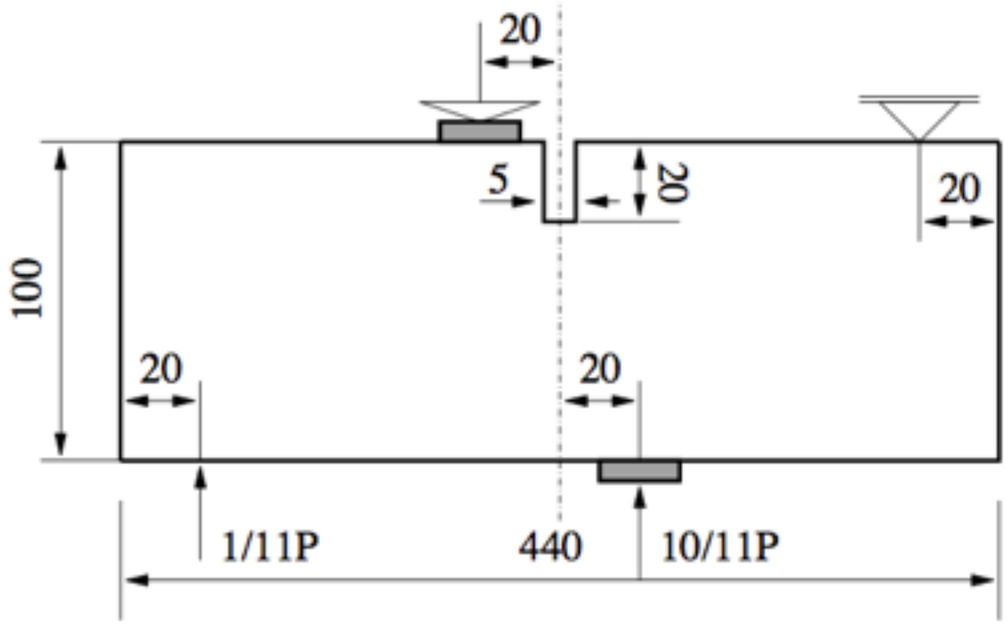
straight line

Assumption: secant unloading!!!

$$\Delta E = \frac{1}{2}\left(\lambda_0 \Delta u - \Delta\lambda u_0\right)$$

$$\phi = \underbrace{\boxed{\frac{1}{2}\left[\lambda^{(n)}(\mathbf{a}^{\mathrm{T}}_{(n+1)} - \mathbf{a}^{\mathrm{T}}_{(n)}) - \Delta\lambda^{(n)}\mathbf{a}^{\mathrm{T}}_{(n)}\right]}}_{\Delta E}\mathbf{g} - \Delta\tau = 0$$

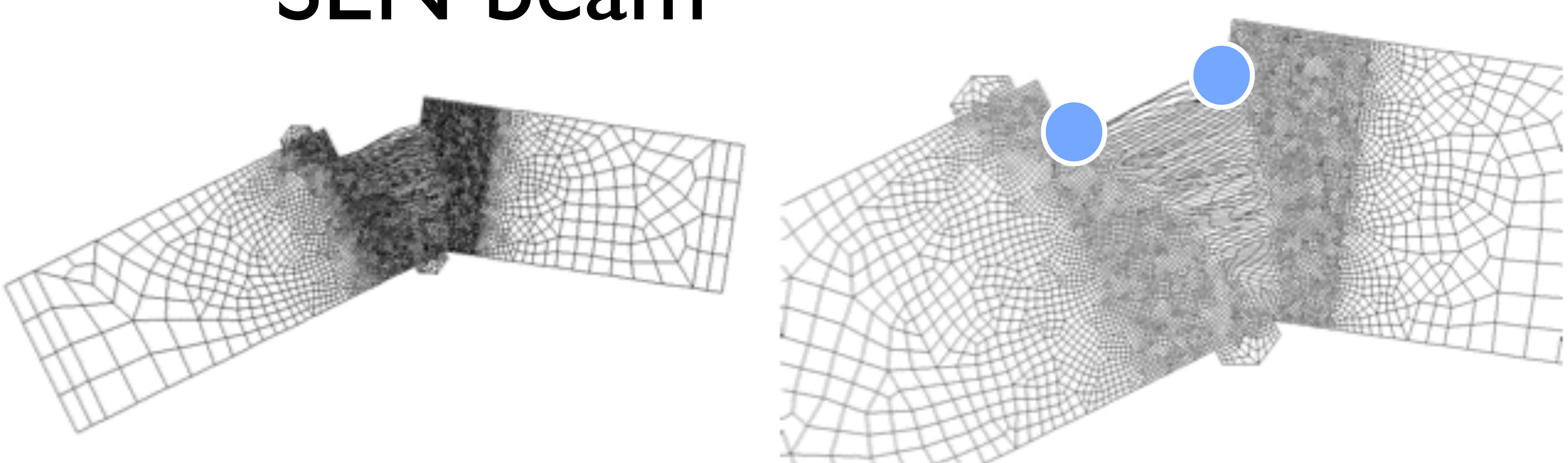
Indirect displacement control [de Borst 1986]



SEN beam

Indirect displacement control

$\phi(||u_A - u_B||, \Delta l)$ local quantity!!!

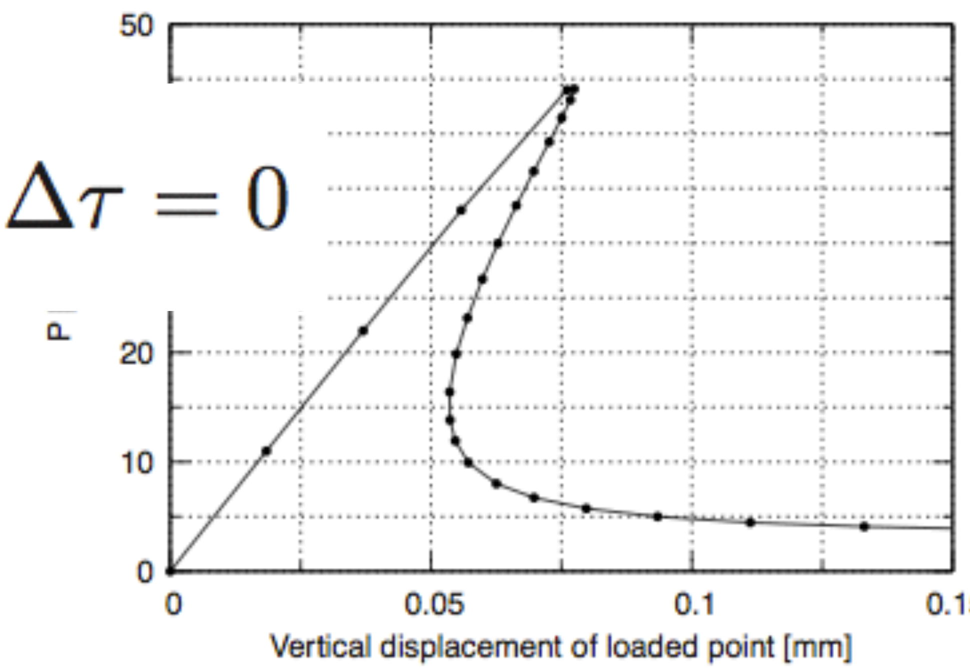


(a) (b)

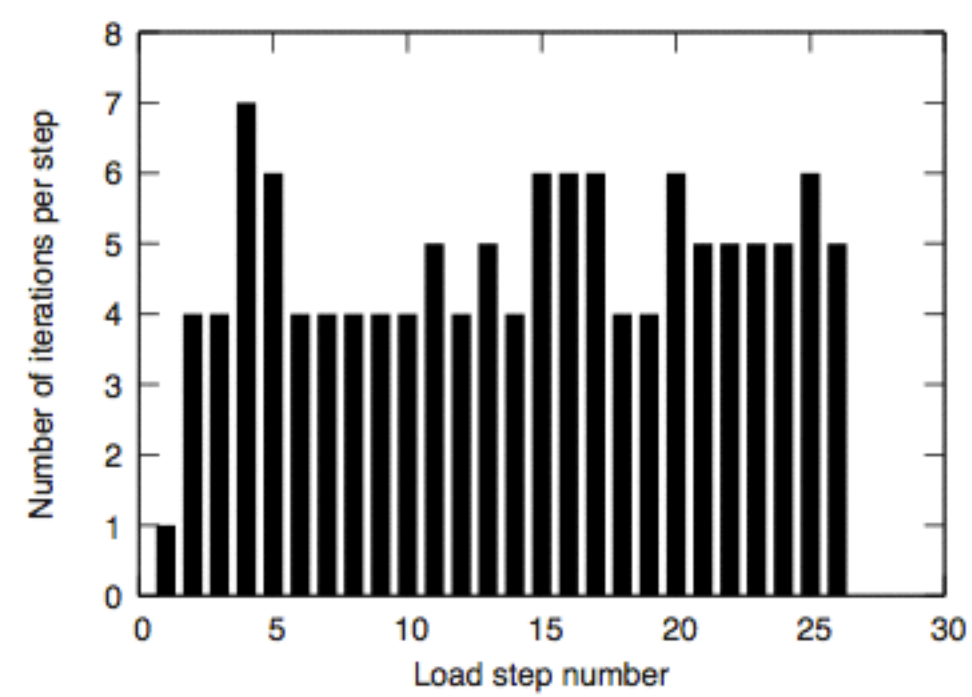
imagine what if
there are 2 cracks???

Advantages of energy control

SEN beam



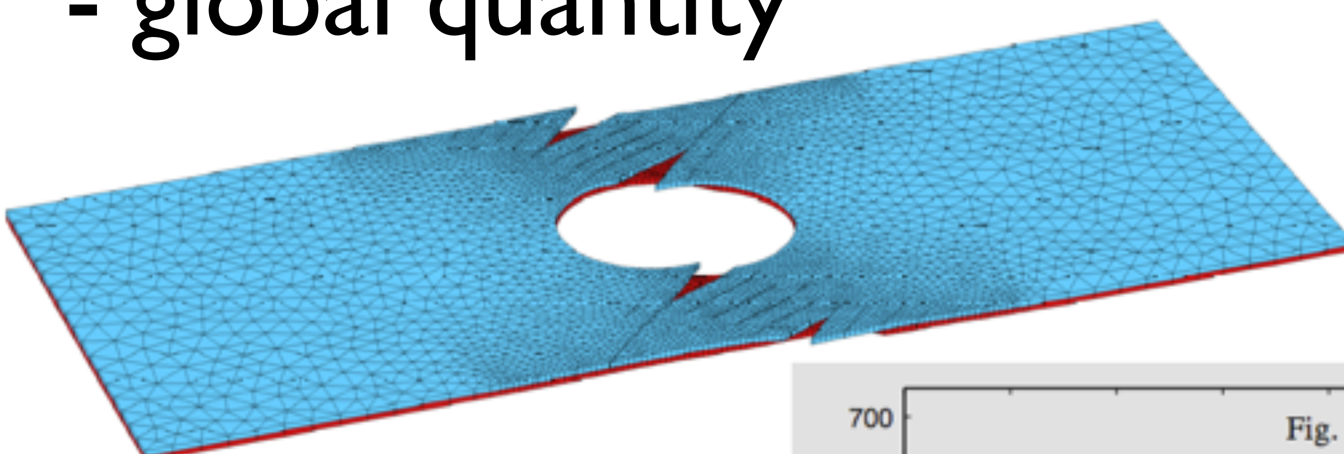
(a)



Energy control

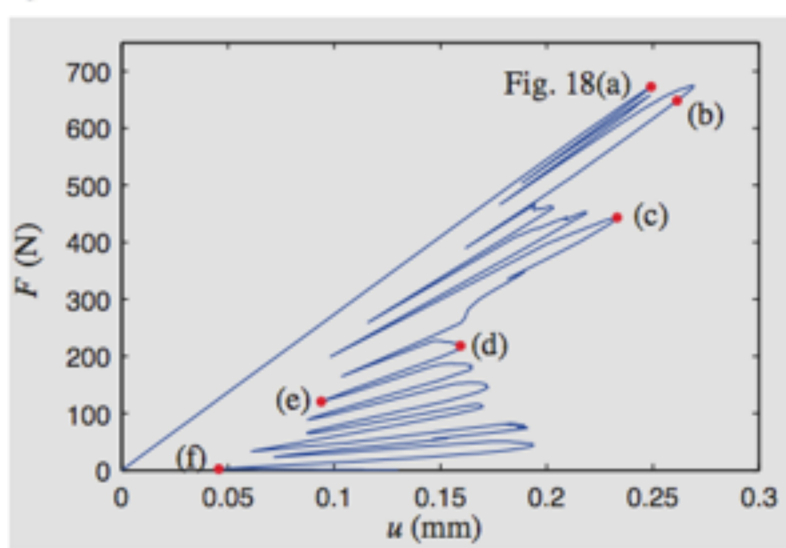
$$\phi = \frac{1}{2}\left[\lambda^{(n)}(\mathbf{a}^{\mathrm{T}}_{(n+1)} - \mathbf{a}^{\mathrm{T}}_{(n)}) - \Delta\lambda^{(n)}\mathbf{a}^{\mathrm{T}}_{(n)}\right]\mathbf{g} - \Delta\tau = 0$$

- fast to evaluate
- global quantity

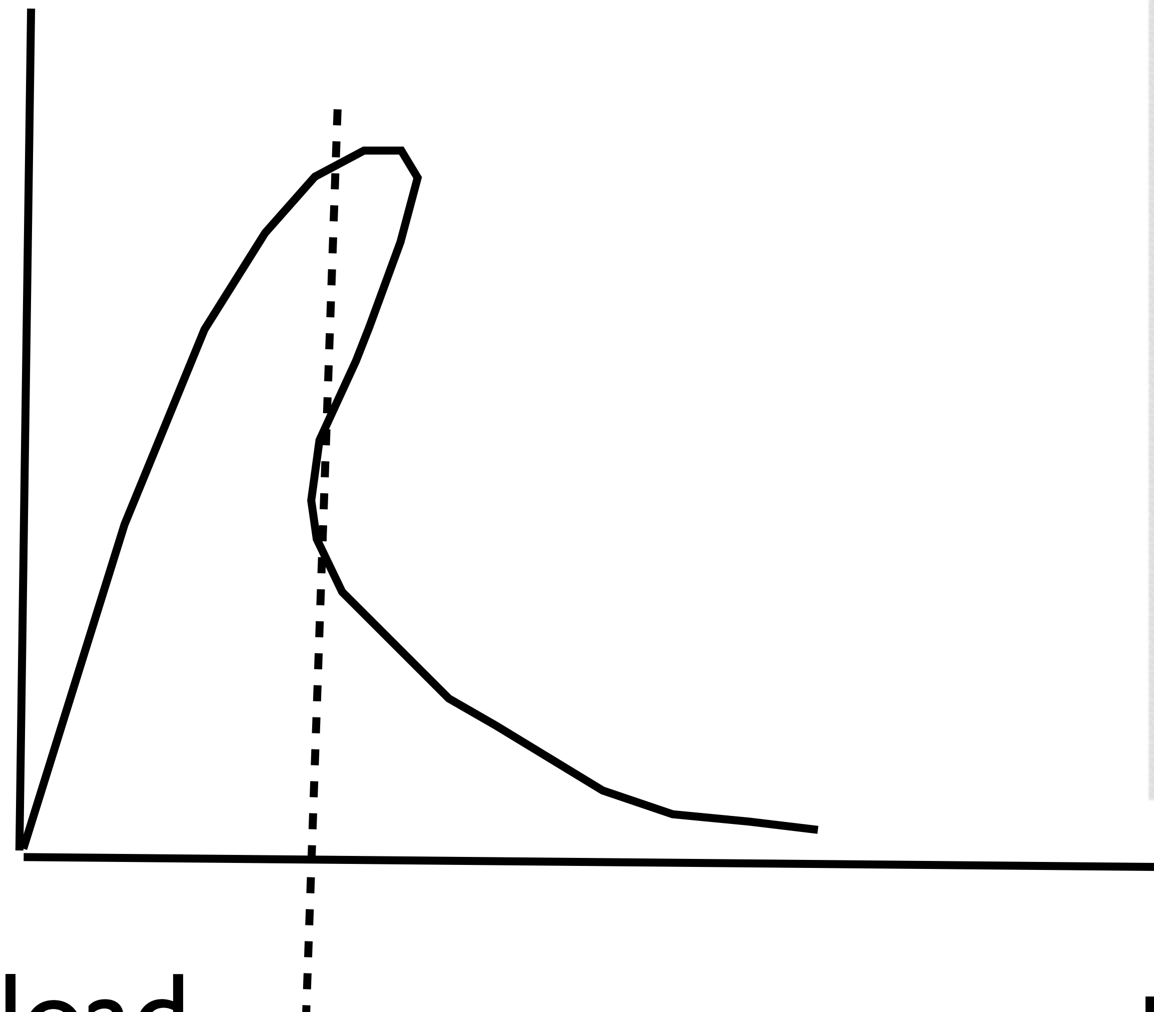


multiple cracks

FP van der Mer
EFM, 2008

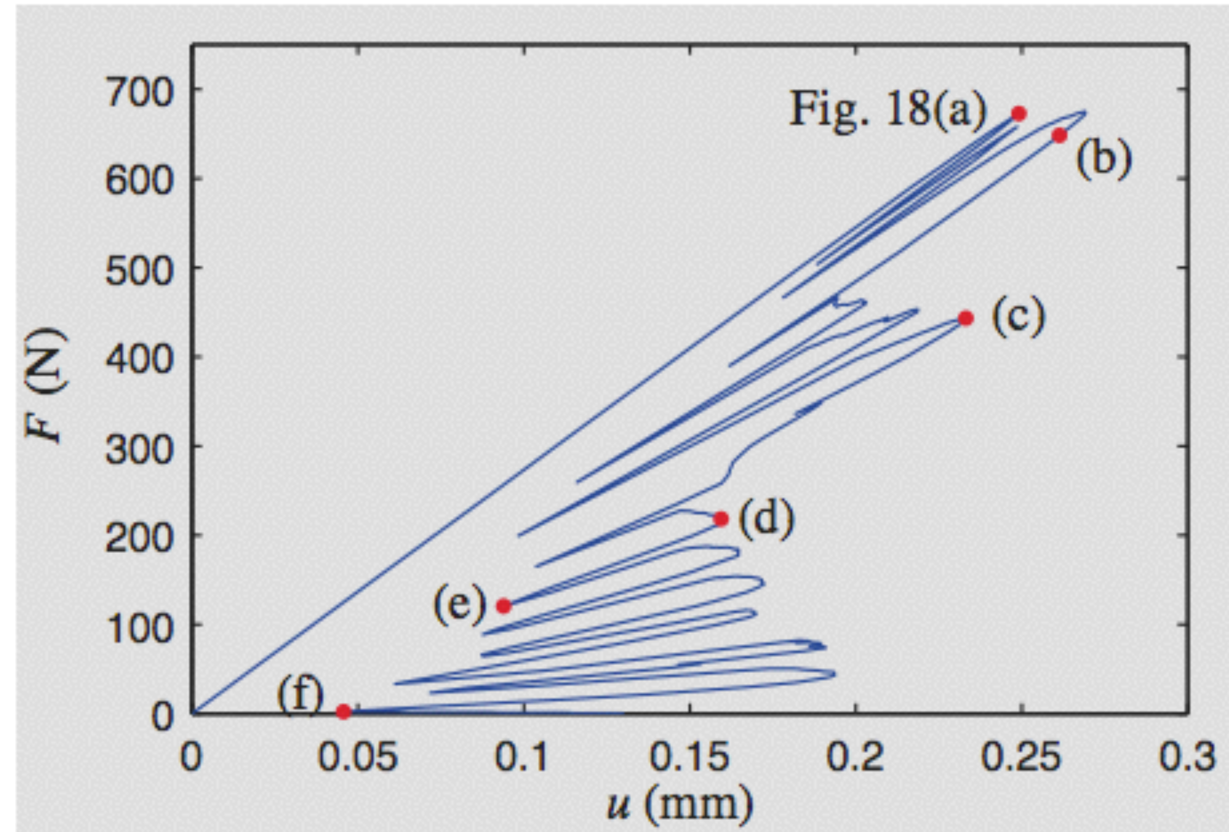


Solution procedure

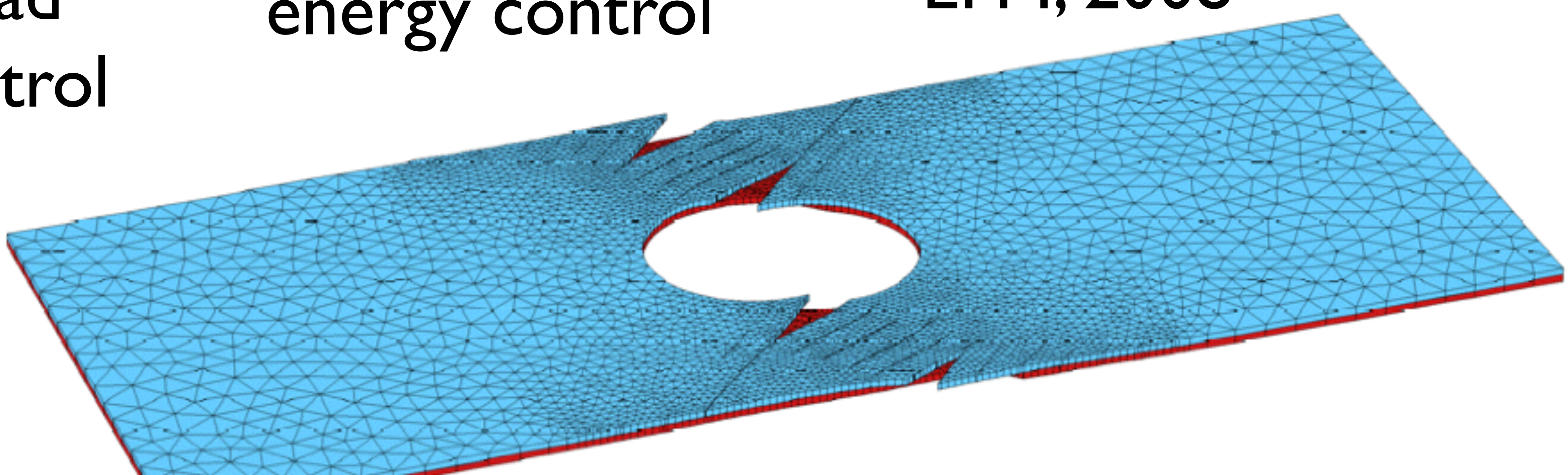


load control

energy control



FP van der Mer
EFM, 2008



Box 4 Flowchart for solution procedure with energy-based control

1. Initialization: $\lambda = 0$, set n_d, $\Delta\tau_0$, $\Delta\tau_{\min}$, $\Delta\tau_{\max}$
2. Solve the elastic branch using load control
   i. Solving the equilibrium $\mathbf{f}^{\text{int}} = \lambda\mathbf{g}$ using Newton-Raphson method
   ii. Store the load scale: $\lambda_0 = \lambda$
   iii. Check number of iterations: $n > n_d$. If no, $\lambda = \lambda + \Delta\tau_0$, goto 2i. Else
   iv. Compute the released energy $G = 0.5\left[\lambda_0(\mathbf{u}^{\text{T}} - \mathbf{u}_0^{\text{T}}) - \Delta\tau_0\mathbf{u}_0^{\text{T}}\right]\mathbf{g}$
   v. $\Delta\tau = G$
   vi. $\Delta\tau_{\min} \leq \Delta\tau \leq \Delta\tau_{\max}$
3. Switch to arc-length control
   i. Update $\mathbf{K}$, $\mathbf{f}$ from element contributions using a FE formulation
   ii. Update $\mathbf{v}$, ω, ϕ

$$\mathbf{f}^{\text{int}} + \mathbf{f}^{\text{coh}}$$

$$\begin{aligned}
\mathbf{v} &= 0.5\lambda_0\mathbf{g} \\
\omega &= -0.5\mathbf{u}_0^{\text{T}}\mathbf{g} \\
\phi &= 0.5\left[\lambda_0(\mathbf{u}^{\text{T}} - \mathbf{u}_0^{\text{T}}) - \Delta\lambda\mathbf{u}_0^{\text{T}}\right]\mathbf{g} - \Delta\tau
\end{aligned}$$

   iii. Start the iteration procedure

$$\begin{aligned}
\mathbf{r} &\leftarrow \lambda\mathbf{g} - \mathbf{f} \\
\mathbf{u}_I &\leftarrow \mathbf{K}^{-1}\mathbf{r} \\
\mathbf{u}_{II} &\leftarrow \mathbf{K}^{-1}\mathbf{g} \\
\alpha &\leftarrow \frac{\mathbf{v}^{\text{T}}\mathbf{u}_I + \phi}{\mathbf{v}^{\text{T}}\mathbf{u}_{II} + w} \\
\mathbf{u} &\leftarrow \mathbf{u} + \mathbf{u}_I - \alpha\mathbf{u}_{II} \\
\lambda &\leftarrow \lambda - \alpha
\end{aligned}$$

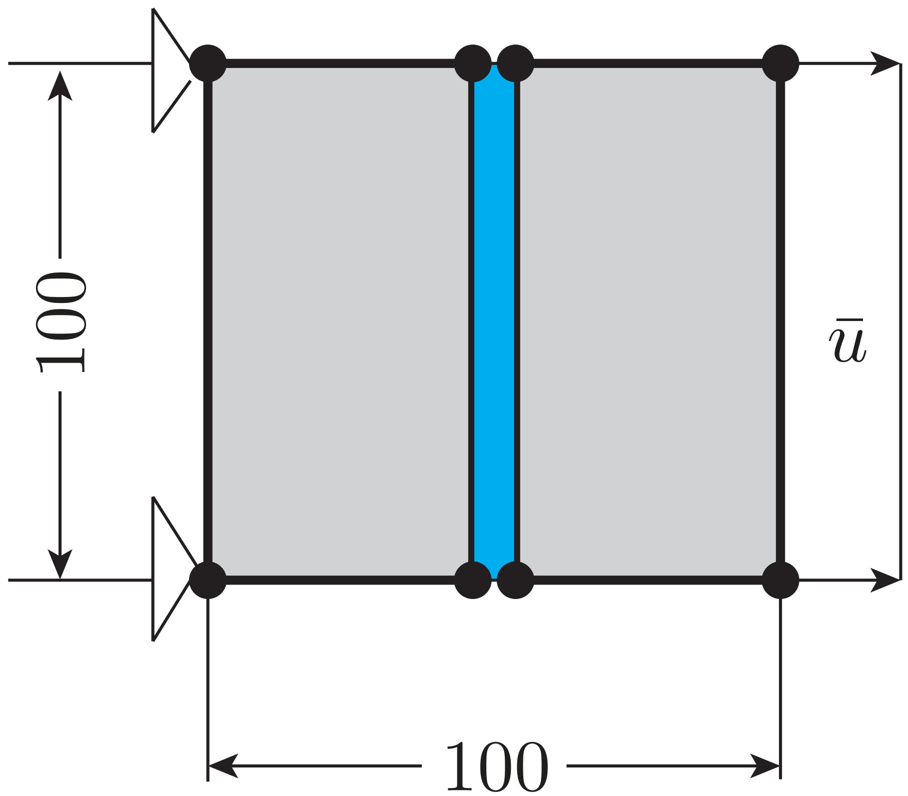
   iv. Check convergence, if not return to step 3i
   v. Adjust the path following parameter $\Delta\tau = \Delta\tau * 0.5^{\gamma}$, $\gamma = \frac{n_j - n_d}{4}$ **
   vi. $\Delta\tau_{\min} \leq \Delta\tau \leq \Delta\tau_{\max}$
   vii. Store the load scale: $\lambda_0 = \lambda$

* In the above the subscript 0 denotes converged values of previous load step.
** This is a kind of automatic incrementation which calculate the complete behavior in as few steps as possible. n_d is a user-defined number of desired iterations per step.

Numerical examples

- Simple tests (to debug code)
- Material interface debonding
- Multi-delamination of a composite DCB
- Delamination of the DCB

Simple test (2D)



$E = 1000$

$\nu = 0.0$

$f_n = 1.0$

$f_s = 1.0$

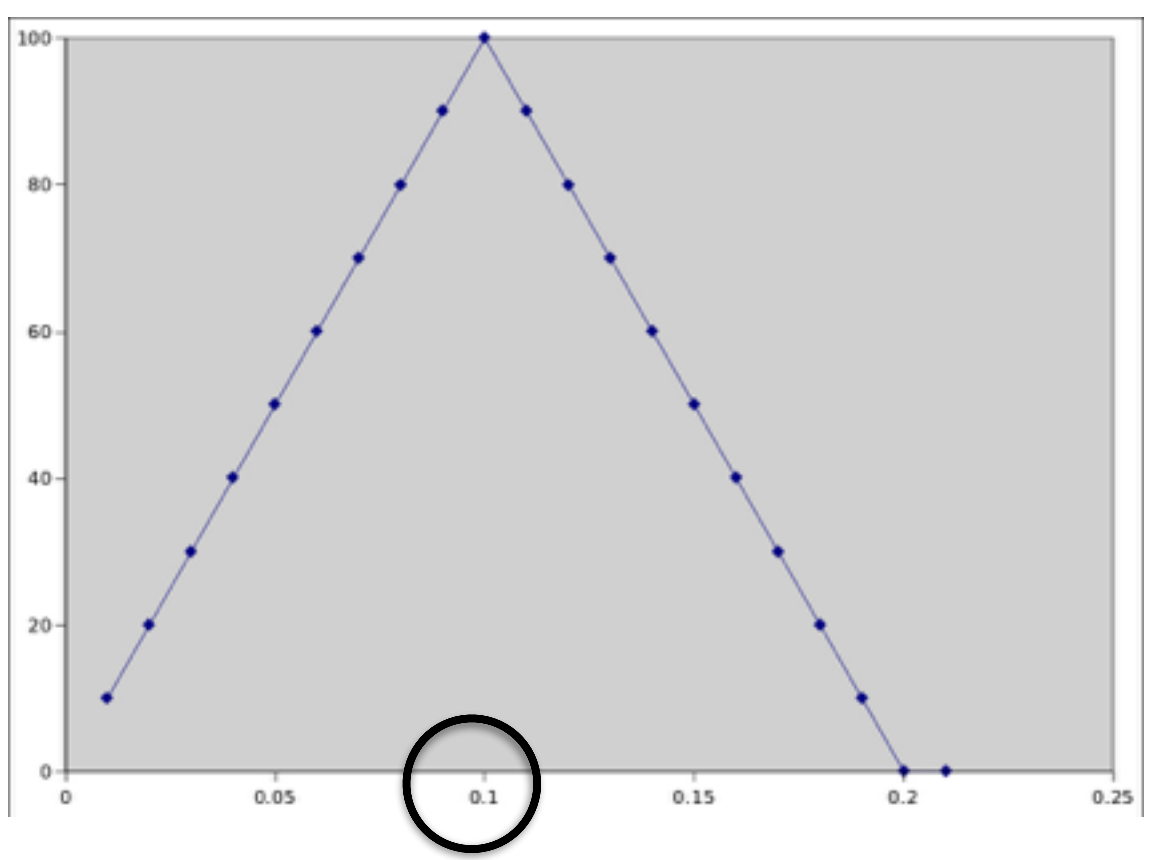
$g_I = 0.1$

$g_{II} = 0.1$

$K = 10^6$

$\bar{u} = 0.01 \times i$

plane strain



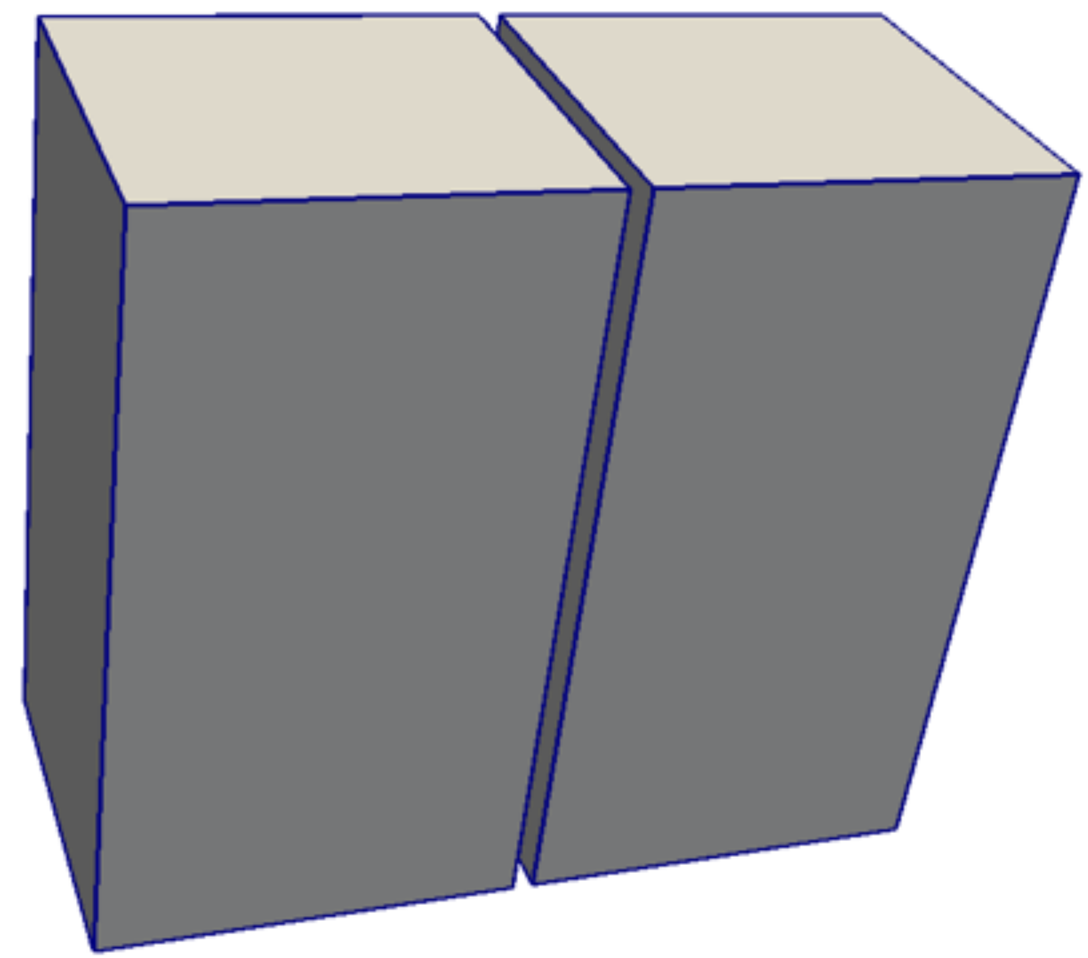
$$\sigma^c_{xx} = t^c = 1$$

$$\epsilon^c_{xx} = \frac{\sigma^c_{xx}}{E} = 0.001$$

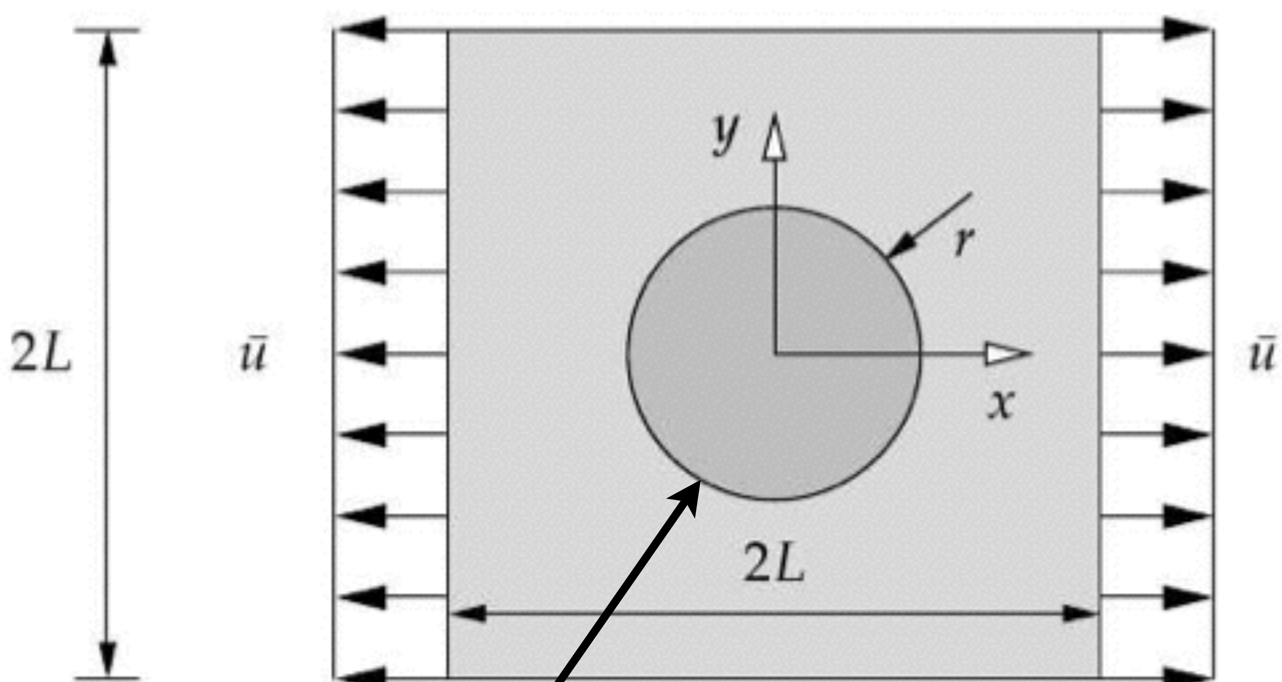
$$u^c_x = \epsilon^c_{xx} \times L = 0.1$$

Simple test (3D)

- As previous 2D example
- Thickness: 50
- Solved with Hex8 and Tet4



Debonding of a material interface



$L = 15\mu m$

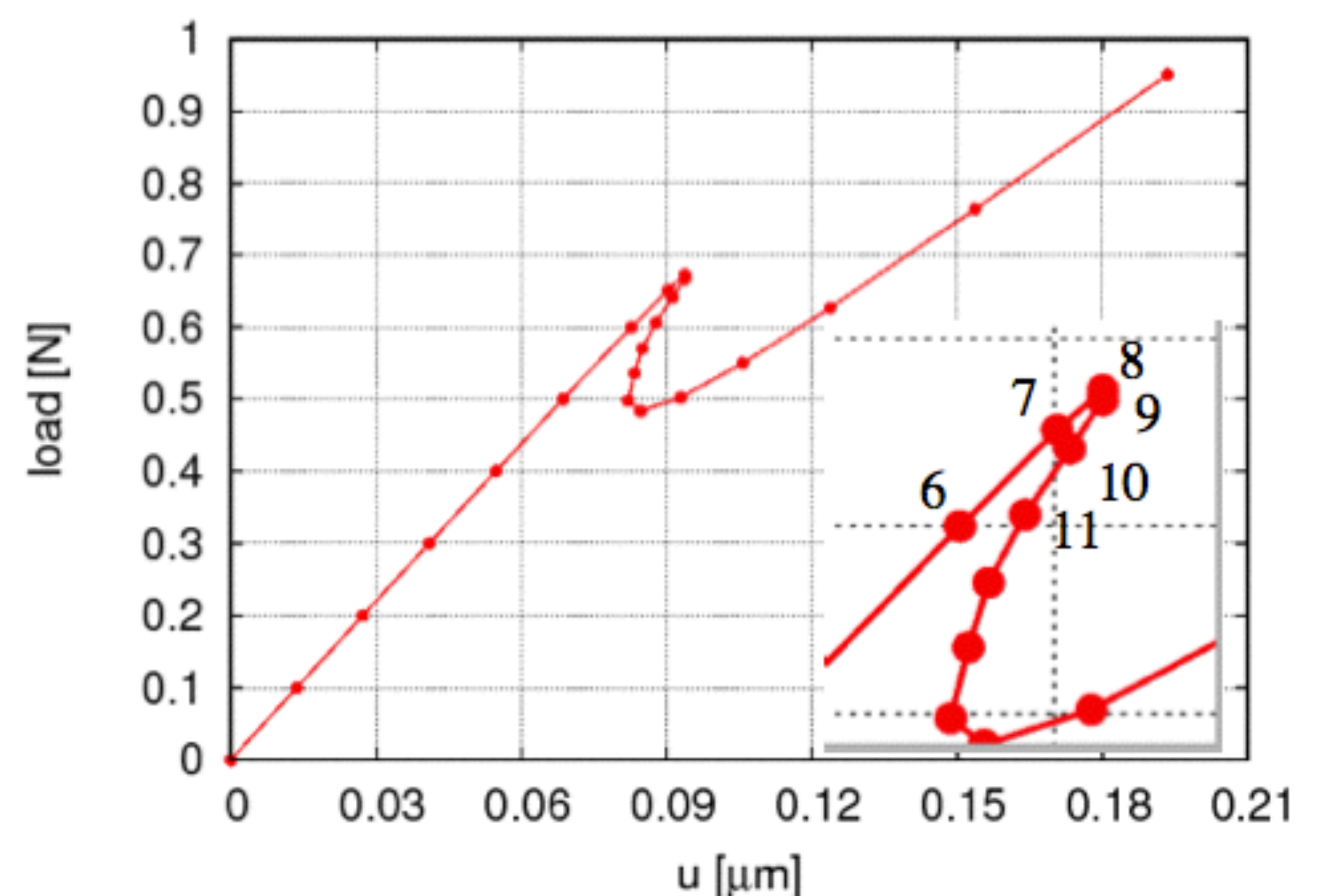
$r = 5\mu m$

$E_{\text{fib}} = 225$ GPa

$\nu_{\text{fib}} = 0.2$

$E_{\text{mat}} = 4.3$ GPa

ν_{m}

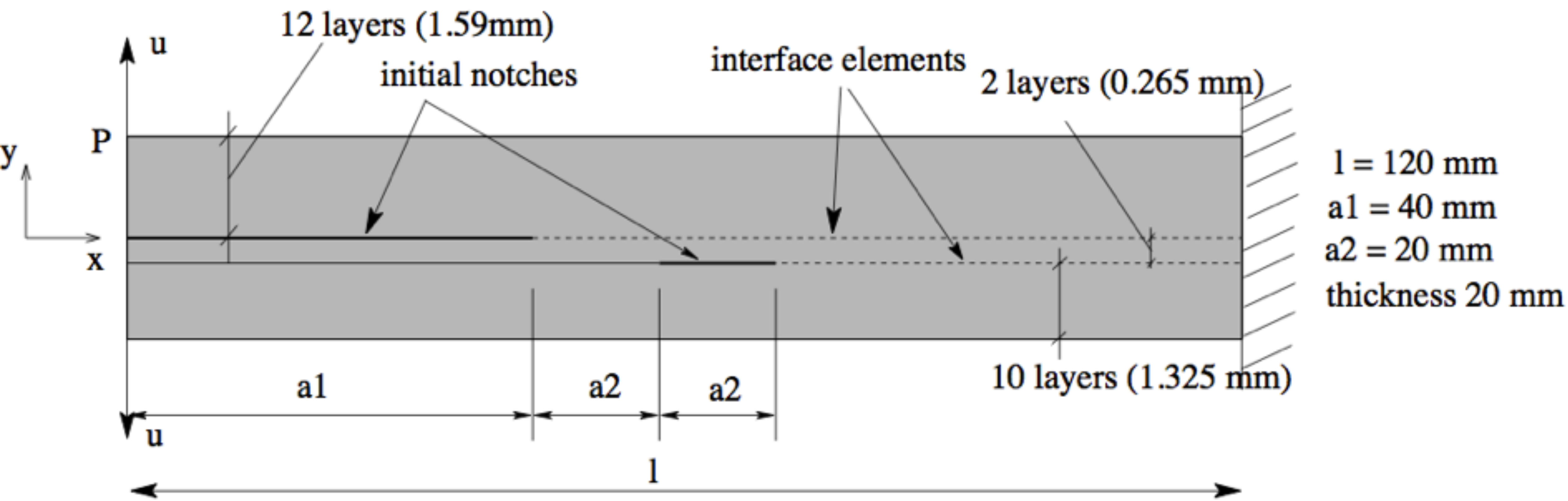


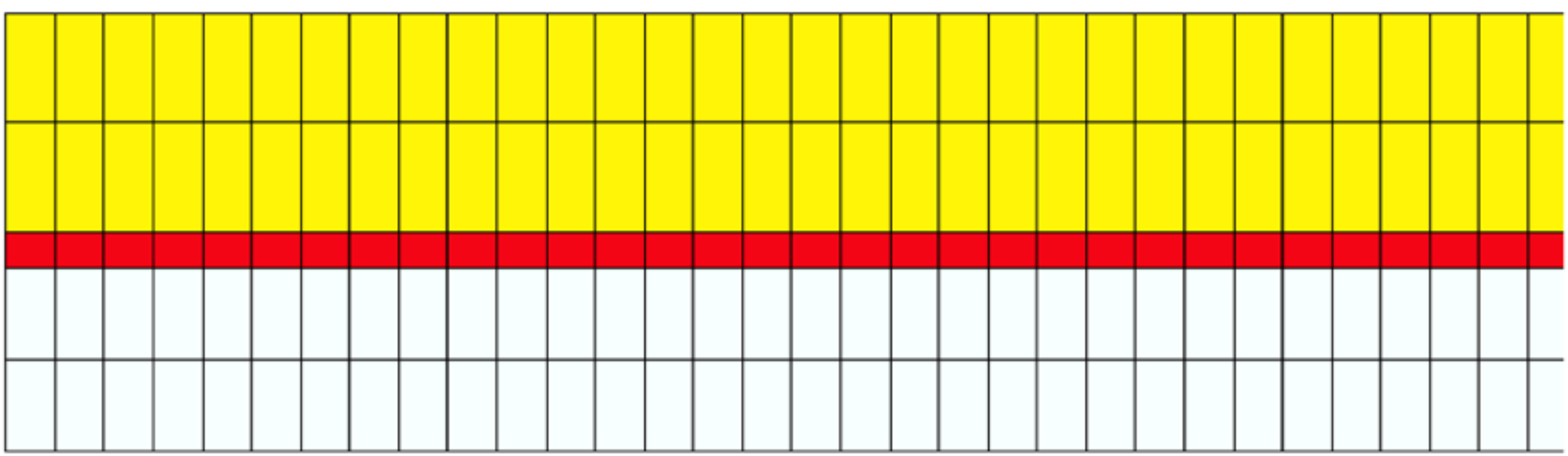
interface: linear 4-node

continuum: T3 elements

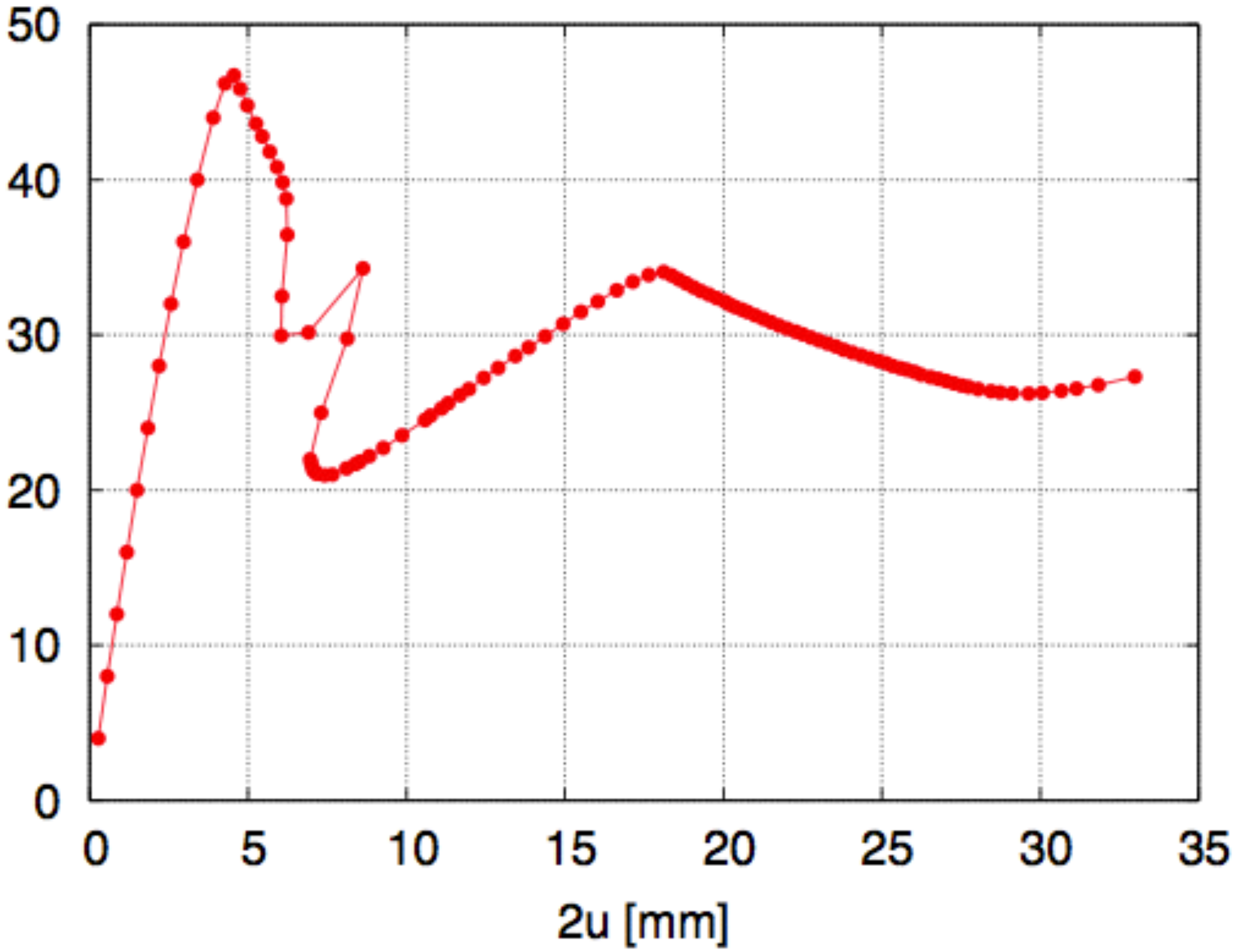
Cohesive law: Xu-Needleman

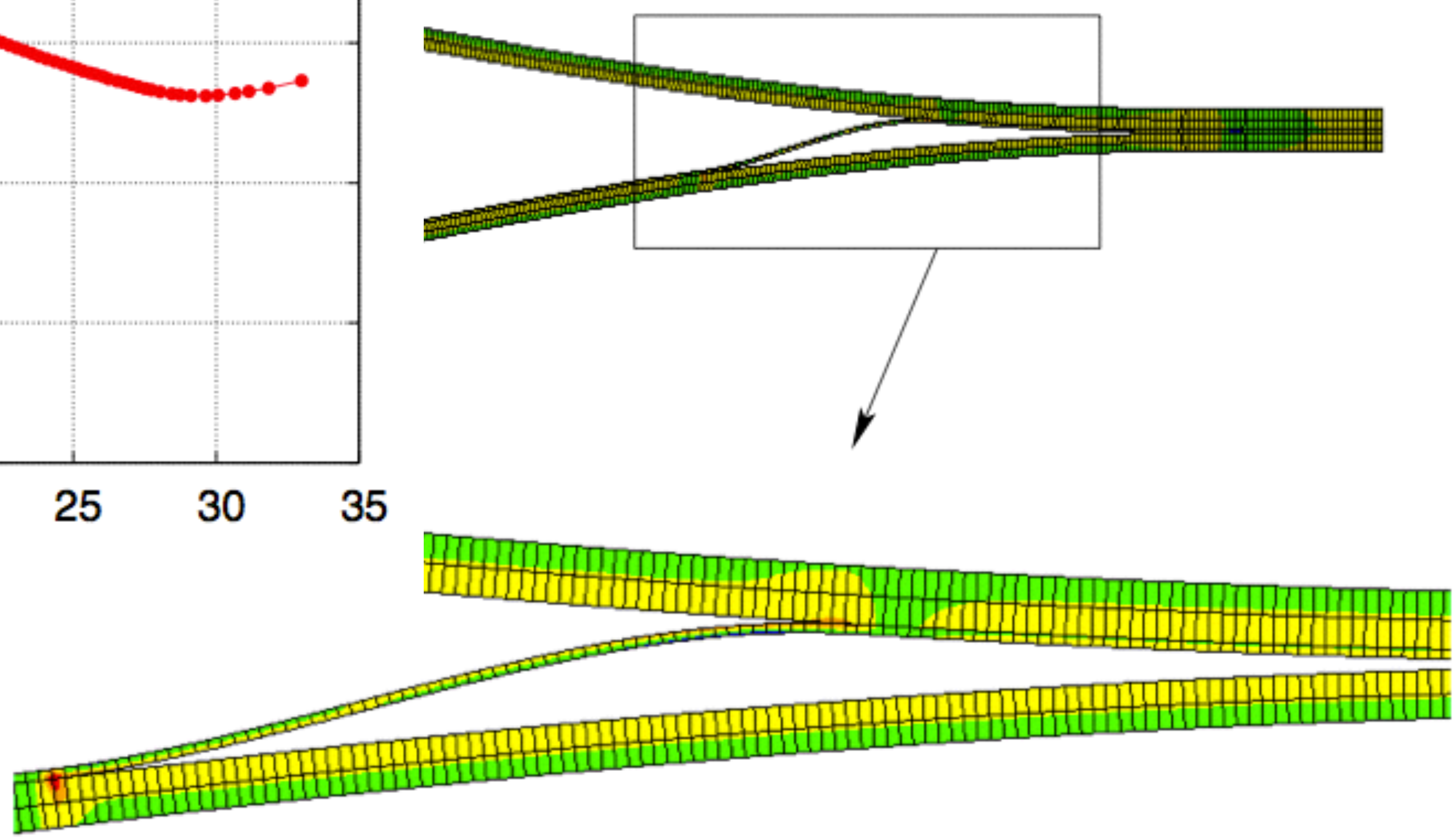
Multi-delamination



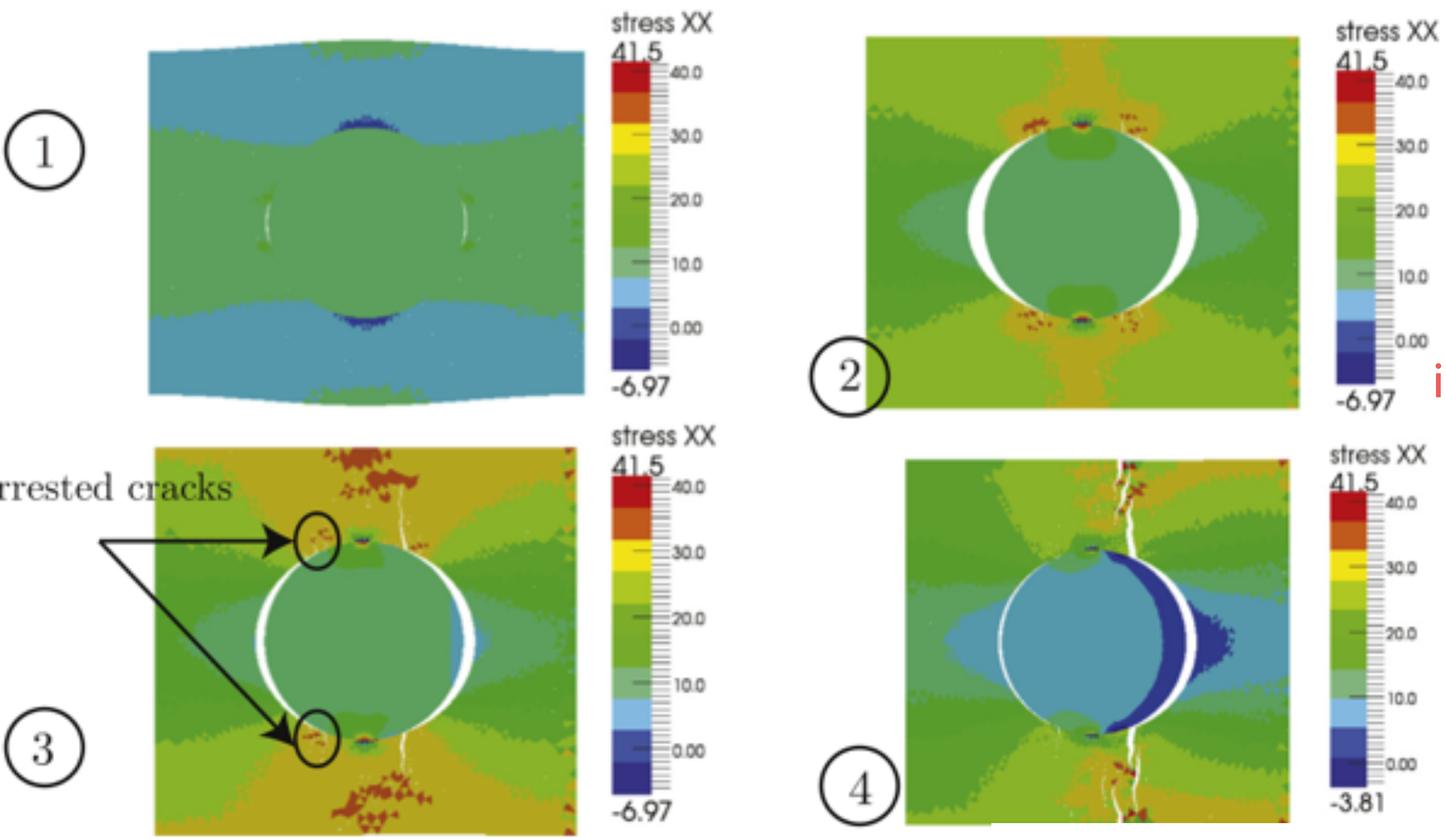


Multi-delamination

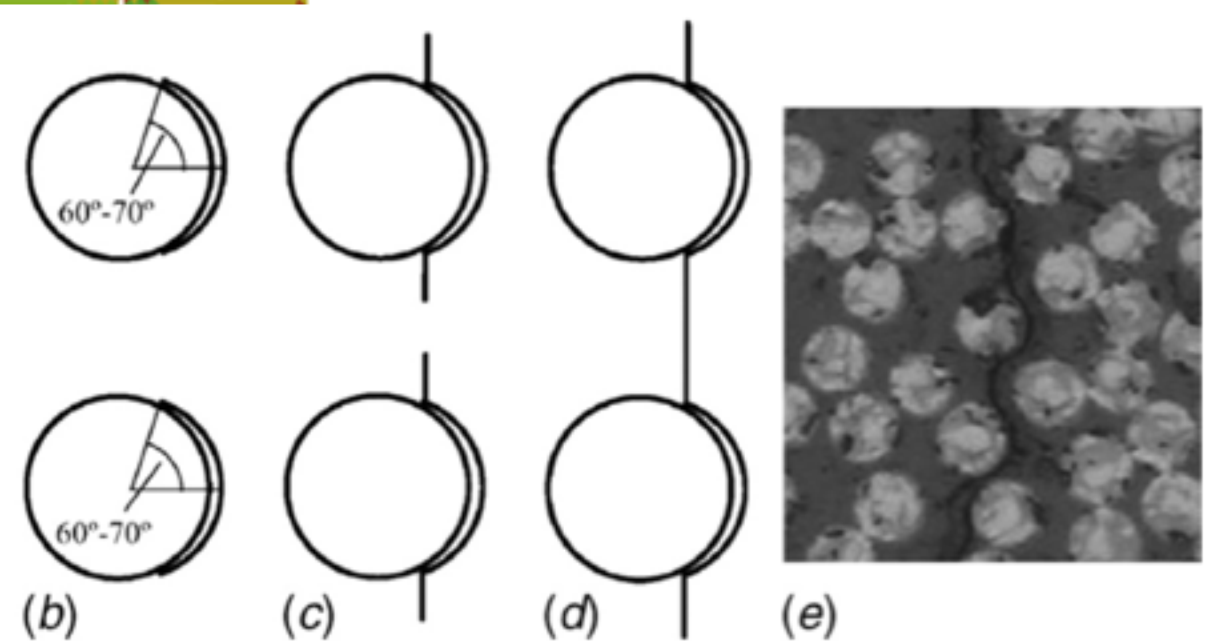




Matrix kinking

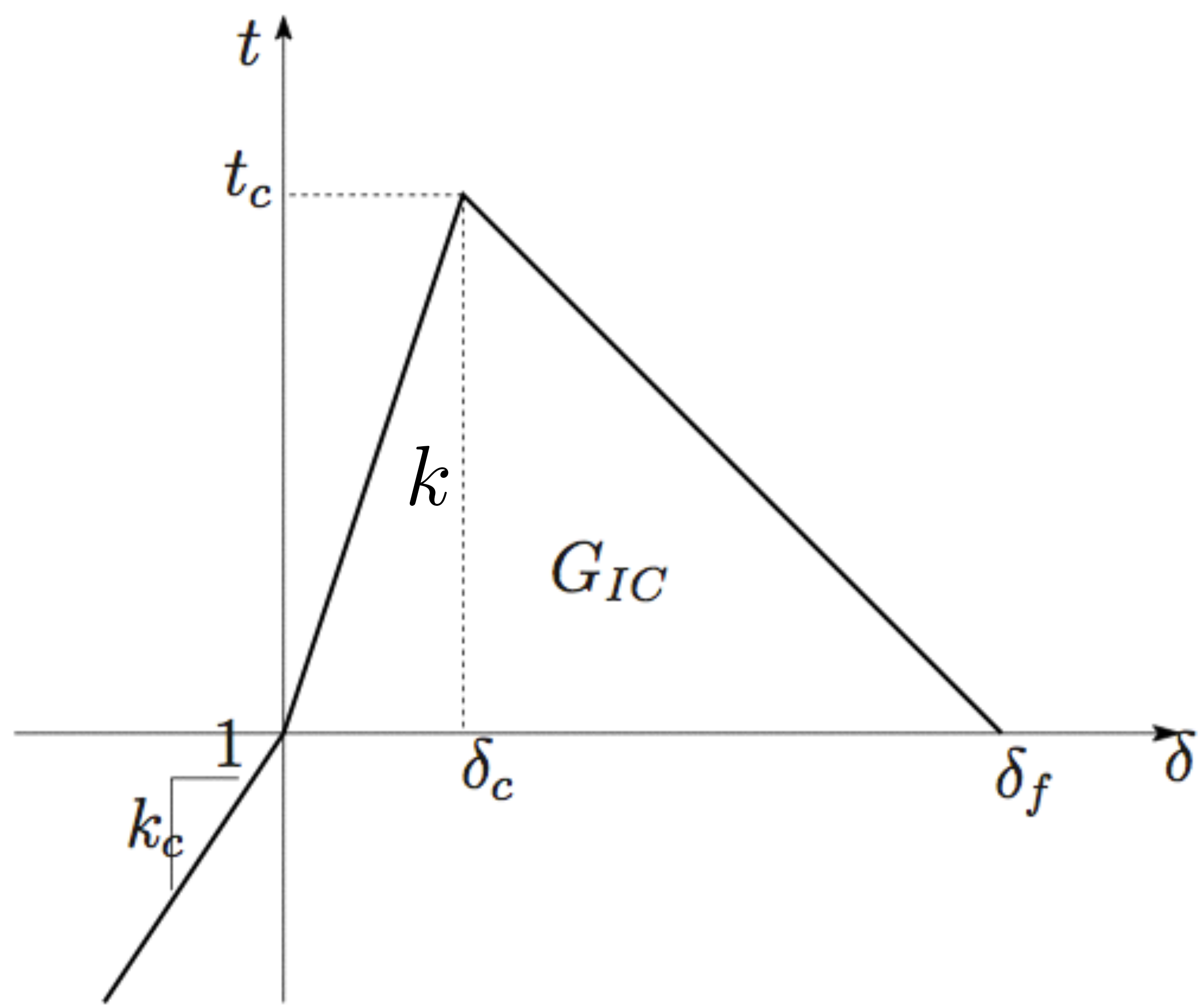


interface elements everywhere
except for the hard inclusion



"Discontinuous Galerkin/extrinsic cohesive zone modeling: Implementation caveats and applications in computational fracture mechanics", VP Nguyen, Engineering Fracture Mechanics, 2014.

Discontinuous Galerkin and cohesive interface elements



k: sufficiently large not to reduce the compliance of solid

Weibull distribution

$$P(\sigma_f) = 1 - \exp\left[-\left(\frac{\sigma_f}{\sigma_0}\right)^m\right]$$ (two-parameter version)

$$P(\sigma_f) = 1 - \exp\left[-\left(\frac{\sigma_f - \sigma_\mathrm{m}}{\sigma_0}\right)^m\right]$$ (three-param. version)

Stresses smaller than sigma_m, no failure: P=0

Probability Density Function (PDF)

$$f(\sigma_f) \equiv \frac{dP}{d\sigma_f} = \frac{m}{\sigma_0}\left(\frac{\sigma_f}{\sigma_0}\right)^{m-1} \exp\left[-\left(\frac{\sigma_f}{\sigma_0}\right)^m\right]$$

Account for the fact that larger samples more prone to failure

$$P(\sigma_f) = 1 - \exp\left[-\left(\frac{V}{V_0}\right)\left(\frac{\sigma_f}{\sigma_0}\right)^m\right]$$

Weibull distribution: implementation

$$P(\sigma_f) = 1 - \exp\left[-\left(\frac{\sigma_f - \sigma_\mathrm{m}}{\sigma_0}\right)^m\right]$$

$$\exp\left[-\left(\frac{\sigma_f - \sigma_m}{\sigma_0}\right)^m\right] = 1 - P(\sigma_f) \equiv P_s(\sigma_f)$$

$$-\left(\frac{\sigma_f - \sigma_m}{\sigma_0}\right)^m = -\log\left[P_s(\sigma_f)\right]$$

$$\sigma_f = \sigma_0\left(-\log(rand)\right)^{1/m} + \sigma_m$$

where *rand* is a random number between 0 and 1.

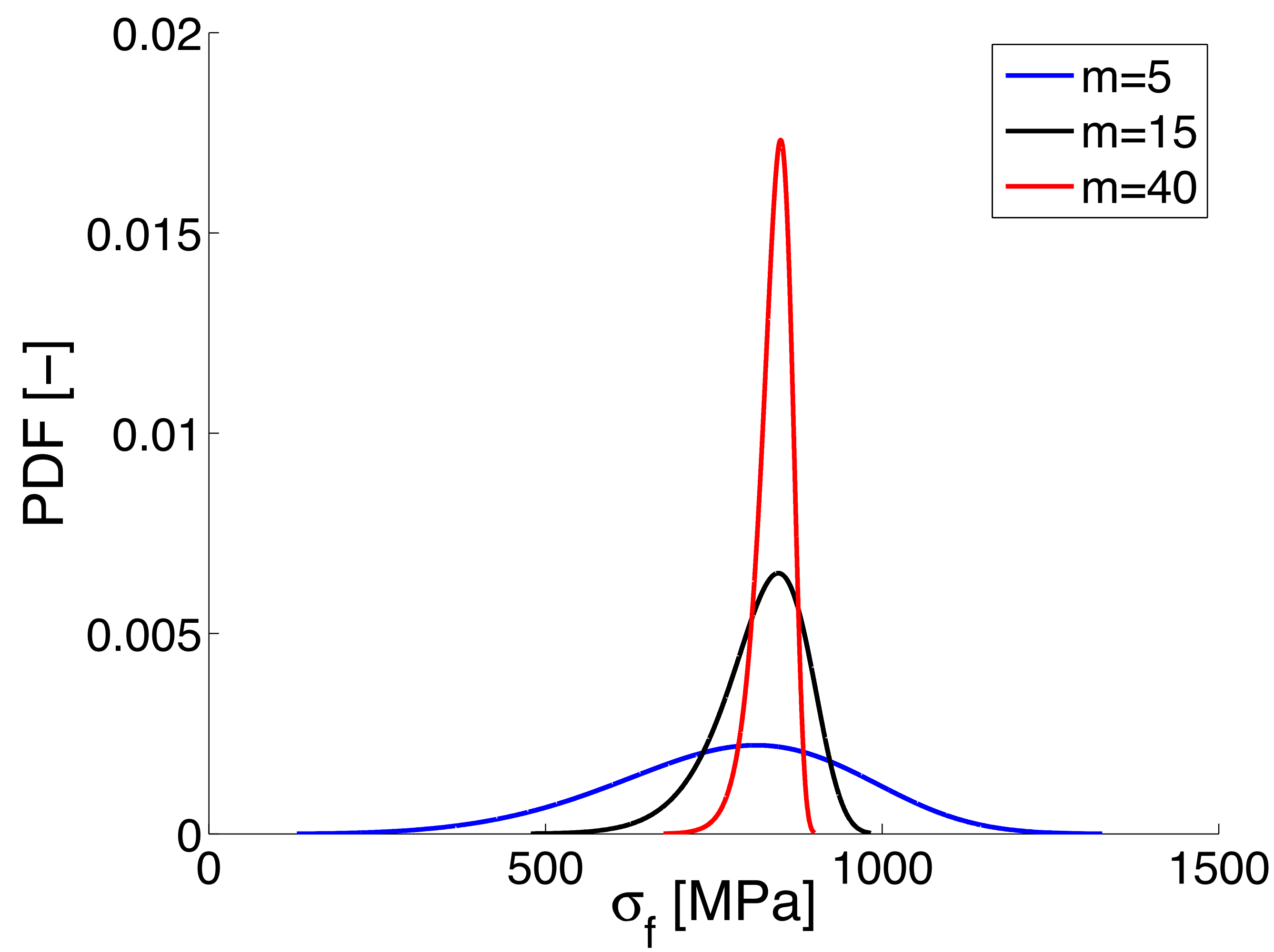
Weibull distribution: implementation

$$\sigma_f = \sigma_0 \left(-\log(rand)\right)^{1/m} + \sigma_m$$

where *rand* is a random number between 0 and 1.

```
double xrand;
double value, f2t;

std::random_device rd;
std::mt19937 gen(rd());
std::uniform_real_distribution<> dis(0, 1);

int ipCount   = 2;
int elemCount = count / ipCount;

for ( int i = 0; i < elemCount; i++ )
{
  xrand = dis(gen);
  value  = pow(-log(xrand),1/m_) ;
  f2t     = f2t_ * value + sigmaMin_;
  f2ts_.pushBack ( f2t, ipCount );
}
```

each IE has the same strength



$\sigma_0 = 850\ \mathrm{MPa} \quad \sigma_\mathrm{m} = 0\ \mathrm{MPa}$

Things to explore

- New cohesive laws
- New (better) interface element formulations (current element technology does not allow industrial applications to be realized.)
- Mesh topology such that <u>mesh bias</u> can be avoided.