Ph.D. Dissertation Presentation

페이즈 필드 모델을 위한 효율적인 수치적 방법 개발

Development of efficient numerical methods for the phase field model

Gihwan Kim

Department of Mechanical Engineering, KAIST

2022. 09. 23

KAIST

CMSS
Computational Mechanics and Structural Systems.

Contents

1. Introduction

Fracture phenomena

- Failures due to fracture

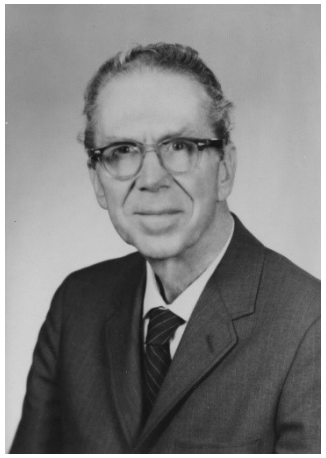
To prevent catastrophic failures,
it is important to understand fracture behavior.

Brief history

❖ Theoretical approaches

A. A. Griffith(1920)

- Approach in thermodynamic aspects.
- Introduce “Energy release rate”.



G. R. Irwin(1957)

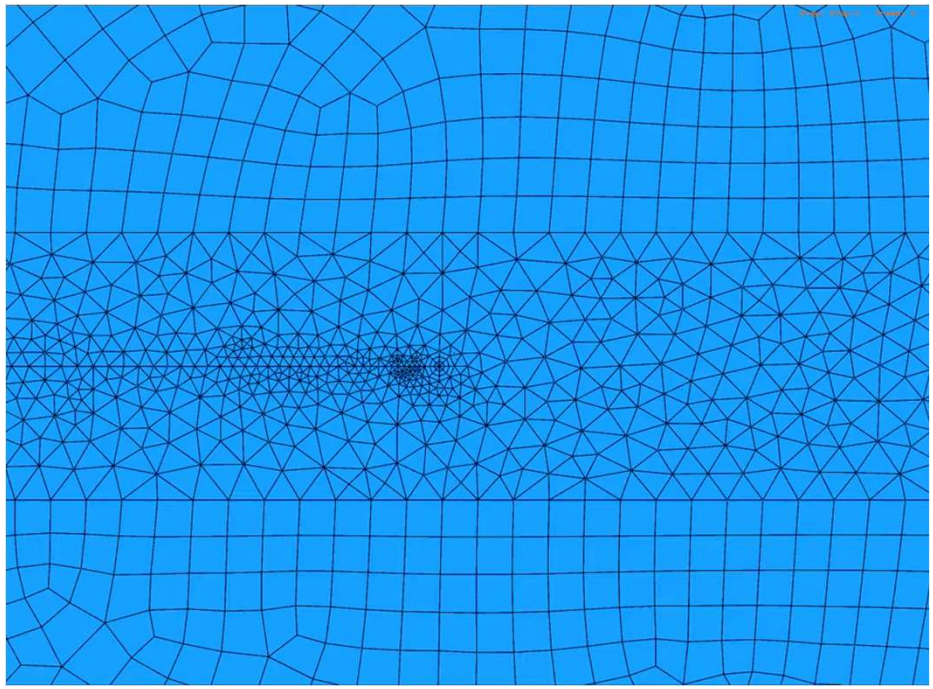
- Introduce “Stress intensity factor”.

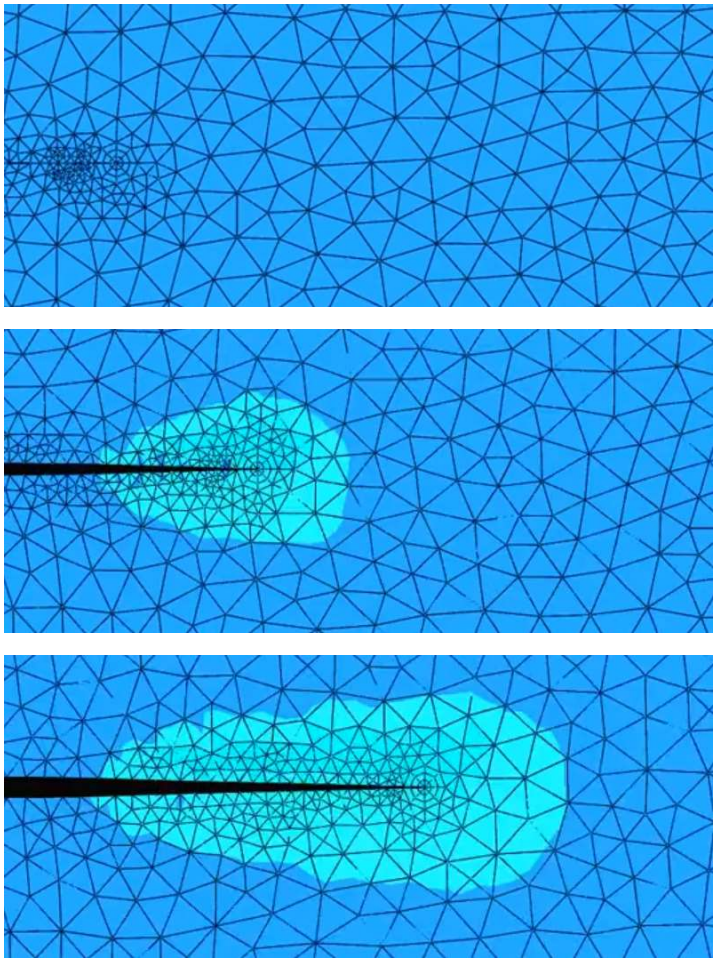
➡ **Linear Elastic Fracture Mechanics (LEFM)**, the foundation of fracture mechanics, is born.

Griffith, A. A. (1920). VI. The phenomena of rupture and flow in solids. *Philosophical transactions of the royal society of london. Series A, containing papers of a mathematical or physical character*, *221*(582-593), 163-198.
Irwin, G. R. (1957). Analysis of stresses and strains near the end of a crack traversing a plate.

Brief history

- Numerical approaches
  - Finite element method

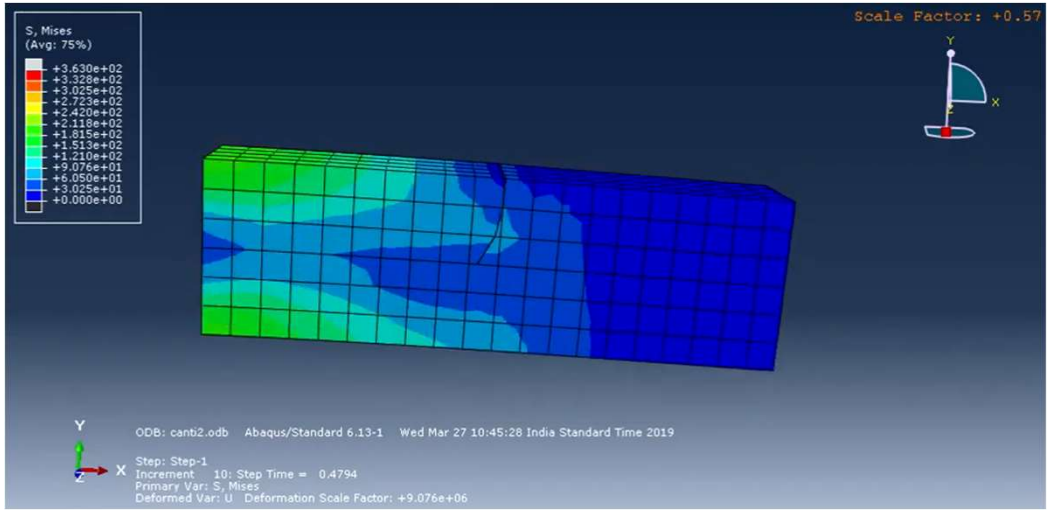




- **Re-meshing** near the crack tip due to mesh conformance
- Computational cost is too **expensive**.

Brief history

- Numerical approaches
  - XFEM (1999, Moës et al.) / Phantom-node method (2004, Hansbo et al.)



- Modeling **discontinuities**
- Explicitly **tracking** the crack surface
- Challenges in dealing with **complex and arbitrary crack paths**

Moës, N., Dolbow, J., and Belytschko, T. (1999). A finite element method for crack growth without remeshing. *International journal for numerical methods in engineering*, *46*(1), 131-150.

Hansbo, A., and Hansbo, P. (2004). A finite element method for the simulation of strong and weak discontinuities in solid mechanics. Computer methods in applied mechanics and engineering, 193(33-35), 3523-3540.

Brief history

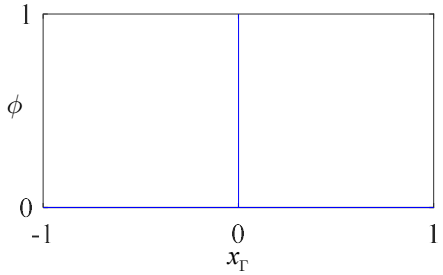

G.A. Francfort


J.-J. Marigo

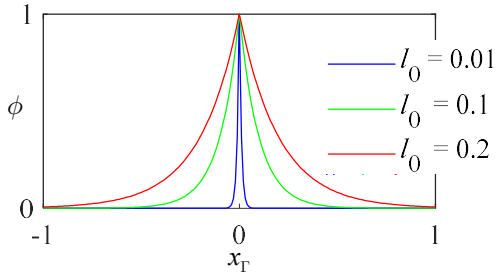
❖ Numerical approaches

- Phase field model
  - Francfort and Marigo (1998)

    - Variational approach to brittle fracture (Potential energy minimization)



Sharp crack model



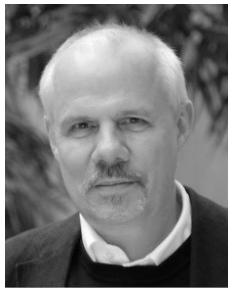
Diffusive crack model


B. Bourdin

  - Bourdin et al. (2000, 2007, 2008)

    - Regularization of the sharp crack topology into a diffusive crack

    - Introduce a scalar variable, called “damage parameter (phase field parameter)”.


C. Miehe

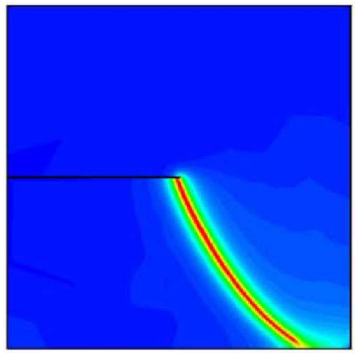
  - Miehe et al. (2010)

    - Introduce crack surface density function (Robust implementation).

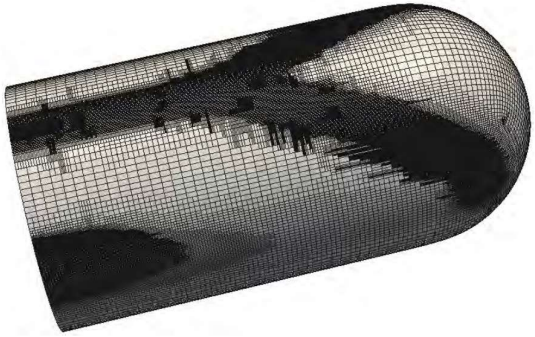
    - Strain energy decomposition

Phase field model

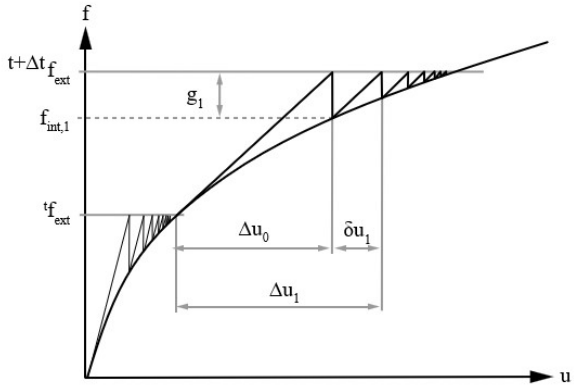
- Advantages
  - Handling arbitrary and complex crack paths
  - No additional crack propagation criteria
  - Not using sharp crack model
- Motivation
  - High computational cost



Phase field method (Miehe et al. 2010)



Many DOFs due to mesh refinement
(Borden et al. 2012)

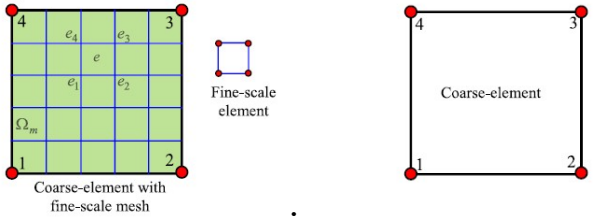


Iterative solution procedure
(e.g. Newton-Raphson method)

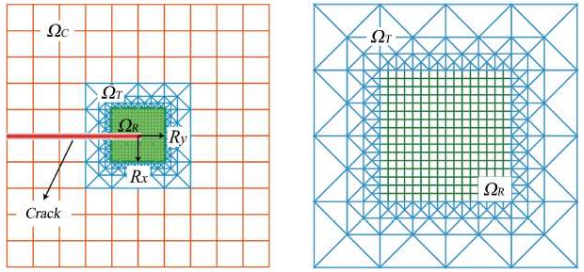
Is there a method to improve computational efficiency?

Related studies

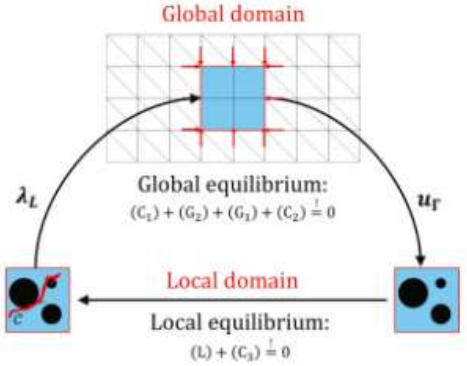
- Application of adaptive mesh refinements to the phase field model
  - Patil et al. (2018)
    - Multiscale approach
    - Equivalent stiffness matrix $\overline{K}_e^{\mathbf{u}} = G_e^{\mathbf{u}T} K_e^{\mathbf{u}} G_e^{\mathbf{u}}$
    - High computational cost due to constructing transformation matrix



  - Tian et al. (2019)
    - Mesh transition (triangular elements)
  - Muixí et al. (2020)
    - Nitsche's method
    - Enforcing the compatibility at non-matching mesh interface without Lagrange multiplier
    - Depending on stability parameter

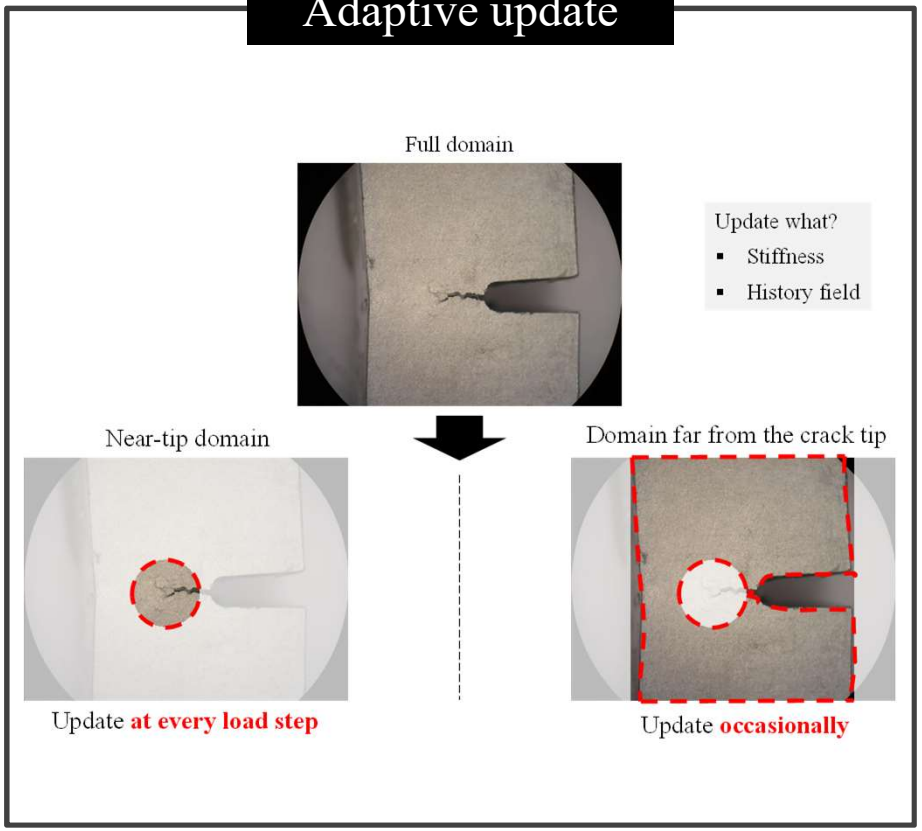


  - Noii et al. (2020)
    - Global-local approach
    - Enforcing the compatibility at non-matching mesh interface using Lagrange multiplier

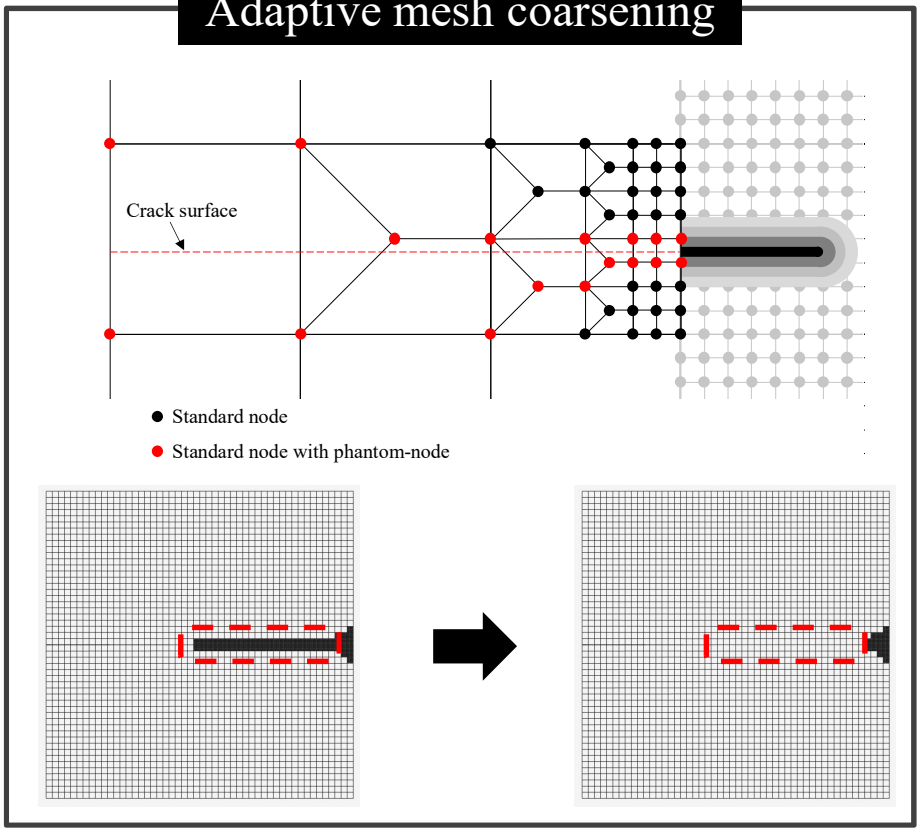


Research objective

Adaptive update



Adaptive mesh coarsening



Maintaining accuracy, improving computational efficiency!

2. Research background

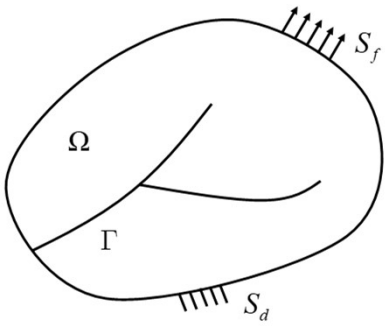
Phase field approximation

- Potential energy functional based on Griffith's theory
  - $\Pi(\mathbf{u},\Gamma) = \int_V \psi_e(\boldsymbol{\varepsilon})dV + G_c\int_\Gamma d\Gamma - \int_V \mathbf{f}_b\cdot\mathbf{u}dV - \int_{S_f}\mathbf{f}_s\cdot\mathbf{u}dS$

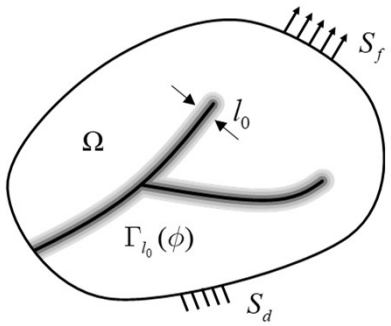
$$G_c\int_\Gamma d\Gamma \approx G_c\int_V \gamma(\phi,\nabla\phi)dV$$
$$\gamma(\phi,\nabla\phi) = \frac{1}{2}l_0\nabla\phi\cdot\nabla\phi + \frac{\phi^2}{2l_0}$$

Crack surface density function

- Potential energy functional proposed by Miehe et al.
  - $\Pi(\mathbf{u},\phi) = \int_V \psi_e(\boldsymbol{\varepsilon})dV + \int_V G_c\left[\frac{1}{2}l_0\nabla\phi\cdot\nabla\phi + \frac{\phi^2}{2l_0}\right]dV - \int_V \mathbf{f}_b\cdot\mathbf{u}dV - \int_{S_f}\mathbf{f}_s\cdot\mathbf{u}dS$



Sharp crack model



Diffusive crack model

$$0 \le \phi < 1$$
$\phi = 0$: undagamed
$\phi = 1$: fully cracked

G_c : Critical energy release rate
ϕ : Damage or phase field parameter
γ : Crack surface density function
ψ_e : Elastic strain energy density
l_o : Characteristic length

Miehe, C., Welschinger, F., and Hofacker, M. (2010). Thermodynamically consistent phase-field models of fracture: Variational principles and multi-field FE implementations. International journal for numerical methods in engineering, 83(10), 1273-1311.

Phase field approximation

❖ Decomposition of strain tensor proposed by Miehe et al.

- $\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}_- + \boldsymbol{\varepsilon}_+$

  where

$$\boldsymbol{\varepsilon}_\pm = \sum \langle \varepsilon_p \rangle_\pm \, \mathbf{n}_p \otimes \mathbf{n}_p$$

$$\langle x \rangle_\pm = \frac{1}{2}\left(x \pm |x|\right)$$

ε_a : Eigenvalue of the strain tensor

$\mathbf{n}_a$: Eigenvector of the strain tensor

$\langle\ \rangle$: Macaulay bracket operator

❖ Decomposition of elastic strain energy density function

- $\psi_e(\boldsymbol{\varepsilon}, \phi) = \psi_e^-(\boldsymbol{\varepsilon}) + [(1-\phi)^2 + k]\psi_e^+(\boldsymbol{\varepsilon})$

  where

$$\psi_e^\pm(\boldsymbol{\varepsilon}) = \frac{1}{2}\lambda \langle \mathrm{tr}(\boldsymbol{\varepsilon}) \rangle_\pm^2 + \mu \mathrm{tr}(\boldsymbol{\varepsilon}_\pm^2)$$

λ : 1st Lamé's constant

μ : 2nd Lamé's constant

$(1-\phi)^2 + k$: Degradation function

Miehe, C., Hofacker, M., & Welschinger, F. (2010). A phase field model for rate-independent crack propagation: Robust algorithmic implementation based on operator splits. Computer Methods in Applied Mechanics and Engineering, 199(45-48), 2765-2778.

Governing equations

- Strong form governing equations
  - $\nabla \cdot \boldsymbol{\sigma} + \mathbf{f}_b = 0$
  - $\boldsymbol{\sigma} = [(1-\phi)^2 + k]\dfrac{\partial \psi_e(\boldsymbol{\varepsilon})}{\partial \boldsymbol{\varepsilon}}$
  - $\left[\dfrac{G_c}{l_0} + 2\psi_e^+\right]\phi - G_c l_0 \nabla\phi \cdot \nabla\phi = 2\psi_e^+$
- Boundary conditions
  - $\mathbf{u} = \mathbf{u}_d \quad \text{on} \quad S_d$
  - $\boldsymbol{\sigma} \cdot \mathbf{n} = \mathbf{f}_s \quad \text{on} \quad \mathrm{S}_f$
  - $\nabla\phi \cdot \mathbf{n} = 0 \quad \text{on} \quad S$
- Irreversibility of crack growth
  - $H(\mathbf{x},t) = \max_{s\in[0,t]} \psi_e^+(\boldsymbol{\varepsilon}(\mathbf{x},s))$

Anisotropic formulation

$$\boldsymbol{\sigma} = \frac{\partial \psi_e^-(\boldsymbol{\varepsilon})}{\partial \boldsymbol{\varepsilon}} + [(1-\phi)^2 + k]\frac{\partial \psi_e^+(\boldsymbol{\varepsilon})}{\partial \boldsymbol{\varepsilon}}$$

Hybrid formulation

$$\boldsymbol{\sigma} = [(1-\phi)^2 + k]\frac{\partial \psi_e(\boldsymbol{\varepsilon})}{\partial \boldsymbol{\varepsilon}}$$

$H(\mathbf{x},t)$: Local history field

Finite element discretization

❖ Incremental equilibrium equation for displacement field at time t

- $^{t}\mathbf{K}_{u}^{(n)}\Delta\mathbf{U}^{(n)} = {}^{t}\mathbf{R} - {}^{t}\mathbf{F}_{u}^{(n)}$ with

$$^{t}\mathbf{K}_{u}^{(n)} = \mathsf{A}_{m=1}^{e}\, {}^{t}\mathbf{K}_{u}^{(m)(n)}, \quad {}^{t}\mathbf{R} = \mathsf{A}_{m=1}^{e}\, {}^{t}\mathbf{R}^{(m)} \quad \text{and} \quad {}^{t}\mathbf{F}_{u}^{(n)} = \mathsf{A}_{m=1}^{e}\, {}^{t}\mathbf{F}_{u}^{(m)(n)}$$

$$^{t}\mathbf{K}_{u}^{(m)(n)} = \int_{V^{(m)}} [(1 - {}^{t}\phi^{(m)})^2 + k](\mathbf{B}_{u}^{(m)})^{\mathrm{T}}\mathbf{C}^{(m)}\mathbf{B}_{u}^{(m)} dV^{(m)}$$

$$^{t}\mathbf{R}^{(m)} = \int_{V^{(m)}} (\mathbf{H}_{u}^{(m)})^{\mathrm{T}}\, {}^{t}\mathbf{f}_{b}^{(m)} dV^{(m)} + \int_{S_f{}^{(m)}} (\mathbf{H}_{u}^{(m)})^{\mathrm{T}}\, {}^{t}\mathbf{f}_{s}^{(m)} dS^{(m)}$$

$$^{t}\mathbf{F}_{u}^{(m)(n)} = \int_{V^{(m)}} [(1 - {}^{t}\phi^{(m)})^2 + k](\mathbf{B}_{u}^{(m)})^{\mathrm{T}}\, {}^{t}\boldsymbol{\sigma}^{(m)(n)} dV^{(m)}$$

$$^{t}\boldsymbol{\sigma}^{(m)(n)} = \begin{bmatrix} {}^{t}\sigma_{xx}^{(m)(n)} & {}^{t}\sigma_{yy}^{(m)(n)} & {}^{t}\sigma_{xy}^{(m)(n)} \end{bmatrix}^{\mathrm{T}}$$

- Displacement field

$^{t}\mathbf{K}_{u}^{(n)}$: Tangential stiffness matrix

$\Delta\mathbf{U}^{(n)}$: Incremental displacement vector

$^{t}\mathbf{R}$: External force vector

$^{t}\mathbf{F}_{u}^{(n)}$: Internal force vector

Finite element discretization

❖ Incremental equilibrium equation for phase field at time t

- $^{t}\mathbf{K}_{\phi}^{(n)}\Delta\mathbf{\Phi}^{(n)} = -{}^{t}\mathbf{F}_{\phi}^{(n)}$ with

$$^{t}\mathbf{K}_{u}^{(n)} = \mathsf{A}_{m=1}^{e}\,{}^{t}\mathbf{K}_{u}^{(m)(n)} \quad \text{and} \quad {}^{t}\mathbf{F}_{\phi}^{(n)} = \mathsf{A}_{m=1}^{e}\,{}^{t}\mathbf{F}_{\phi}^{(m)(n)}$$

$$^{t}\mathbf{K}_{\phi}^{(m)(n)} = \int_{V^{(m)}} \left(G_c l_0 (\mathbf{B}_{\phi}^{(m)})^{\mathrm{T}} \mathbf{B}_{\phi}^{(m)} + \left[\frac{G_c}{l_0} + 2\,{}^{t}H^{(m)} \right] (\mathbf{H}_{\phi}^{(m)})^{\mathrm{T}} \mathbf{H}_{\phi}^{(m)} \right) dV^{(m)}$$

$$^{t}\mathbf{F}_{\phi}^{(m)(n)} = -\int_{V^{(m)}} 2\,{}^{t}H^{(m)} (\mathbf{H}_{\phi}^{(m)})^{\mathrm{T}}\, dV^{(m)}$$

$$+ \int_{V^{(m)}} \left(G_c l_0 (\mathbf{B}_{\phi}^{(m)})^{\mathrm{T}} \nabla\, {}^{t}\phi^{(m)(n)} + \left[\frac{G_c}{l_0} + 2\,{}^{t}H^{(m)} \right] (\mathbf{H}_{\phi}^{(m)})^{\mathrm{T}}\, {}^{t}\phi^{(m)(n)} \right) dV^{(m)}$$

- Phase field

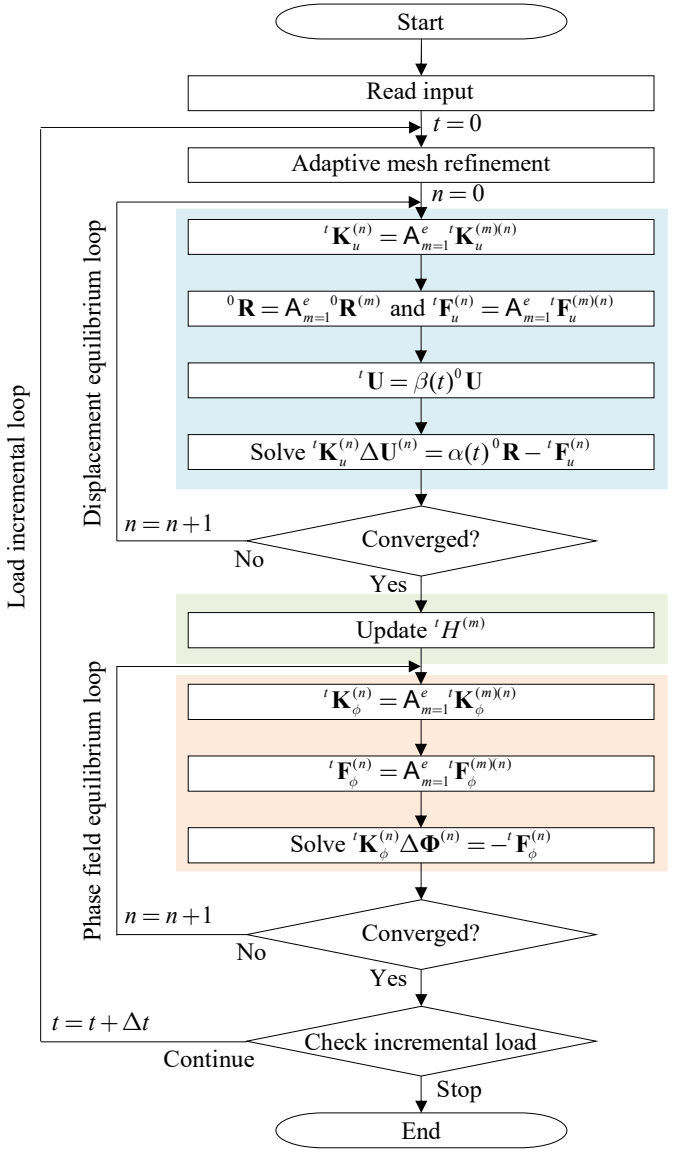
$^{t}\mathbf{K}_{\phi}^{(n)}$: Tangential stiffness matrix

$\Delta\mathbf{\Phi}^{(n)}$: Incremental displacement vector

$^{t}\mathbf{F}_{\phi}^{(n)}$: Internal force vector

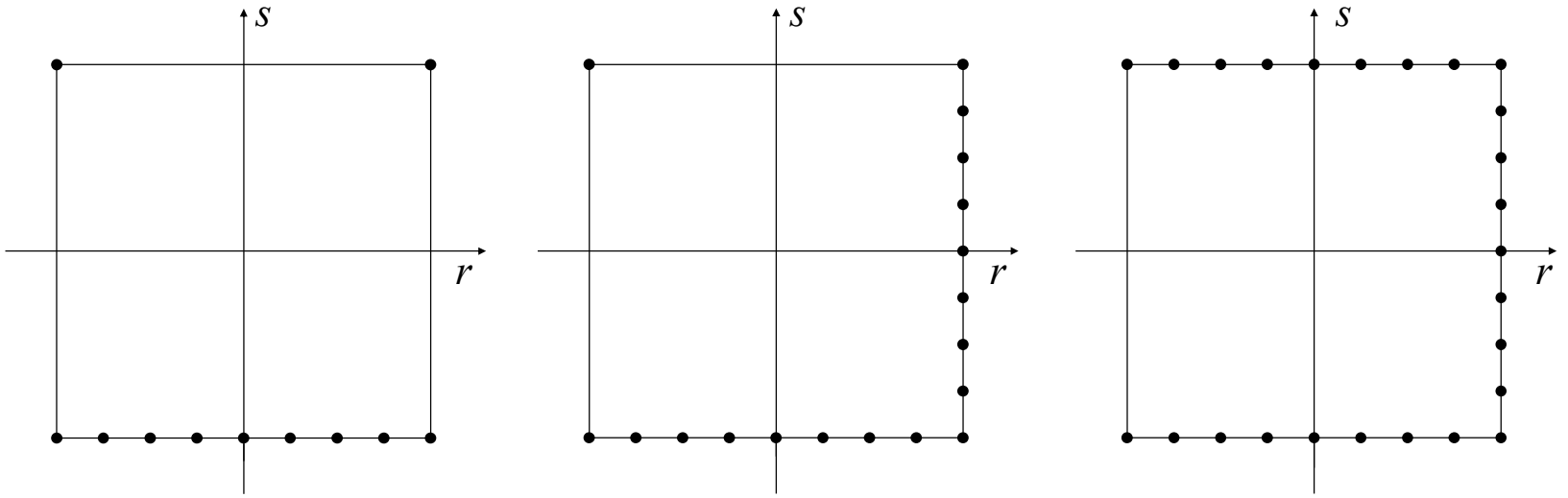
Whole numerical procedure

- Equilibrium for displacement field
  - $^{t}\mathbf{K}_{u}^{(n)} = \mathsf{A}_{m=1}^{e}\, {}^{t}\mathbf{K}_{u}^{(m)(n)}$
  - $^{0}\mathbf{R} = \mathsf{A}_{m=1}^{e}\, {}^{0}\mathbf{R}^{(m)}$ and $^{t}\mathbf{F}_{u}^{(n)} = \mathsf{A}_{m=1}^{e}\, {}^{t}\mathbf{F}_{u}^{(m)(n)}$
  - Solve $^{t}\mathbf{K}_{u}^{(n)}\Delta\mathbf{U}^{(n)} = \alpha(t)\,{}^{0}\mathbf{R} - {}^{t}\mathbf{F}_{u}^{(n)}$
- History field
  - Update $^{t}H^{(m)}$
- Equilibrium for phase field
  - $^{t}\mathbf{K}_{\phi}^{(n)} = \mathsf{A}_{m=1}^{e}\, {}^{t}\mathbf{K}_{\phi}^{(m)(n)}$
  - $^{t}\mathbf{F}_{\phi}^{(n)} = \mathsf{A}_{m=1}^{e}\, {}^{t}\mathbf{F}_{\phi}^{(m)(n)}$
  - Solve $^{t}\mathbf{K}_{\phi}^{(n)}\Delta\mathbf{\Phi}^{(n)} = -{}^{t}\mathbf{F}_{\phi}^{(n)}$



Adaptive mesh refinement

- Variable-node elements
  - To maximize the computational efficiency, variable-node elements are used.
  - It is possible to add nodes at edge of elements.
  - Shape functions are obtained using the moving least square (MLS).

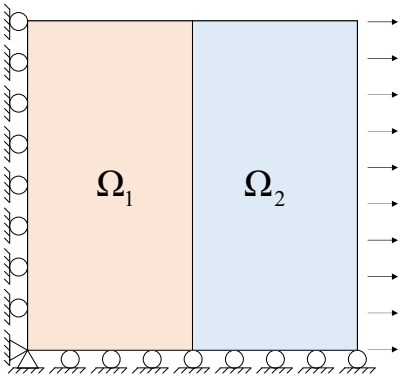


11-node element 18-node element 25-node element

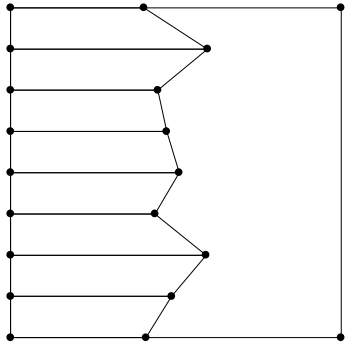
Cho, Y. S., Jun, S., Im, S., and Kim, H. G. (2005). An improved interface element with variable nodes for non-matching finite element meshes. *Computer Methods in Applied Mechanics and Engineering*, *194*(27-29), 3022-3046.

Adaptive mesh refinement

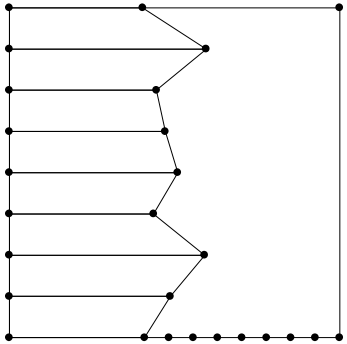
❖ Patch test for variable-node elements



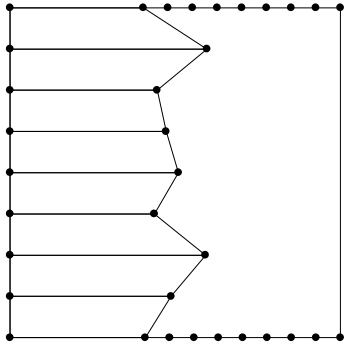
(a)



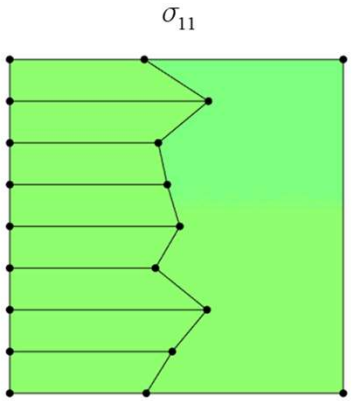
(b)
11-node element



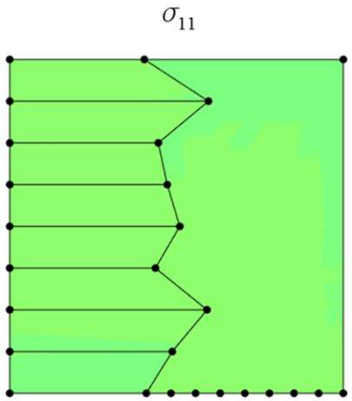
(c)
18-node element



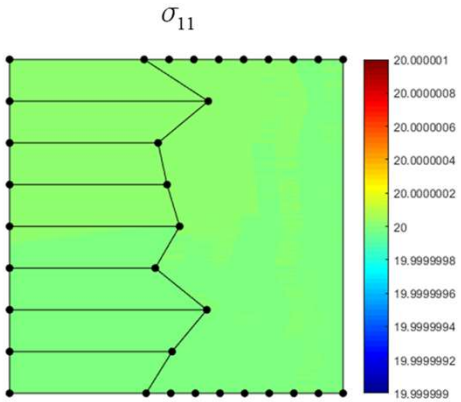
(d)
25-node element



(a)
11-node element



(b)
18-node element

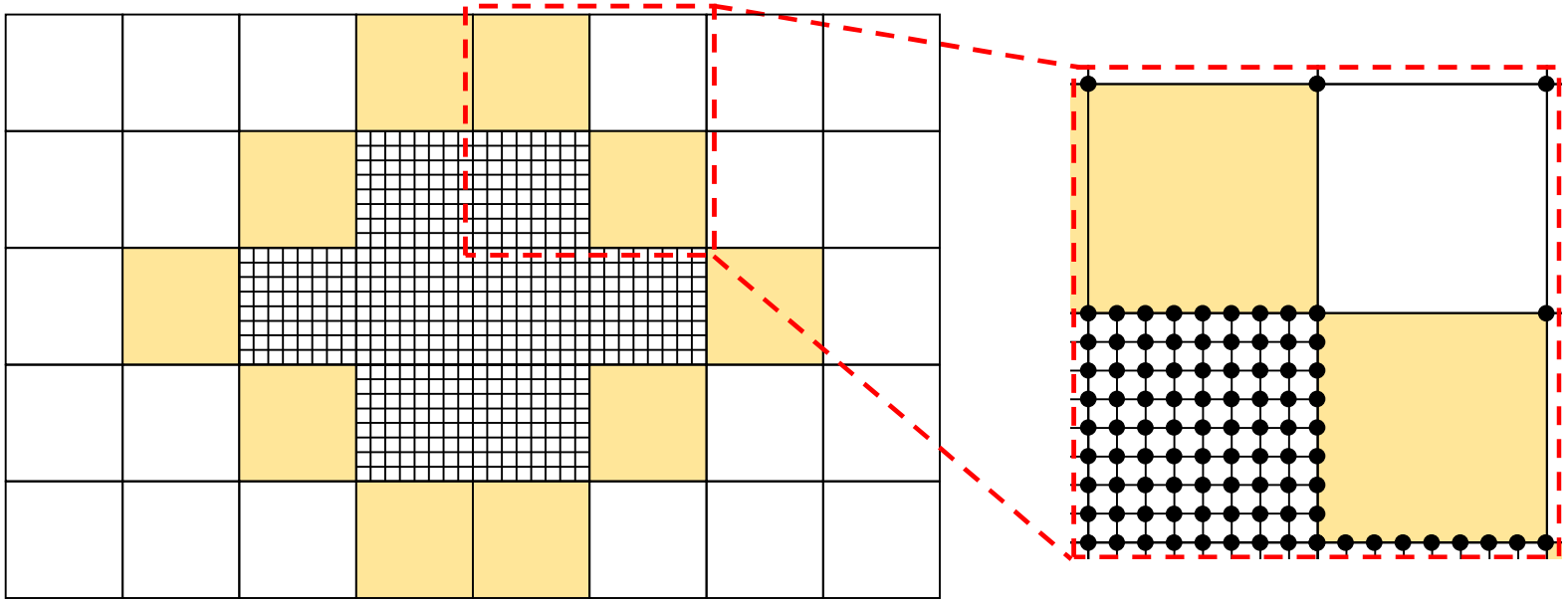


(c)
25-node element

Adaptive mesh refinement

❖ Variable-node elements

- Variable-node elements are utilized as transient elements.
- Coarse mesh is converted into 8×8 fine mesh.



Variable-node elements

Adaptive mesh refinement

❖ Domain identification

- $\Omega_{ct}^{t+\Delta t} = \{\mathbf{x}_i \mid \mathbf{x}_i \in \Omega_{nt}^{t},\ \phi_i^{t} < \phi_c\ ,\ \phi_i^{t+\Delta t} \geq \phi_c\}$
- $\Omega_{cpp}^{t+\Delta t} = \{\mathbf{x}_i \mid \mathbf{x}_i \notin \Omega_{ct}^{t+\Delta t},\ \phi_i^{t+\Delta t} \geq \phi_c\}$
- $\Omega_{nt}^{t+\Delta t} = \{\mathbf{x}_i \mid \|\mathbf{x}_i - \mathbf{x}_c\| \leq R_c,\ \mathbf{x}_c \in \Omega_{ct}^{t+\Delta t}\}$

❖ Fine mesh ⟷ Coarse mesh

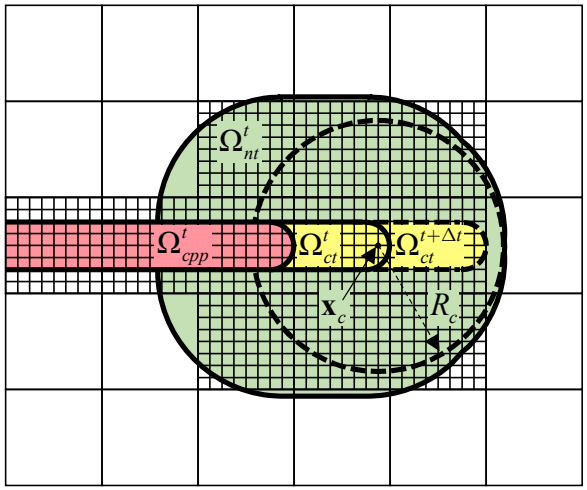
Ω_{ct}^{t} : **Crack tip** domain at time $t+\Delta t$

Ω_{cpp}^{t} : Domain of the **crack propagation path** at time $t+\Delta t$

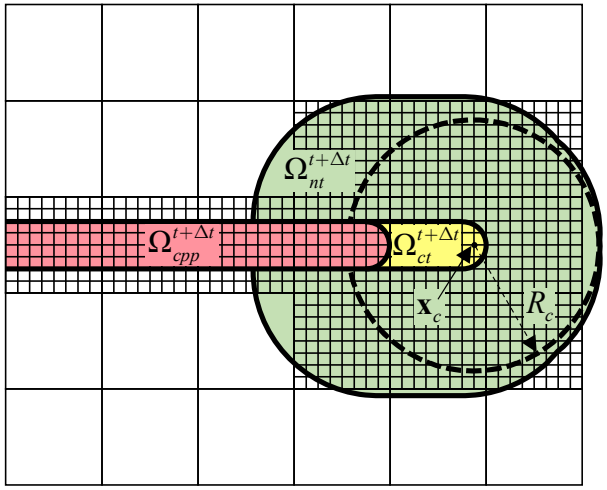
Ω_{nt}^{t} : **Near-tip** domain at time $t+\Delta t$

ϕ_c : Critical value of the damage parameter

R_c : Radius of the near-tip domain



At time t



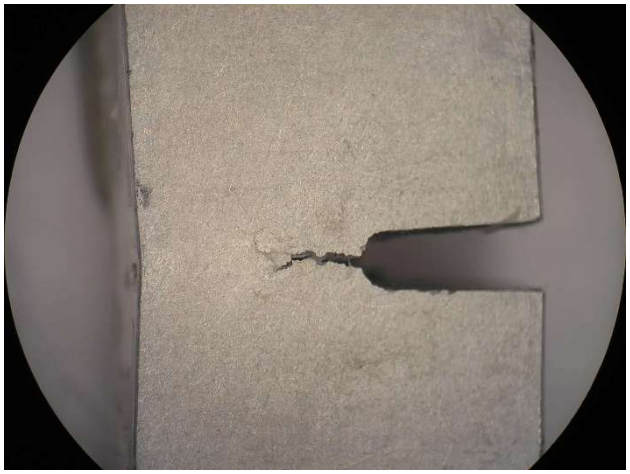
At time $t+\Delta t$

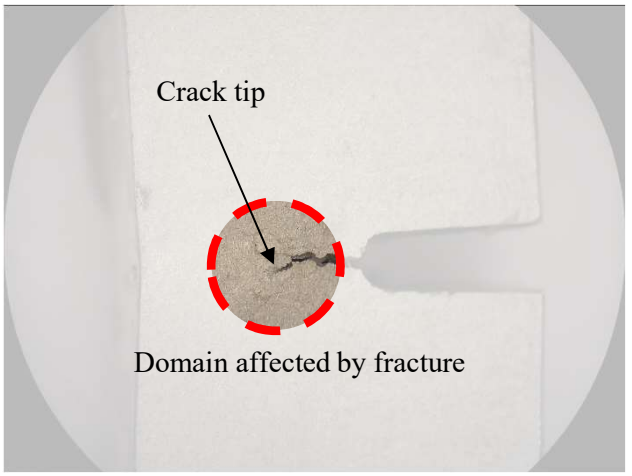
3. Research topics

Topic 1. Adaptive update scheme

Motivation

❖ Observation



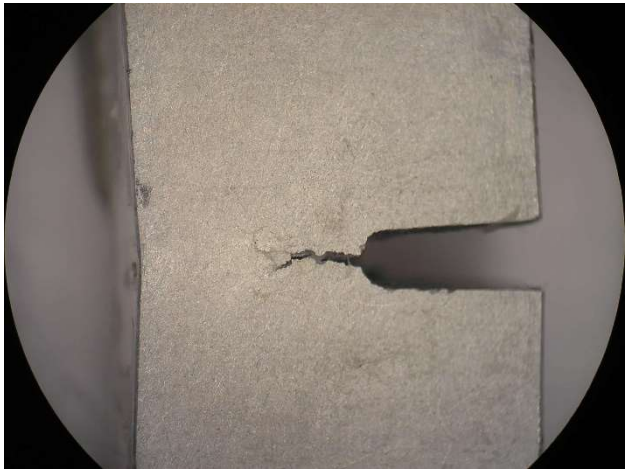


- Fracture is a **local** phenomenon.
- During propagation, structural properties **change rapidly only in the near-tip** domain.

Structural properties rarely vary in the domain far from the crack tip!

Motivation

Full domain



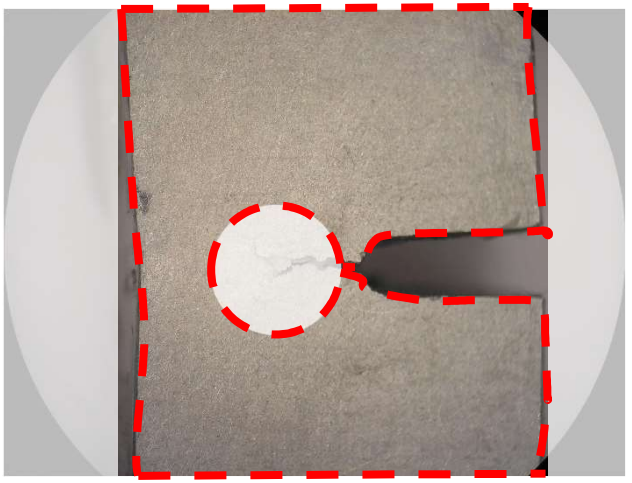
Update what?

- Stiffness
- History field

Near-tip domain

Update **at every load step**

Domain far from the crack tip



Update **occasionally**

Adaptive update

❖ Key idea

- ${}^t\mathbf{K}_t^{(n)} = {}^t\mathbf{K}_u^{nt(n)} + {}^t\mathbf{K}_u^{ft(n)}$ and ${}^t\mathbf{K}_\phi^{(n)} = {}^t\mathbf{K}_\phi^{nt(n)} + {}^t\mathbf{K}_\phi^{ft(n)}$

  with

$$ {}^t\mathbf{K}_u^{nt(n)} = \mathsf{A}_{m=1}^{e^{nt}} {}^t\mathbf{K}_u^{(m)(n)} \quad , \quad {}^t\mathbf{K}_u^{ft(n)} = \mathsf{A}_{m=1}^{e^{ft}} {}^t\mathbf{K}_u^{(m)(n)} $$

$$ {}^t\mathbf{K}_\phi^{nt(n)} = \mathsf{A}_{m=1}^{e^{nt}} {}^t\mathbf{K}_\phi^{(m)(n)} \quad , \quad {}^t\mathbf{K}_\phi^{ft(n)} = \mathsf{A}_{m=1}^{e^{ft}} {}^t\mathbf{K}_\phi^{(m)(n)} $$

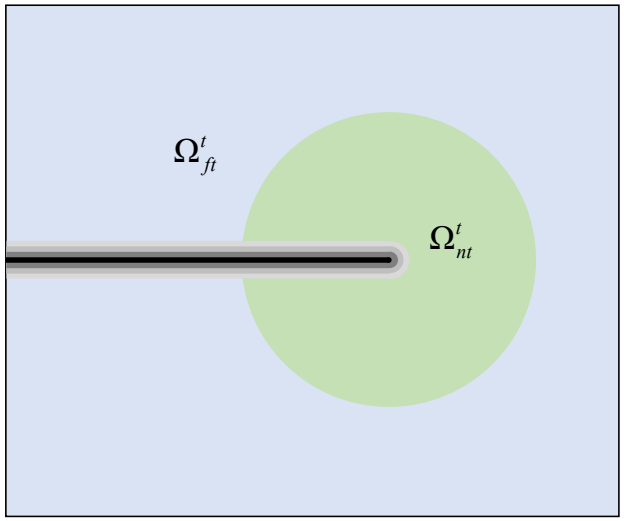
Ω_{ft} : Domain far from the crack tip

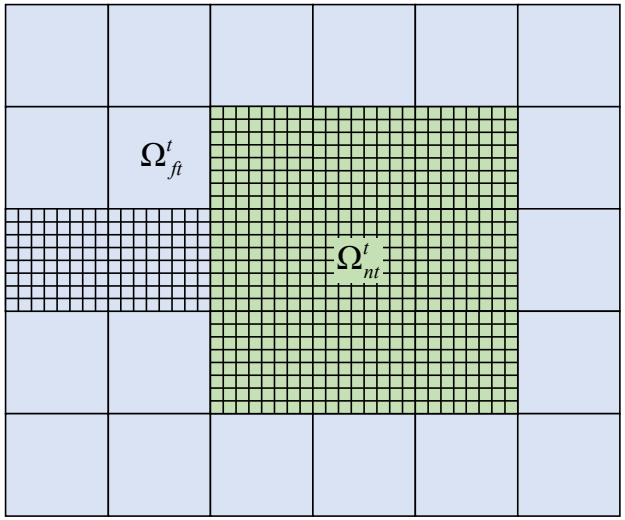
Ω_{nt} : Near-tip domain

- ${}^tH^{(m)}$ on Ω_{nt} , ${}^tH^{(m)}$ on Ω_{ft}

: Update **at every load step**

: Update **occasionally**





Adaptive update

❖ Update criteria

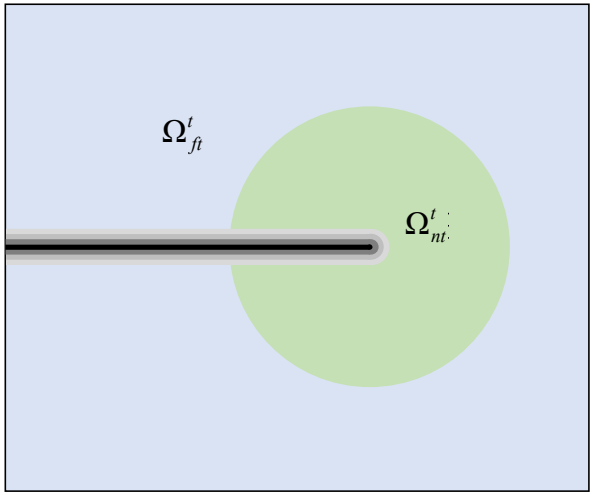
- When the following criteria is satisfied, the update is performed:

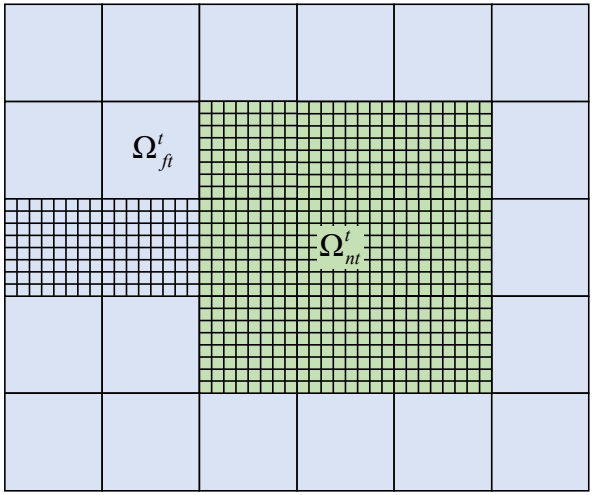
$$\frac{\sum_{k=t_u}^{t} \max_{\phi_i^k \in \Omega_{ct}^k} (\phi_i^k)}{\max_{\phi_i^t \in \Omega_{ct}^t} (\phi_i^t)} \geq \eta$$

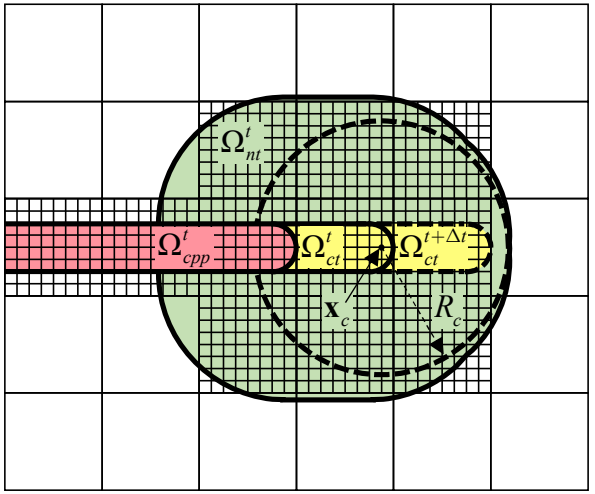
- After the criteria, t_u is set to the next load step.

Ω_{ct} : Crack tip domain
ϕ_i^t : Damage parameter
η : Update parameter
t_u : Initial update load step

- The update is additionally performed when the crack propagates ($\Omega_{ct}^t \neq \Omega_{ct}^{t-\Delta t}$).

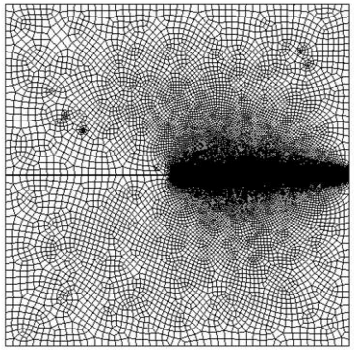




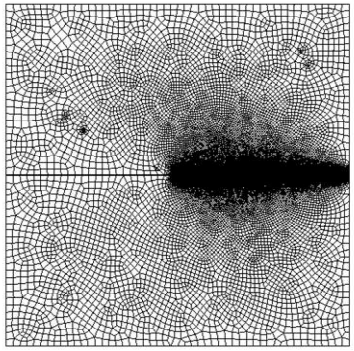


Numerical examples

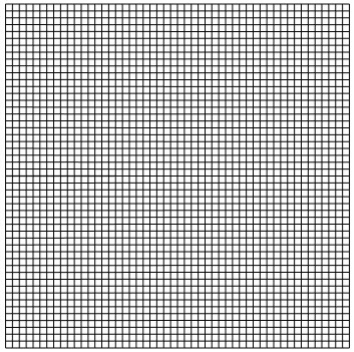
❖ 4 cases



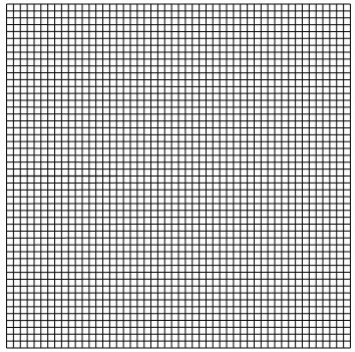
Local mesh refinement



Local mesh refinement
+ **Adaptive update**



Adaptive mesh refinement

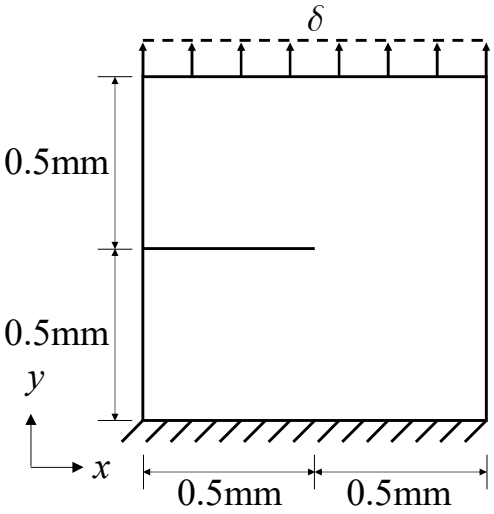


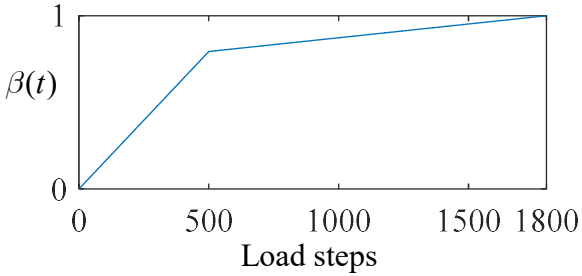
Adaptive mesh refinement
+ **Adaptive update**

- PC : an Intel(R) Core(TM) i7-7700 CPU @ 3.60GHz and 64GB RAM.
- 5 numerical examples
  - Single-edge notched tension problem
  - Notched plate with three holes
  - Single-edge notched shear plate with diamond-shaped holes
  - L-shaped panel
  - Single-edge notched branching problem
- $\phi_c = 0.5$ (Critical value of the damage parameter)
- $\eta = 10$ (Update parameter)
- Element length of fine mesh : $l_e \leq 0.5 l_0$

Numerical examples

❖ 1. Single-edge notched tension problem





$E = 210\text{GPa}$

$v = 0.3$

$G_c = 2.7 \times 10^{-3}\text{kN/mm}$

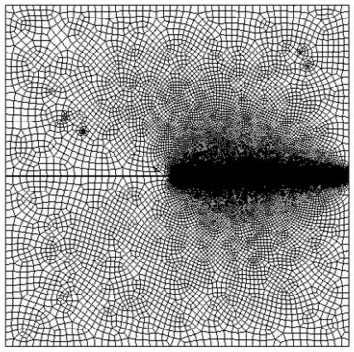
$l_0 = 0.0075\text{mm}$

$R_c = 0.04\text{mm}$

$\eta = 10$

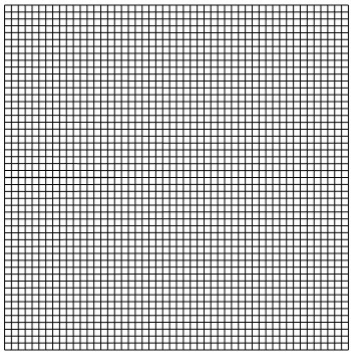
$\phi_c = 0.5$

${}^t\mathbf{U} = \beta(t)\,{}^0\mathbf{U}$



21791 elements

For local mesh refinement

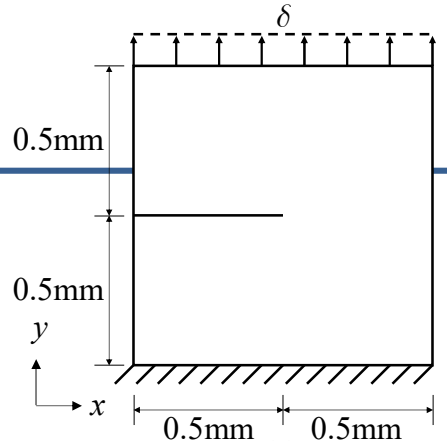


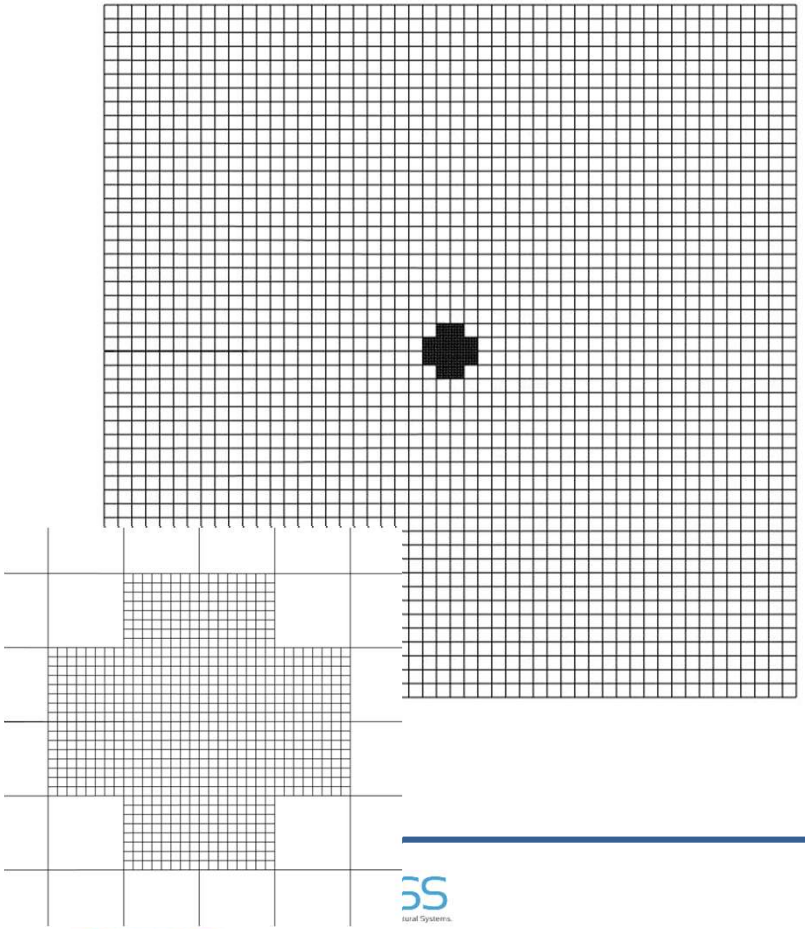
2500 elements

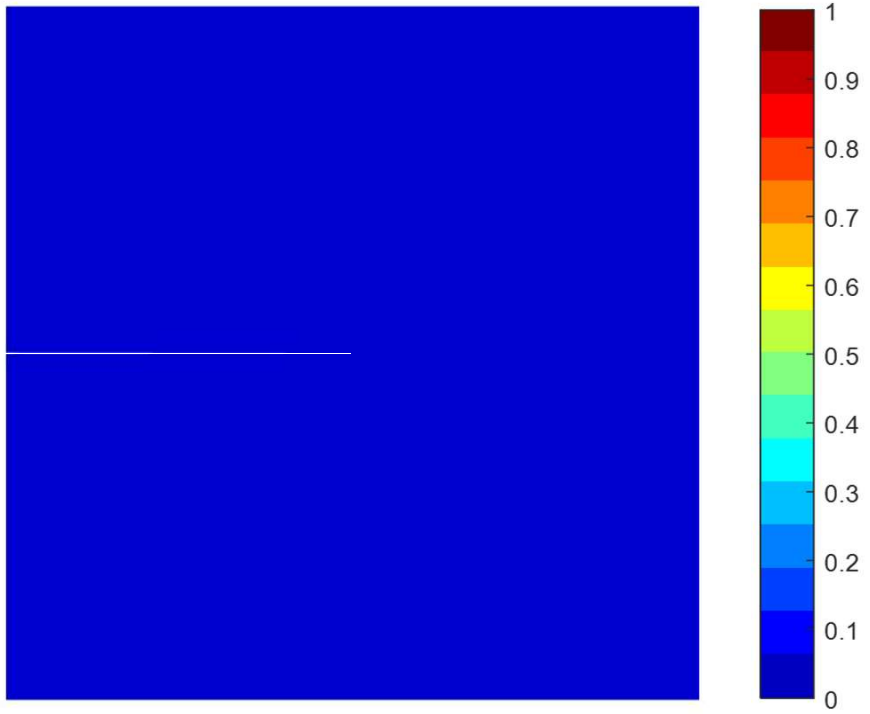
For adaptive mesh refinement

Numerical examples

❖ 1. Single-edge notched tension problem

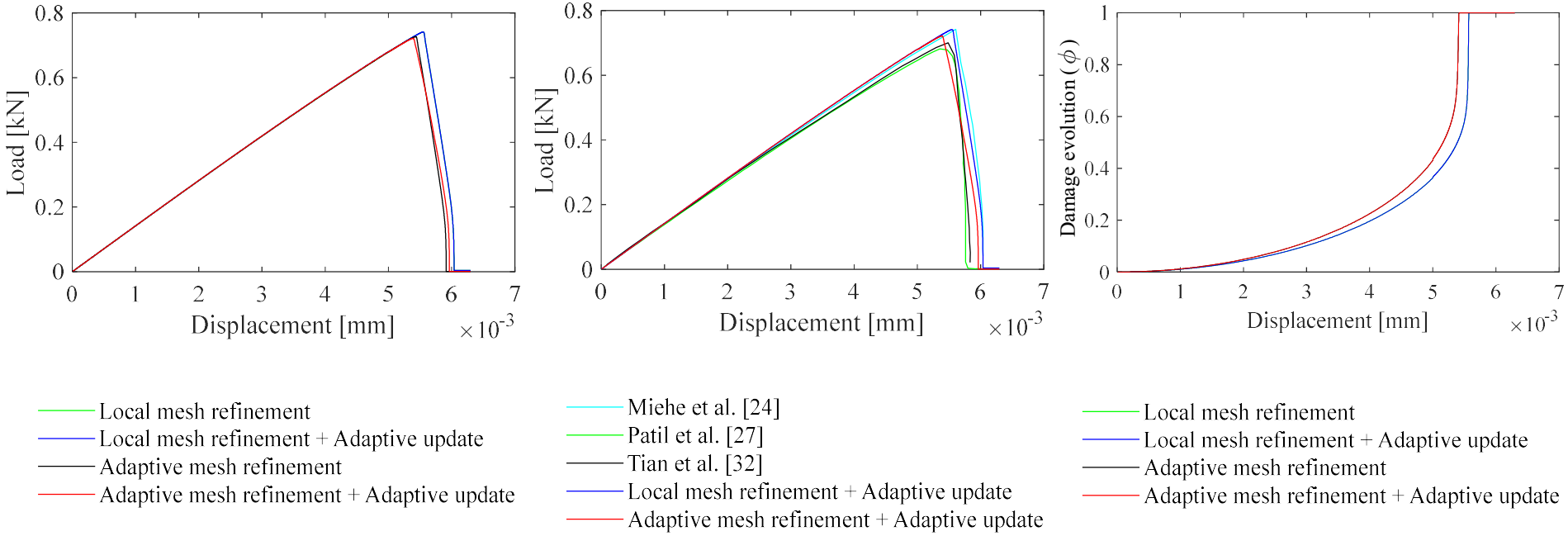






Numerical examples

❖ 1. Single-edge notched tension problem



Numerical examples

❖ 1. Single-edge notched tension problem

local mesh refinement scheme

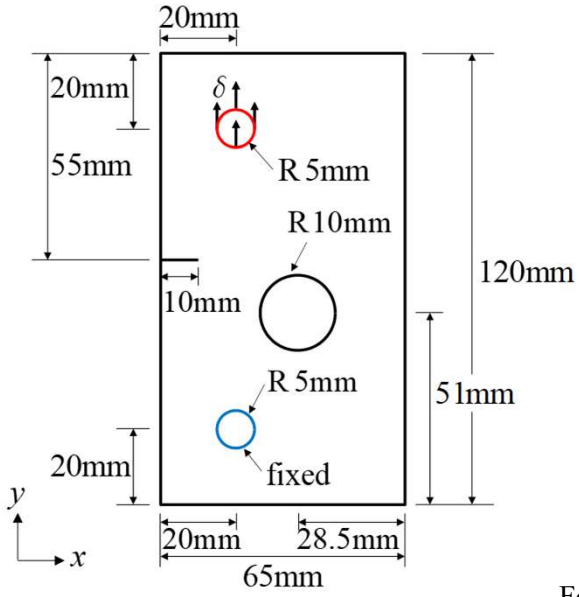
Numerical schemes applied	Items	Computation time	
		[sec]	Ratio [%]
Local mesh refinement	History field update	6863.18	21.36
	Stiffness construction	22250.76	69.23
	Equation solving	2928.56	9.11
	Etc	97.00	0.30
	Total	32139.50	100.00
Local mesh refinement + **Adaptive update**	History field update	2132.38	6.01
	Stiffness construction	16034.04	45.12
	Equation solving	4535.24	12.76
	Etc	364.32	1.02
	Total	23065.98	64.91

adaptive mesh refinement scheme

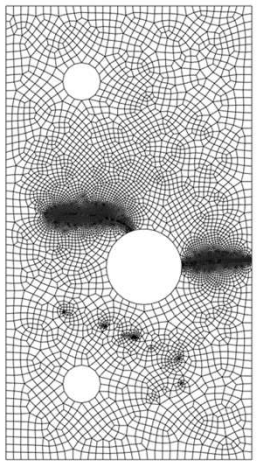
Numerical schemes applied	Items	Computation time	
		[sec]	Ratio [%]
Adaptive mesh refinement	Mesh refinement	25.99	0.54
	History field update	1025.52	21.24
	Stiffness construction	3504.25	72.59
	Equation solving	252.97	5.24
	Etc	18.68	0.39
	Total	4827.41	100.00
Adaptive mesh refinement + **Adaptive update**	Mesh refinement	19.68	0.41
	History field update	154.94	3.21
	Stiffness construction	1886.50	39.08
	Equation solving	343.08	7.11
	Etc	15.83	0.32
	Total	2420.03	50.13

Numerical examples

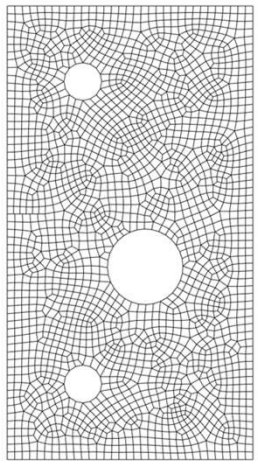
❖ 2. Notched plate with three holes



(a)



7512 elements
For local mesh refinement



2223 elements
For adaptive mesh refinement

$\lambda = 1.94\text{GPa}$

$\mu = 2.45\text{GPa}$

$G_c = 2.28\times10^{-3}\text{kN/mm}$

$l_0 = 0.5\text{mm}$

$R_c = 4\text{mm}$

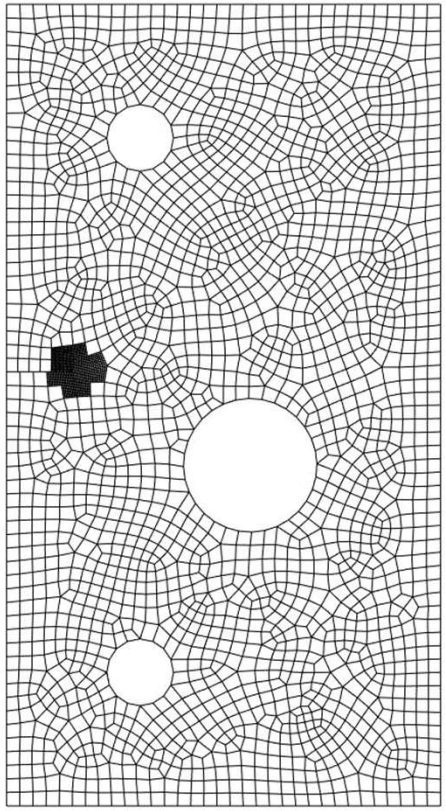
$\eta = 10$

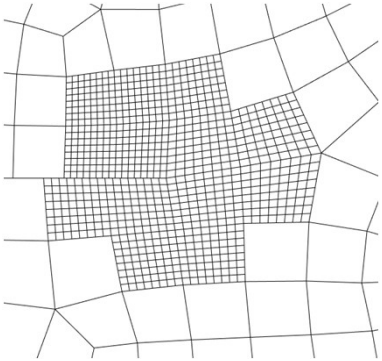
$\phi_c = 0.5$

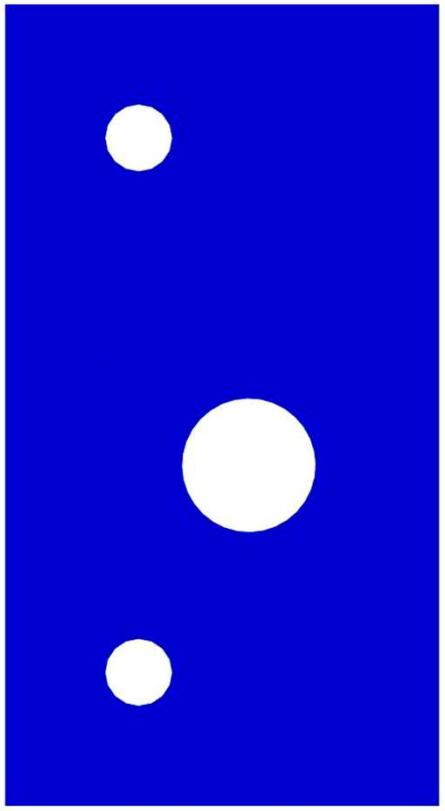
${}^t\mathbf{U} = \beta(t)\,{}^0\mathbf{U}$

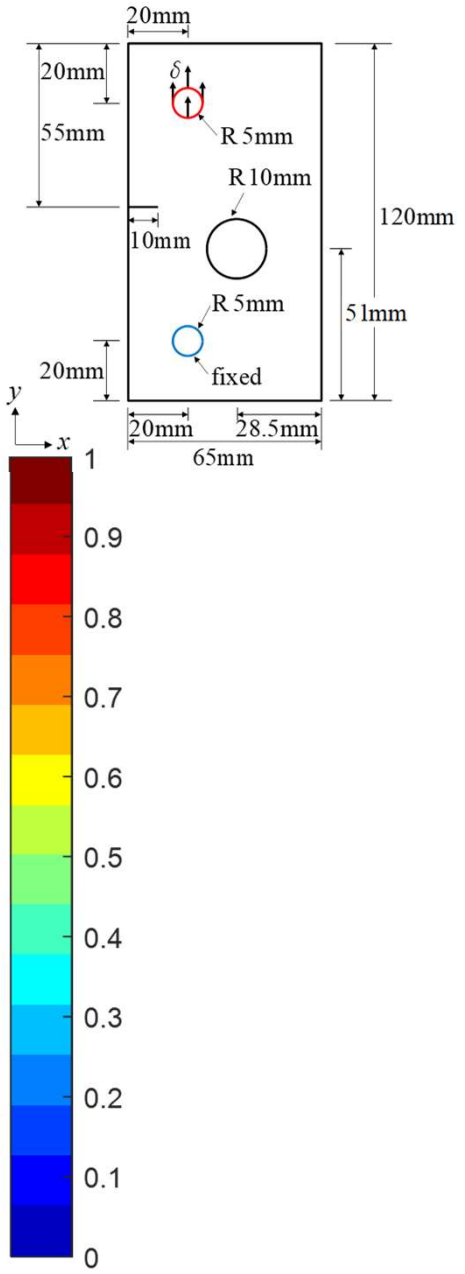
Numerical examples

❖ 2. Notched plate with three holes



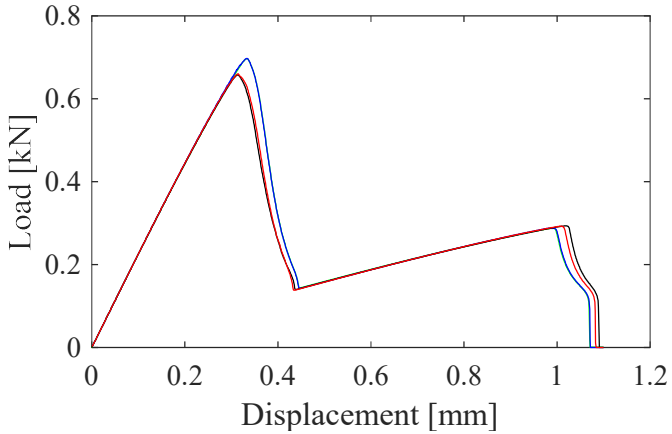


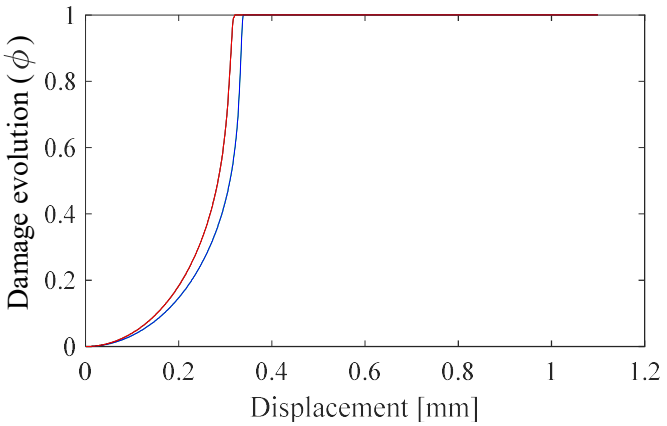




Numerical examples

❖ 2. Notched plate with three holes

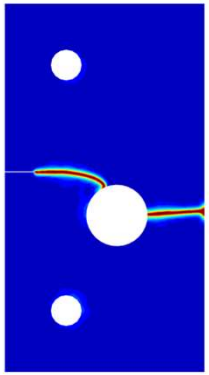




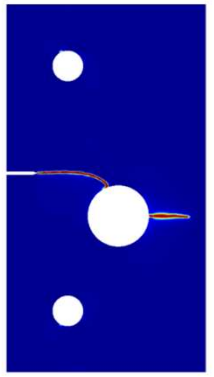
Local mesh refinement
Local mesh refinement + Adaptive update
Adaptive mesh refinement
Adaptive mesh refinement + Adaptive update

Local mesh refinement
Local mesh refinement + Adaptive update
Adaptive mesh refinement
Adaptive mesh refinement + Adaptive update

Numerical schemes applied	Computation time	
	[sec]	Ratio [%]
Local mesh refinement	6917.37	100.00
Local mesh refinement + **Adaptive update**	4565.42	66.00
Adaptive mesh refinement	4146.08	100.00
Adaptive mesh refinement + **Adaptive update**	2358.78	56.92



Numerical result

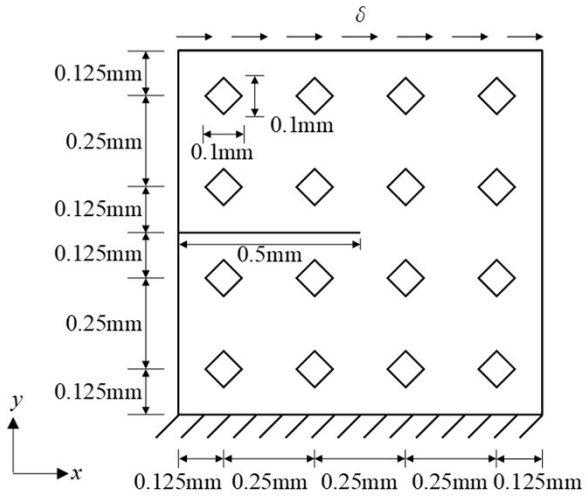


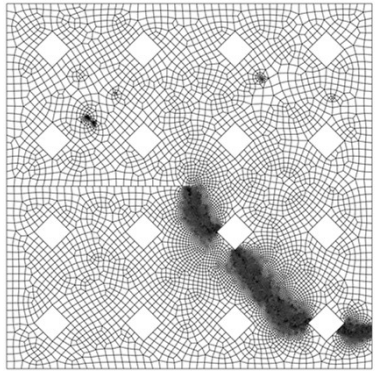
Numerical result by Ambati et al.

Experimental result by Ambati et al.

Numerical examples

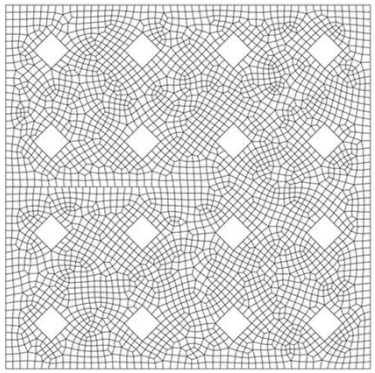
❖ 3. Single-edge notched shear plate with diamond-shaped holes





7512 elements

For local mesh refinement



2223 elements

For adaptive mesh refinement

$E = 210\text{GPa}$

$v = 0.3$

$G_c = 2.7 \times 10^{-3}\text{kN/mm}$

$l_0 = 0.0075\text{mm}$

$R_c = 0.04\text{mm}$

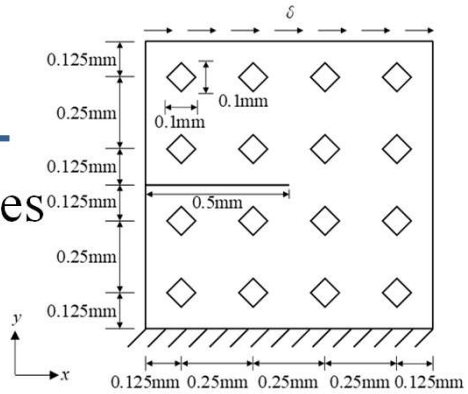
$\eta = 10$

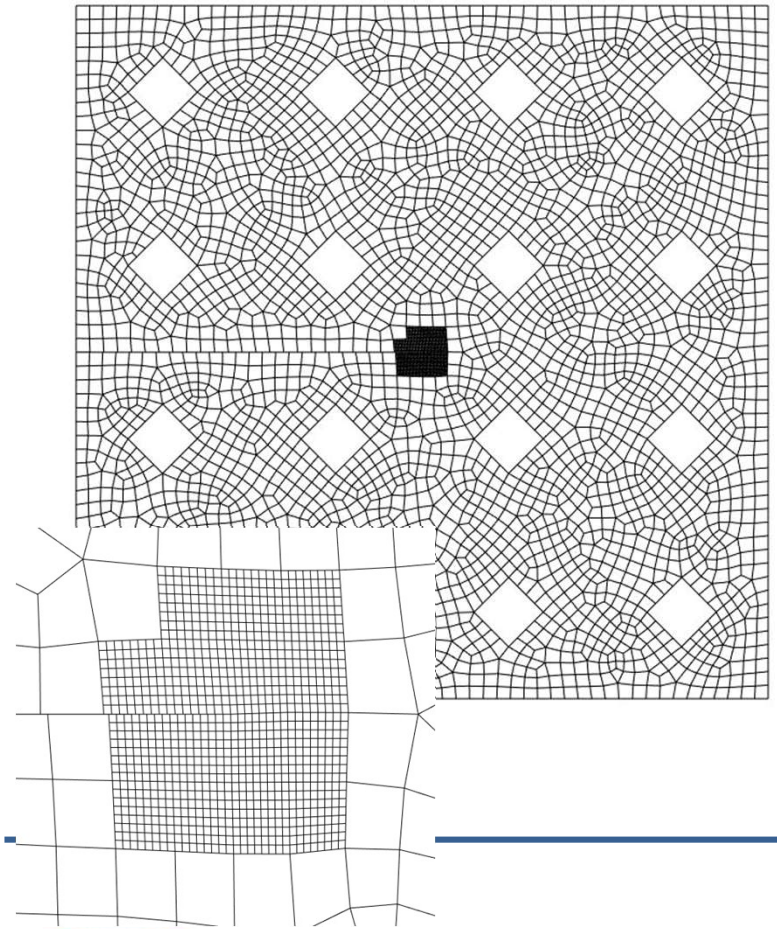
$\phi_c = 0.5$

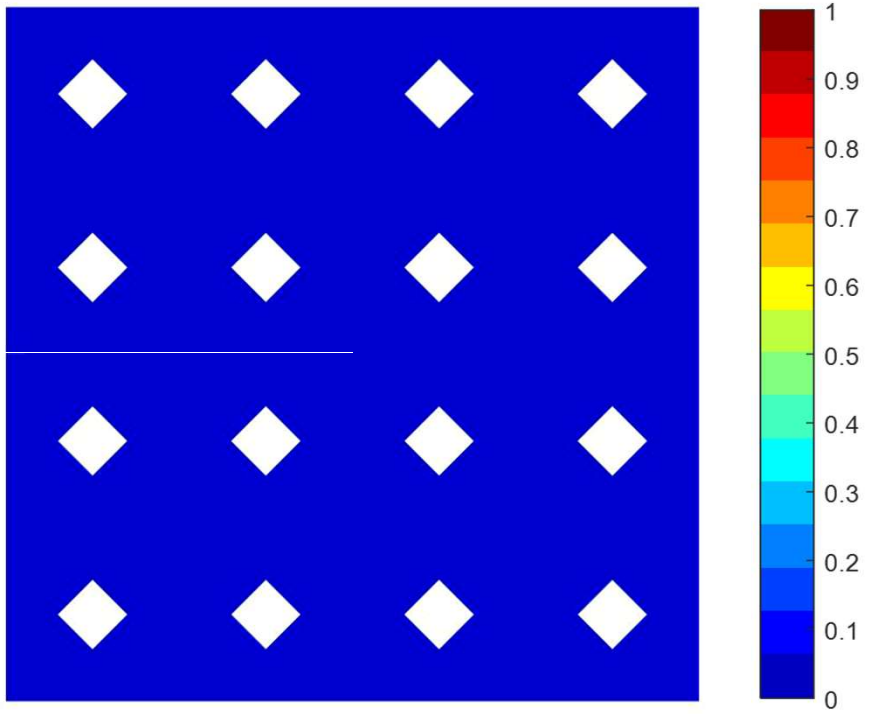
${}^t\mathbf{U} = \beta(t)\,{}^0\mathbf{U}$

Numerical examples

❖ 3. Single-edge notched shear plate with diamond-shaped holes

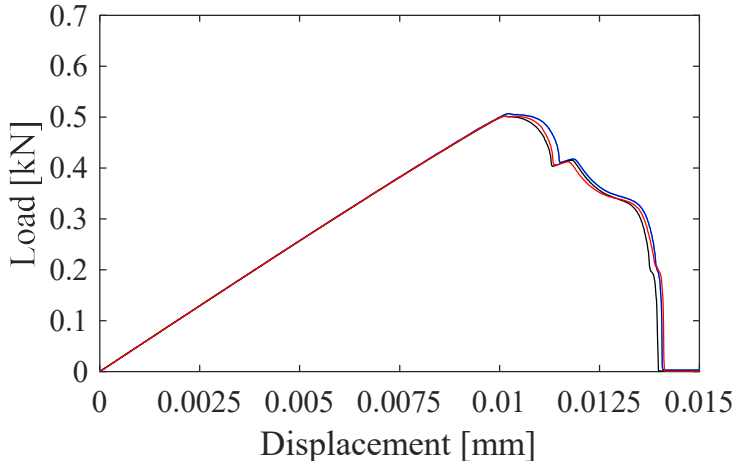


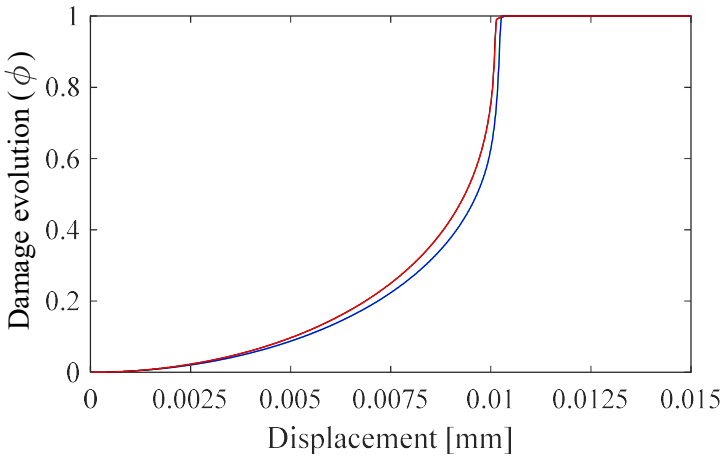




Numerical examples

❖ 3. Single-edge notched shear plate with diamond-shaped holes





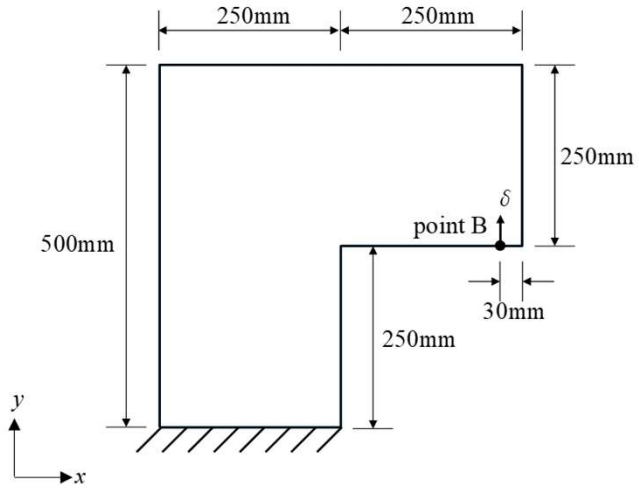
Local mesh refinement
Local mesh refinement + Adaptive update
Adaptive mesh refinement
Adaptive mesh refinement + Adaptive update

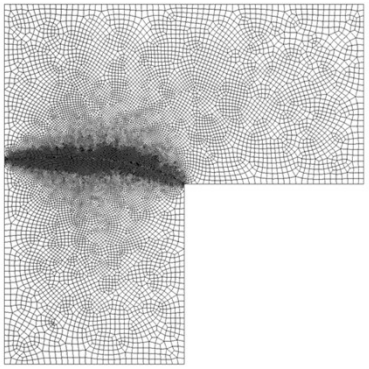
Local mesh refinement
Local mesh refinement + Adaptive update
Adaptive mesh refinement
Adaptive mesh refinement + Adaptive update

Numerical schemes applied	Computation time	
	[sec]	Ratio [%]
Local mesh refinement	14624.72	100.00
Local mesh refinement + **Adaptive update**	8907.93	60.91
Adaptive mesh refinement	5043.26	100.00
Adaptive mesh refinement + **Adaptive update**	2928.41	58.07

Numerical examples

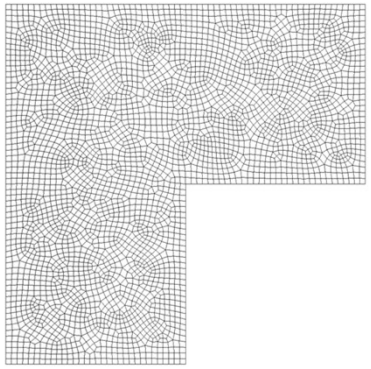
❖ 4. L-shaped panel





19501 elements

For local mesh refinement



3718 elements

For adaptive mesh refinement

$E = 25.85\text{GPa}$

$v = 0.18$

$G_c = 95\text{N/mm}$

$l_0 = 2\text{mm}$

$R_c = 15\text{mm}$

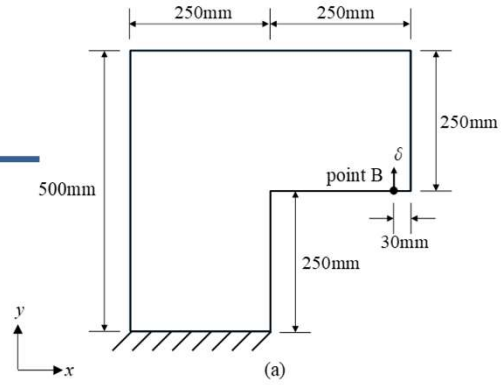
$\eta = 10$

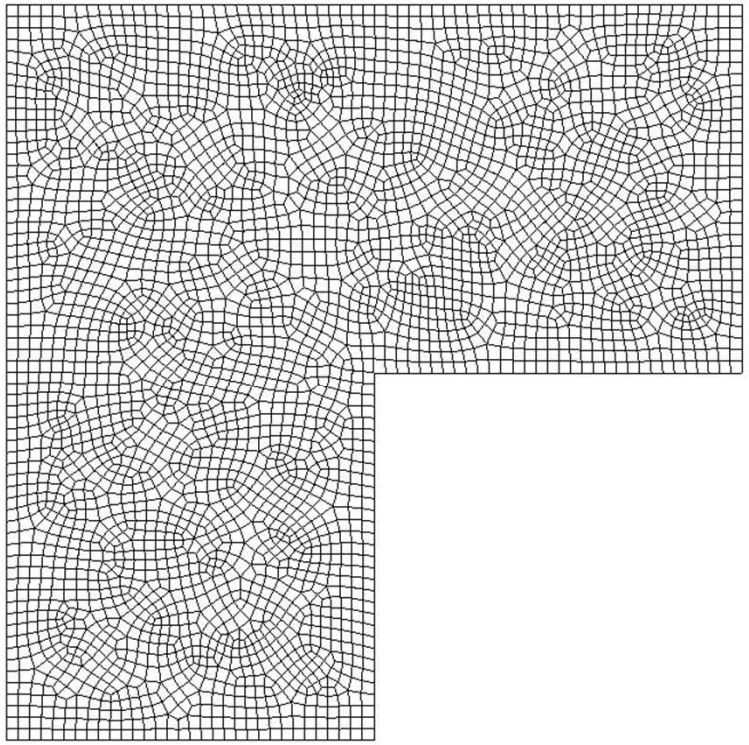
$\phi_c = 0.5$

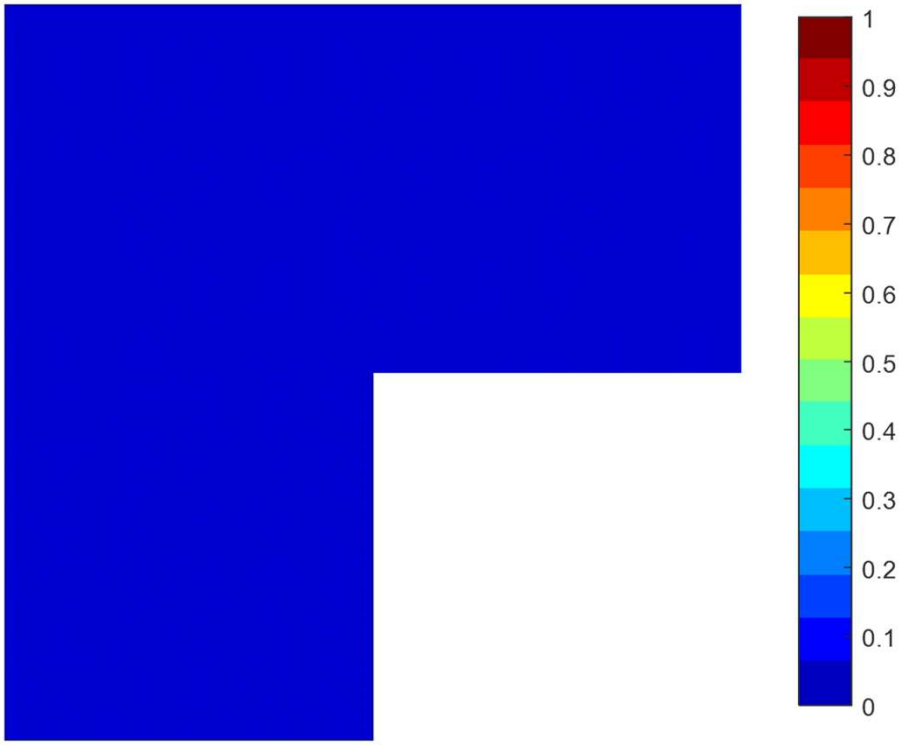
${}^t\mathbf{U} = \beta(t)\,{}^0\mathbf{U}$

Numerical examples

❖ 4. L-shaped panel

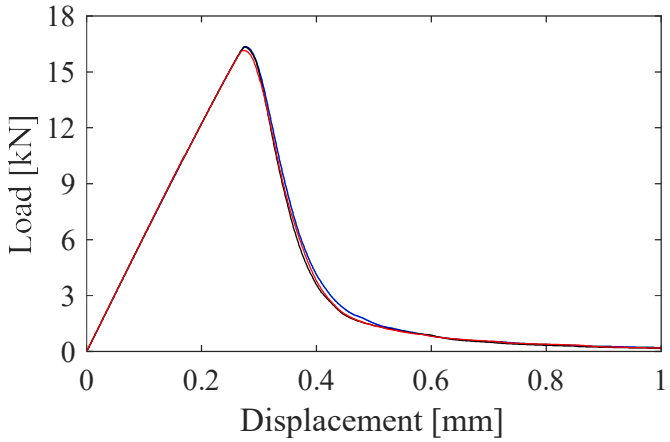


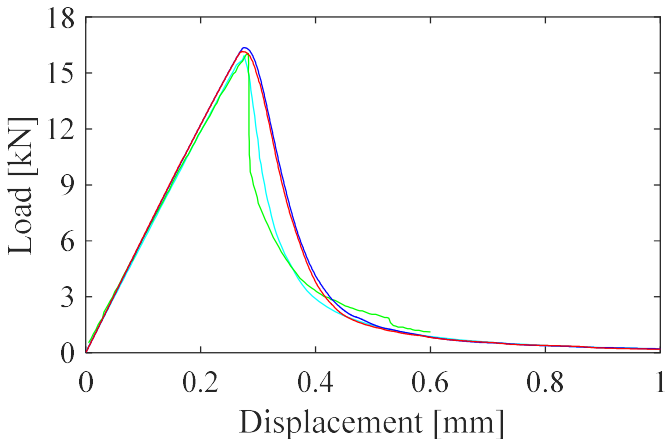




Numerical examples

❖ 4. L-shaped panel

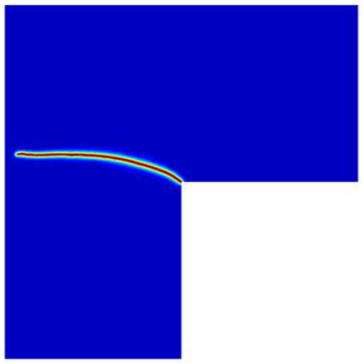




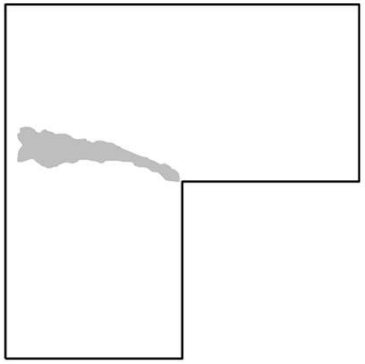
Local mesh refinement
Local mesh refinement + Adaptive update
Adaptive mesh refinement
Adaptive mesh refinement + Adaptive update

Patil et al. [40]
Hirshikesh et al. [42]
Local mesh refinement + Adaptive update
Adaptive mesh refinement + Adaptive update

Numerical schemes applied	Computation time	
	[sec]	Ratio [%]
Local mesh refinement	33347.53	100.00
Local mesh refinement + **Adaptive update**	21775.64	65.30
Adaptive mesh refinement	9719.60	100.00
Adaptive mesh refinement + **Adaptive update**	5733.24	58.99



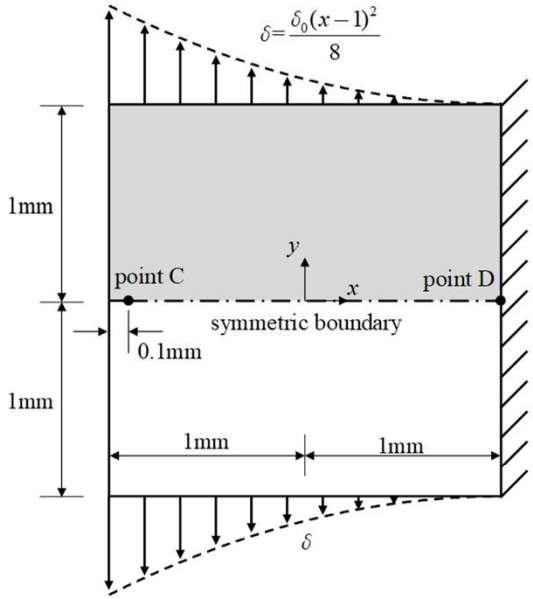
Numerical result

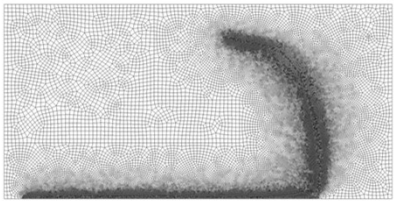


Experimental reuslt by Winkler

Numerical examples

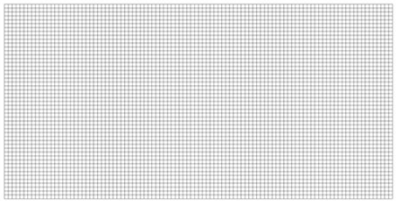
❖ 5. Single-edge notched branching problem





54625 elements

For local mesh refinement



5000 elements

For adaptive mesh refinement

$E = 25.85\text{GPa}$

$v = 0.18$

$G_c = 95\text{N/mm}$

$l_0 = 2\text{mm}$

$R_c = 15\text{mm}$

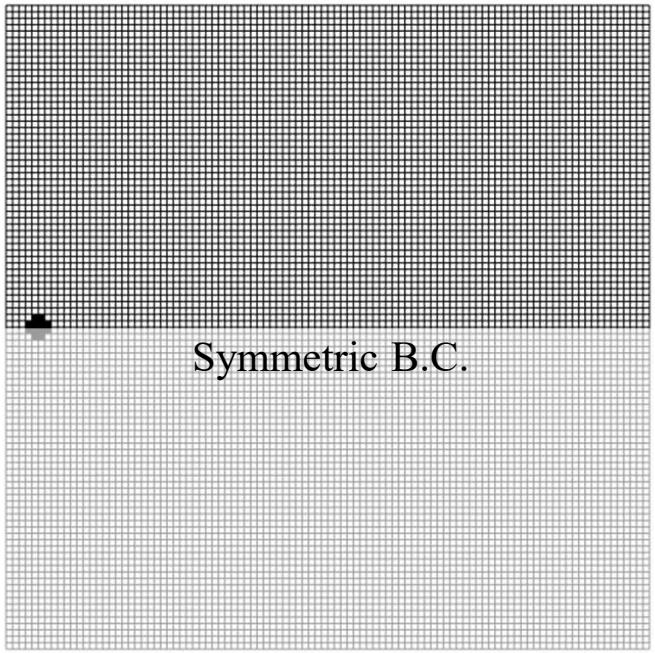
$\eta = 10$

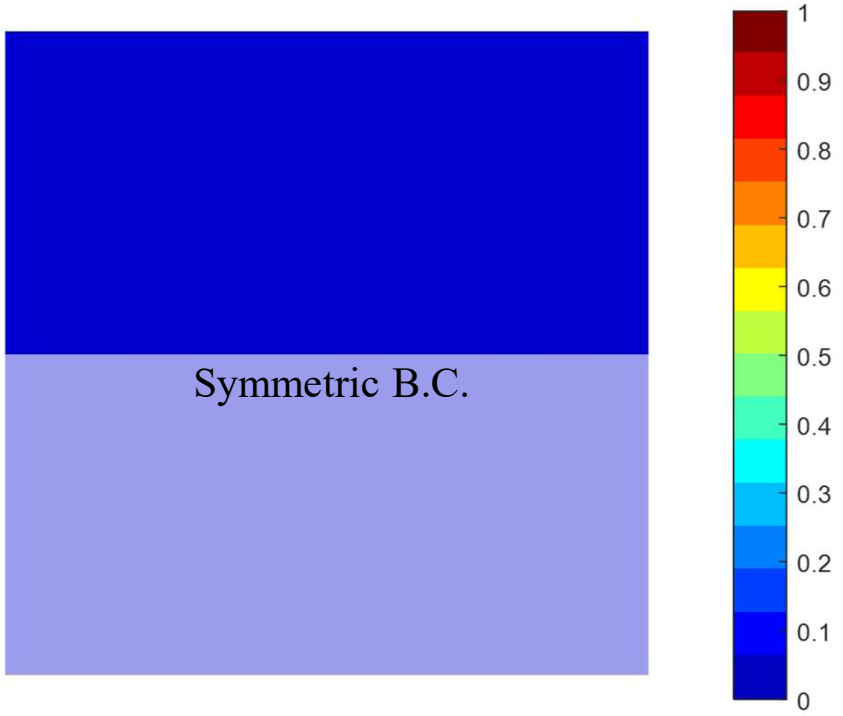
$\phi_c = 0.5$

${}^t\mathbf{U} = \beta(t)\,{}^0\mathbf{U}$

Numerical examples

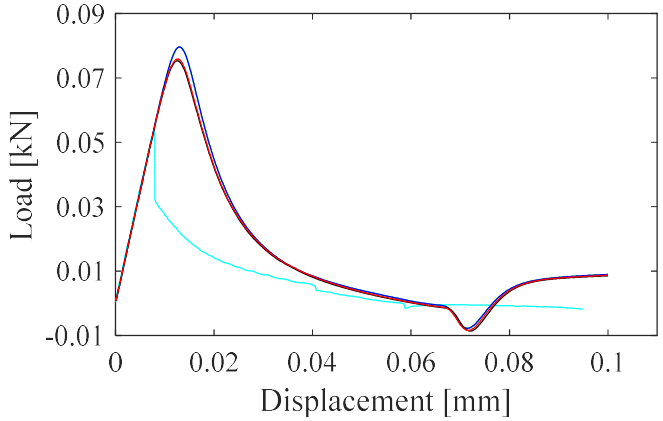
❖ 5. Single-edge notched branching problem

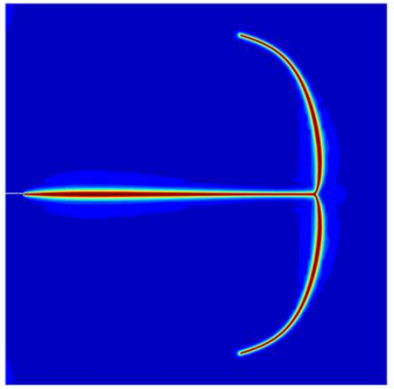




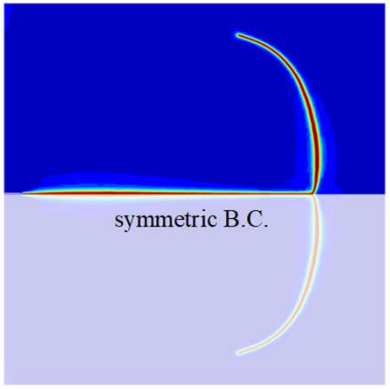
Numerical examples

❖ 5. Single-edge notched branching problem

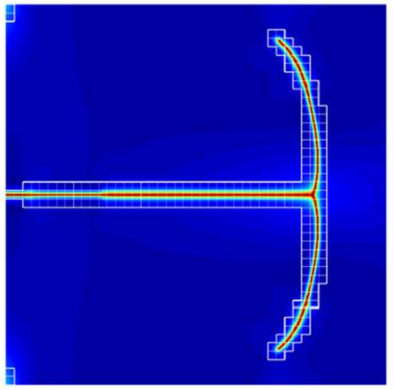




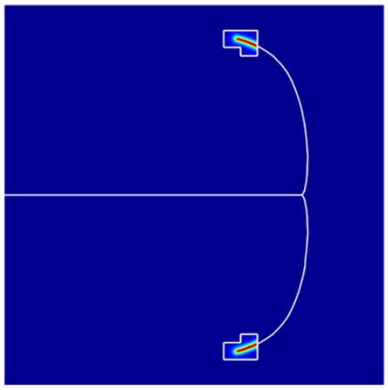
Full model (Proposed)
(a)



Half model (Proposed)
(b)



Full model (Muixí et al., 2020 [45])
(c)



Full model (Muixí et al., 2021 [47])
(d)

Numerical schemes applied	Computation time	
	[sec]	Ratio [%]
Local mesh refinement	31231.21	100.00
Local mesh refinement + **Adaptive update**	21358.76	68.39
Adaptive mesh refinement	10910.54	100.00
Adaptive mesh refinement + **Adaptive update**	6454.63	59.16

Closure (Topic 1)

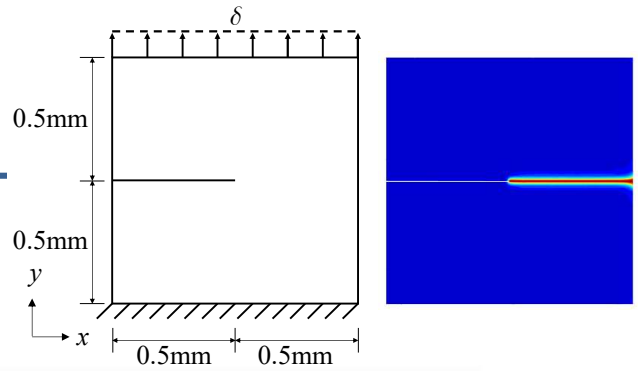
- **Adaptive update scheme** was proposed.
- The performance was investigated through several numerical examples.
- The proposed adaptive update scheme reduced computation time

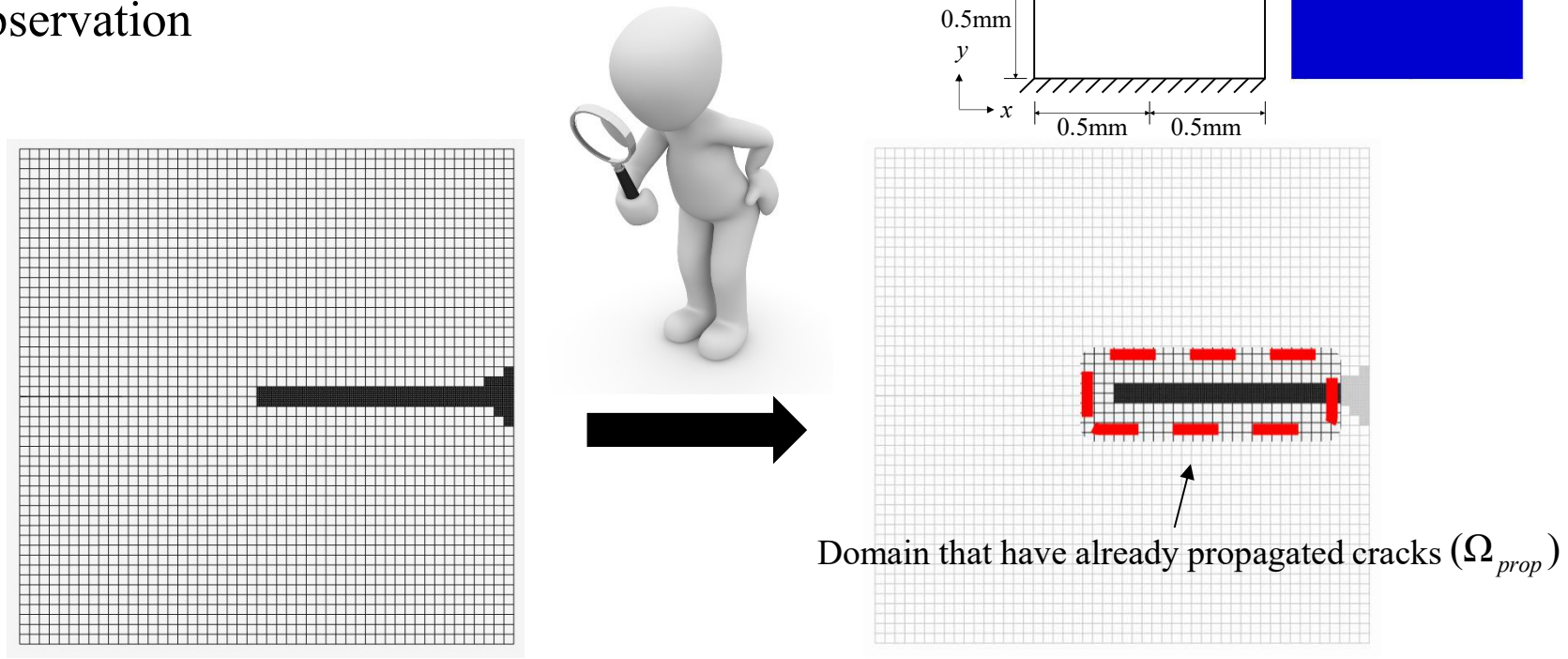
  by **40% (Local mesh refinement)** to **50% (Adaptive mesh refinement)**.

Topic 2. Adaptive mesh coarsening using the phantom-node method

Motivation

❖ Observation

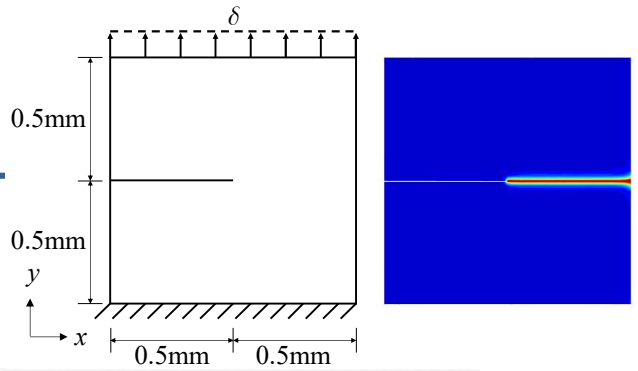




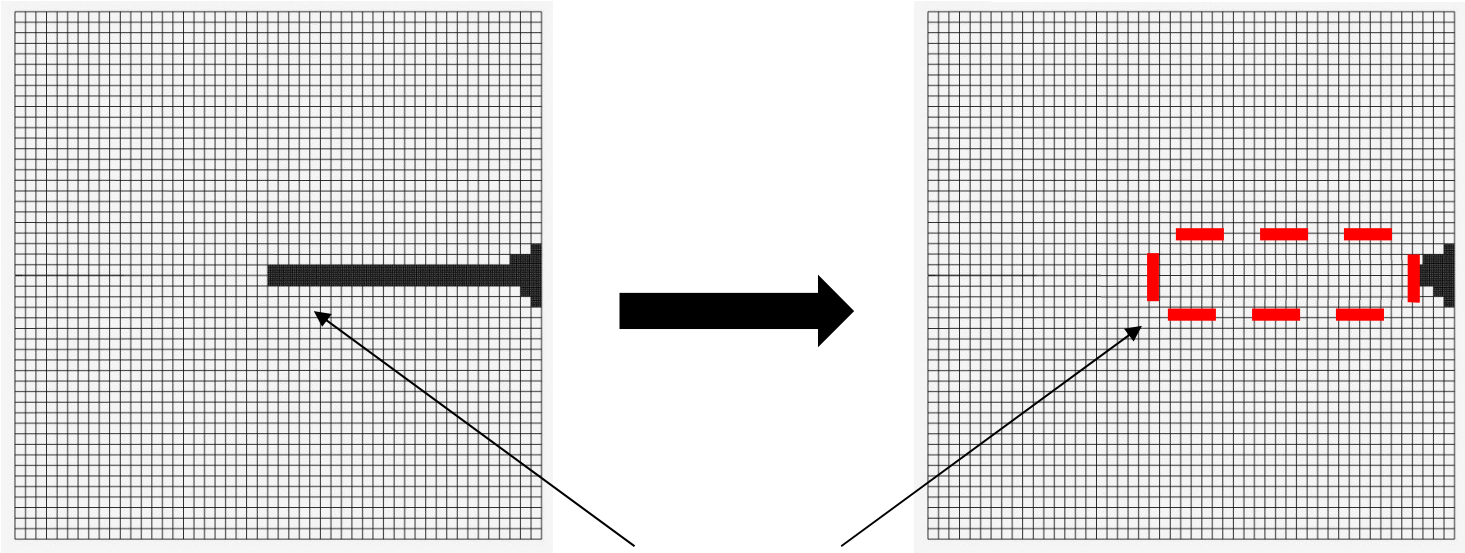
- **Fine mesh** remains in Ω_{prop} after crack propagation.
- During propagation, structural properties **rarely change** in Ω_{prop}.

Fine mesh is not needed in the domain that have already propagated cracks!

Key idea



❖ Adaptive mesh coarsening



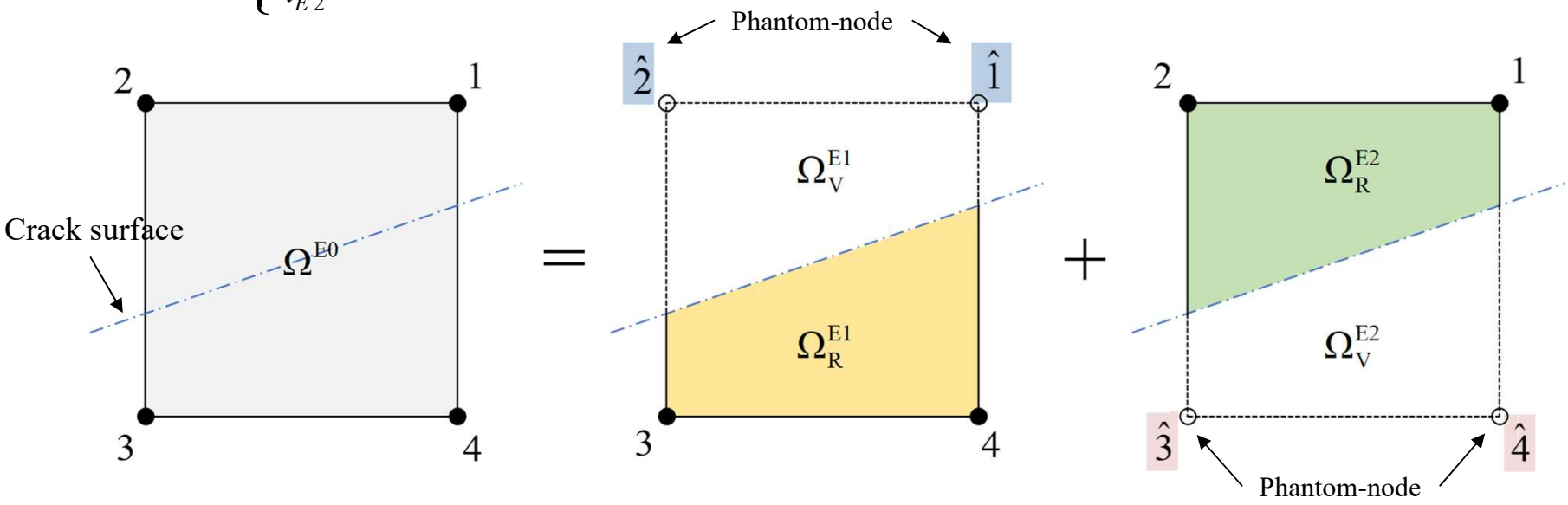
Domain that have already propagated cracks

- Converting **fine mesh** into **coarse mesh** using **the phantom-node method**
- Improving the computational efficiency due to reducing DOFs

Phantom-node method

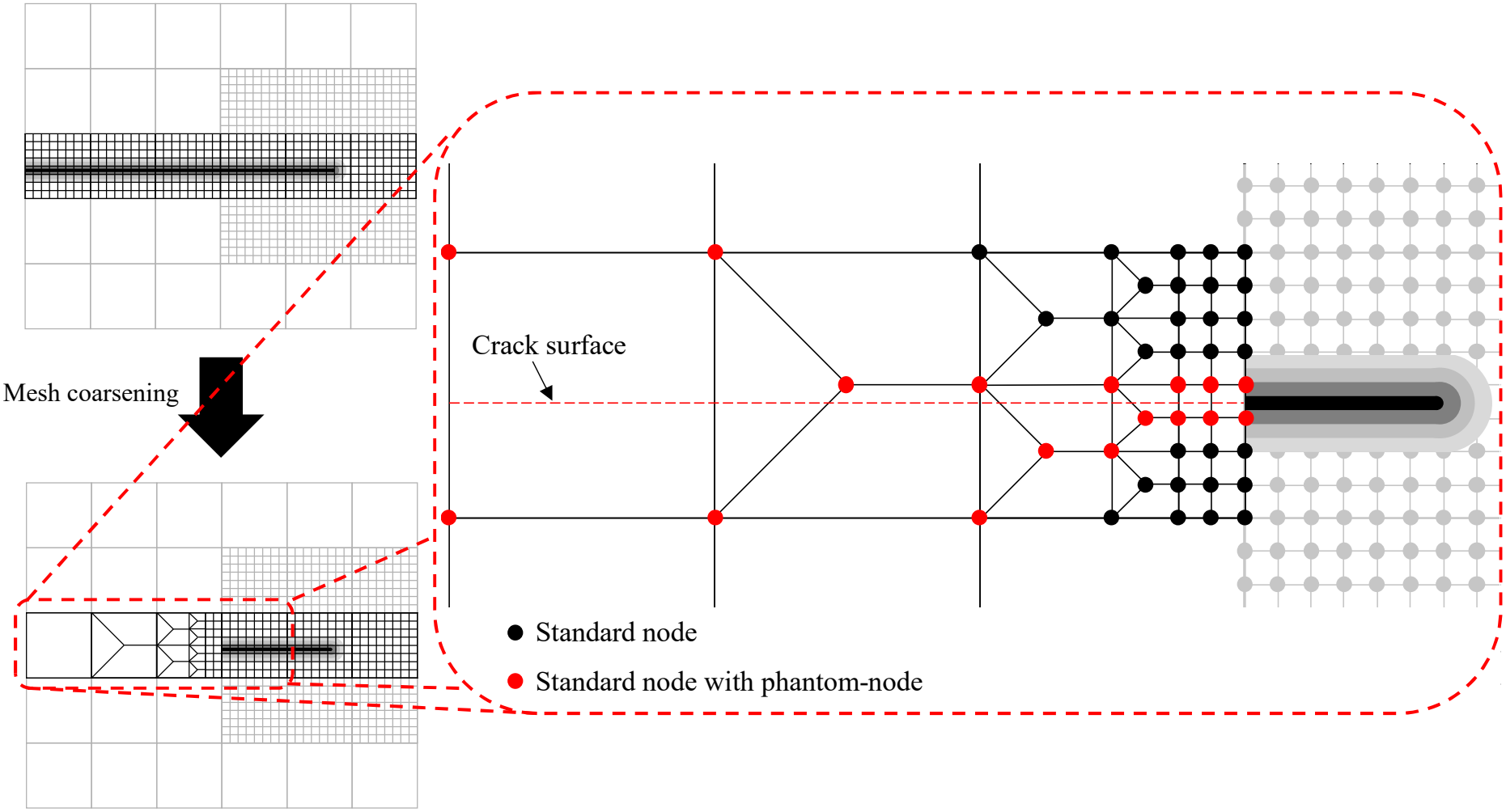
❖ Formulation

$$\mathbf{u}^{(m)} = \begin{cases} \sum_{i_{E1}}^{4} h_i \begin{bmatrix} u_i & v_i \end{bmatrix}^{\mathrm{T}} & \text{in} \quad \Omega_{\mathrm{R}}^{\mathrm{E1}} \quad \text{where} \quad i_{\mathrm{E1}} \in \{\hat{1}, \hat{2}, 3, 4\} \\ \sum_{i_{E2}}^{4} h_i \begin{bmatrix} u_i & v_i \end{bmatrix}^{\mathrm{T}} & \text{in} \quad \Omega_{\mathrm{R}}^{\mathrm{E2}} \quad \text{where} \quad i_{\mathrm{E2}} \in \{1, 2, \hat{3}, \hat{4}\} \end{cases}$$



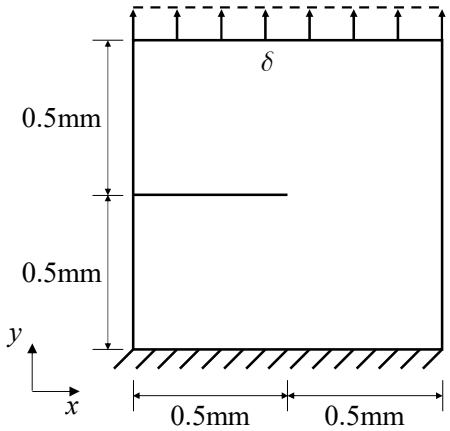
Phantom-node method

❖ Adaptive mesh coarsening

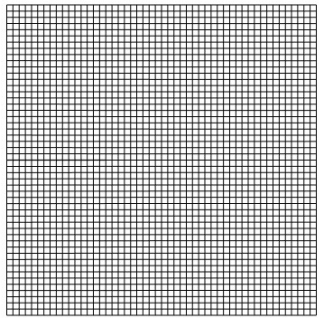


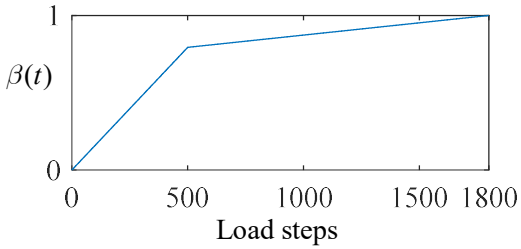
Numerical examples

❖ 1. Single-edge notched tension problem



(a)


(b)



(c)

$E = 210\text{GPa}$

$v = 0.3$

$G_c = 2.7 \times 10^{-3}\text{kN/mm}$

$l_0 = 0.0075\text{mm}$

$R_c = 0.04\text{mm}$

$\eta = 10$

$\phi_c = 0.5$

${}^t\mathbf{U} = \beta(t)\,{}^0\mathbf{U}$

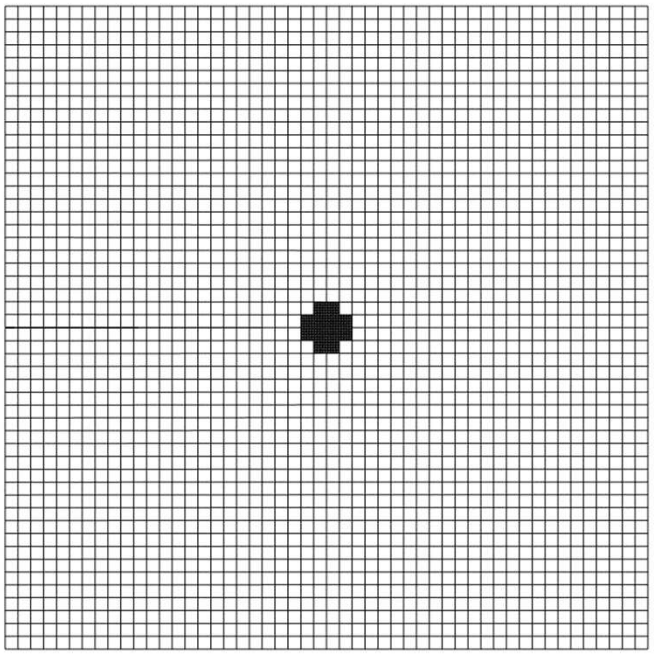
2500 elements

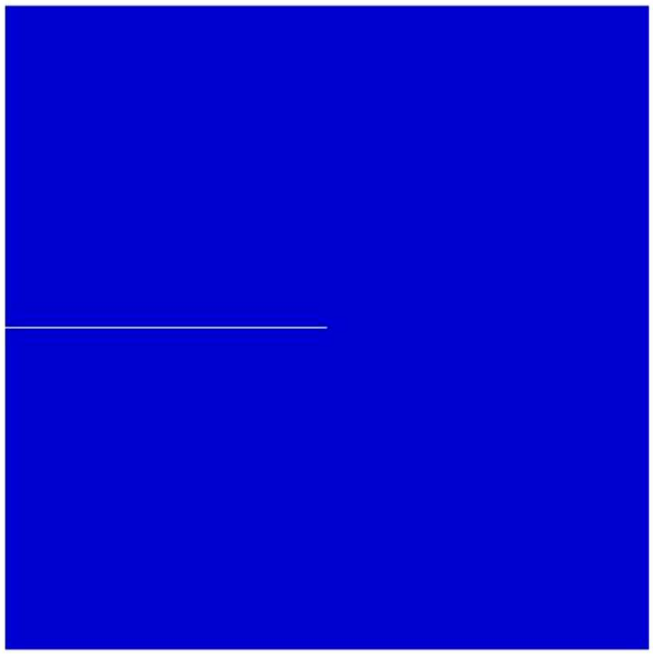
Reference case : Adaptive mesh refinement + Adaptive update

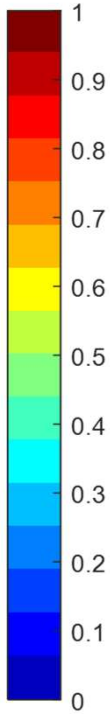
Proposed case : Adaptive mesh refinement + Adaptive update + **Phantom-node method**

Numerical examples

❖ 1. Single-edge notched tension problem

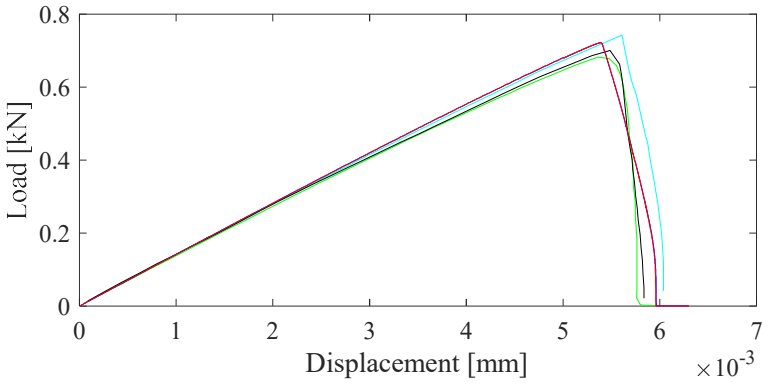


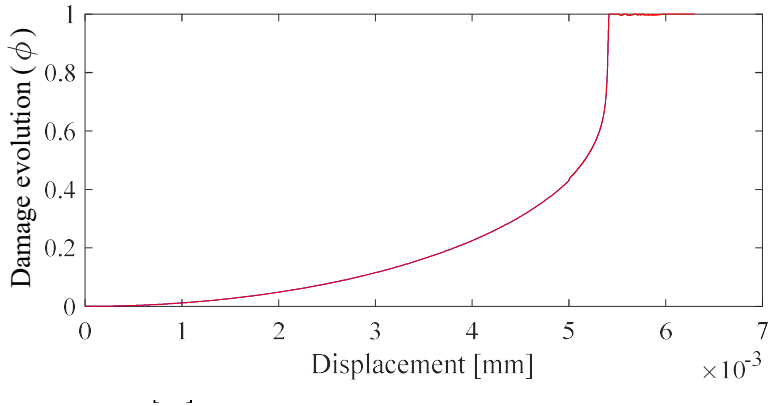




Numerical examples

❖ 1. Single-edge notched tension problem

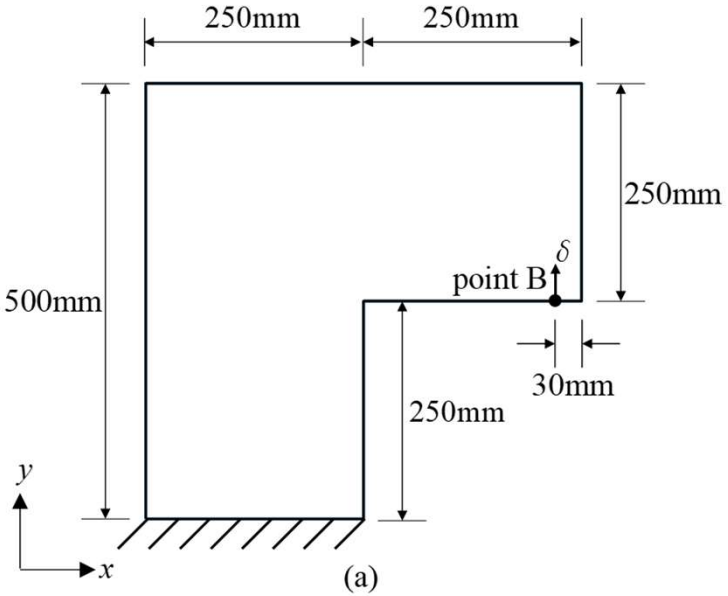




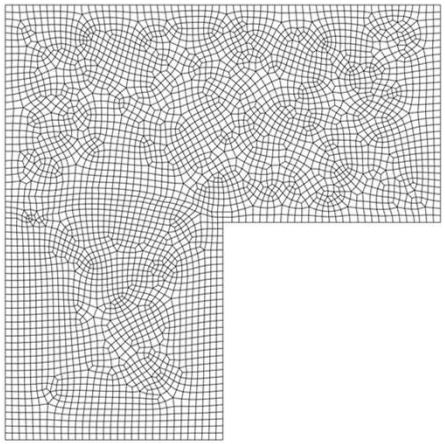
Numerical schemes applied	Normalized computation time			
	Whole time step		After crack propagation	
	Total	Equation solving	Total	Equation solving
Adaptive mesh refinement Adaptive update	100%	100%	100%	100%
Adaptive mesh refinement Adaptive update **Phantom-node method**	87.73%	74.90%	77.76%	63.01%

Numerical examples

❖ 2. L-shaped panel



(a)



(b)

3718 elements

$E = 25.85\text{GPa}$

$v = 0.18$

$G_c = 95\text{N/mm}$

$l_0 = 2\text{mm}$

$R_c = 15\text{mm}$

$\eta = 10$

$\phi_c = 0.5$

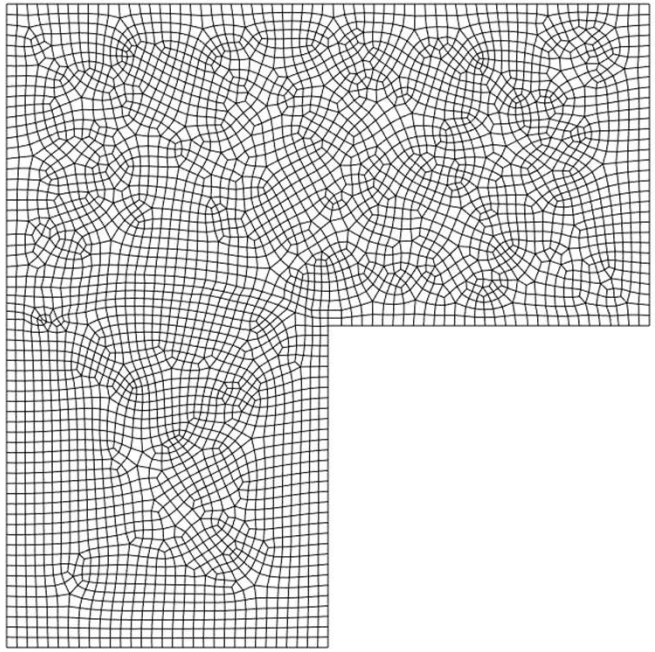
$^t\mathbf{U} = \beta(t)\,{}^0\mathbf{U}$

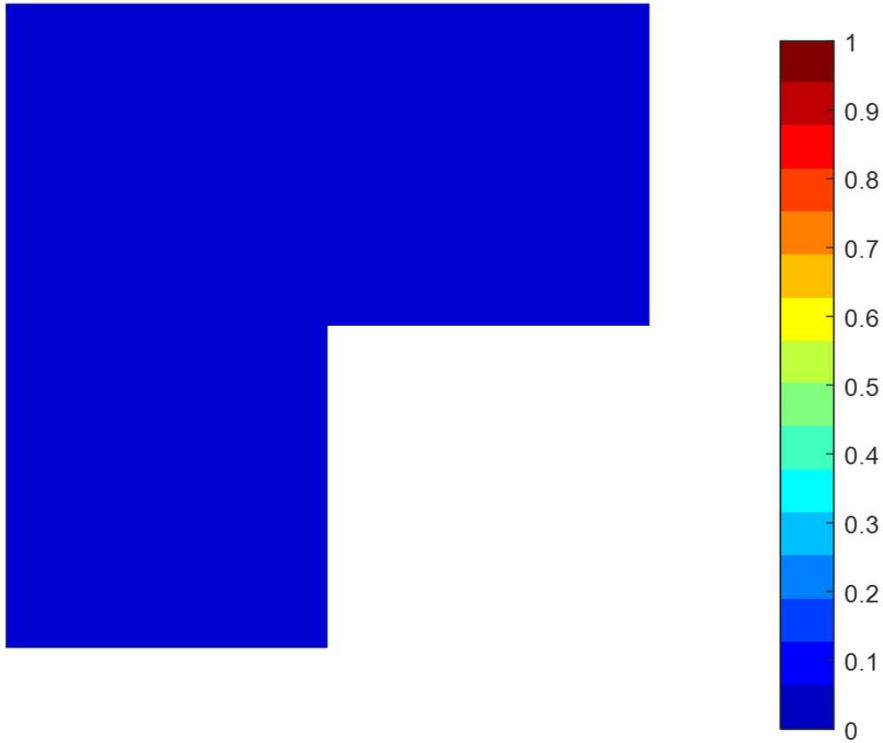
Reference case : Adaptive mesh refinement + Adaptive update

Proposed case : Adaptive mesh refinement + Adaptive update + **Phantom-node method**

Numerical examples

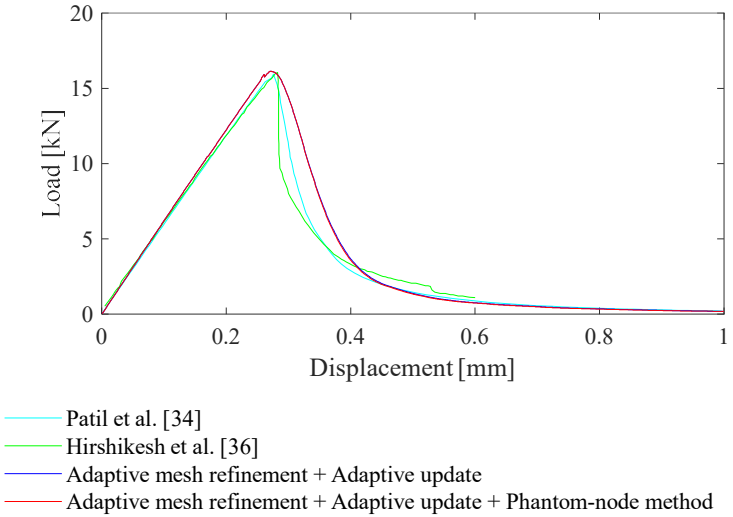
❖ 2. L-shaped panel

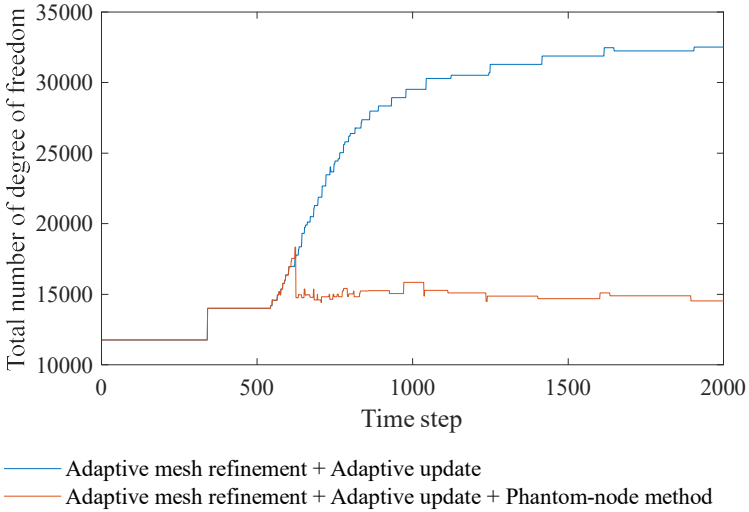




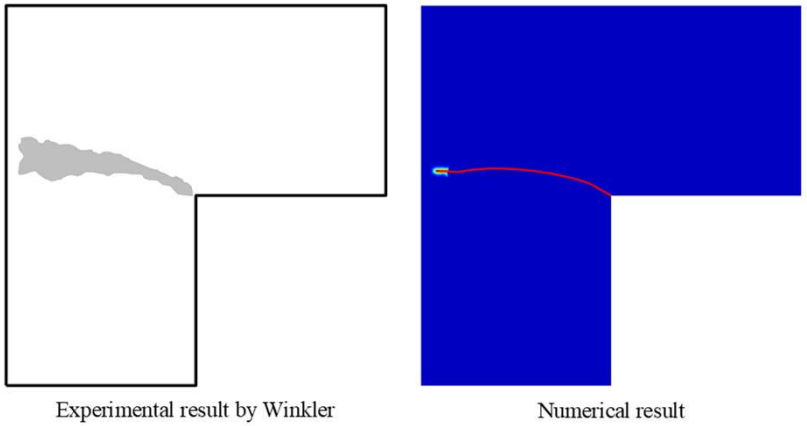
Numerical examples

❖ 2. L-shaped panel



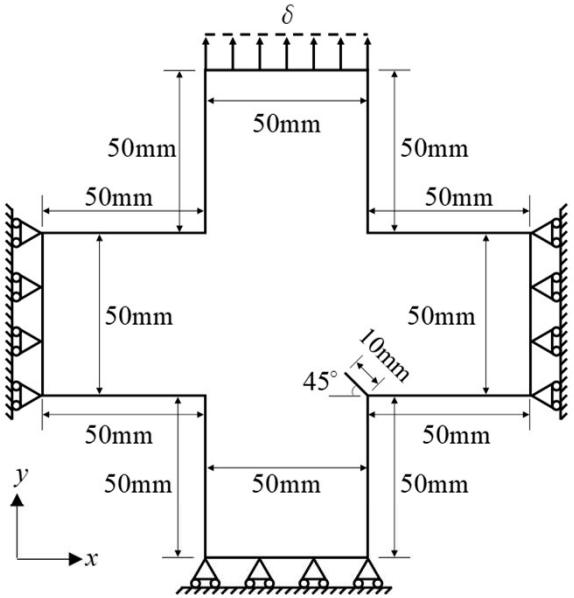


Numerical schemes applied	Normalized computation time			
	Whole time step		After crack propagation	
	Total	Equation solving	Total	Equation solving
Adaptive mesh refinement Adaptive update	100 %	100 %	100 %	100 %
Adaptive mesh refinement Adaptive update **Phantom-node method**	66.36 %	49.01 %	59.15 %	42.85 %

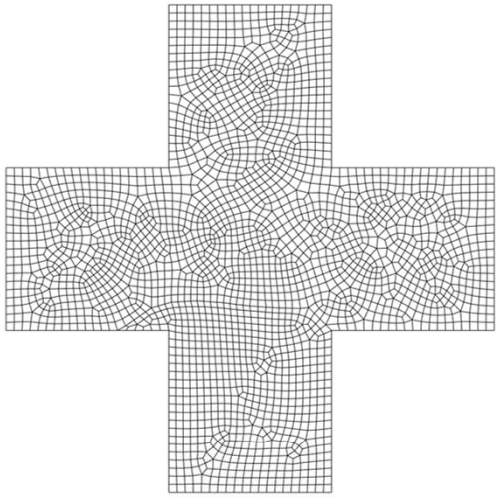


Numerical examples

❖ 3. Notched cruciform plate

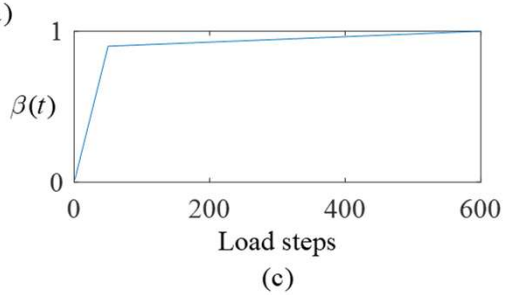


(a)



(b)

2527 elements



(c)

$E = 210\text{GPa}$

$v = 0.3$

$G_c = 2.7 \times 10^{-3}\text{kN/mm}$

$l_0 = 0.0075\text{mm}$

$R_c = 0.04\text{mm}$

$\eta = 10$

$\phi_c = 0.5$

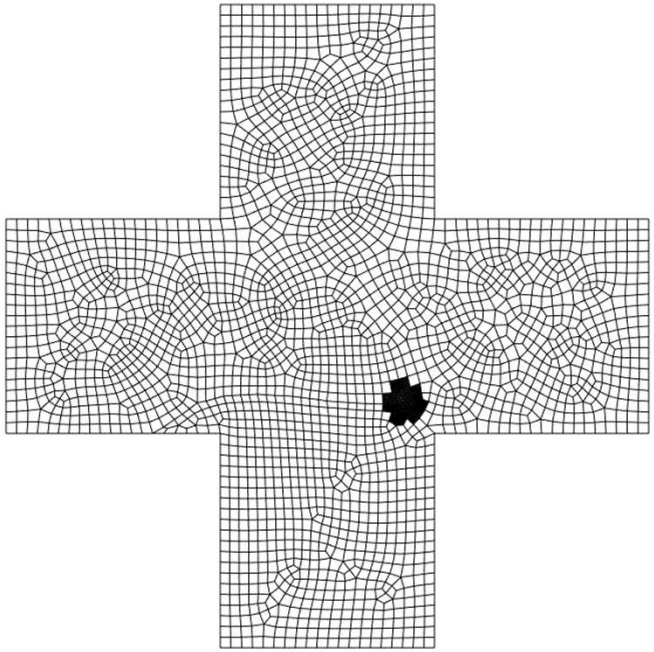
${}^t\mathbf{U} = \beta(t)\,{}^0\mathbf{U}$

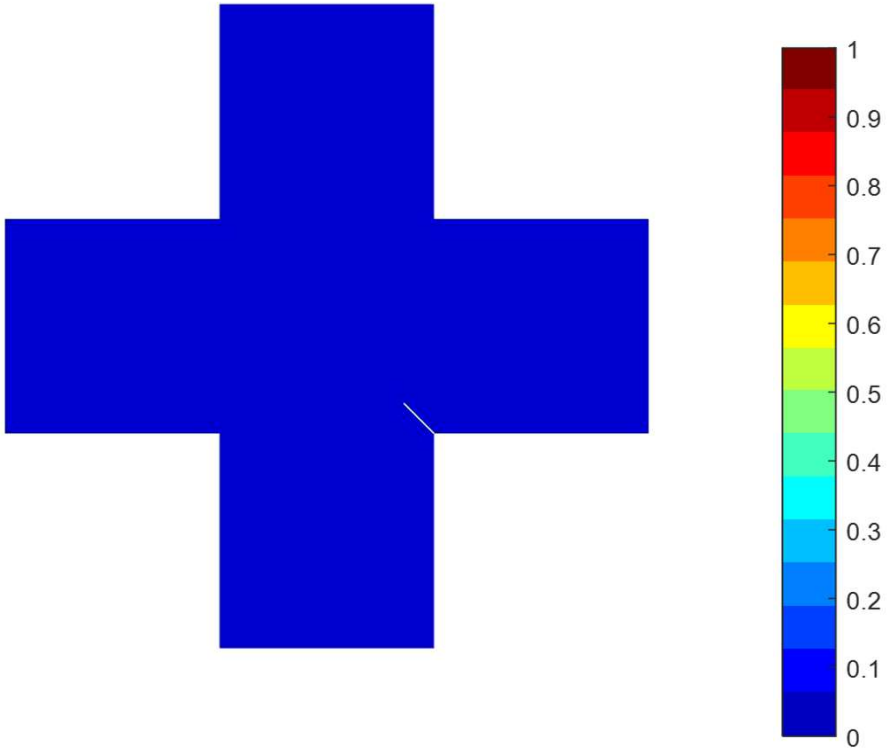
Reference case : Adaptive mesh refinement + Adaptive update

Proposed case : Adaptive mesh refinement + Adaptive update + **Phantom-node method**

Numerical examples

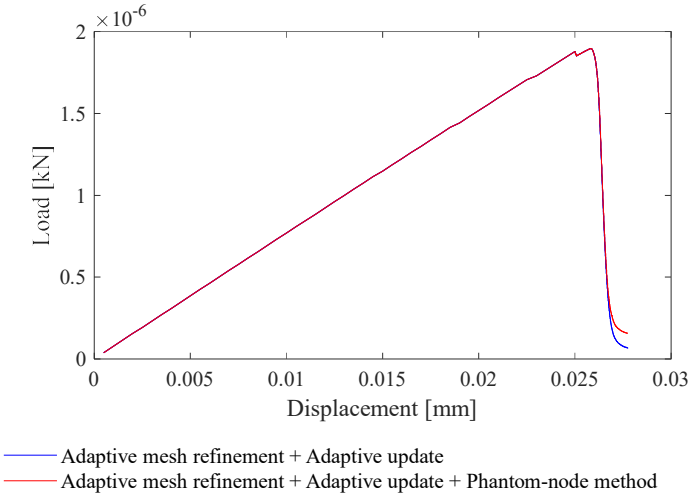
❖ 3. Notched cruciform plate

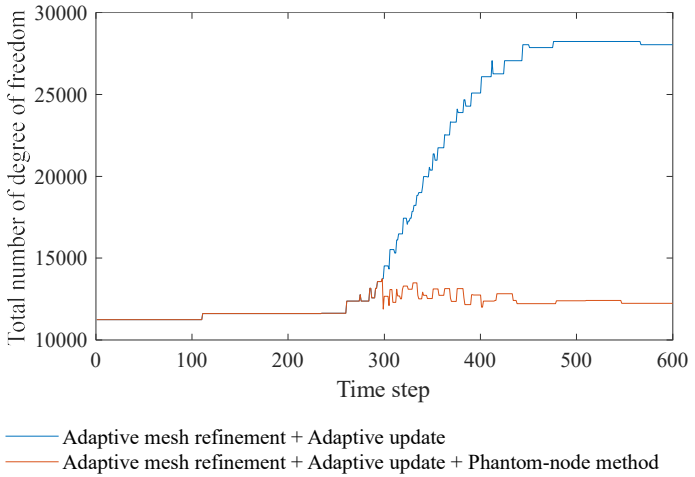




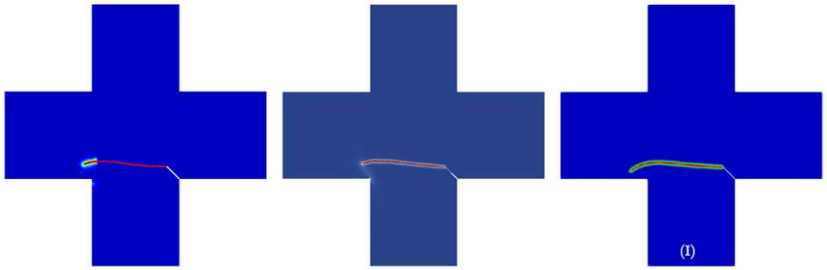
Numerical examples

❖ 3. Notched cruciform plate



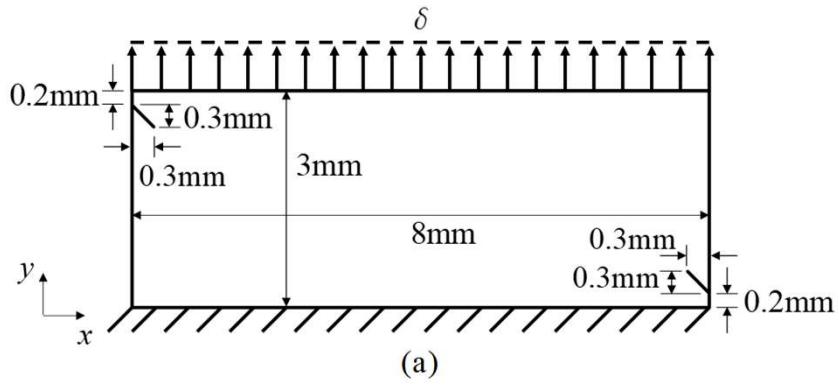


Numerical schemes applied	Normalized computation time			
	Whole time step		After crack propagation	
	Total	Equation solving	Total	Equation solving
Adaptive mesh refinement Adaptive update	100 %	100 %	100 %	100 %
Adaptive mesh refinement Adaptive update **Phantom-node method**	78.86 %	59.28 %	72.30 %	49.89 %

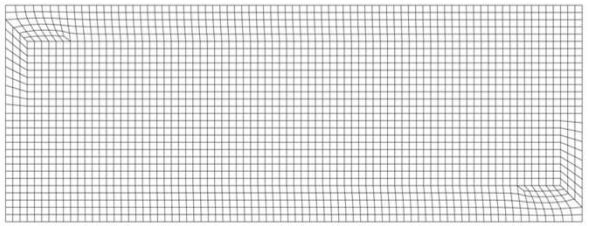


Numerical examples

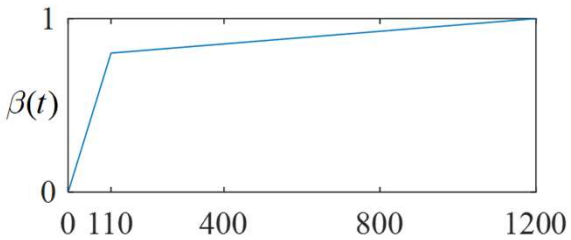
❖ 4. Double notched tension problem



(a)



(b)
2392 elements



(c)

$E = 210\text{GPa}$

$v = 0.3$

$G_c = 2.7 \times 10^{-3}\text{kN/mm}$

$l_0 = 0.0075\text{mm}$

$R_c = 0.04\text{mm}$

$\eta = 10$

$\phi_c = 0.5$

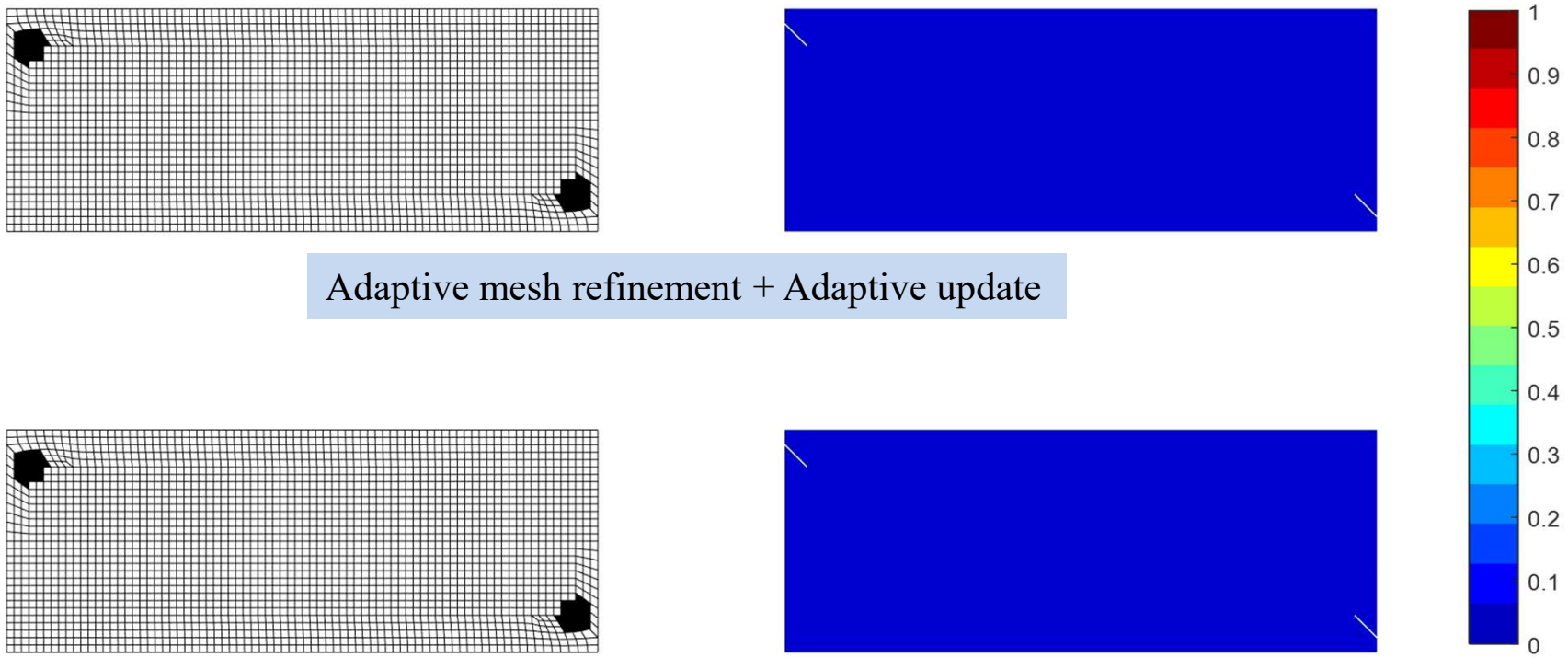
$^{t}\mathbf{U} = \beta(t)\,^{0}\mathbf{U}$

Reference case : Adaptive mesh refinement + Adaptive update

Proposed case : Adaptive mesh refinement + Adaptive update + **Phantom-node method**

Numerical examples

❖ 4. Double notched tension problem

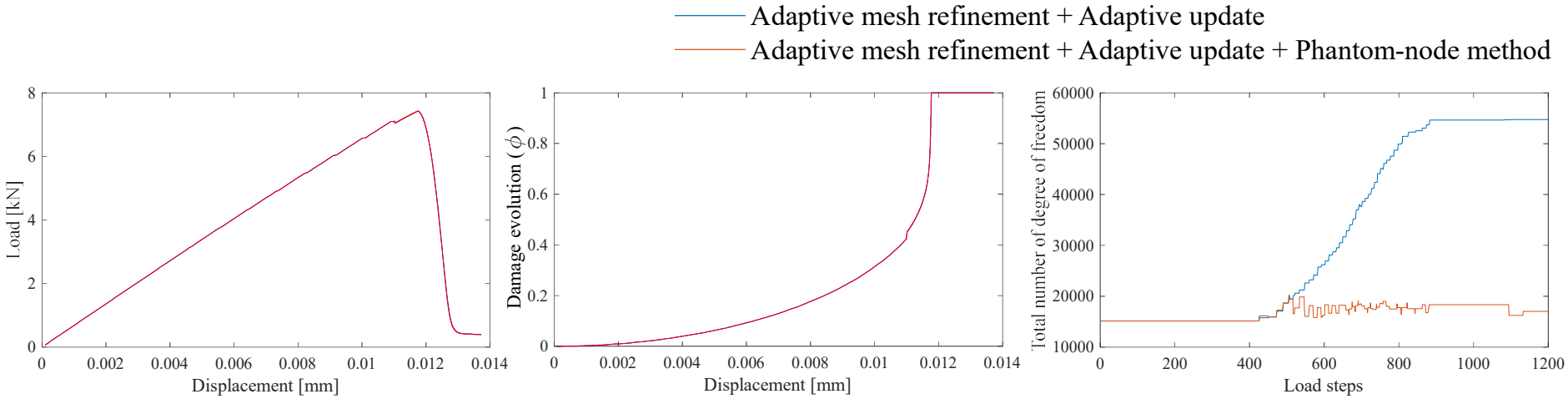


Adaptive mesh refinement + Adaptive update

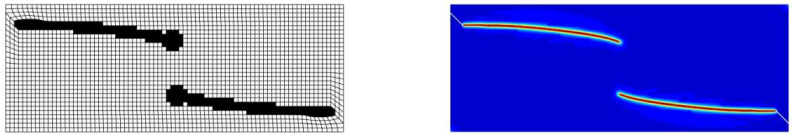
Adaptive mesh refinement + Adaptive update + **Phantom-node method**

Numerical examples

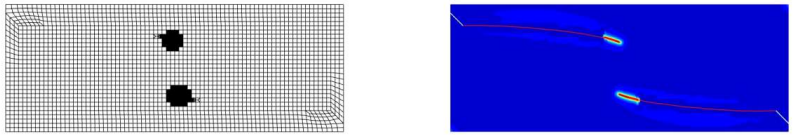
❖ 4. Double notched tension problem



Numerical schemes applied	Normalized computation time			
	Whole time step		After crack propagation	
	Total	Equation solving	Total	Equation solving
Adaptive mesh refinement Adaptive update	100 %	100 %	100 %	100 %
Adaptive mesh refinement Adaptive update **Phantom-node method**	69.26 %	47.48 %	62.44 %	39.03 %



Adaptive mesh refinement + Adaptive update



Adaptive mesh refinement + Adaptive update + Phantom-node method

Closure (Topic 2)

- **Adaptive mesh coarsening** was proposed.
- **The phantom-node method** was applied to the phase field model.
- **Fine mesh** in the domain that have already propagated cracks was **converted** into **coarse mesh**.
- Through various numerical examples, it is shown that adaptive mesh coarsening **reduced computation time** by **60~70%** .

4. Conclusions & Future works

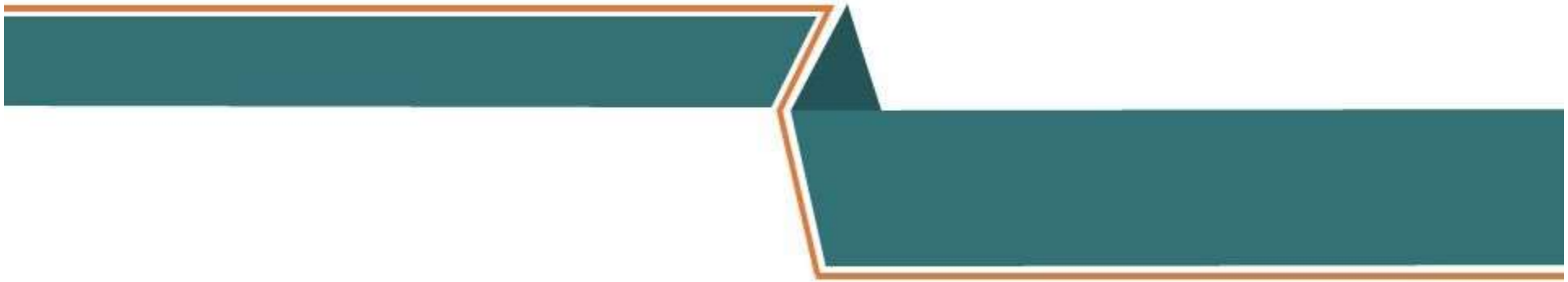
Conclusions

- **Efficient methods** for the phase field model were developed.
- First, **the adaptive update scheme** was proposed.
- Adaptive update scheme largely **reduced computation times** for local and adaptive mesh refinements by **40~50%**.
- Second, **adaptive mesh coarsening** using the phantom-node method was also proposed.
- As **fine mesh** was converted into **coarse mesh**, adaptive mesh coarsening **reduced computation times by 60~70%** .

Future works

- 3D or shell fracture
  - It is hard to extend to 3D / shell fracture.
- Material modelling
  - Extension to anisotropic, composite and ductile material
- Dynamic fracture
  - Dynamic analysis
- Thermo-mechanical fracture

“Thank You”



Question and Answer