Cut Finite Element Methods for Contact Problems

Susanne Claus

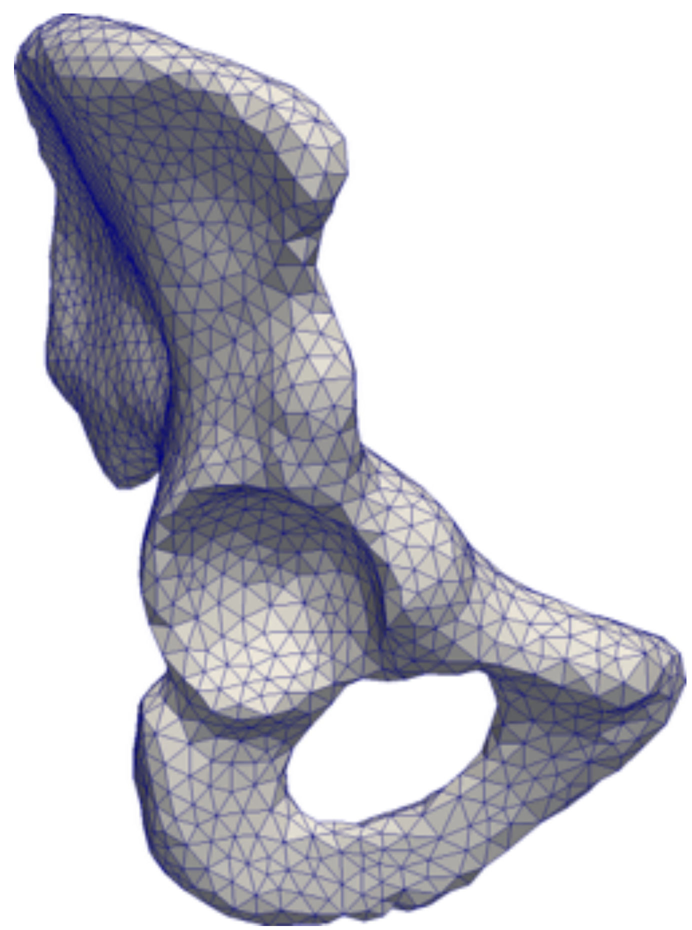
Department of Computer Science, University of Copenhagen, Denmark.

AI Seminar Series, Copenhagen, Jan 2019

THE ALFRED BENZON FOUNDATION

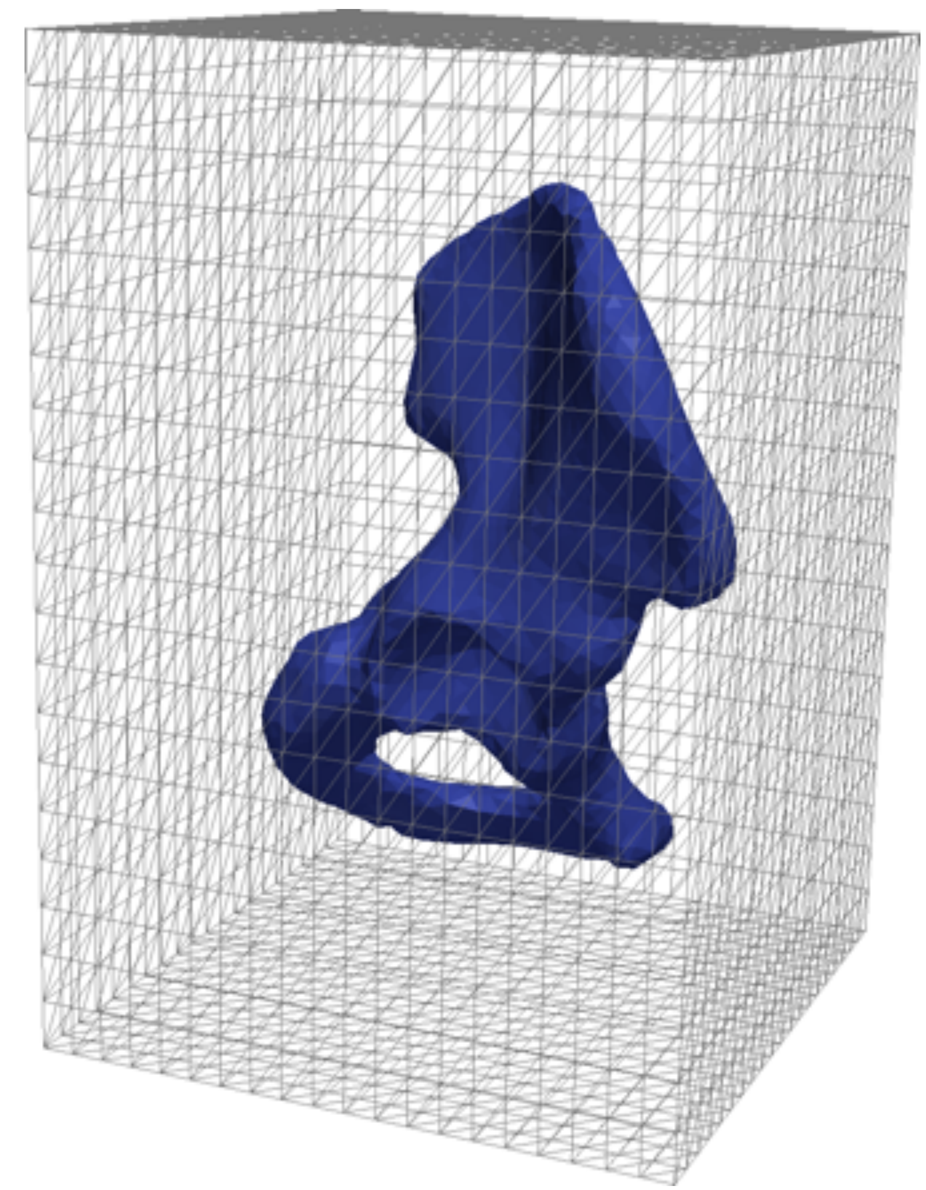
Geometry Discretisation in Finite Element Methods

Classical FEM



Geometry is meshed

CutFEM



Geometry is embedded in fixed background grid and described by a function (e.g. level set function)

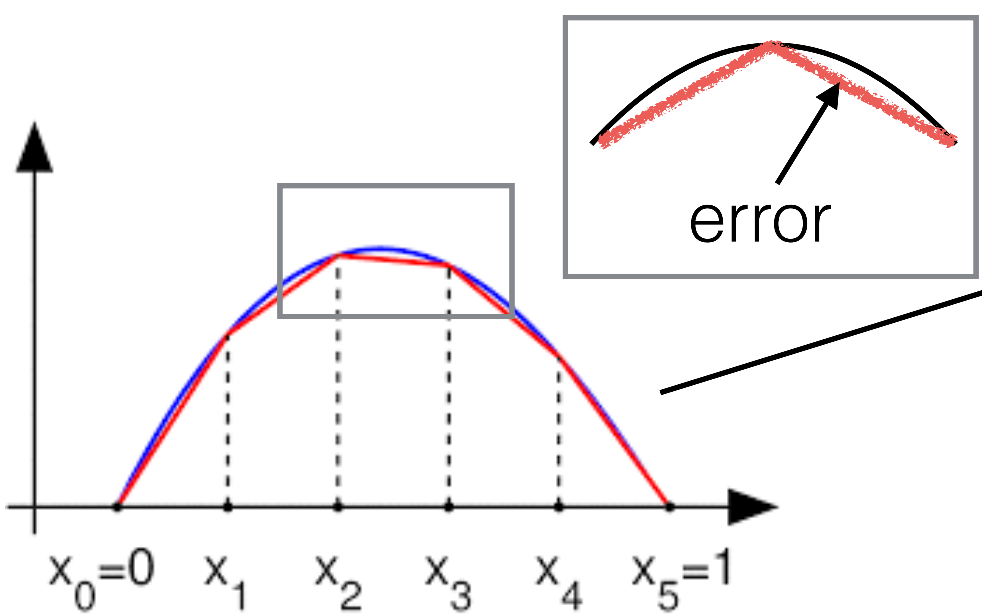
Finite Element Methods

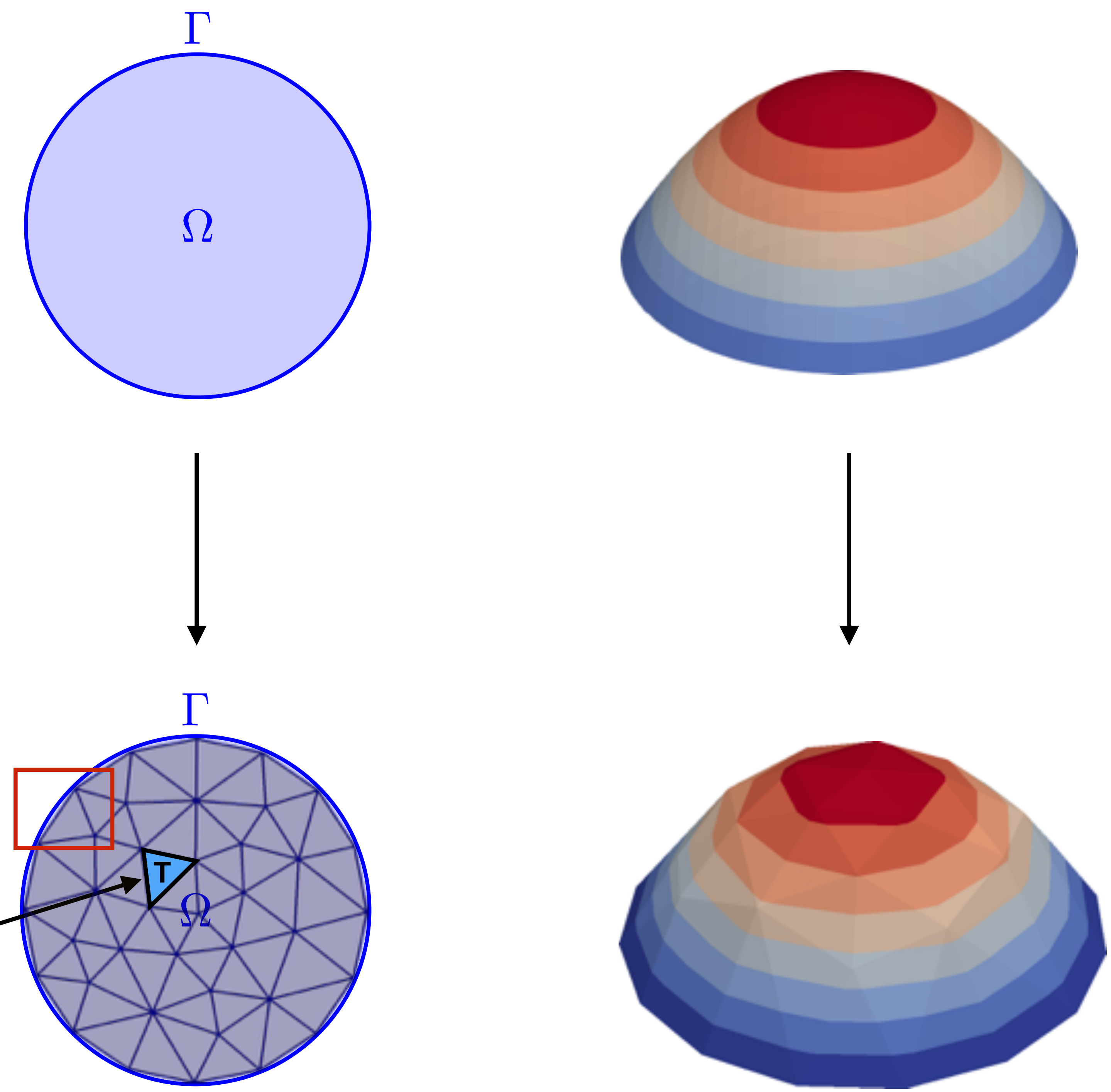
Consider the following diffusion partial differential equation (PDE)

$$-\Delta u = f \text{ in } \Omega$$
$$u = 0 \text{ on } \Gamma$$

Find $u \in V$ such that

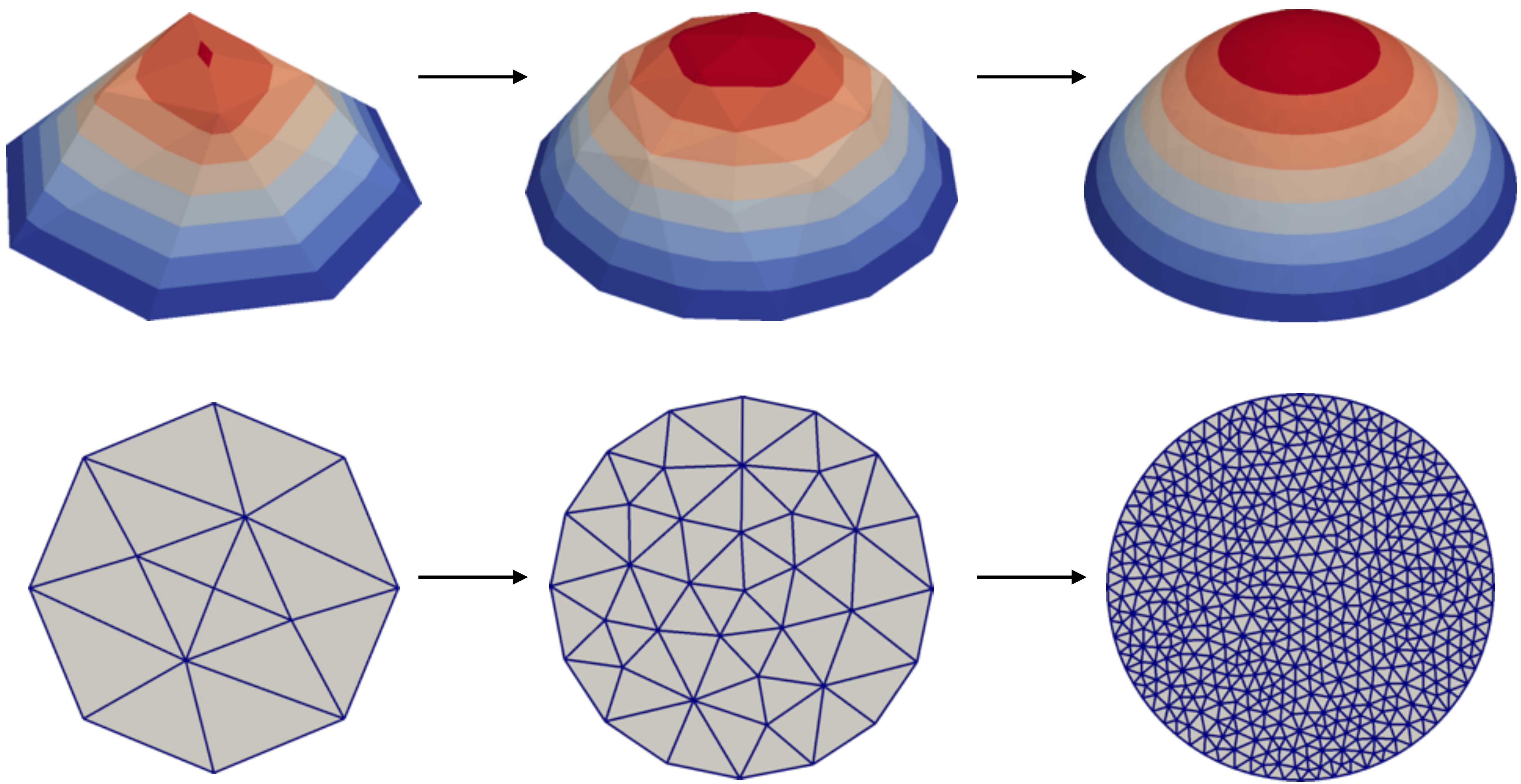
$$\int_\Omega \nabla u \cdot \nabla v dx = \int_\Omega f v dx \quad \forall v \in V$$





Convergence with mesh refinement

The error decreases with mesh refinement. However, how fast the error decreases with mesh refinement (convergence order) depends on multiple factors.



Convergence strongly depends on

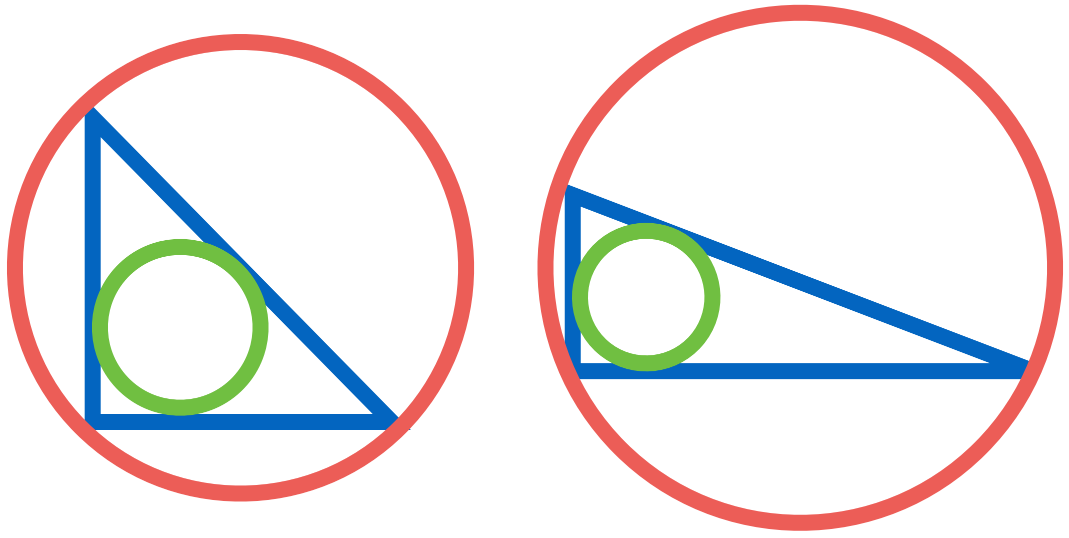
Accuracy

- **Numerical error**: error from piecewise polynomial approximation of the solution and of the geometry.
- **Mesh Quality:** The quality and size of the mesh has a significant impact on the accuracy of the solution

Stability

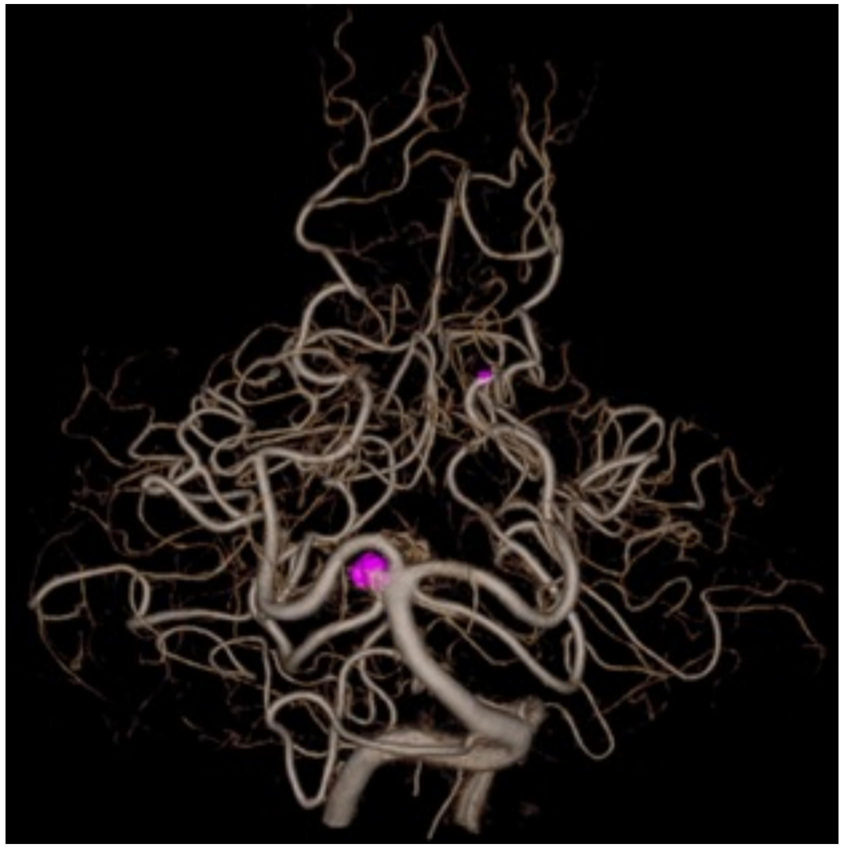
- **Instabilities** frequently occur in **simulations as numerical errors can grow in the solution process. Numerical error growth needs to be controlled and stabilised** carefully. Too much stabilisation leads to inaccuracies.

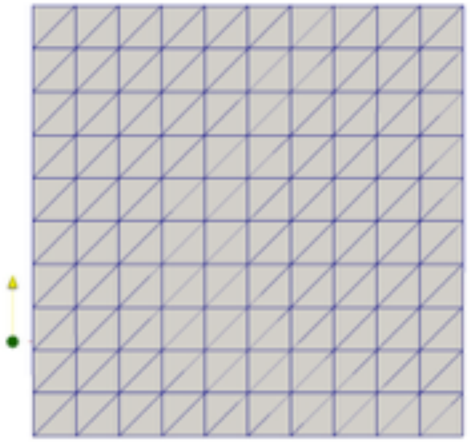
Good Element Bad Element

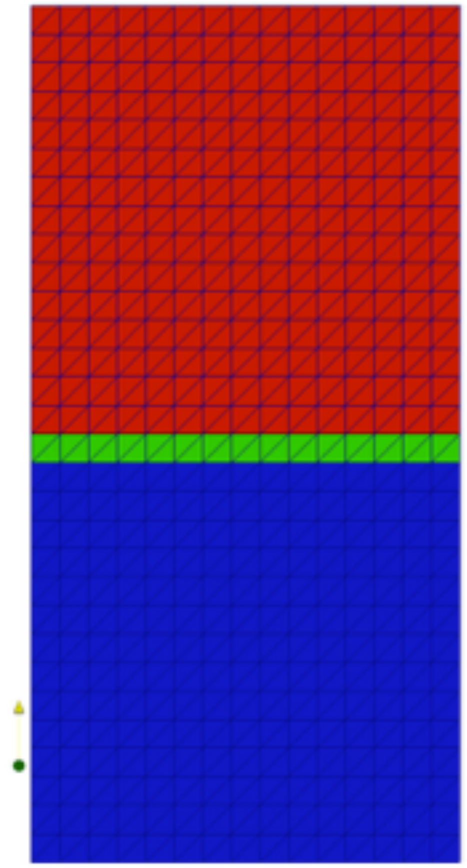


Convergence = Accuracy + Stability

Difficulty of maintaining a high quality mesh







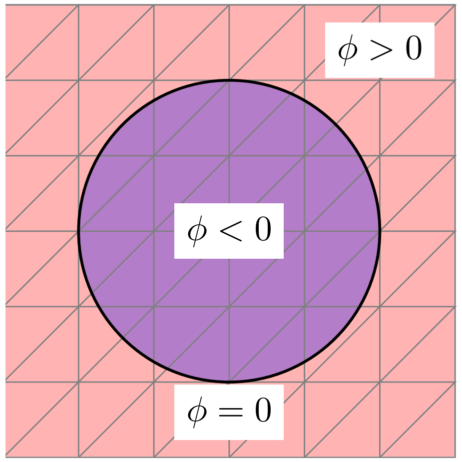
Advantages of Mesh Independent Geometry Descriptions

1. reduces the computational cost for preprocessing or transformation of acquired geometry descriptions
2. efficient and robust for problems involving evolving geometries undergoing large deformations

Important aspects of implicit geometry/cut finite element methods

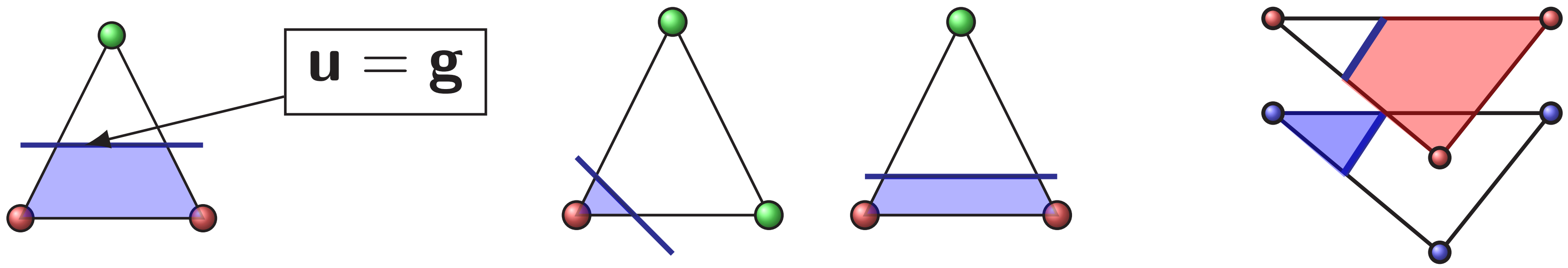
Geometry Algorithms

Discretisation of the geometry based on implicit interface description



Accuracy and Stabilisation

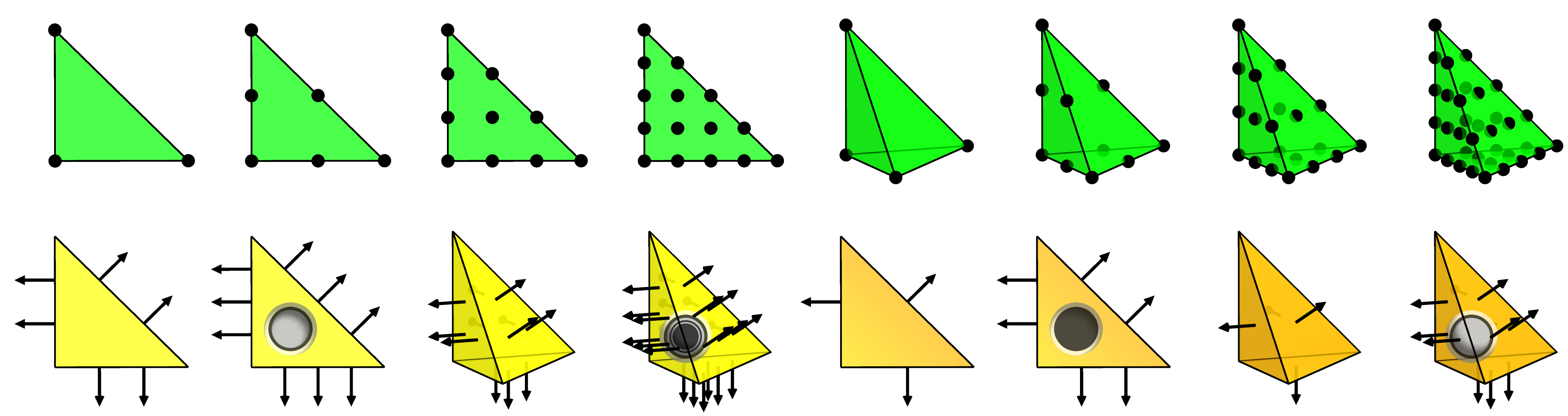
Construction of stable and accurate finite element methods independent of how the interface intersects the mesh.



FEniCS

Open Source Finite Element Library for the Automated Solution of PDEs

- high level mathematical input language
- generates efficient C++ code from these mathematical inputs
- supports a wide range of different finite element types
- supports simulations in 2D and 3D
- fully parallelised
- active world wide developer community, e.g. Simula Research Laboratory, University of Cambridge, University of Chicago, University of Texas at Austin, KTH Royal Institute of Technology, Chalmers University of Technology.



http://fenicsproject.org

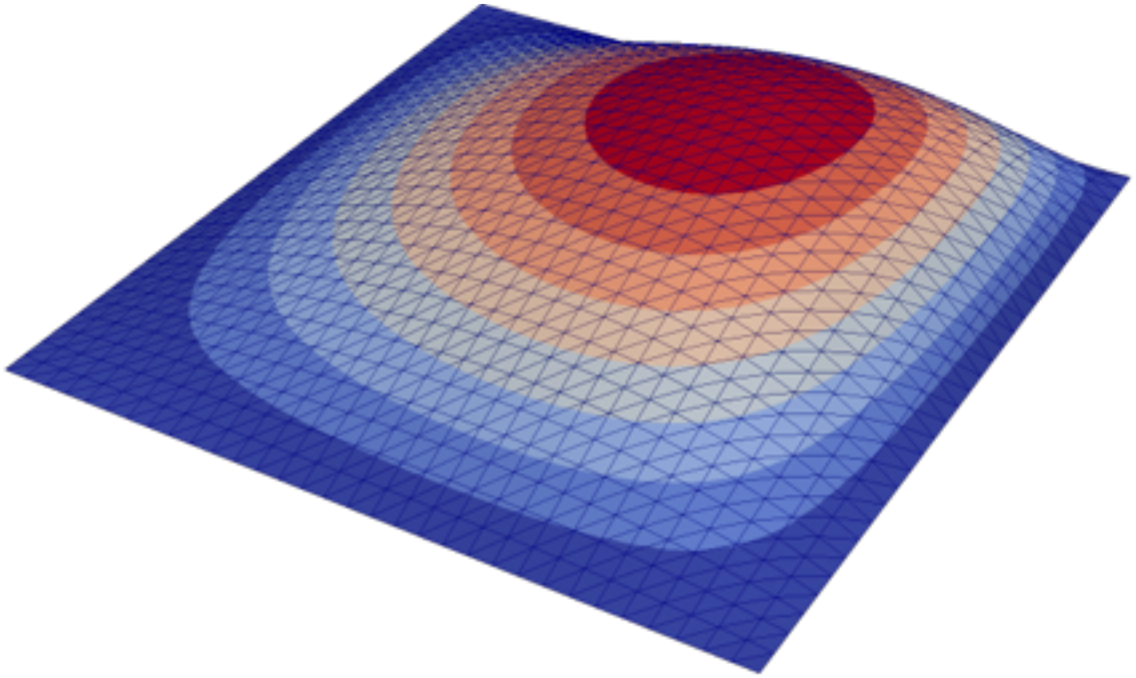
FEniCS Example: Poisson Equation

Consider the elliptic problem

$$\boxed{\begin{aligned} -\Delta u &= f \text{ in } \Omega, \\ u &= 0 \text{ on } \Gamma. \end{aligned}}$$

Find $u \in V$ such that

$$\int_\Omega \nabla u \cdot \nabla v dx = \int_\Omega f v dx \quad \forall v \in V$$



```python
from dolfin import *

# Create mesh and define function space
mesh = UnitSquareMesh(32, 32)
V = FunctionSpace(mesh, "CG", 1)

# Define boundary condition
bc = DirichletBC(V, 0.0, DomainBoundary())

# Define variational problem
u = TrialFunction(V)
v = TestFunction(V)
f = Expression(“x[0]*x[1]”)

a = inner(grad(u), grad(v))*dx
L = f*v*dx

# Compute solution
u = Function(V)
solve(a == L, u, bc)
```

FEniCS: Under The Hood

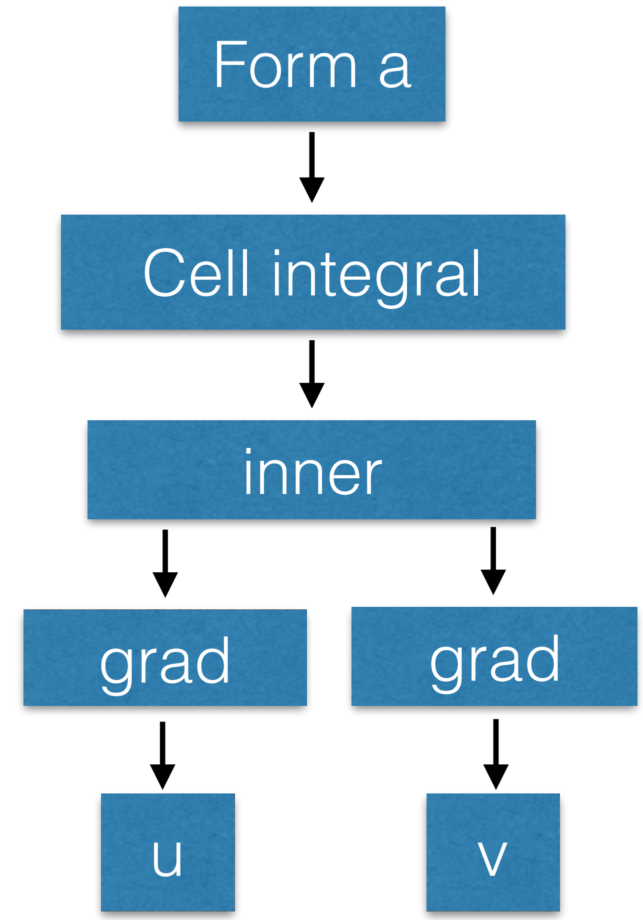
Input: File in Python and Unified From Language

Unified Form Language (UFL)

Interprets expressions close to mathematical notation

```
a=inner(grad(u), grad(v))*dx
```

$$a(u,v) = \int \nabla u \nabla v \, dx$$



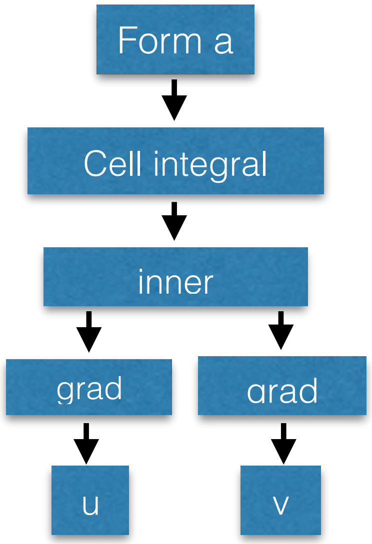
FEniCS: Under The Hood

Input: File in Python and Unified From Language

a=inner(grad(u), grad(v))*dx

$$a(u,v) = \int \nabla u \nabla v \, dx$$

Unified Form Language (UFL)
Interprets expressions close to mathematical notation



FEniCS Form Compiler (FFC)
Generates Header file with information about
Elemental matrices (form)
Degrees of Freedom Map (element)

Poisson.h

```
/// Evaluate basis function i at given point x in cell
_evaluate_basis(std::size_t i, double* values, const double* x, const double* coordinate_dofs)

 /// Tabulate the tensor for the contribution from a local cell
virtual void tabulate_tensor(double*  A, const double * const *  w, const double* coordinate_dofs)
```

DOLFIN
(Mesh, Communicator and Assembler)

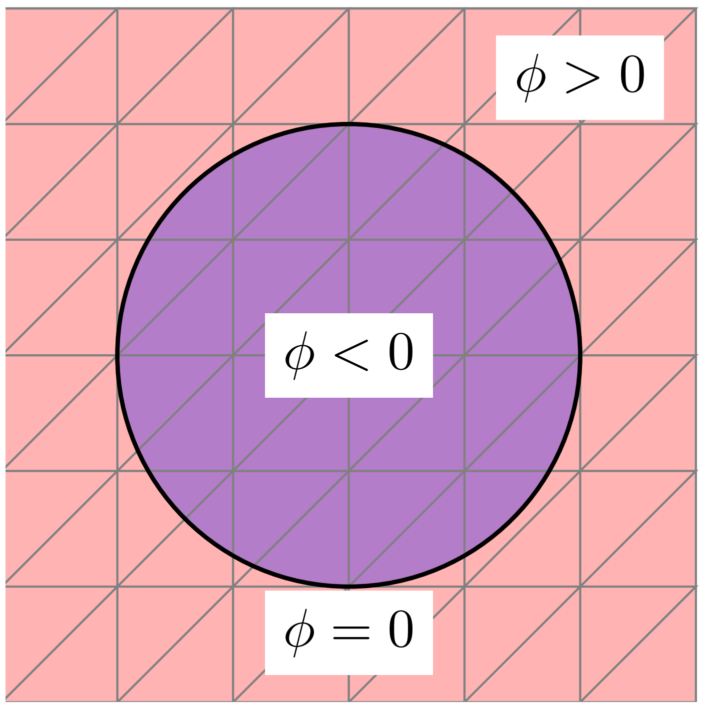
LibCutFEM

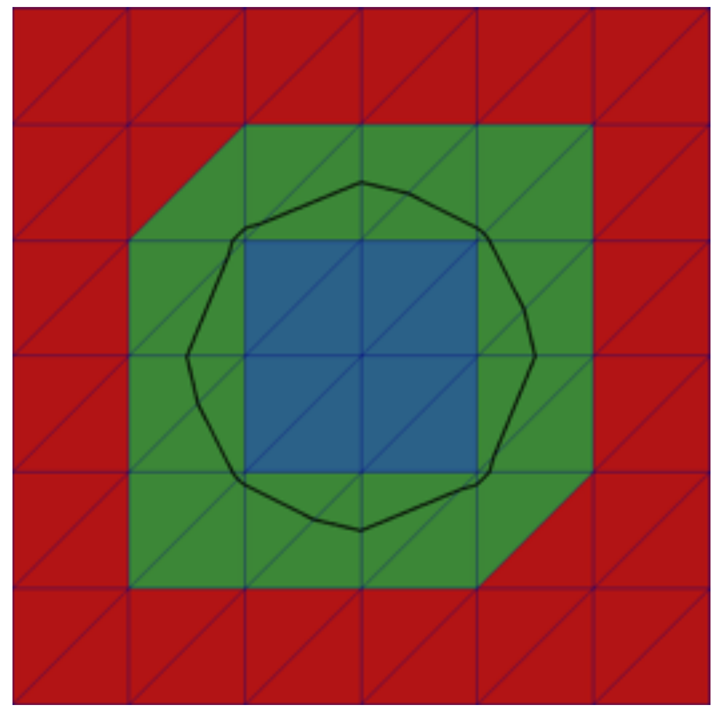
Geometry Algorithm

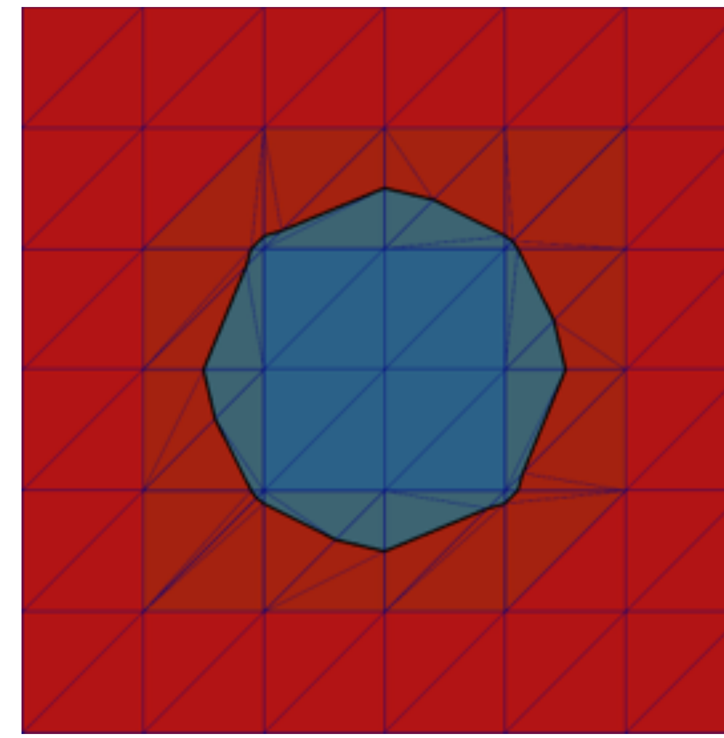
Level Set Function

Finite Element Approximation

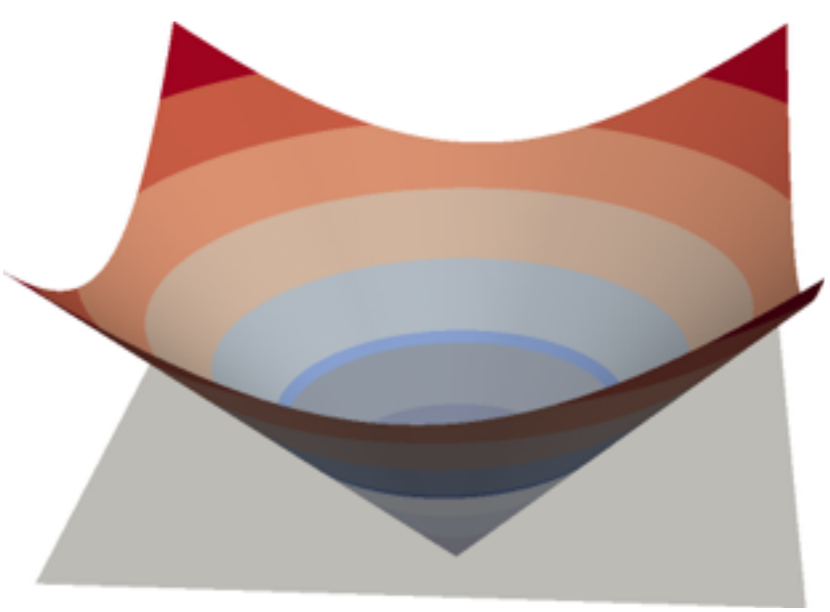
Mesh/Levelset intersection for integration

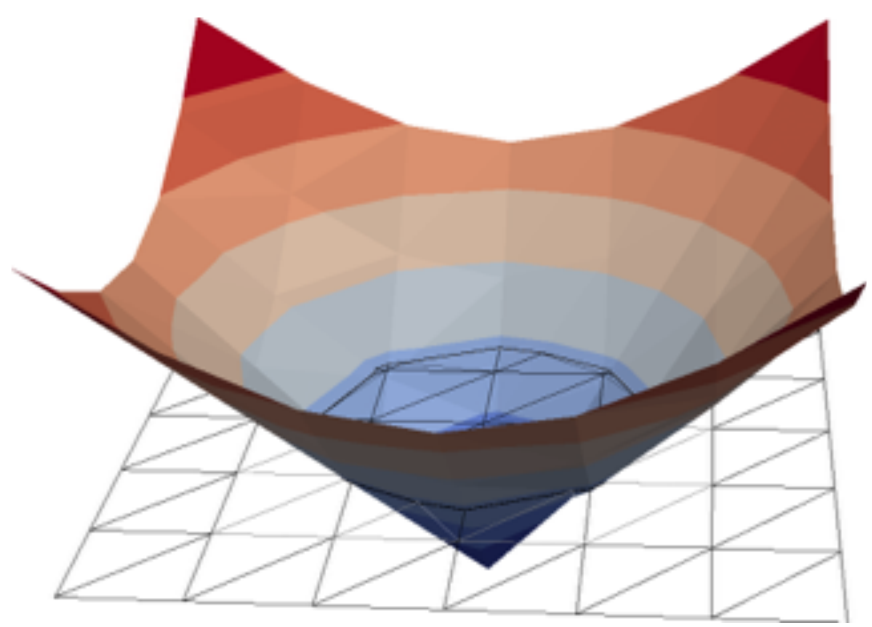


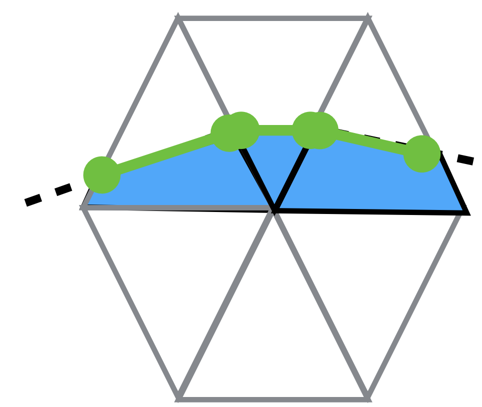


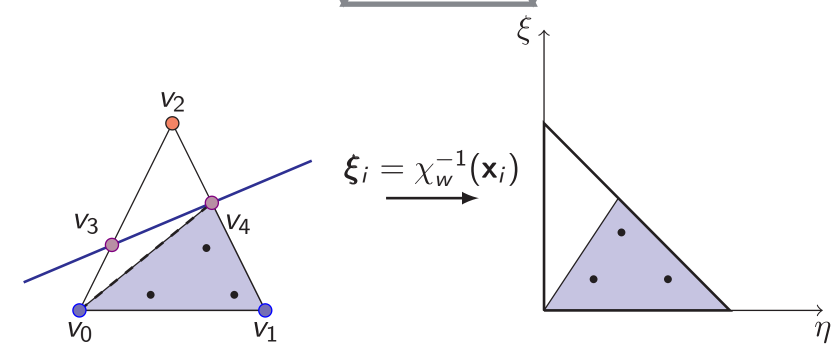


Signed Distance Function

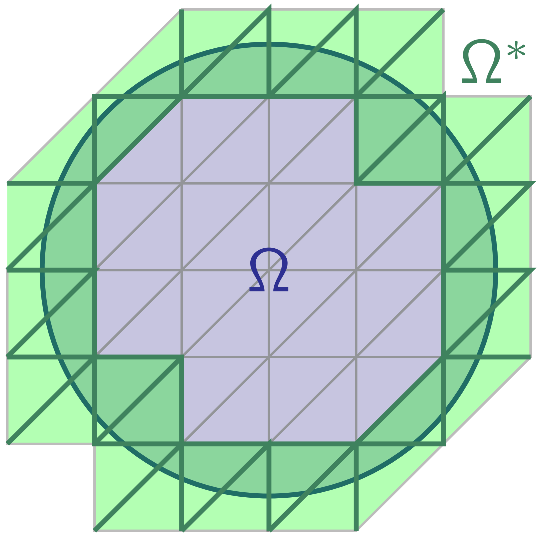


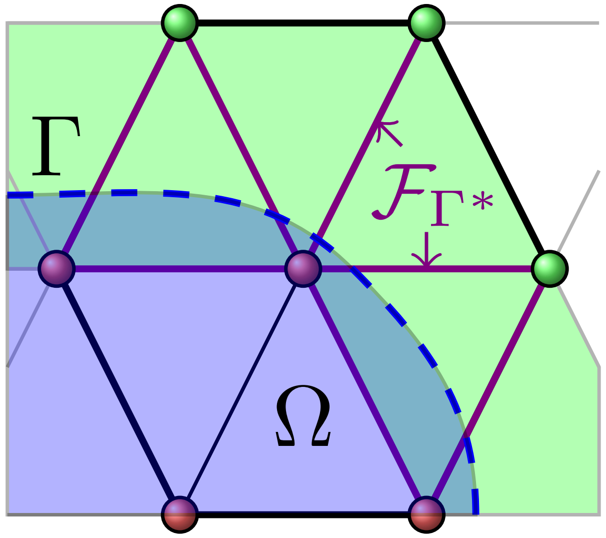






Fictitious Domain Poisson Problem





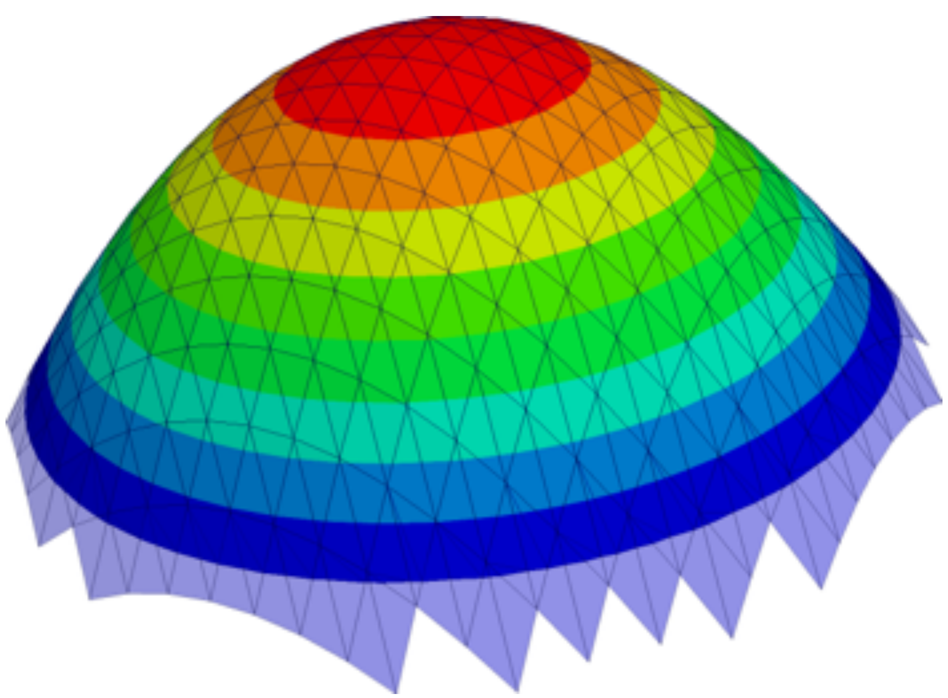
Find $u_h \in V_h$ such that for all $v_h \in V_h$

$$A(u_h, v_h) = a(u_h, v_h) + j(u_h, v_h) = L(v_h)$$

$$a(u_h, v_h) = \int_\Omega \nabla u_h \cdot \nabla v_h \, \mathrm{d}x + \int_\Gamma \Big(-\underbrace{\nabla u_h \cdot n v_h}_{\text{consistency}} - \underbrace{\nabla v_h \cdot n u_h}_{\text{symmetry}} + \underbrace{\frac{\gamma}{h} u_h v_h}_{\text{coercivity}}\Big) \, \mathrm{d}s$$

$$L(v_h) = \int_\Omega f \, v_h \, \mathrm{d}x + \int_\Gamma \Big(-g \, \nabla v_h \cdot n + \frac{\gamma}{h} g \, v_h\Big) \, \mathrm{d}s$$

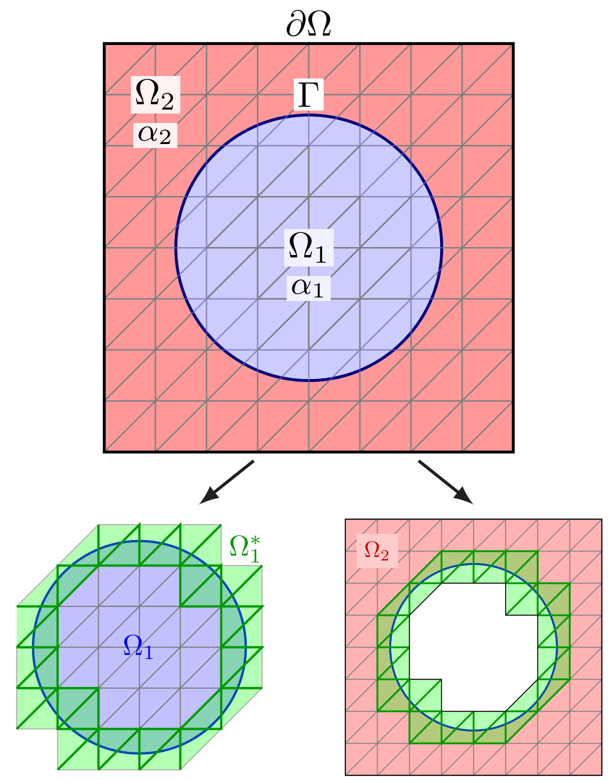
$$j(u_h, v_h) = \gamma_1 \sum_{F \in \mathcal{F}_{\Gamma^*}} h_F([\![\partial_n u_h]\!], [\![\partial_n v_h]\!])_F$$

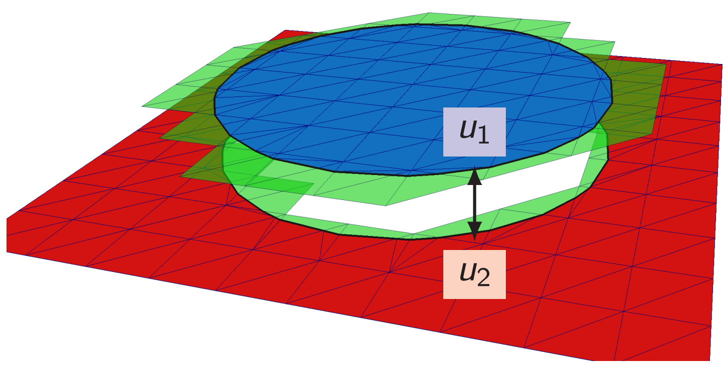


Poisson with contrast in diffusivities

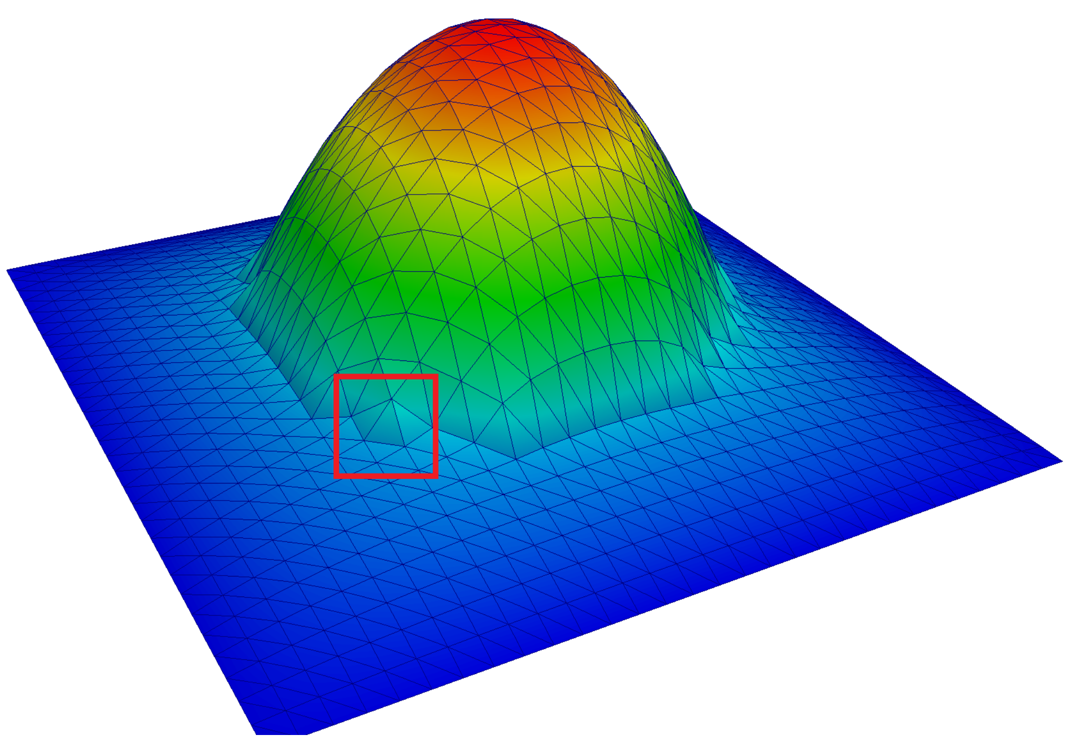
$$
\begin{aligned}
-\nabla \cdot (\alpha \nabla u) &= f \quad \text{in} \quad \Omega_1 \cup \Omega_2, \\
u &= 0 \quad \text{on} \quad \partial\Omega, \\
[\![u]\!] &= 0 \quad \text{on} \quad \Gamma, \\
[\![-\alpha_i \nabla u_i \cdot \mathbf{n}]\!] &= 0 \quad \text{on} \quad \Gamma.
\end{aligned}
$$

Choose $\alpha_1 = 1, \alpha_2 = 10, f_1 = f_2 = 1.$

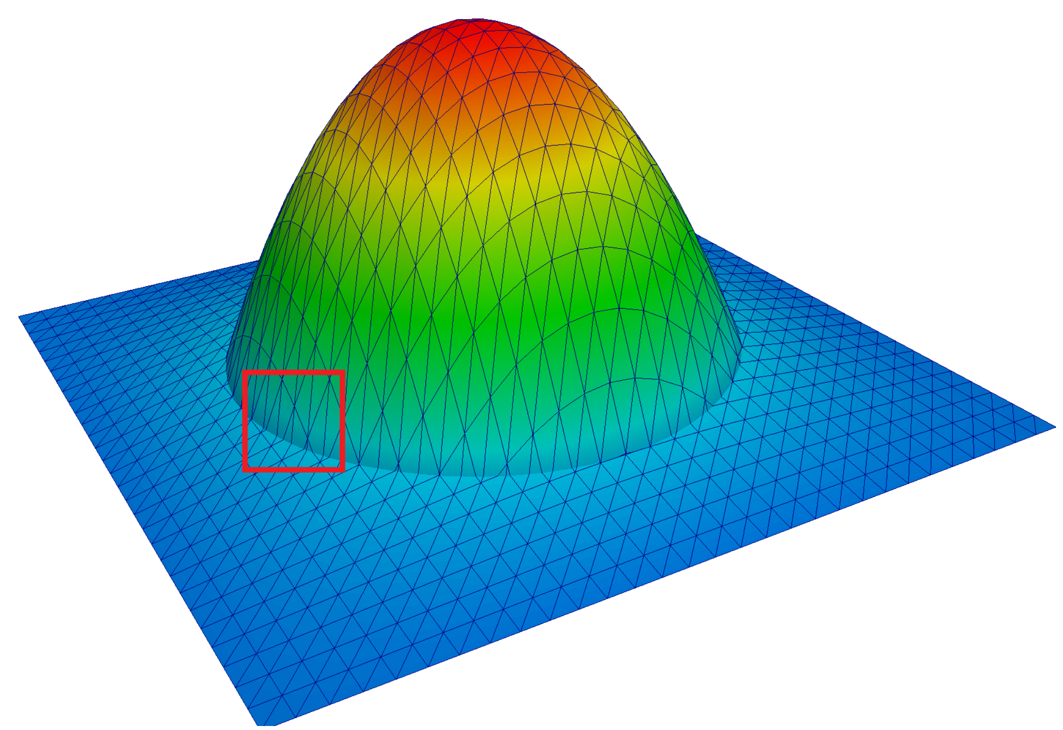




Without Enrichment

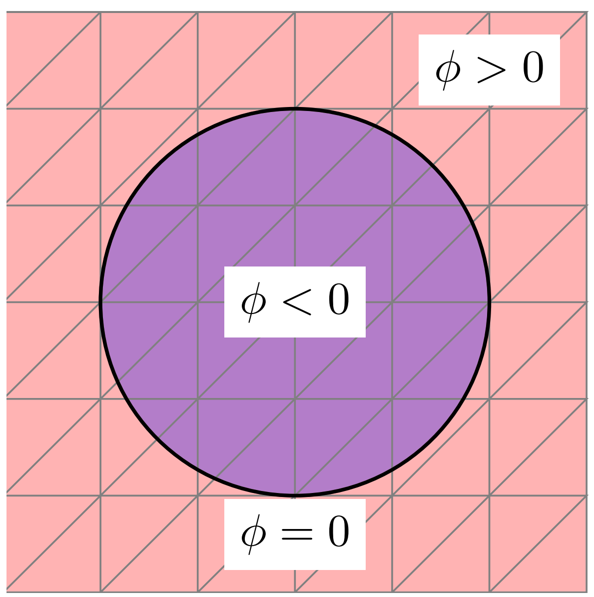


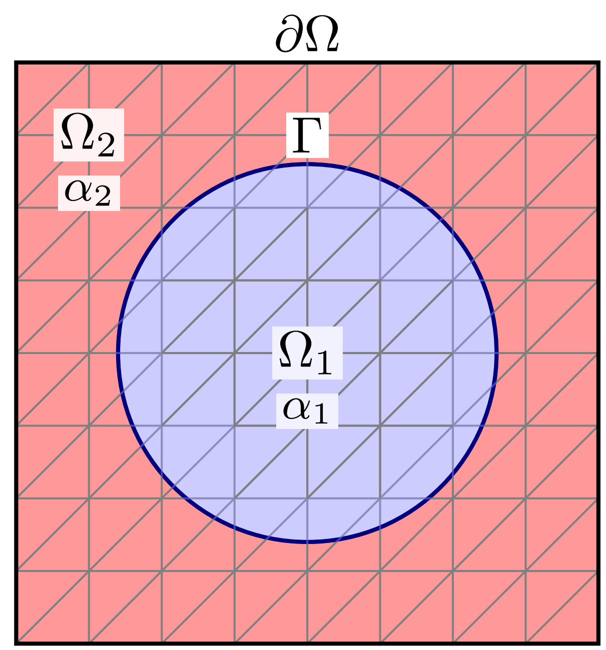
With Enrichment

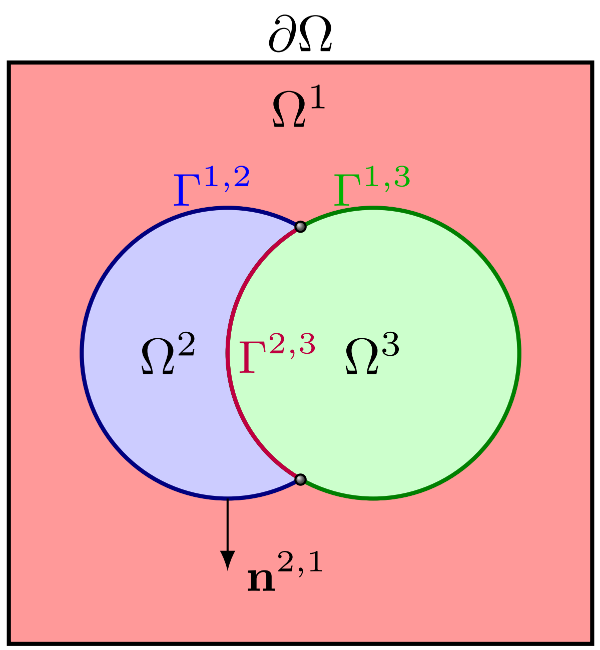


Limitations with single level set function

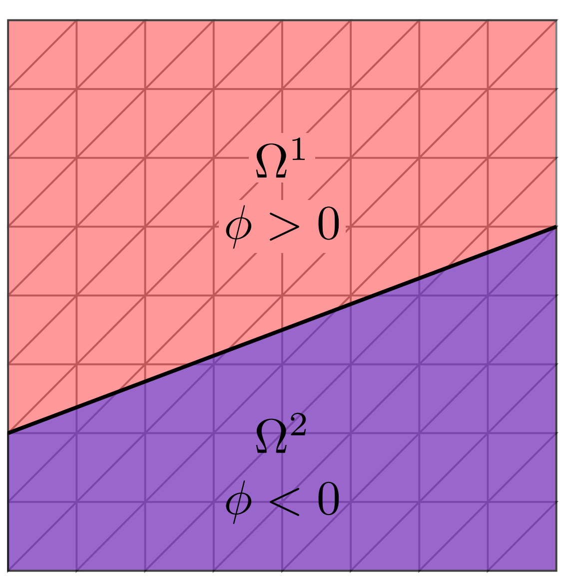
Partitions the domain into one inside and one outside domain (max. 2 different materials)

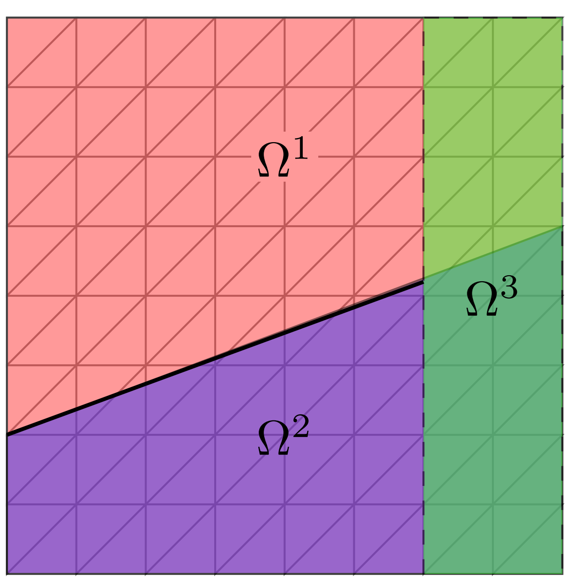






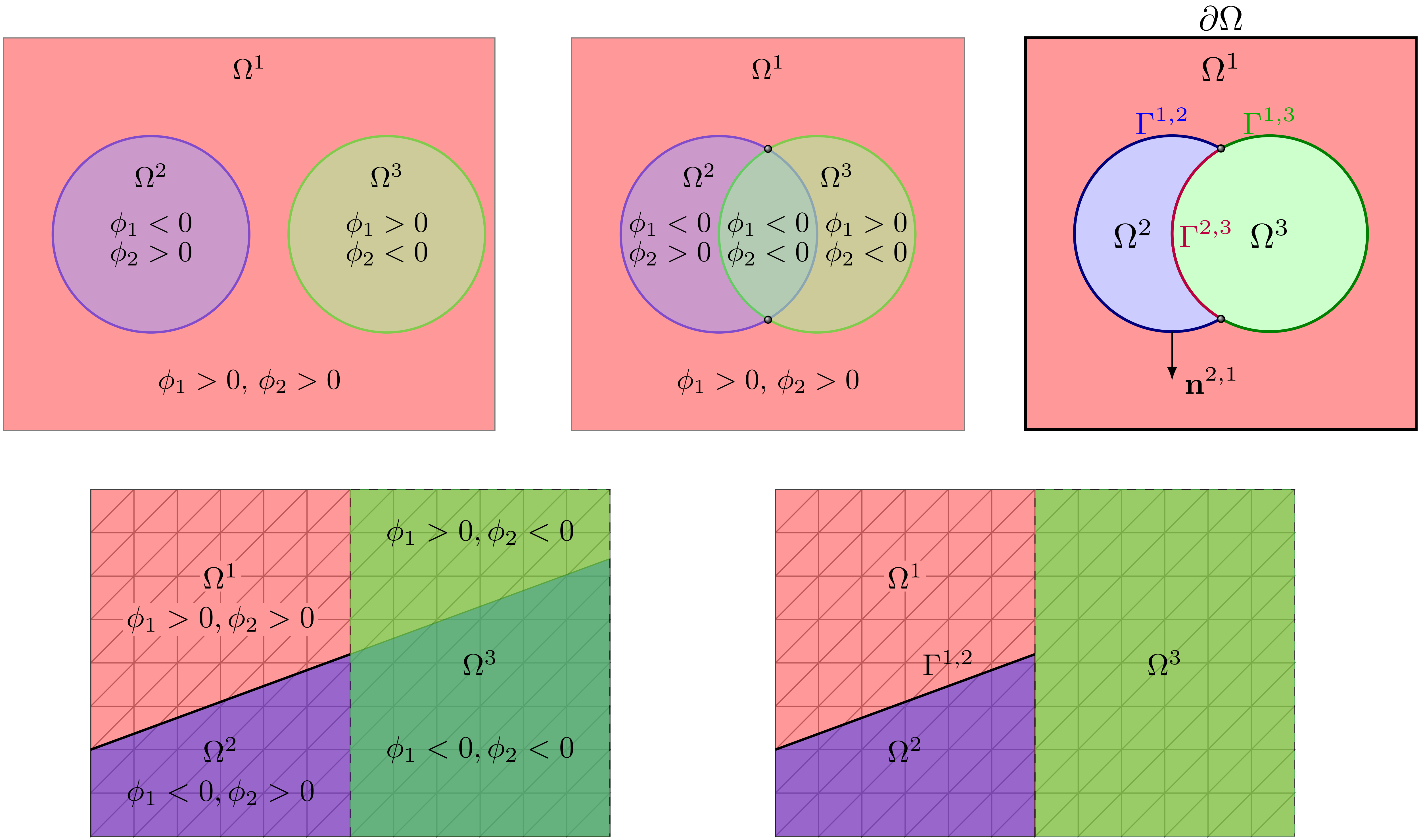
Zero level set surface contour is a closed surface, i.e it is not possible to describe geometries with open boundaries such as cracks



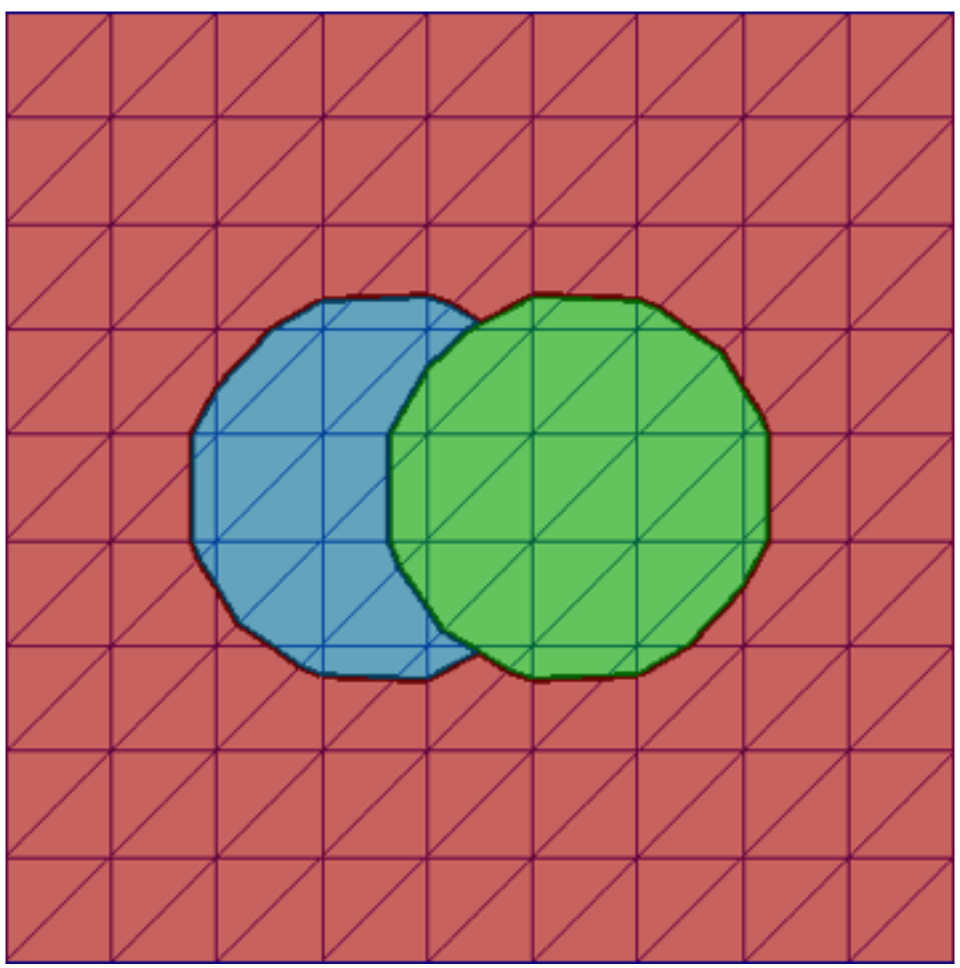


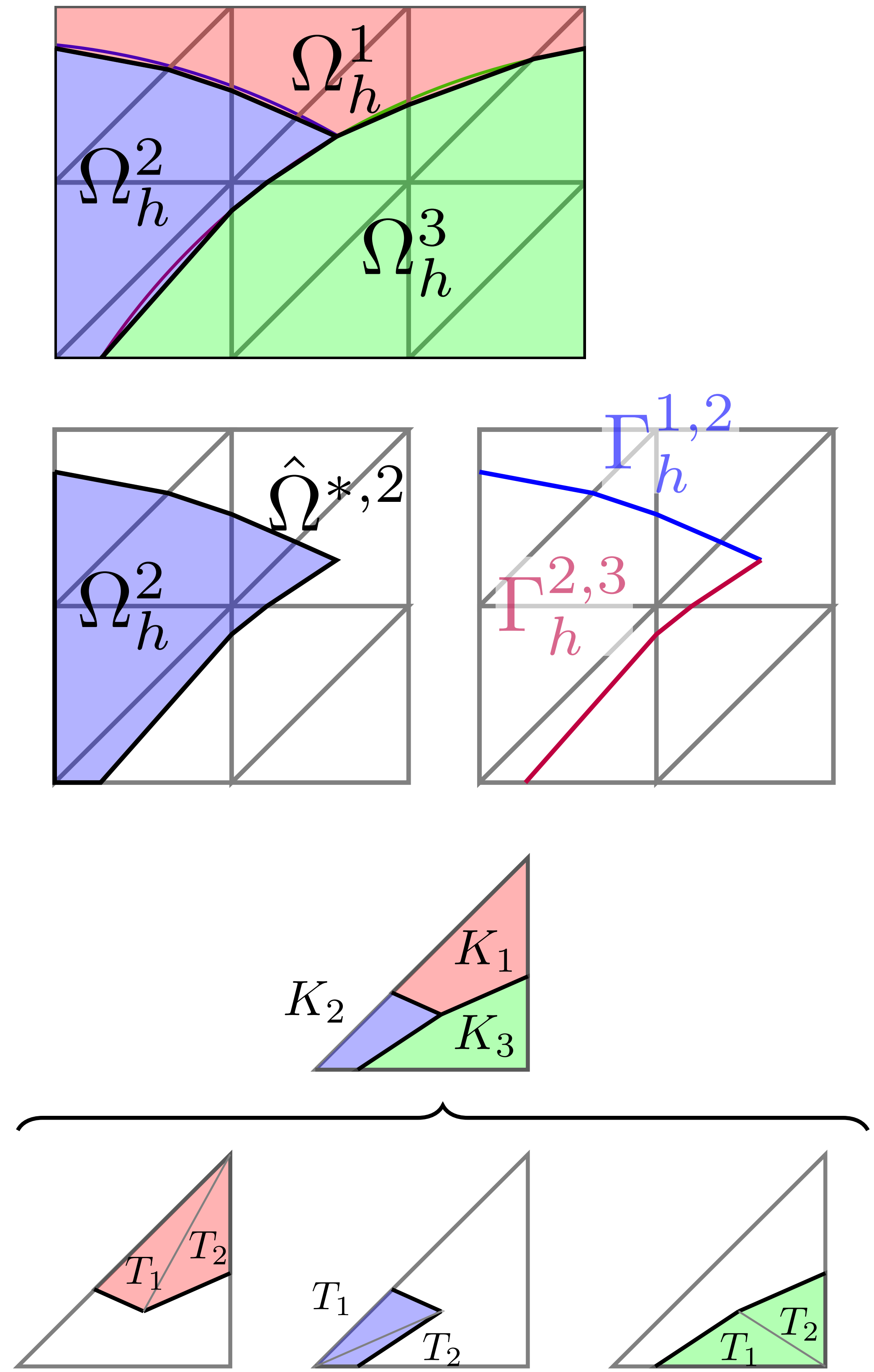
Use multiple level set functions for complex geometries

N-1 level set functions can described N different subdomains

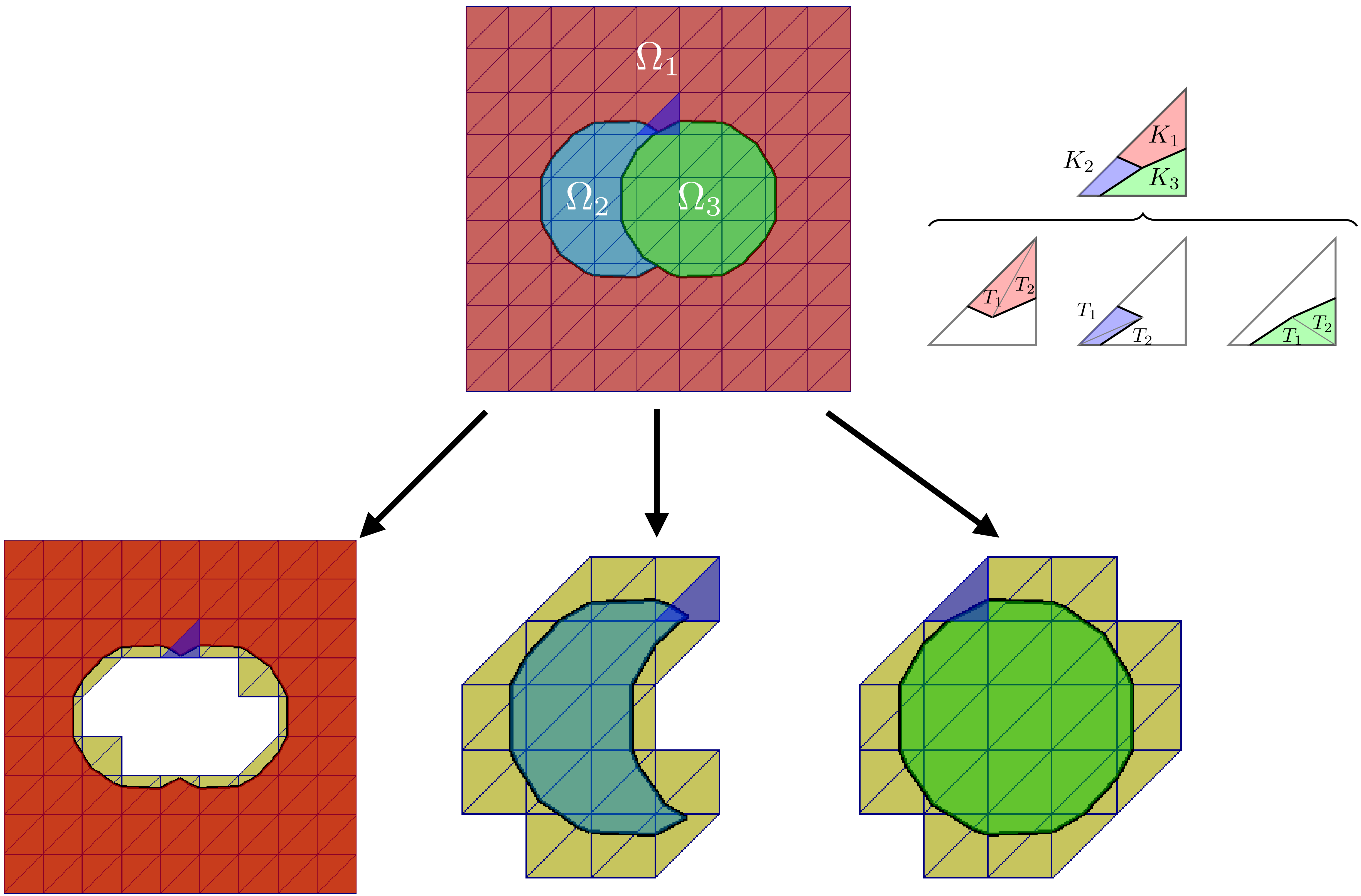


Level Set Mesh Intersection



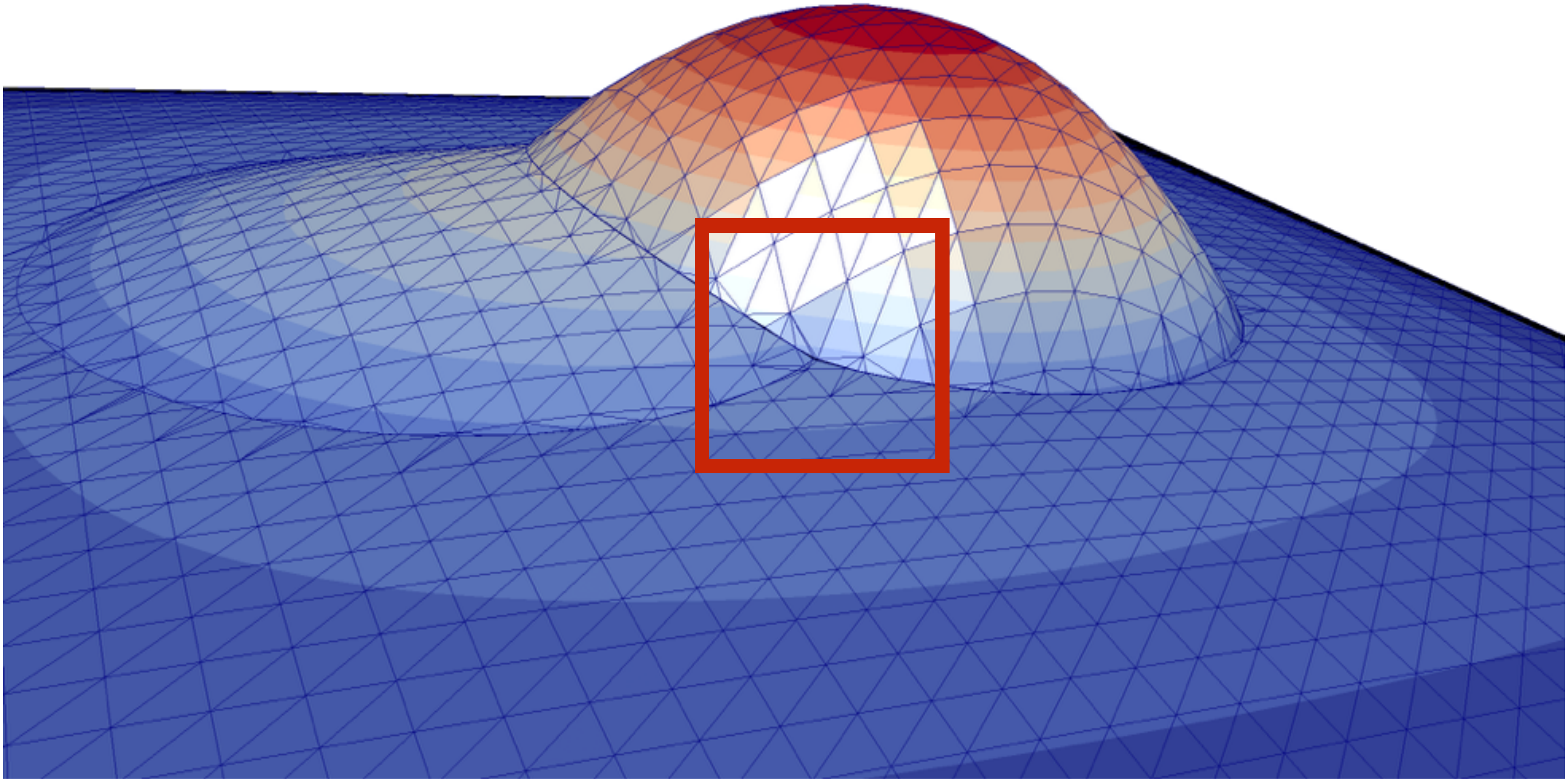


Enrichment for jump and kink representation



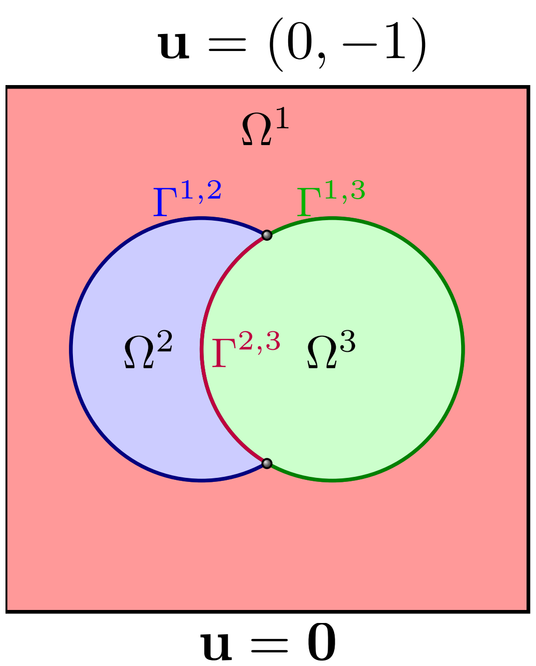
Claus, S., and P. Kerfriden. "A stable and optimally convergent LaTIn-CutFEM algorithm for multiple unilateral contact problems." IJNME 113.6 (2018): 938-966.

Triple Poisson Problem



Contact Problems

Contact Problem in linear Elasticity



Bulk Problem

For all Ω_i, find the displacement fields $\mathbf{u}^i : \Omega_i \to \mathbb{R}$, such that

$$-\nabla \cdot \boldsymbol{\sigma}(\mathbf{u}^i) = \mathbf{f} \qquad \text{in } \Omega_i$$

$$\boldsymbol{\sigma}(\mathbf{u}^i) = \lambda^i \,\mathrm{tr}(\boldsymbol{\epsilon}(\mathbf{u}^i))\, \boldsymbol{I} + 2\,\mu_i\, \boldsymbol{\epsilon}(\mathbf{u}^i) \qquad \text{in } \Omega_i$$

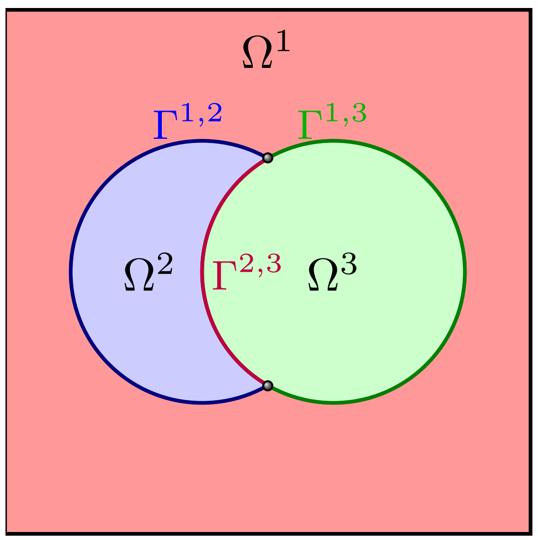
$$\mathbf{u}^i = \mathbf{g} \text{ on } \partial\Omega_D\,,\ \boldsymbol{\sigma}(\mathbf{u}_i)\cdot \mathbf{n} = \mathbf{F_N} \text{ on } \partial\Omega_N$$

Here, $\boldsymbol{\epsilon}(\mathbf{u}) = \frac{1}{2}\left(\nabla \mathbf{u} + \nabla \mathbf{u}^T\right)$ is the strain tensor, $\mathbf{f}$ is the body force, $\mathbf{F_N}$ is the surface load, $\mathbf{g}$ the Dirichlet boundary condition, λ^i and μ^i are the two Lamé coefficients (E^i is the Young's modulus, $\nu = 0.3$ is the Poisson's ratio)

$$\mu^i = \frac{E^i}{2(1+\nu)}, \quad \lambda^i = \frac{E^i \nu}{(1+\nu)(1-2\nu)}.$$

Unilateral contact for isotropic linear elasticity

$\mathbf{u} = (0, -1)$



$\mathbf{u} = \mathbf{0}$

Contact Conditions

For any displacement field $\mathbf{u}_i$, we decompose the surface traction $\mathbf{F}^i = \boldsymbol{\sigma}(\mathbf{u}_i) \cdot \mathbf{n}^{i,j}$ on the interface $\Gamma^{i,j}$ into its normal and tangential components

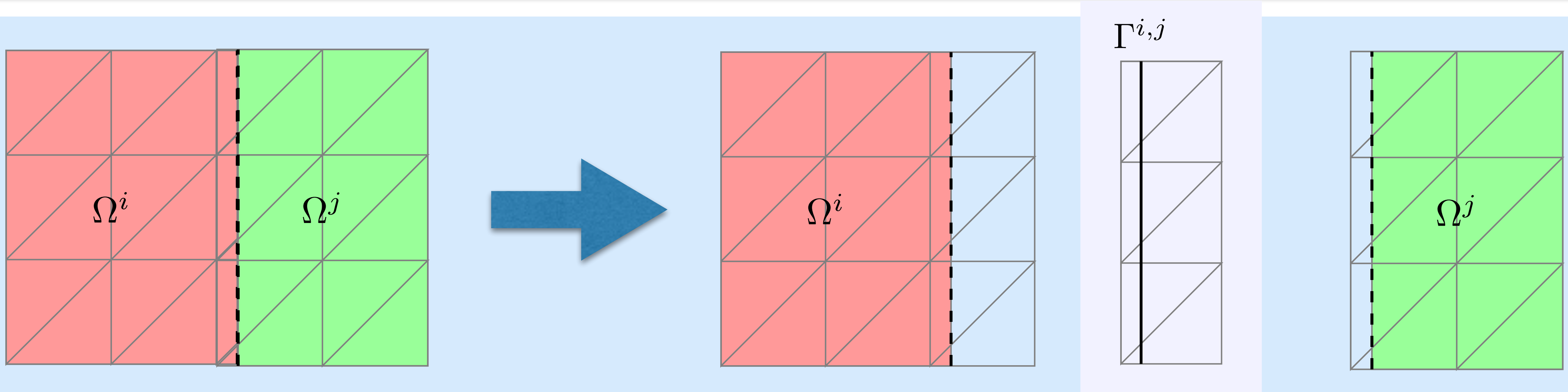
$$\mathbf{F}^i = \mathbf{F_n}^i + \mathbf{F_t}^i .$$

Then, the conditions of contact with Coulomb friction reads

$$\begin{aligned} \left(\mathbf{u}^j - \mathbf{u}^i\right) \cdot \mathbf{n}^{i,j} &\geq 0\,, \\ \mathbf{F}^i \cdot \mathbf{n}^{i,j} &\leq 0\,, \\ \left(\left(\mathbf{u}^j - \mathbf{u}^i\right) \cdot \mathbf{n}^{i,j}\right) \cdot \left(\mathbf{F}^i \cdot \mathbf{n}^{i,j}\right) &= 0\,, \\ \|\mathbf{F_t}^i\| &\leq c\, \mathbf{F}^i \cdot \mathbf{n}^{i,j} && \text{if } \|\hat{\mathbf{g}}_t^i\|_2 = 0 \\ \mathbf{F_t}^i &= -c\, \mathbf{F}^i \cdot \mathbf{n}^{i,j} \frac{\hat{\mathbf{g}}_t^i}{\|\hat{\mathbf{g}}_t^i\|_2} && \text{if } \|\hat{\mathbf{g}}_t^i\|_2 > 0 \end{aligned}$$

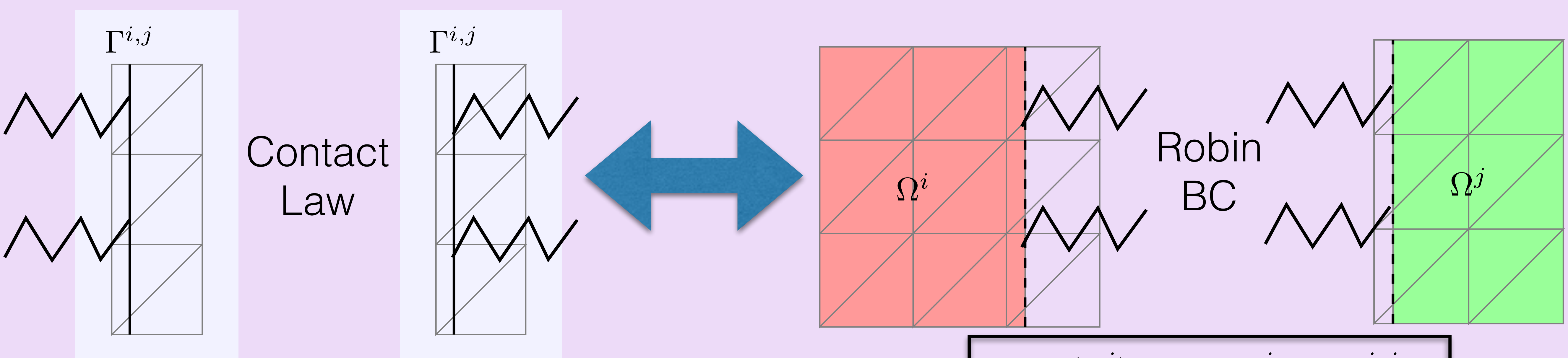
Here, $\mathbf{n}^{i,j}$ is the normal pointing from Ω_i to Ω_j, c is the Coulomb friction coefficient, and $\hat{\mathbf{g}}_t^i := (\mathbf{I} - \mathbf{n}^{i,j} \otimes \mathbf{n}^{i,j}) \cdot (\dot{\mathbf{u}}^j - \dot{\mathbf{u}}^i)$ is the relative tangential velocity

LaTIn Algorithm



Linear Elasticity | Contact | Linear Elasticity

Local Stage | **Linear Stage**

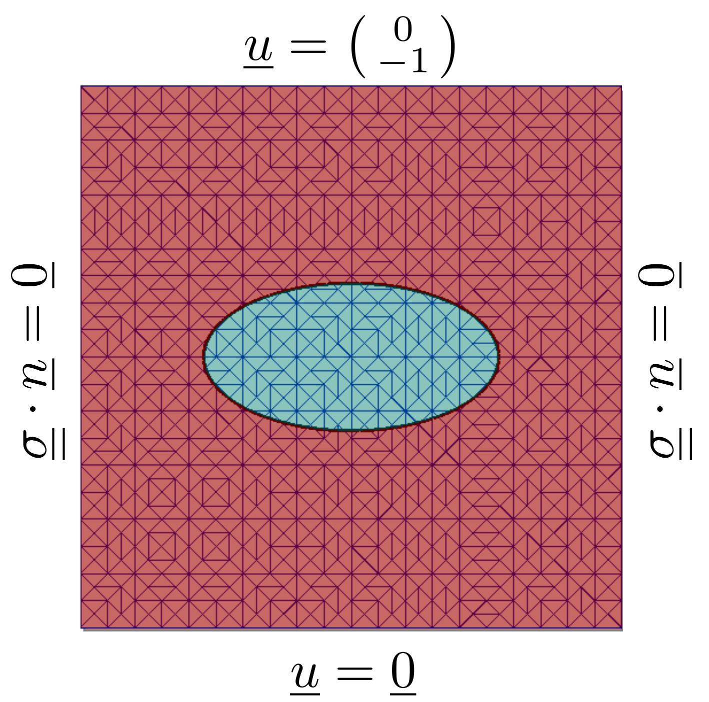


$$\boxed{\hat{\mathbf{F}}^i - k^+ \hat{\mathbf{w}}^i = \mathbf{F}^i - k^+ \mathbf{w}^i}$$

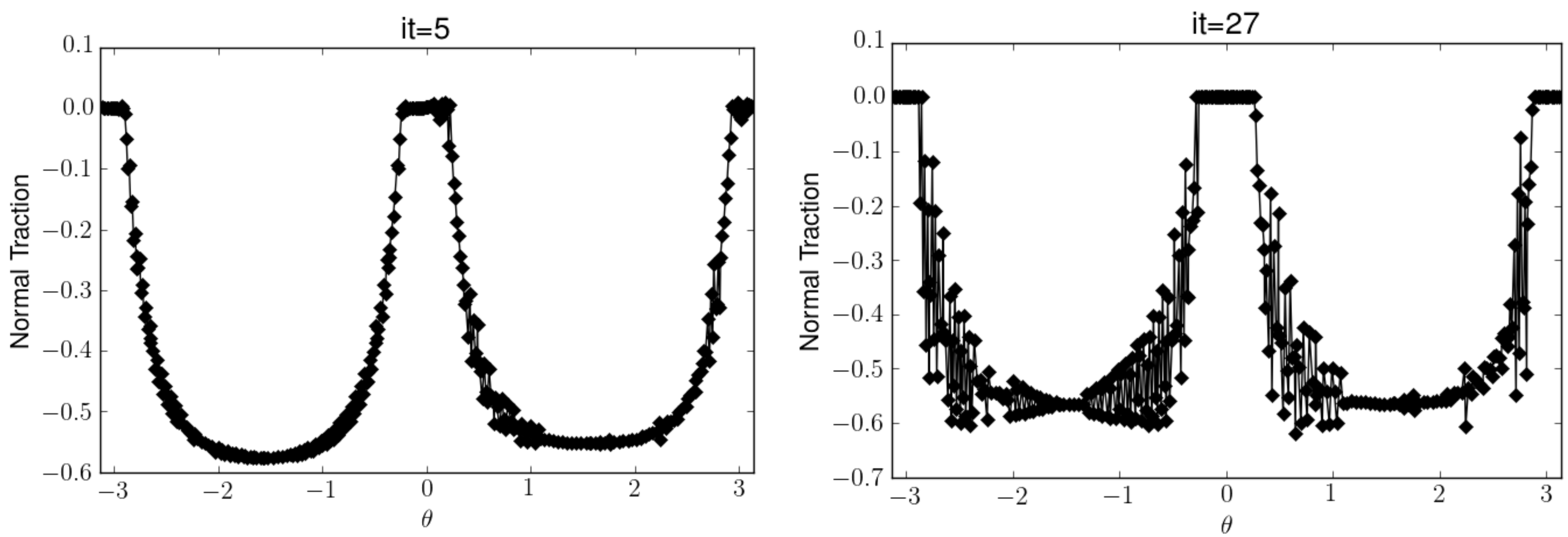
where $\hat{\mathbf{F}}^i$ and $\hat{\mathbf{w}}^i$ satisfy contact

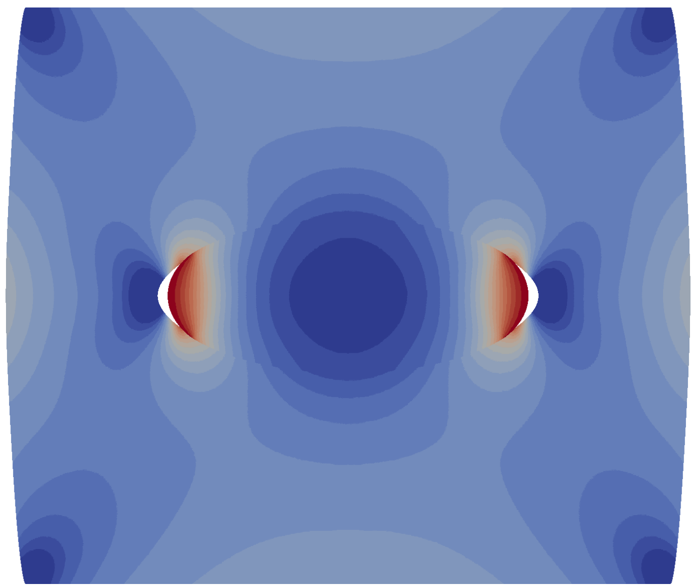
$$\begin{aligned} \boldsymbol{\sigma}(\mathbf{u}^i) \cdot \mathbf{n} &= \mathbf{F}^i \text{ on } \Gamma^{i,j} \\ \mathbf{u}^i &= \mathbf{w}^i \text{ on } \Gamma^{i,j} \\ \mathbf{F}^i + k^- \mathbf{w}^i &= \hat{\mathbf{F}}^i + k^- \hat{\mathbf{w}}^i \end{aligned}$$

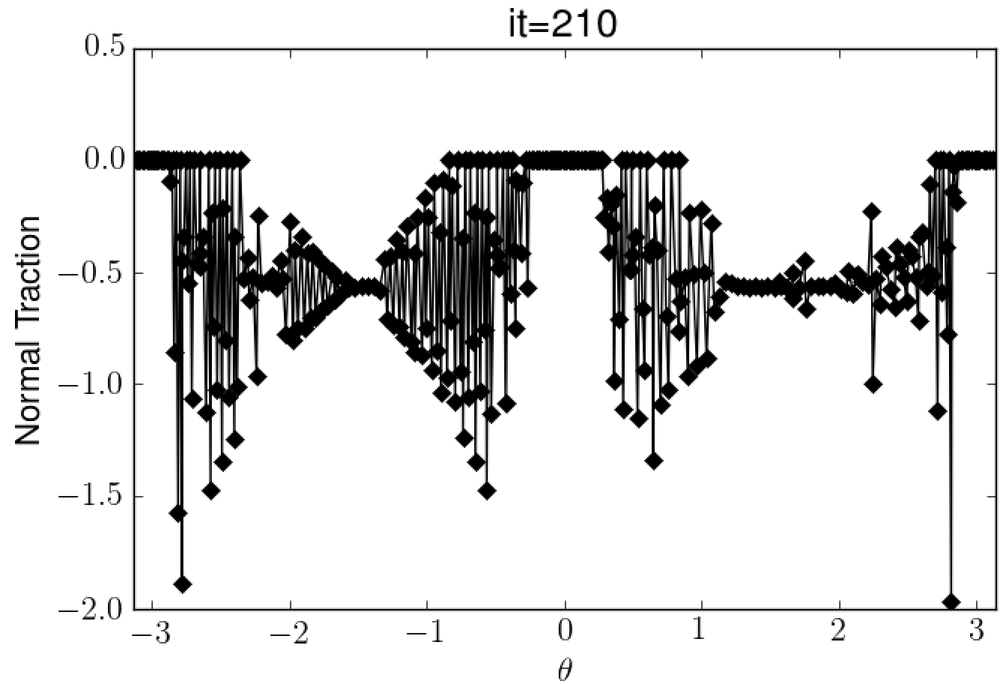
LaTIn Algorithm: Stability



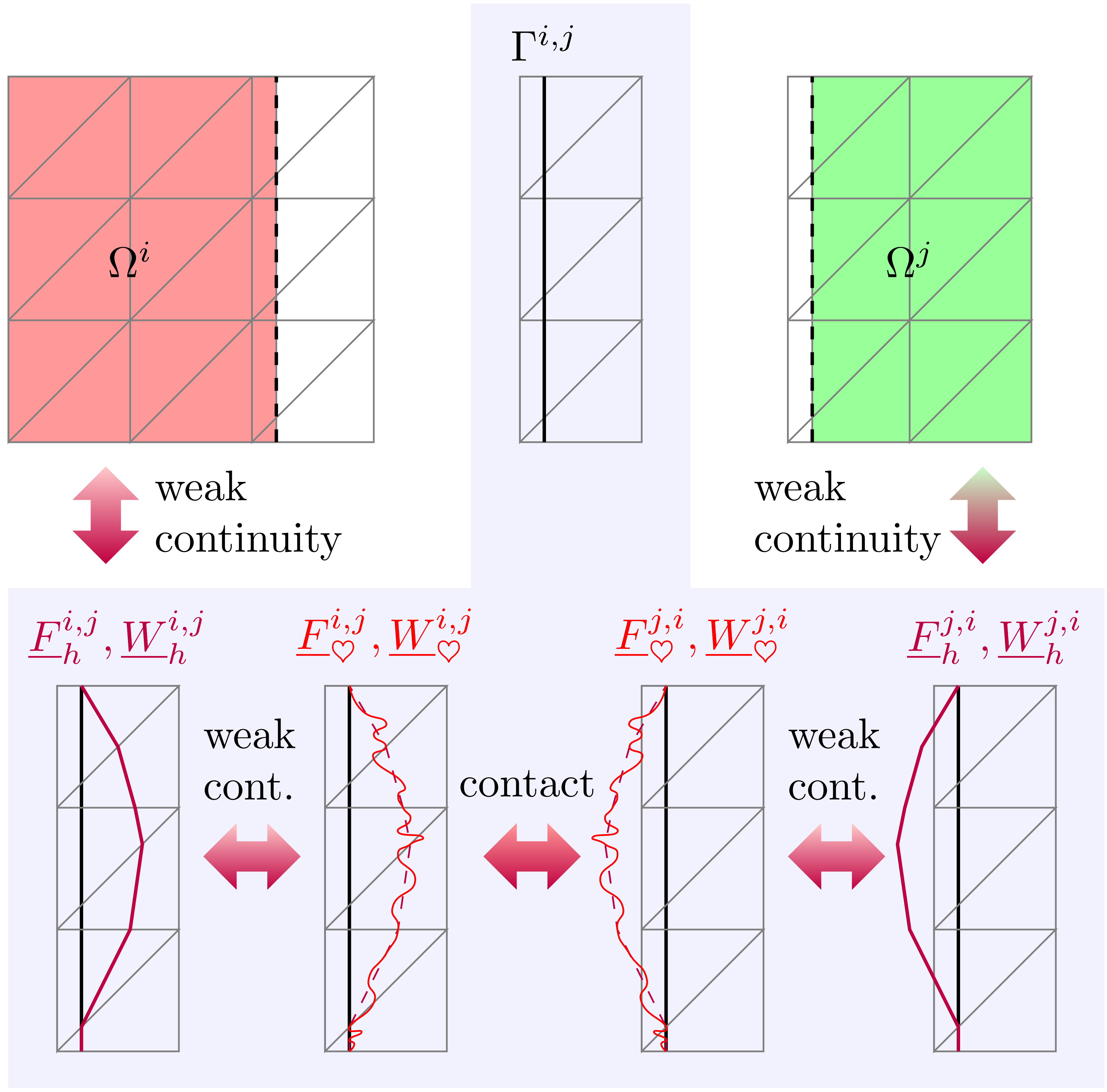
P1/P0 scheme polluted $\mathbf{F}^i$



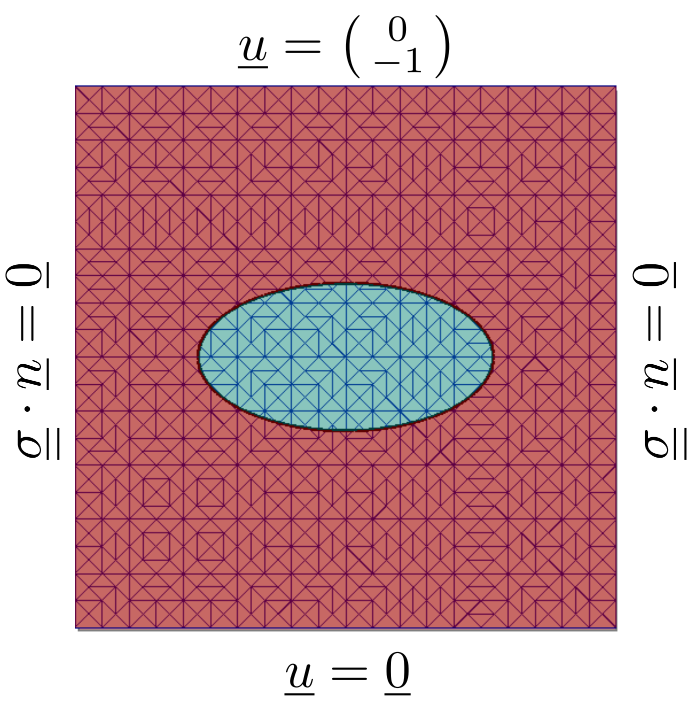




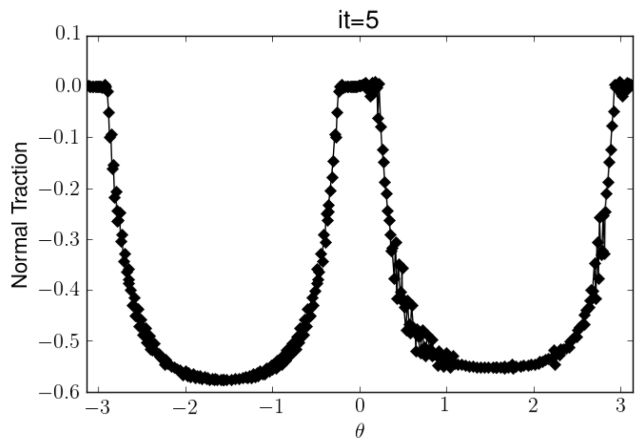
P1/P1 Stabilised Projection



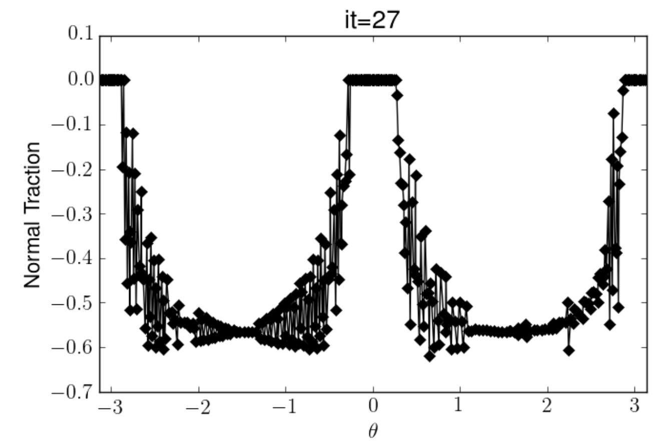
LaTIn Algorithm: Stability



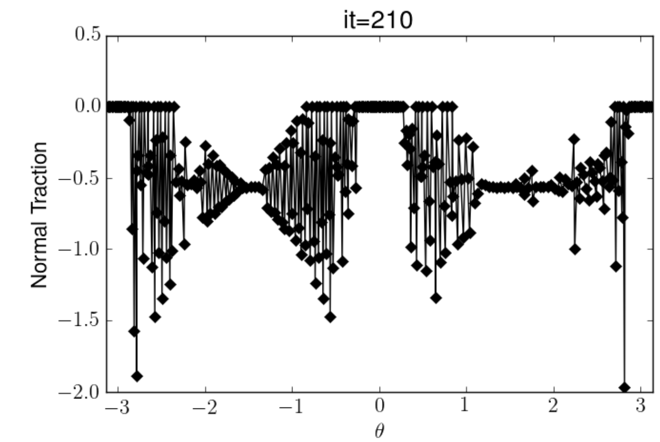
P1/P0 scheme polluted



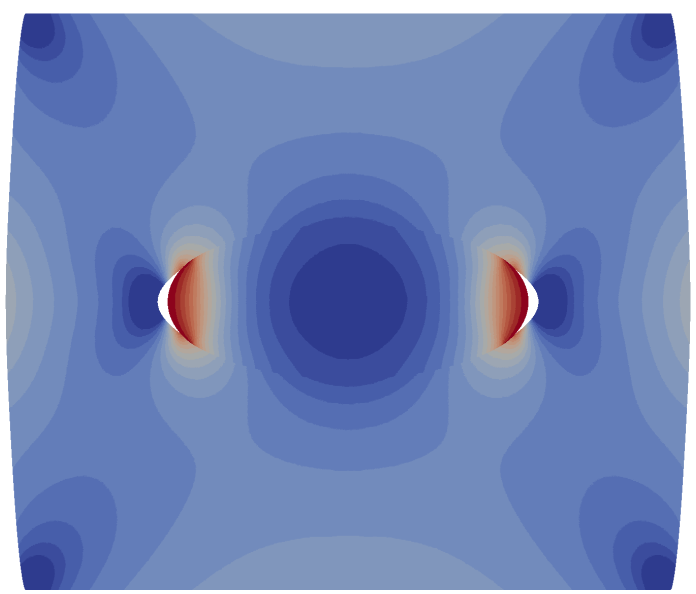
(a) P1/P0



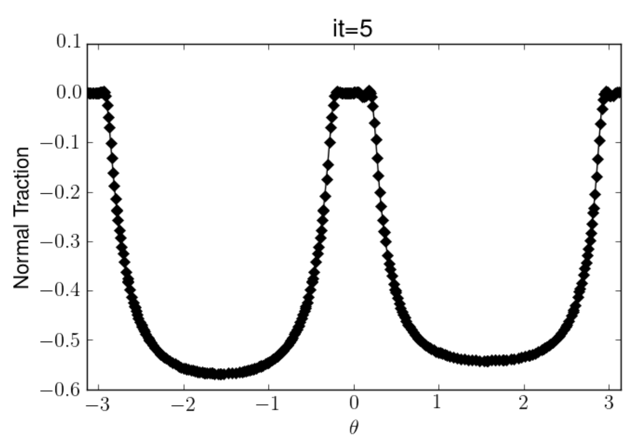
(b) P1/P0



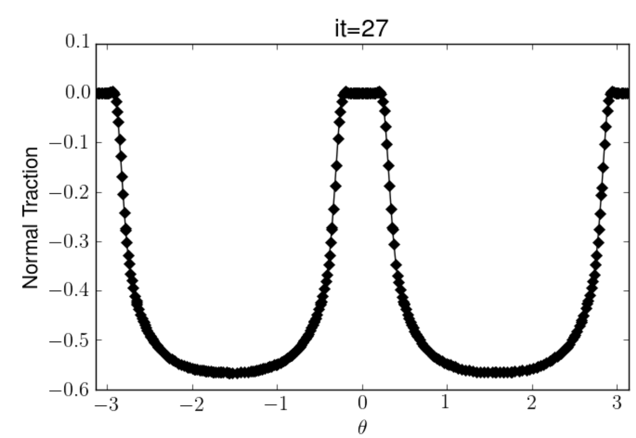
(c) P1/P0



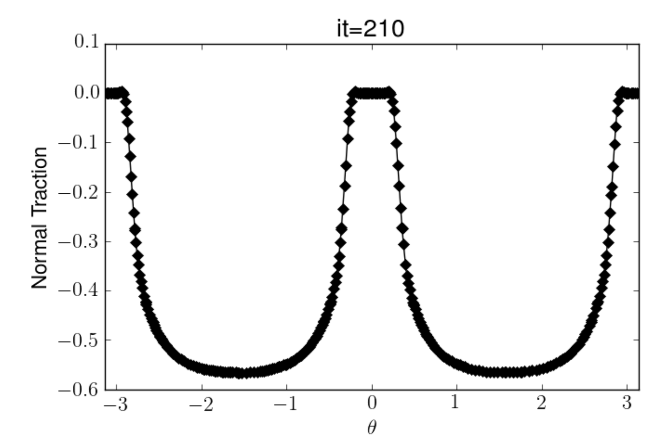
Proposed Solution: P1/P1 scheme with stabilisation



(d) P1/P1

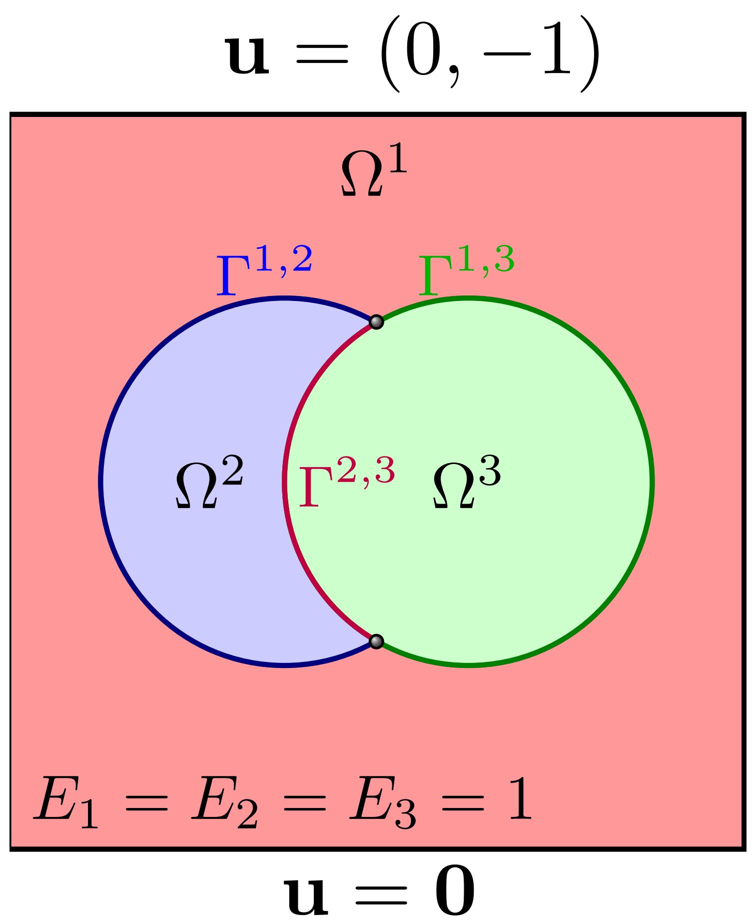


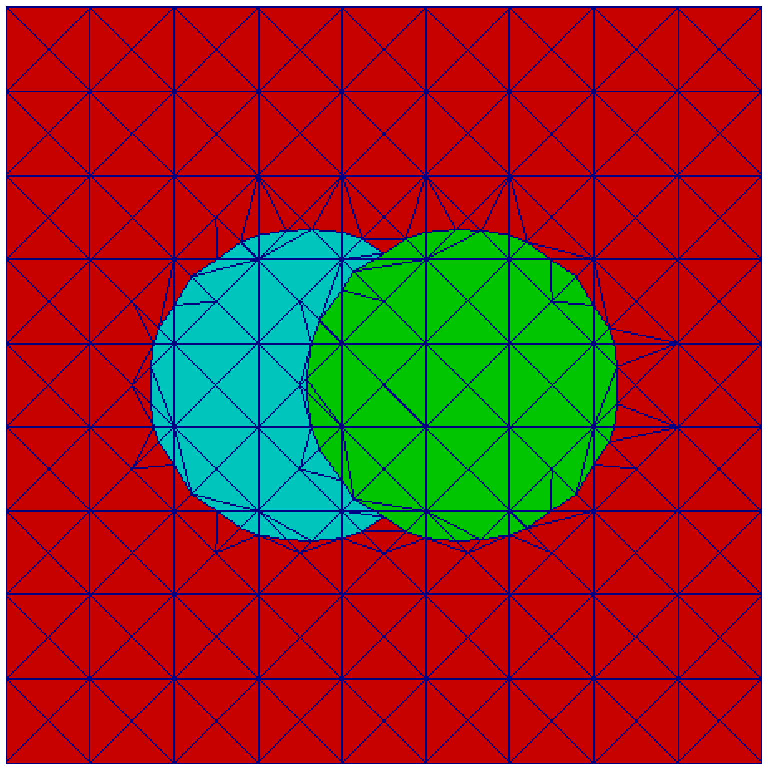
(e) P1/P1

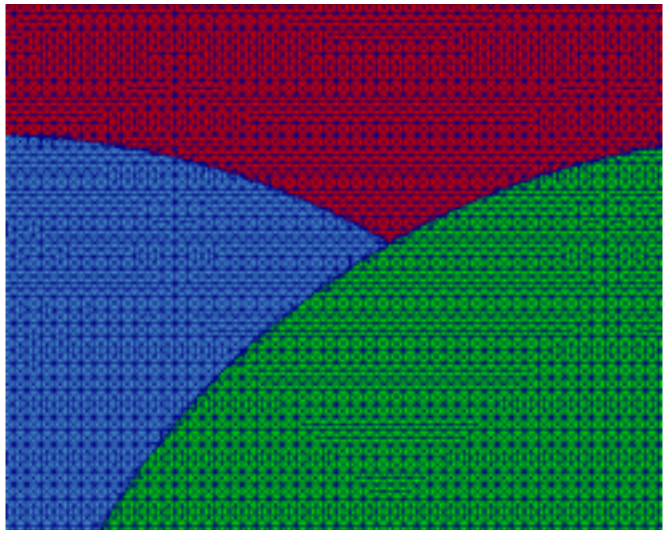


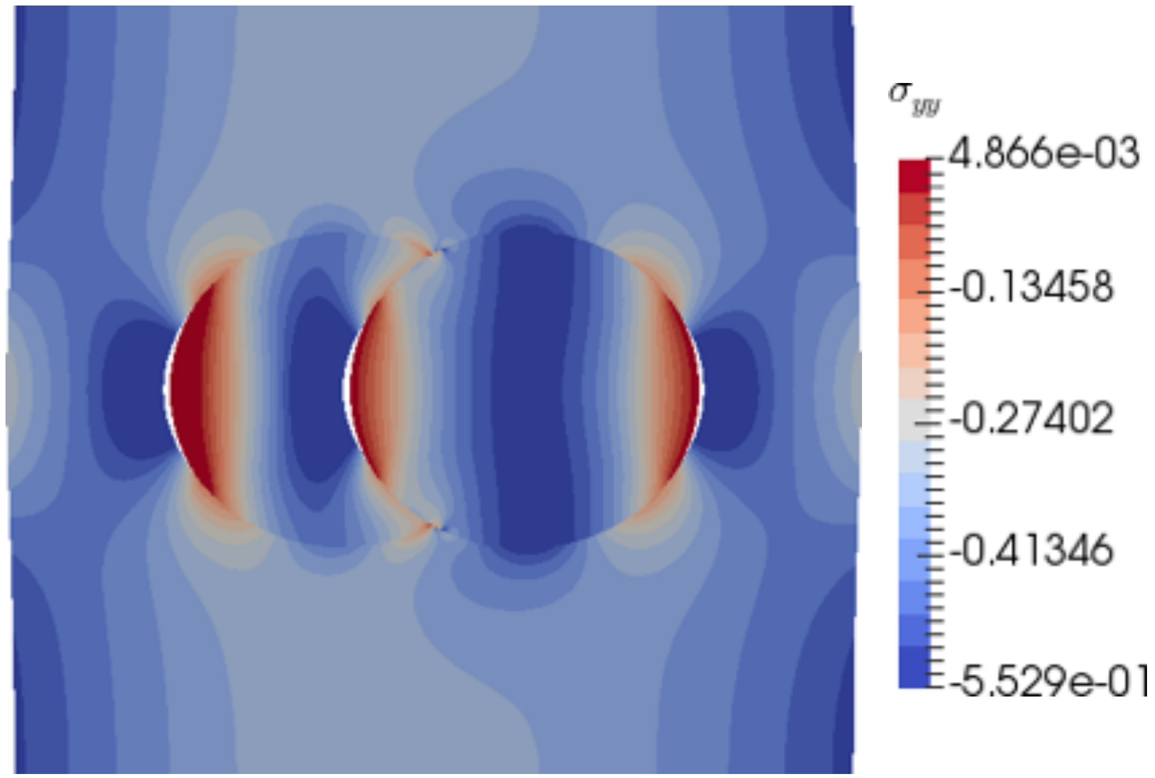
(f) P1/P1

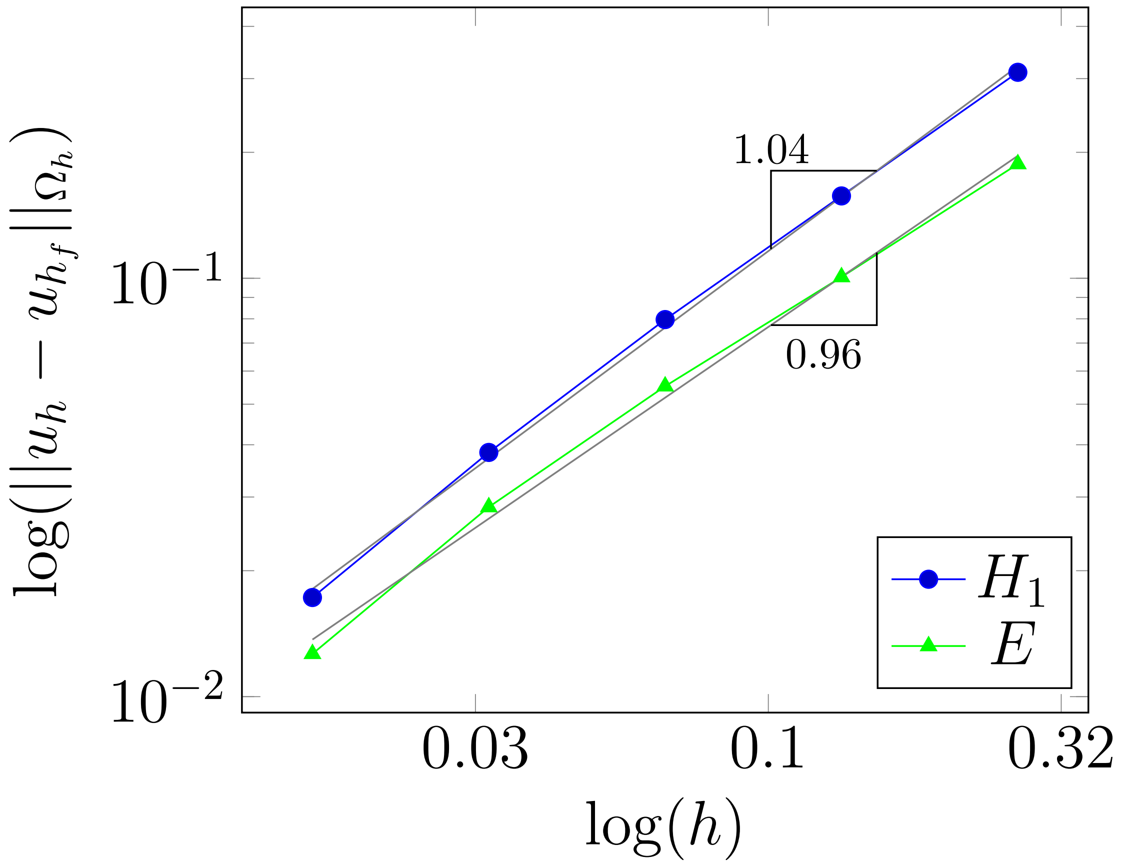
Two Inclusions Frictionless Contact





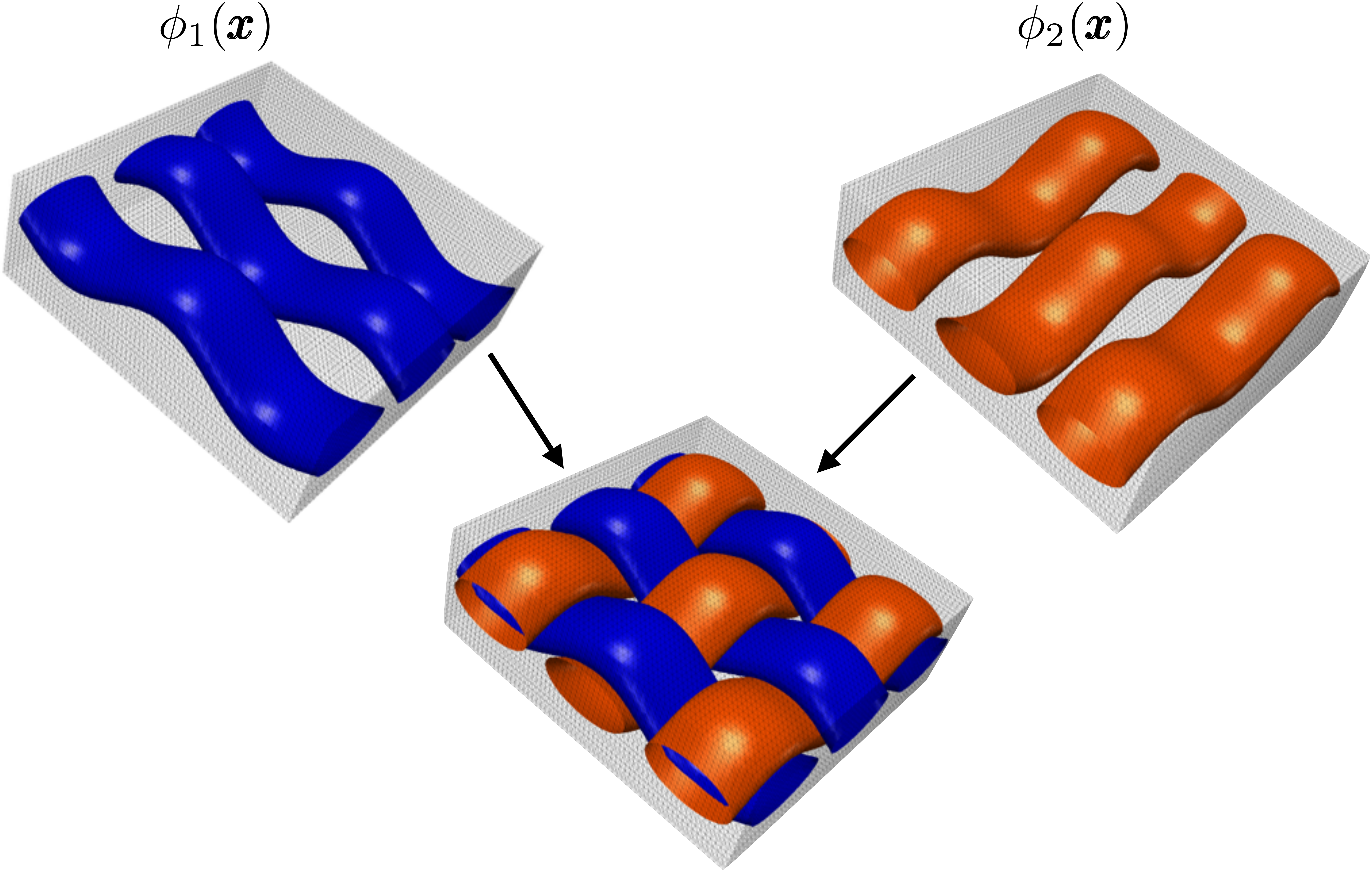




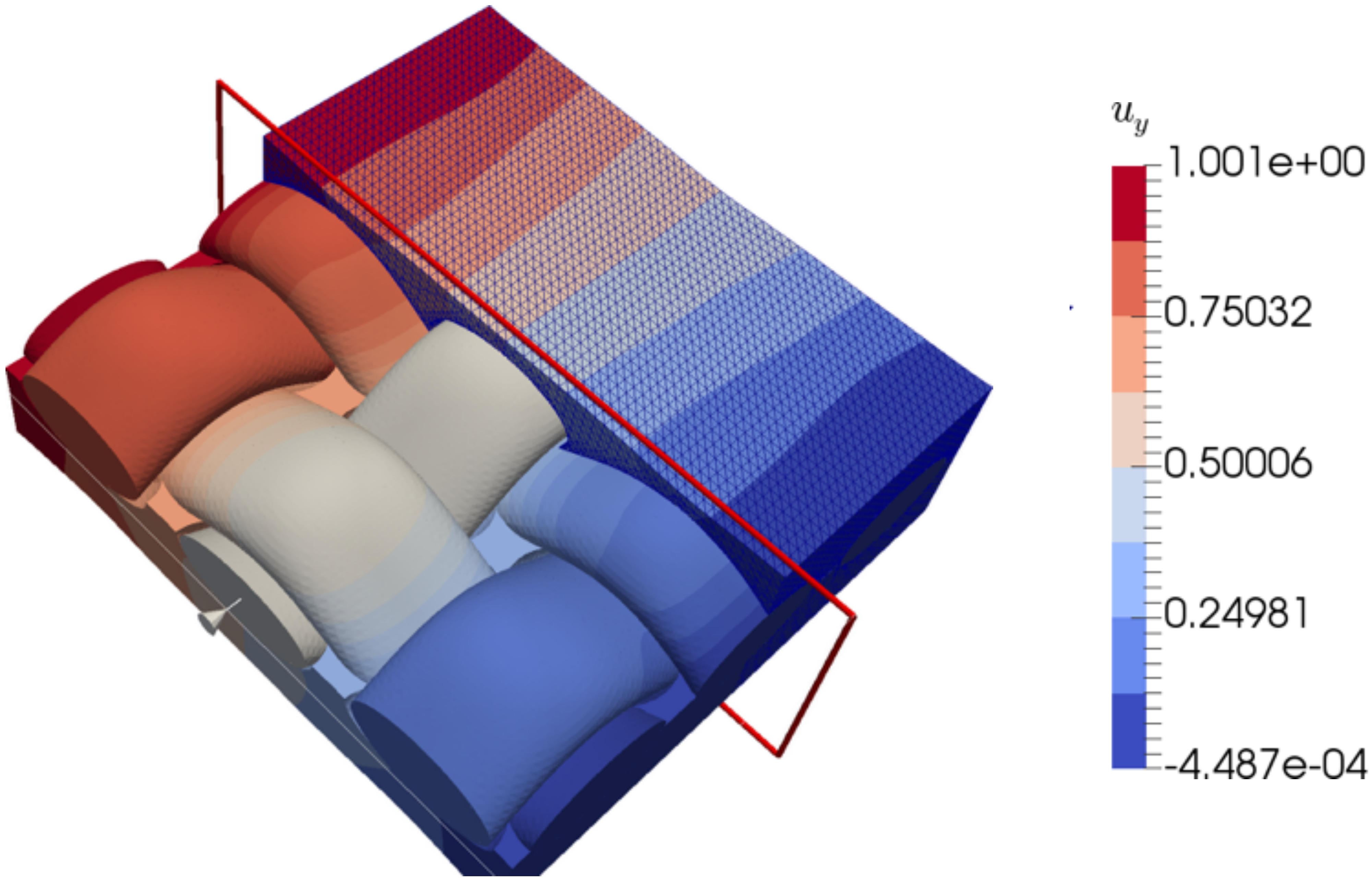


Applications in Engineering

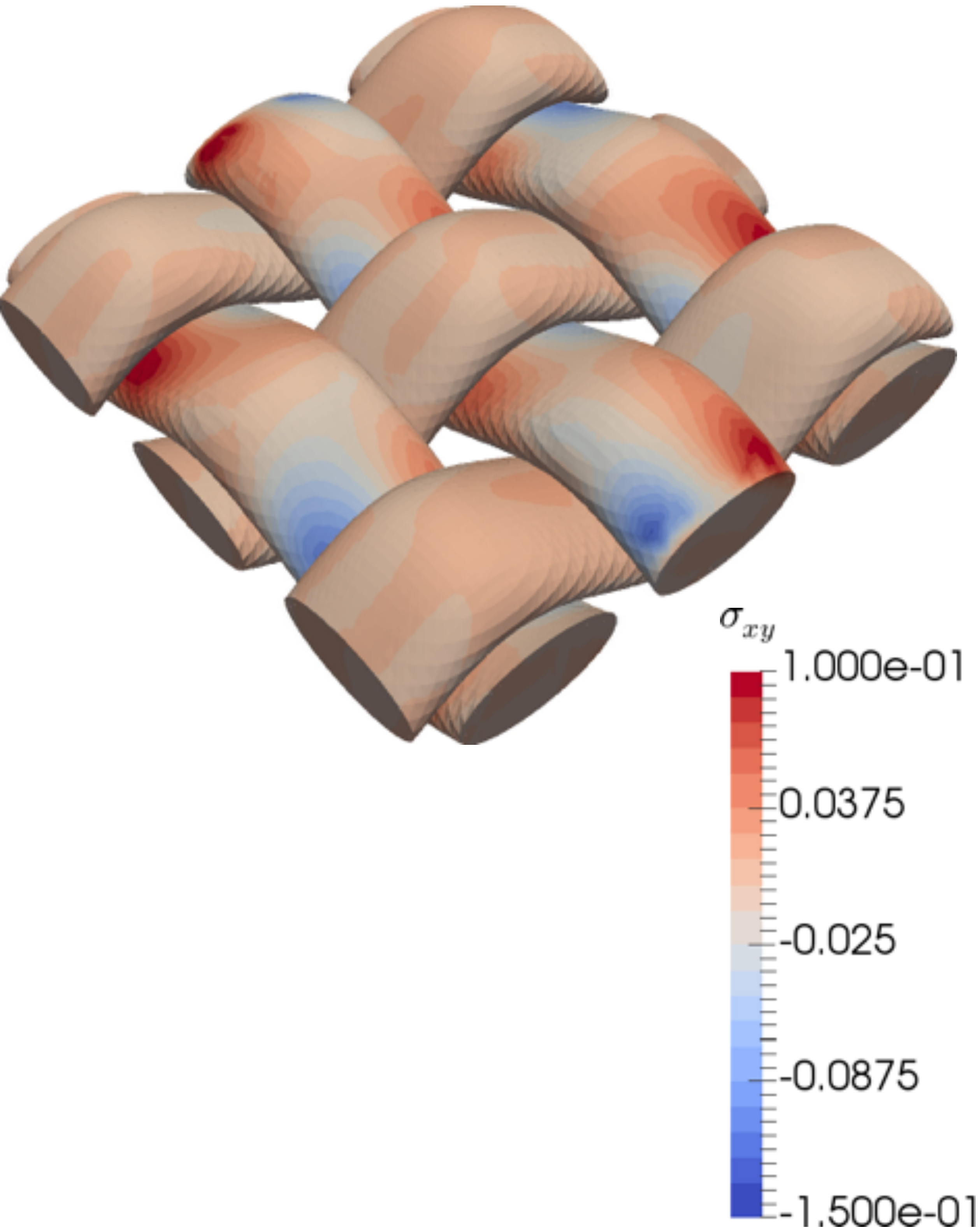
Braided Composite

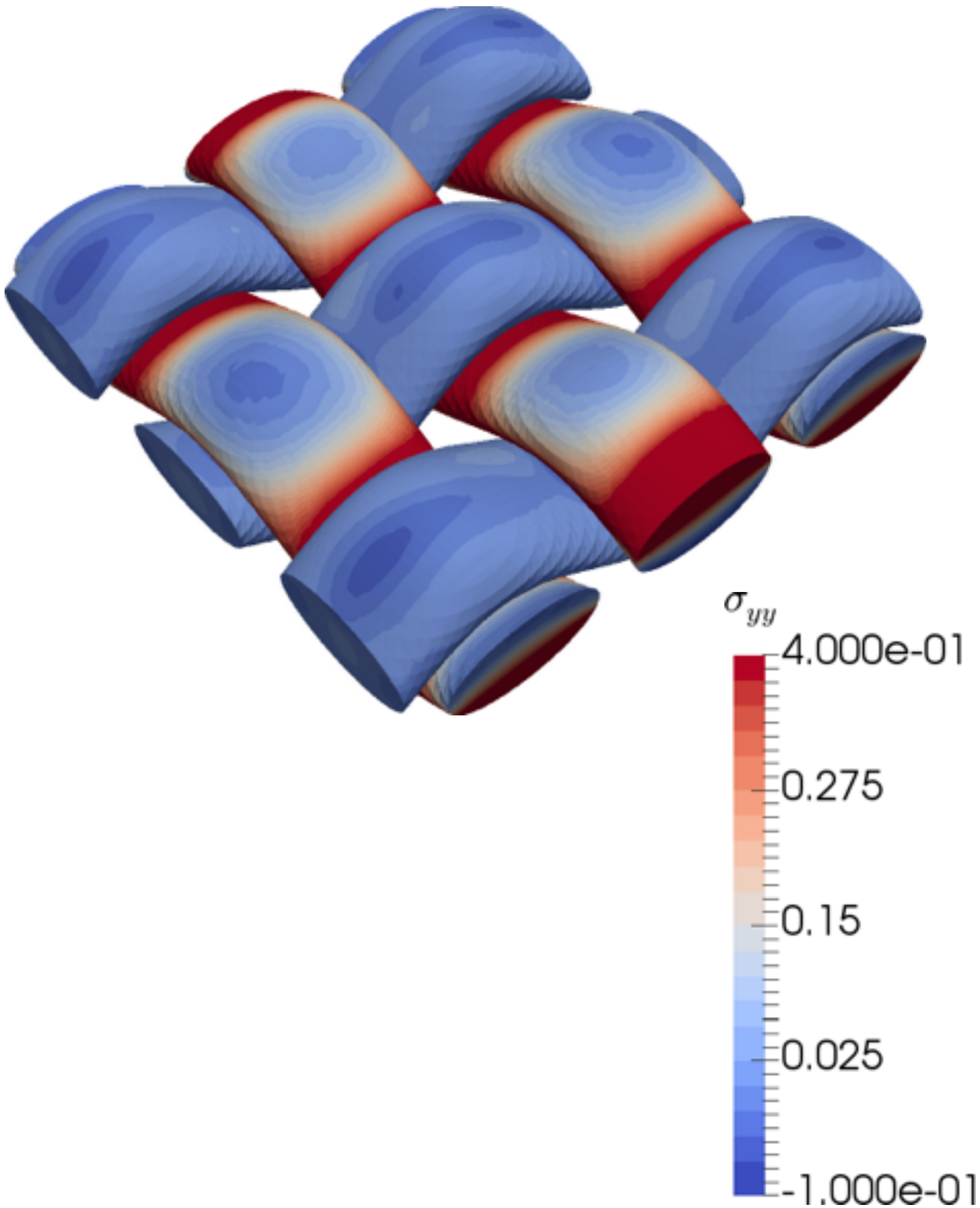


Braided Composite

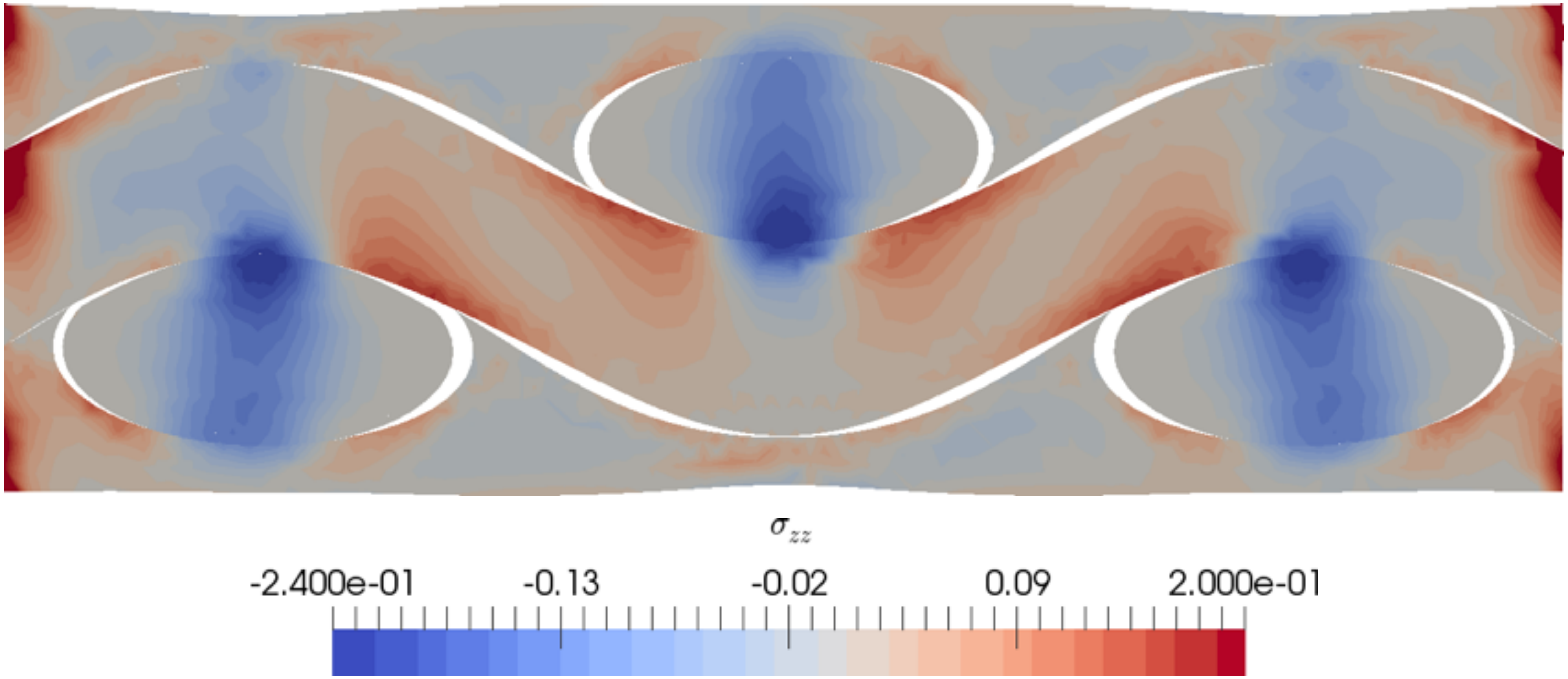


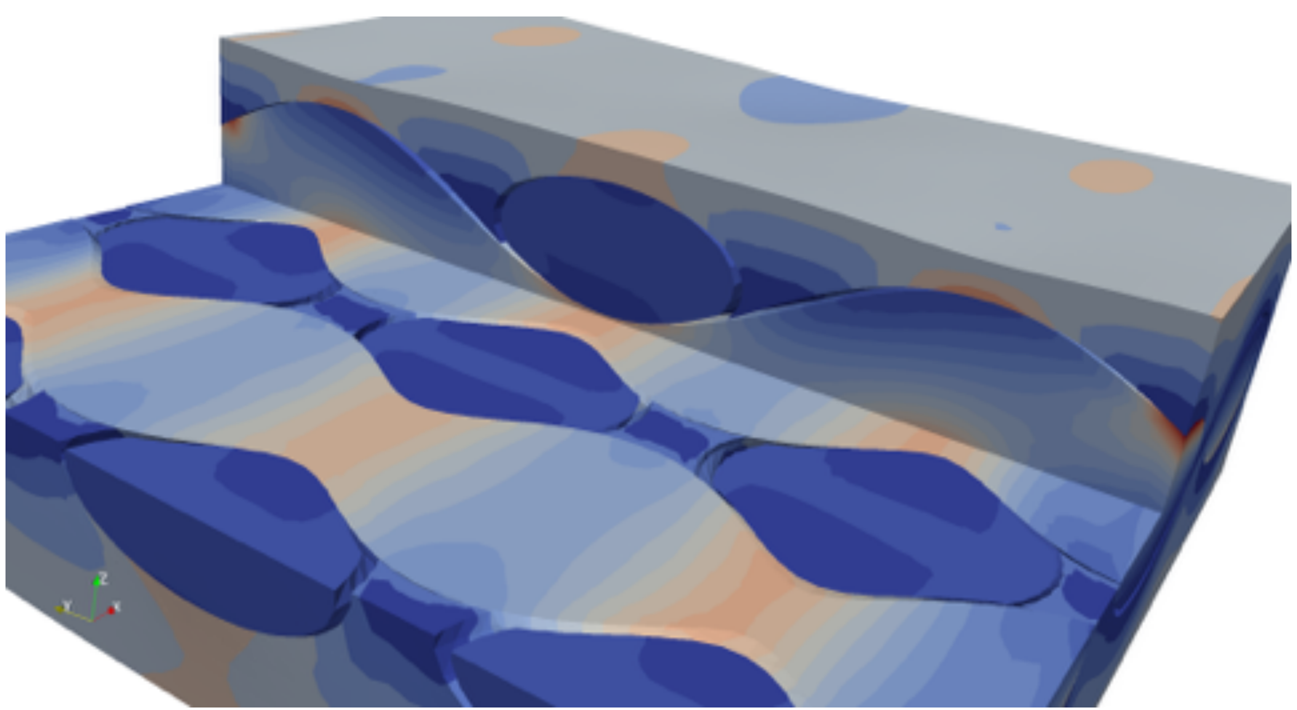
Braided Composite

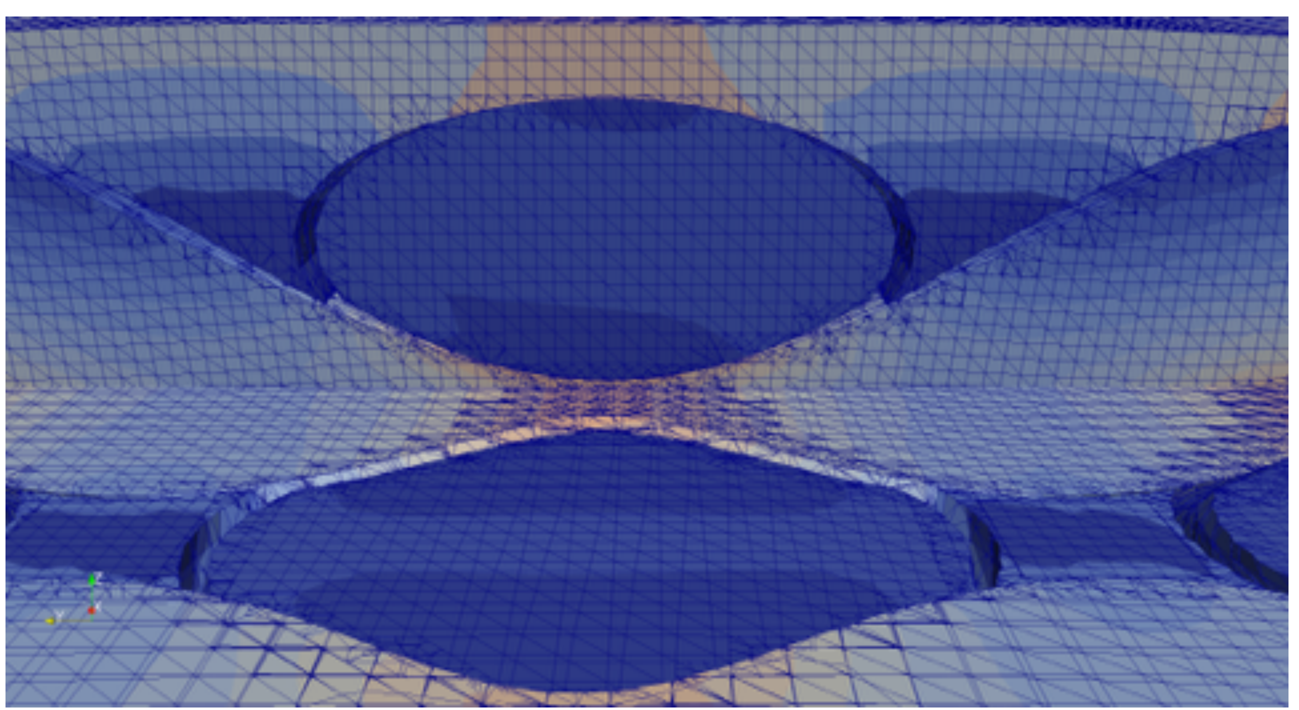




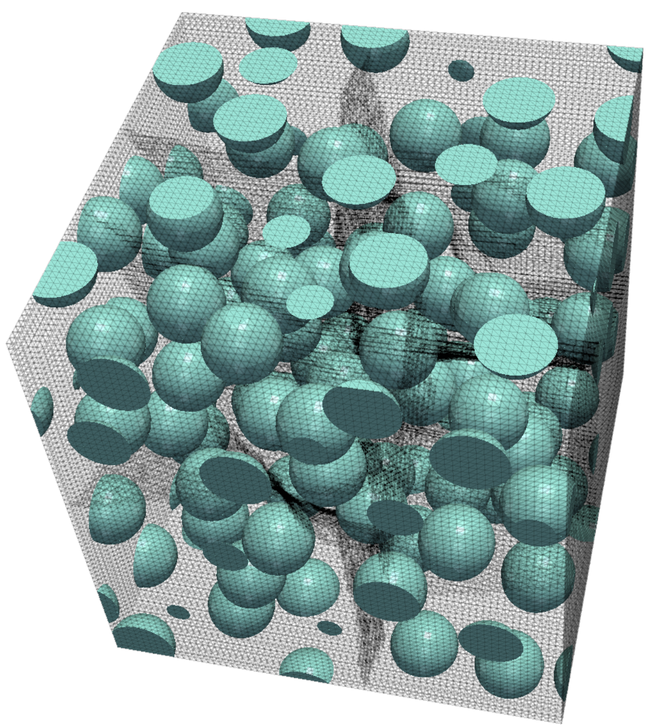
Braided Composite

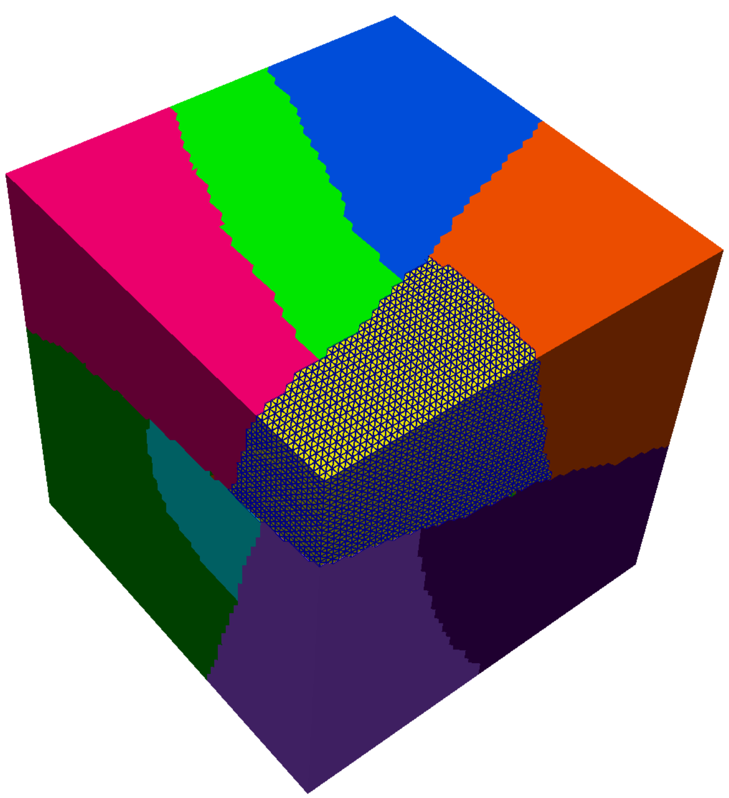




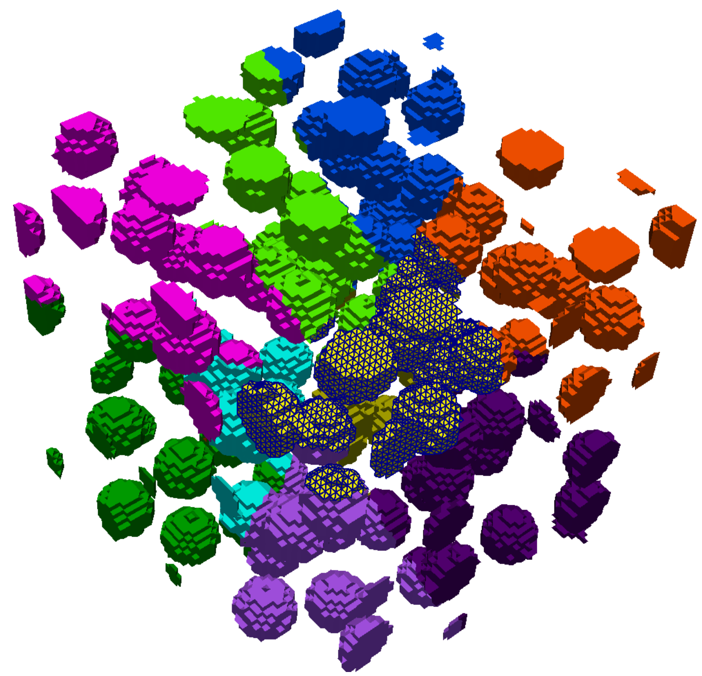


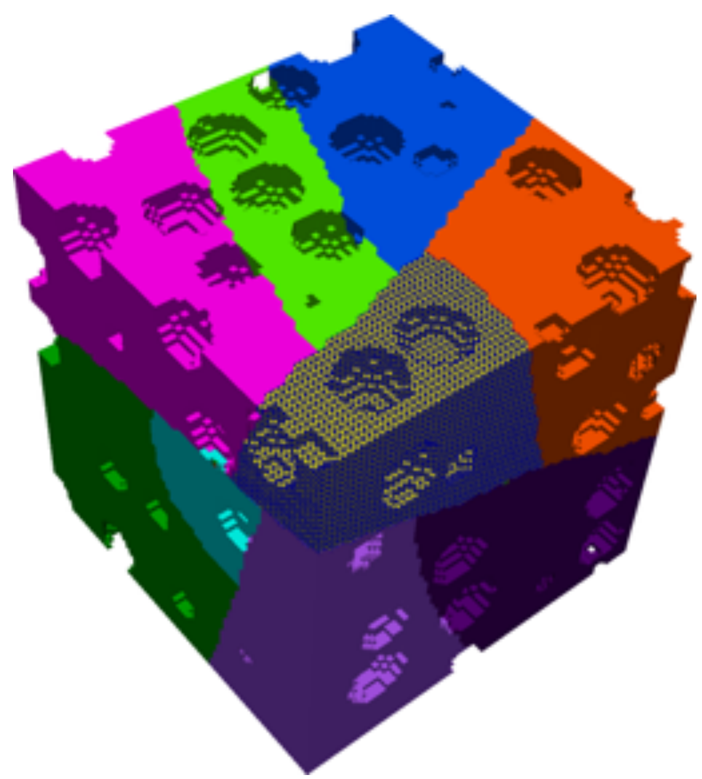
Damage in Concrete: Parallelisation

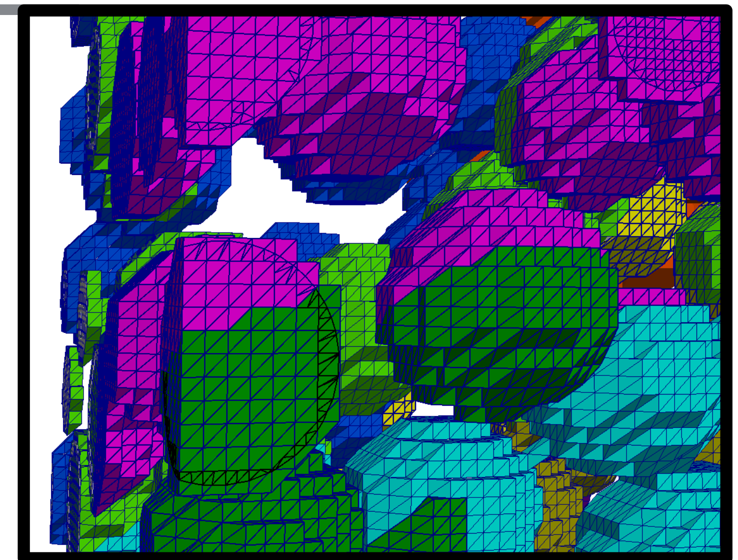




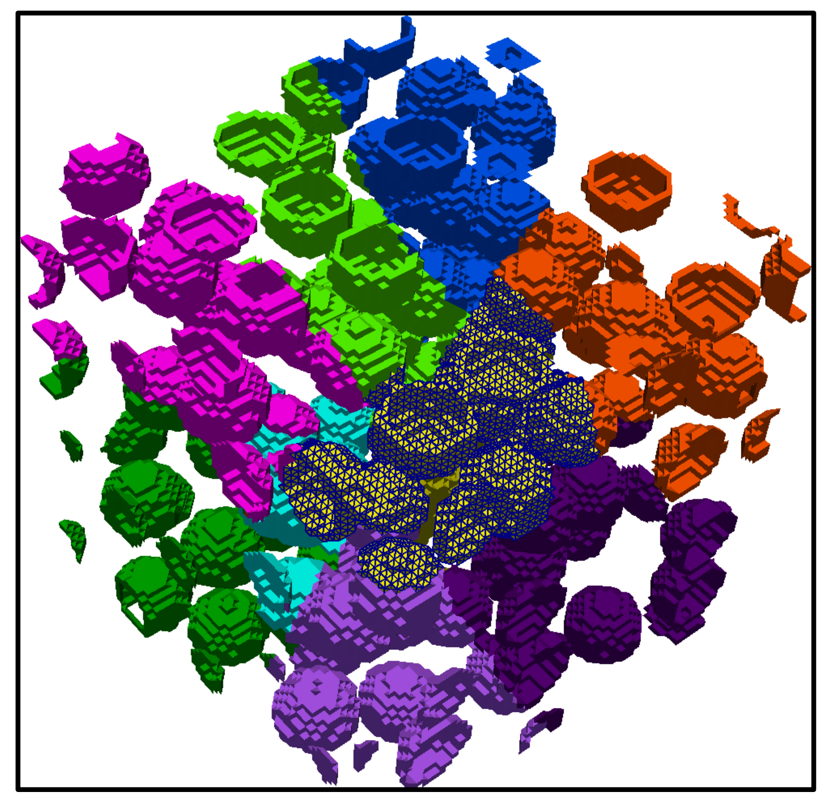
Linear Stage



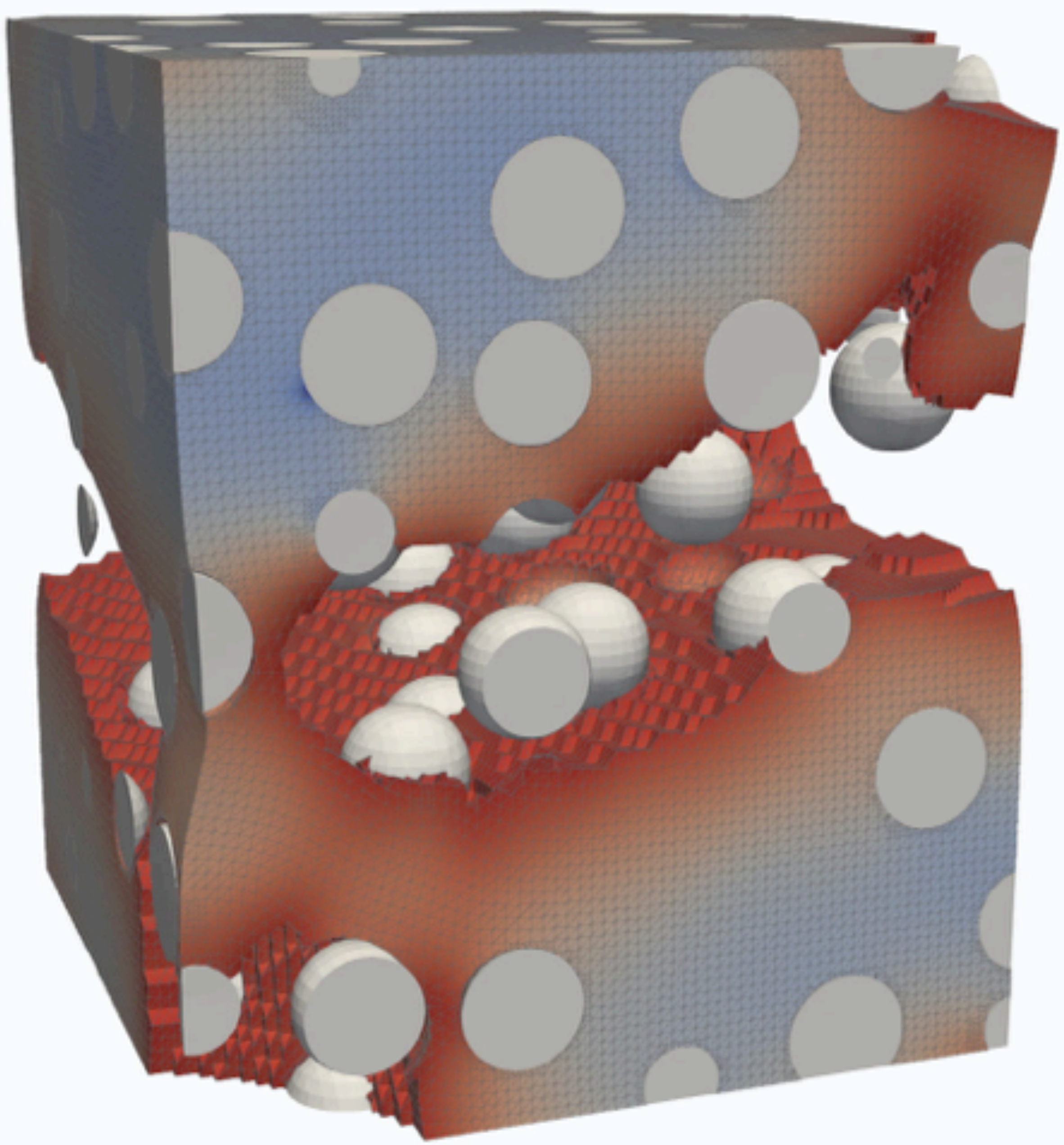


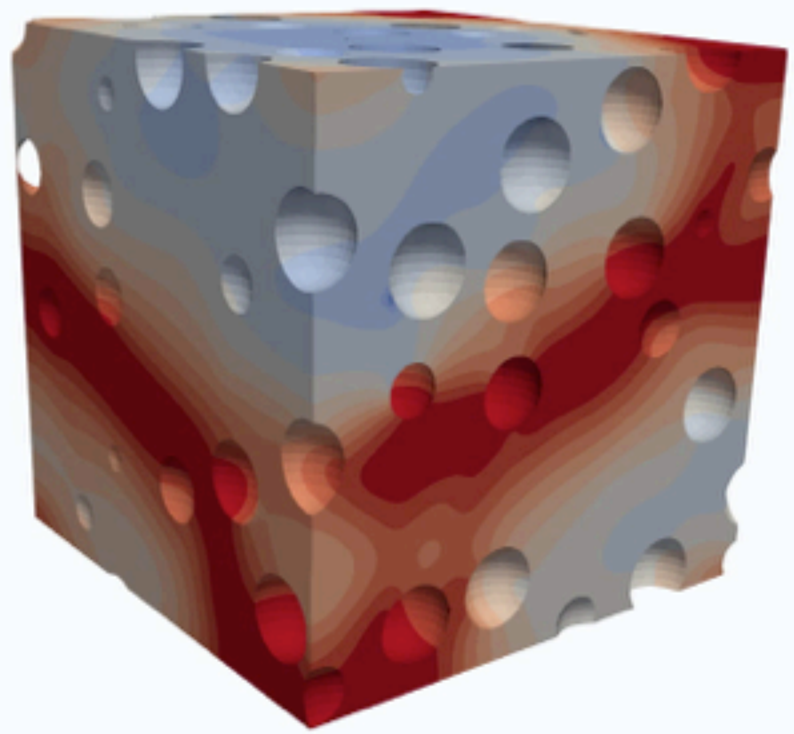


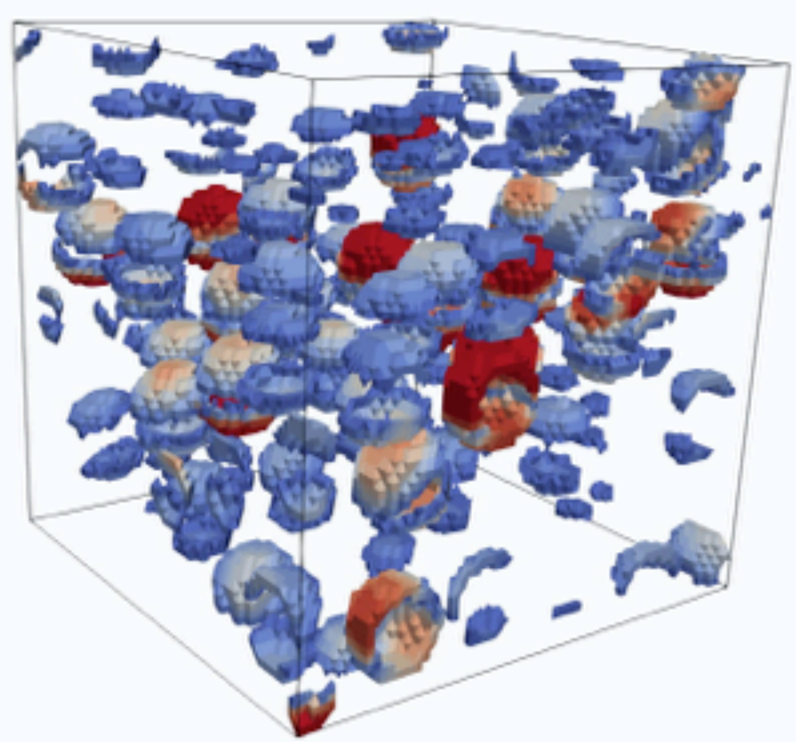
Local Stage



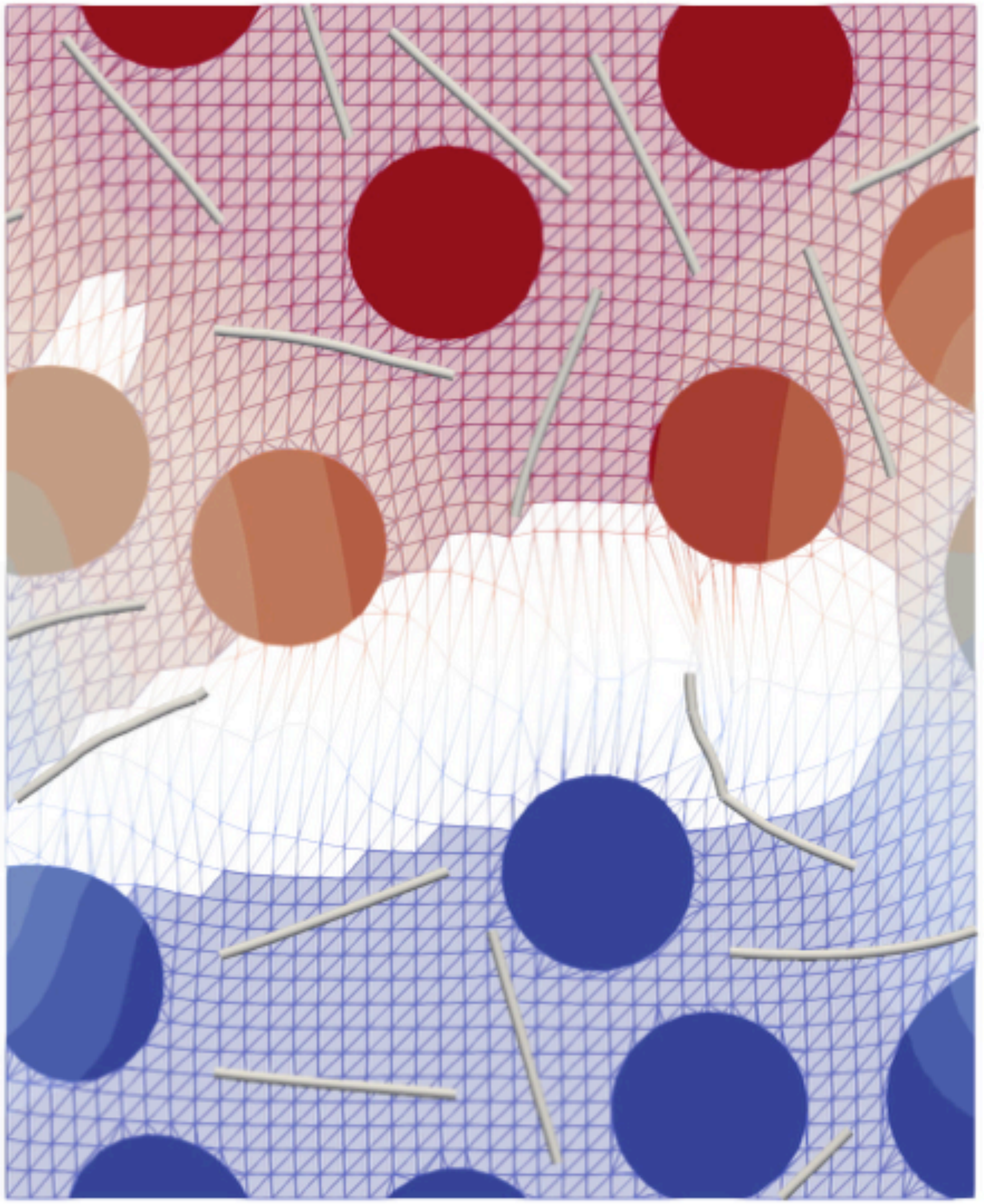
Damage in Concrete

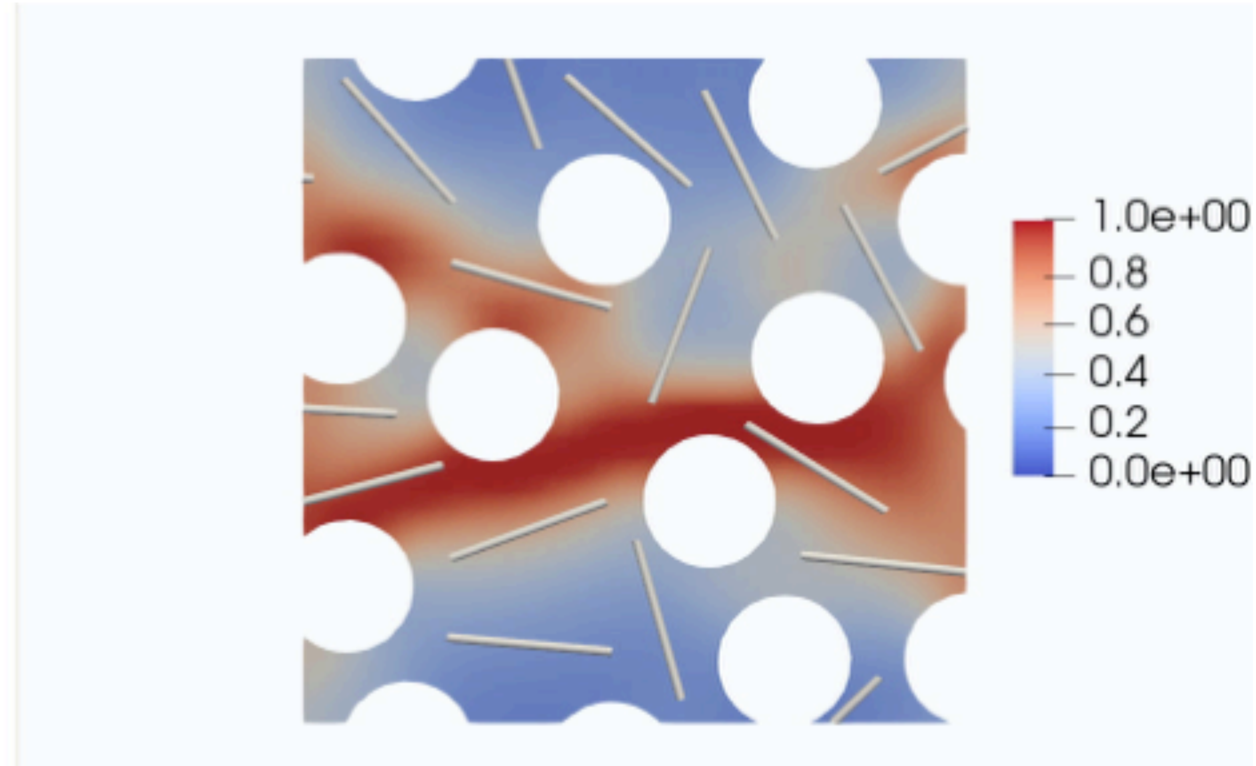


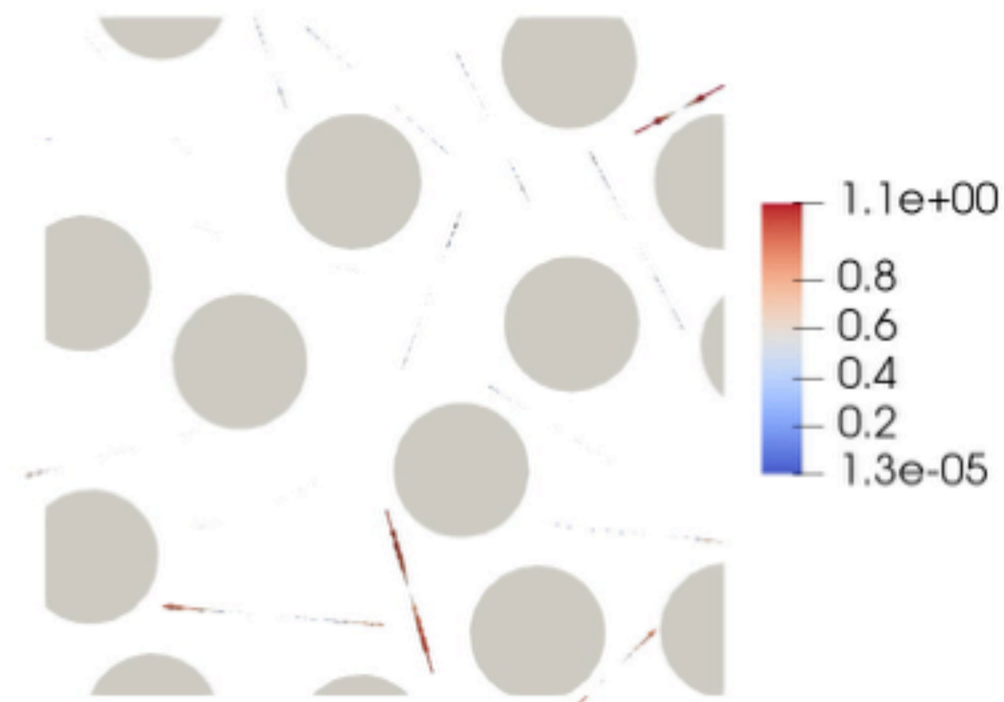




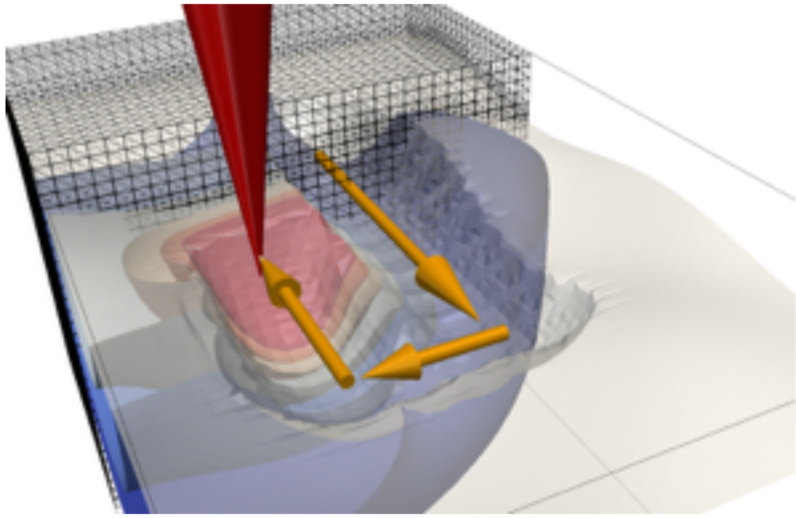
[redacted]

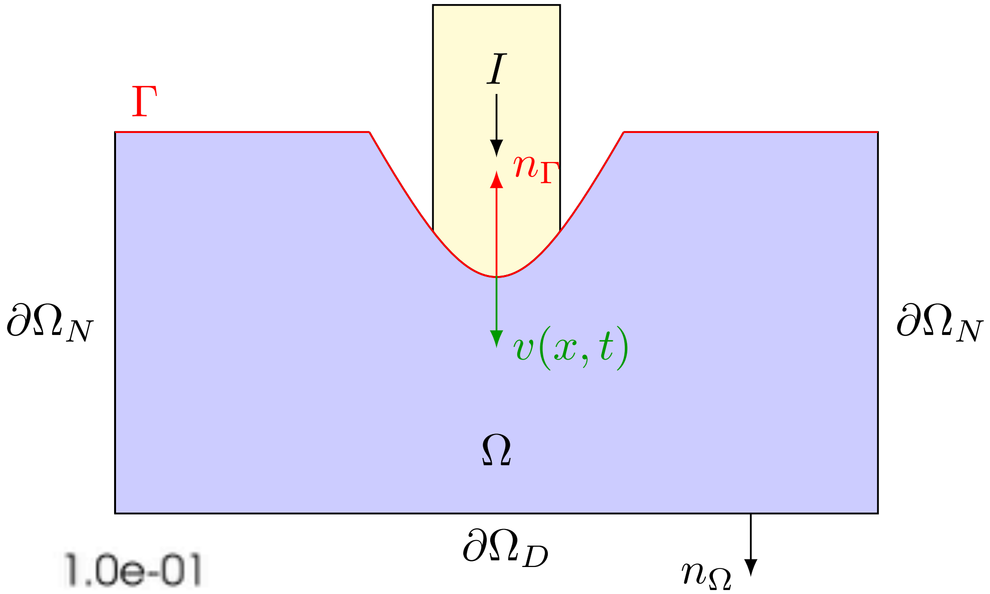


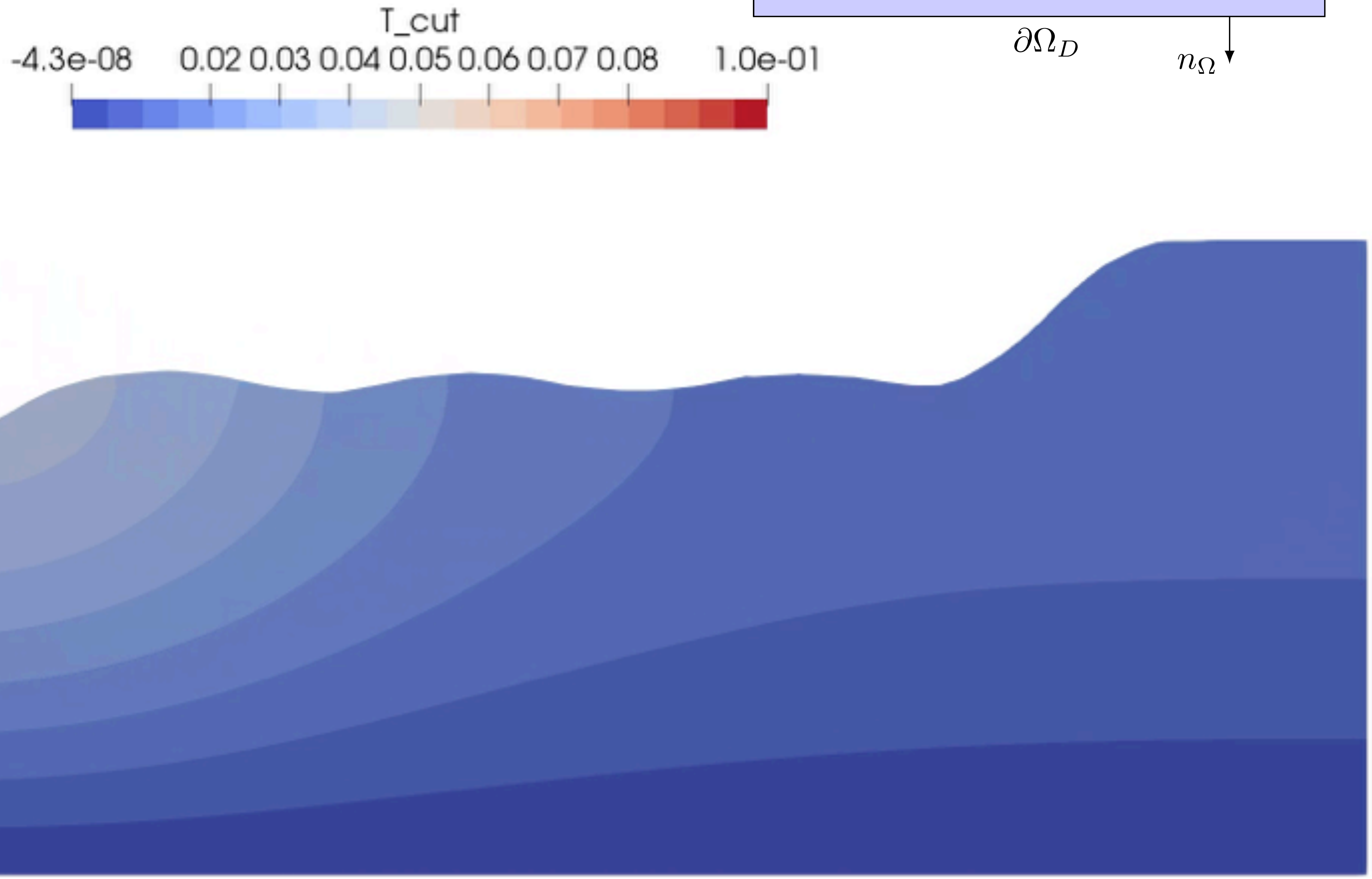




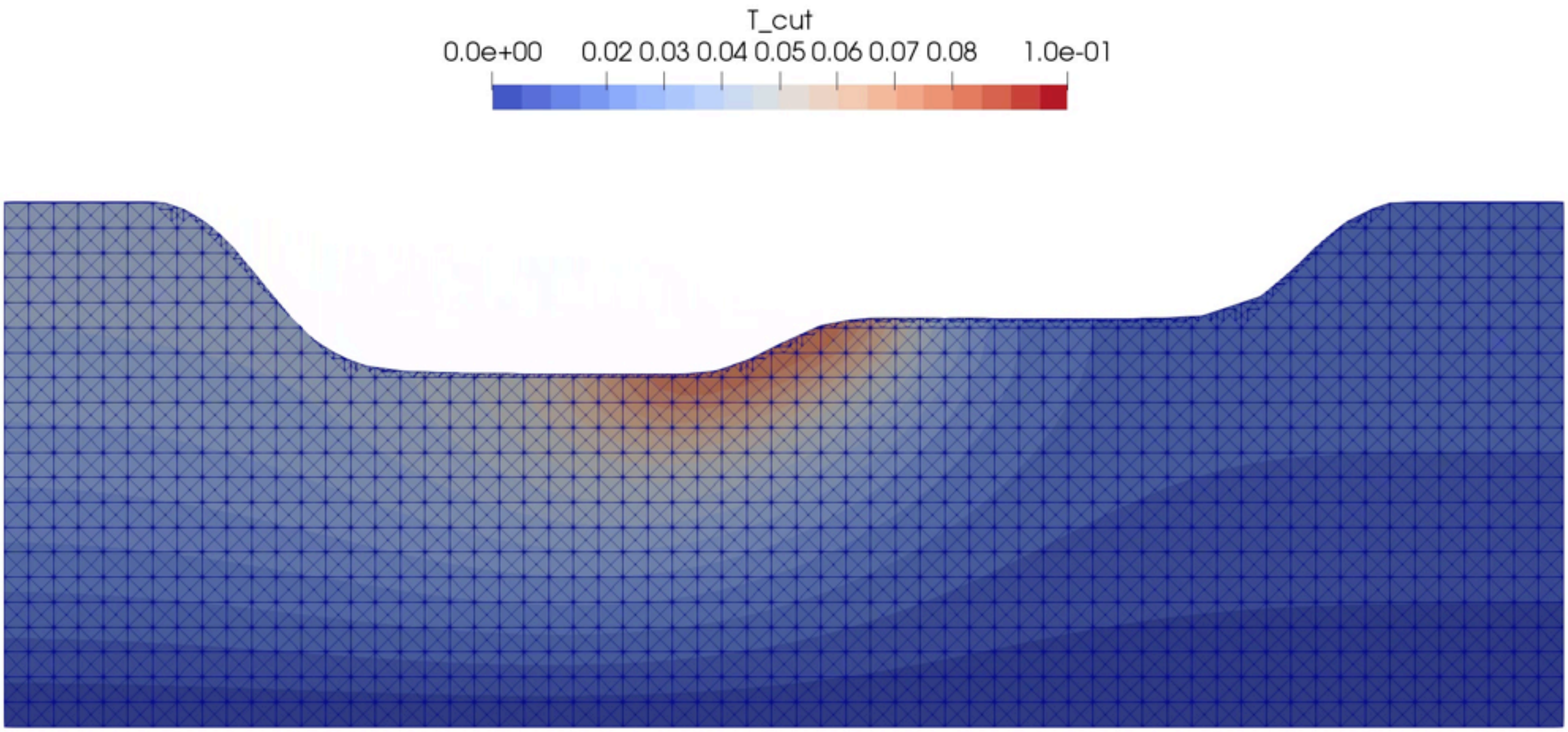
Pulsed Thermal Ablation



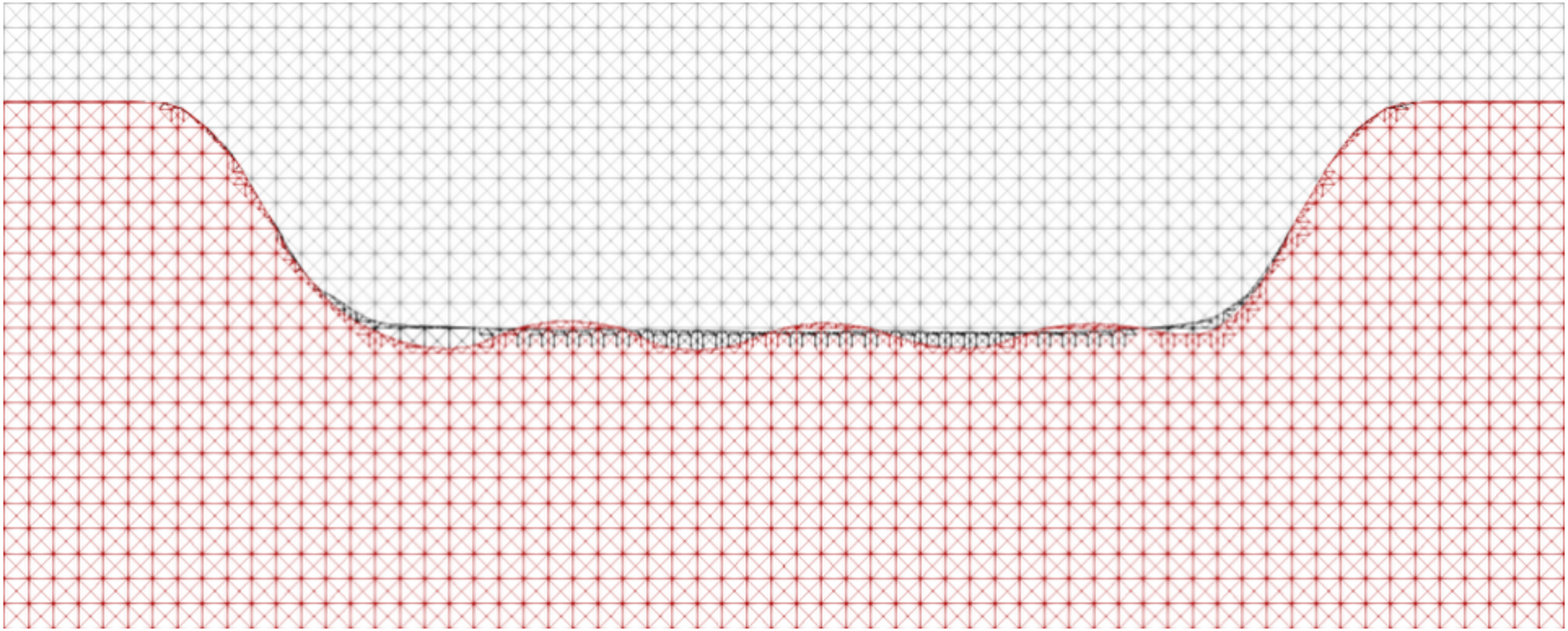




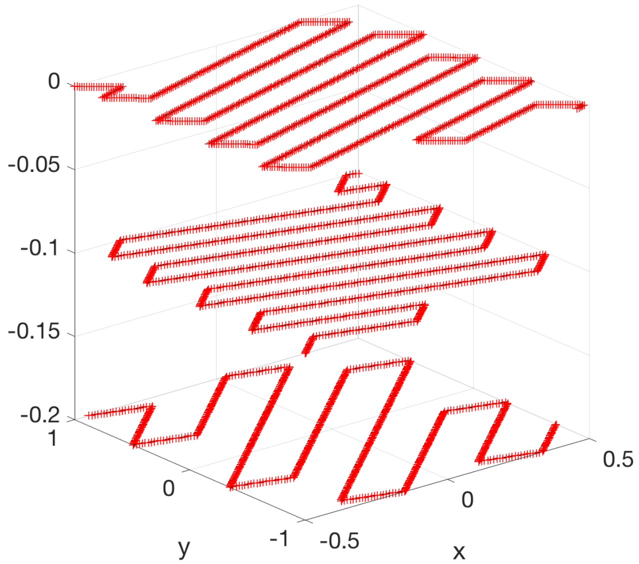
Pulsed Thermal Ablation

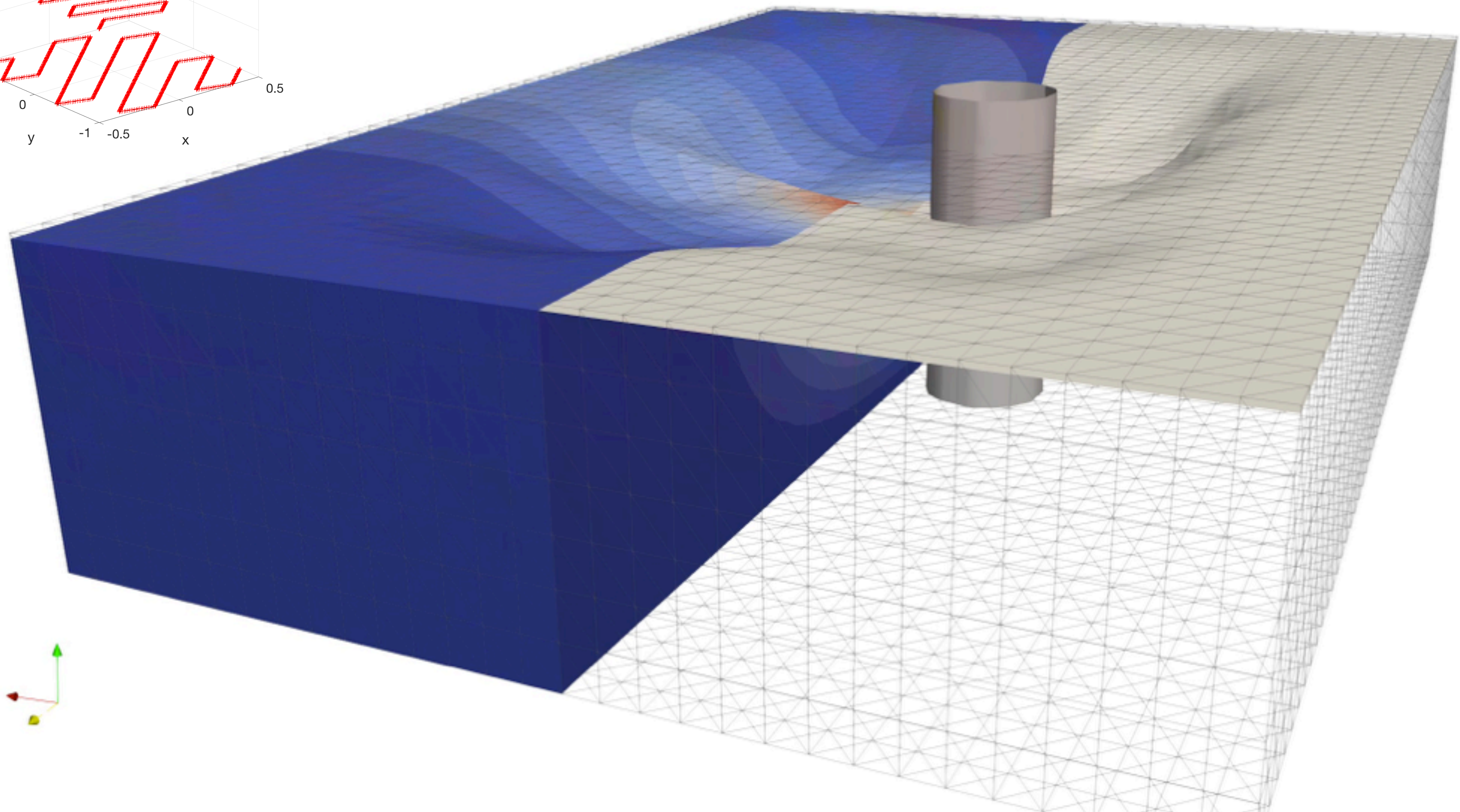


Pulsed Thermal Ablation



3D Machining Path





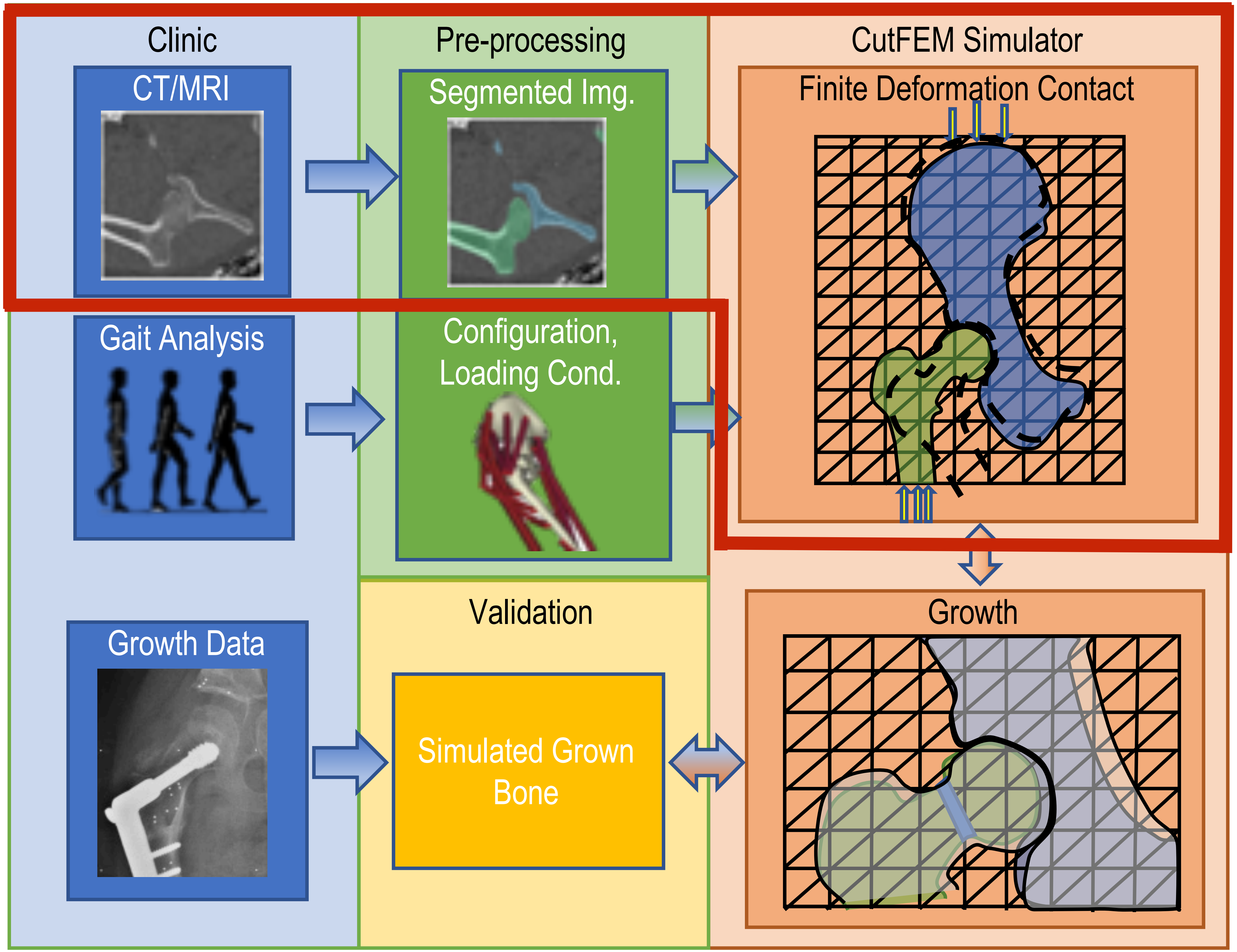
Applications in Biomechanics

Cut Finite Element Hip Modelling Motivation

Treatment options for hip malformations: (left) untreated hip deformity of a 4-year-old child, (middle) well-formed hip 8 years post guided growth surgery[1], i.e. insertion of one screw through the growth plate of the femur bones, (right) well-formed hip after an invasive femur and hip osteotomy, i.e. cutting through the bones and insertion of screws and plates.

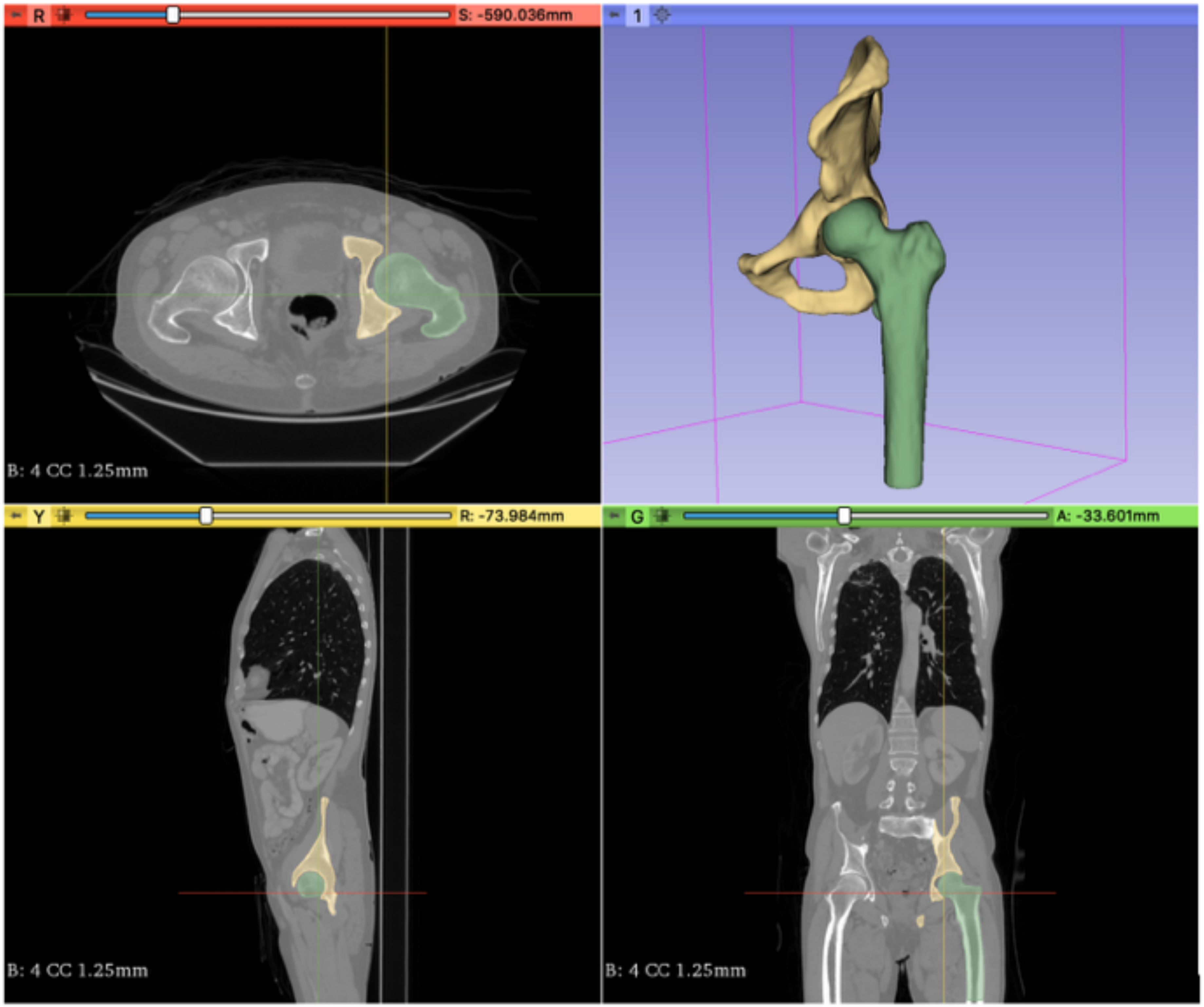
Study stress in hip joint using FE Modelling to enhance understanding of bone growth and bone shape changes

[1] Lee, W-C et al. "Guided growth of the proximal femur for hip displacement in children with cerebral palsy." *Journal of Pediatric Orthopaedics* 36.5, 511-515, (2016).

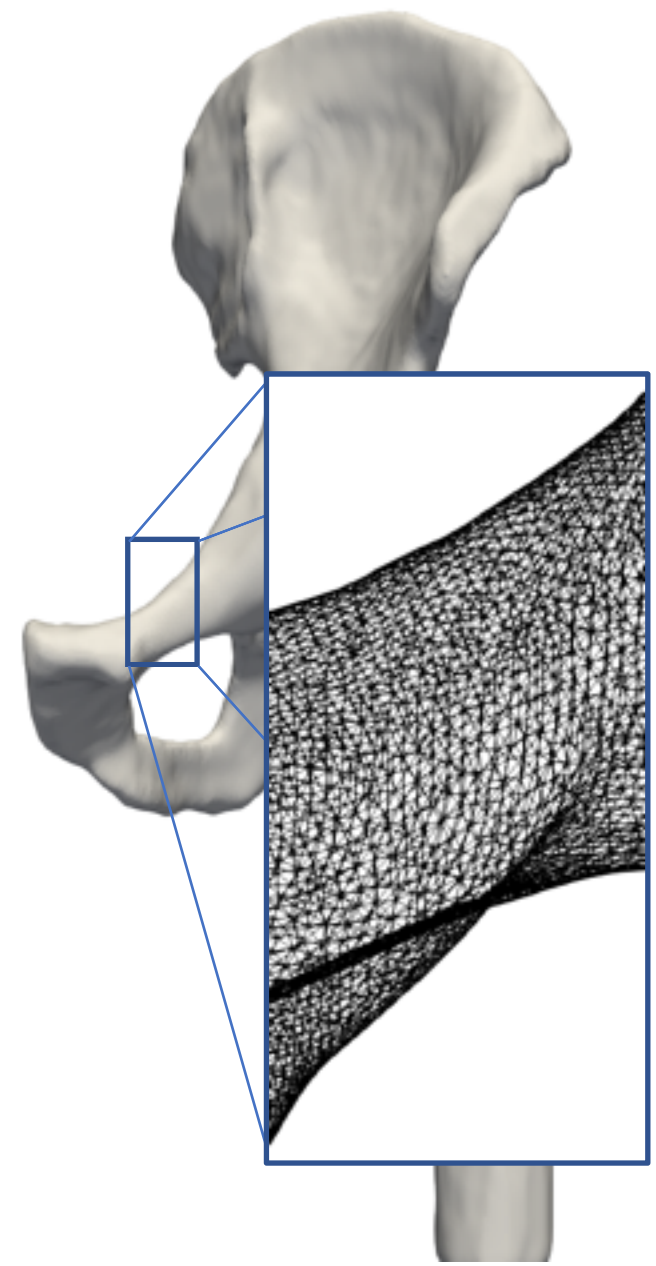

Clinic
CT/MRI
Gait Analysis
Growth Data
Pre-processing
Segmented Img.
Configuration,
Loading Cond.
Validation
Simulated Grown
Bone
CutFEM Simulator
Finite Deformation Contact
Growth

Surface Triangulation to CutFEM pipeline

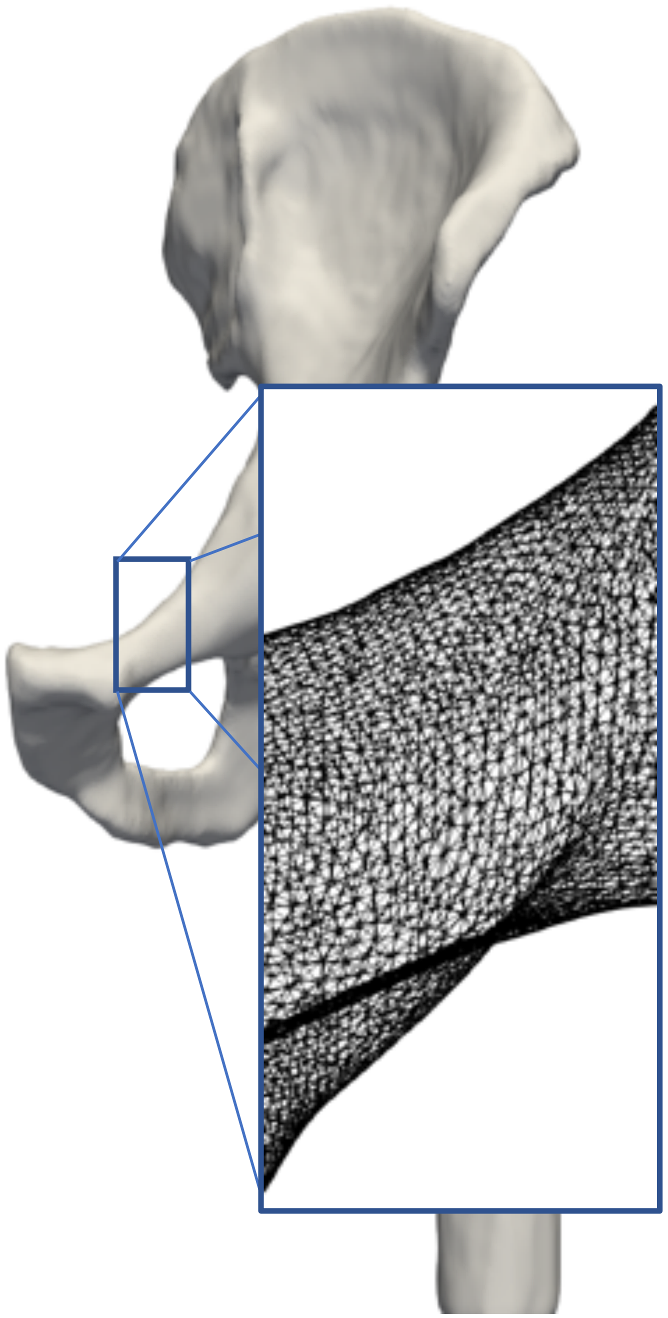
Segmentation with 3D Slicer (Faezeh Moshfeghifar) of CT-image from the cancer imaging archive (TCIA)

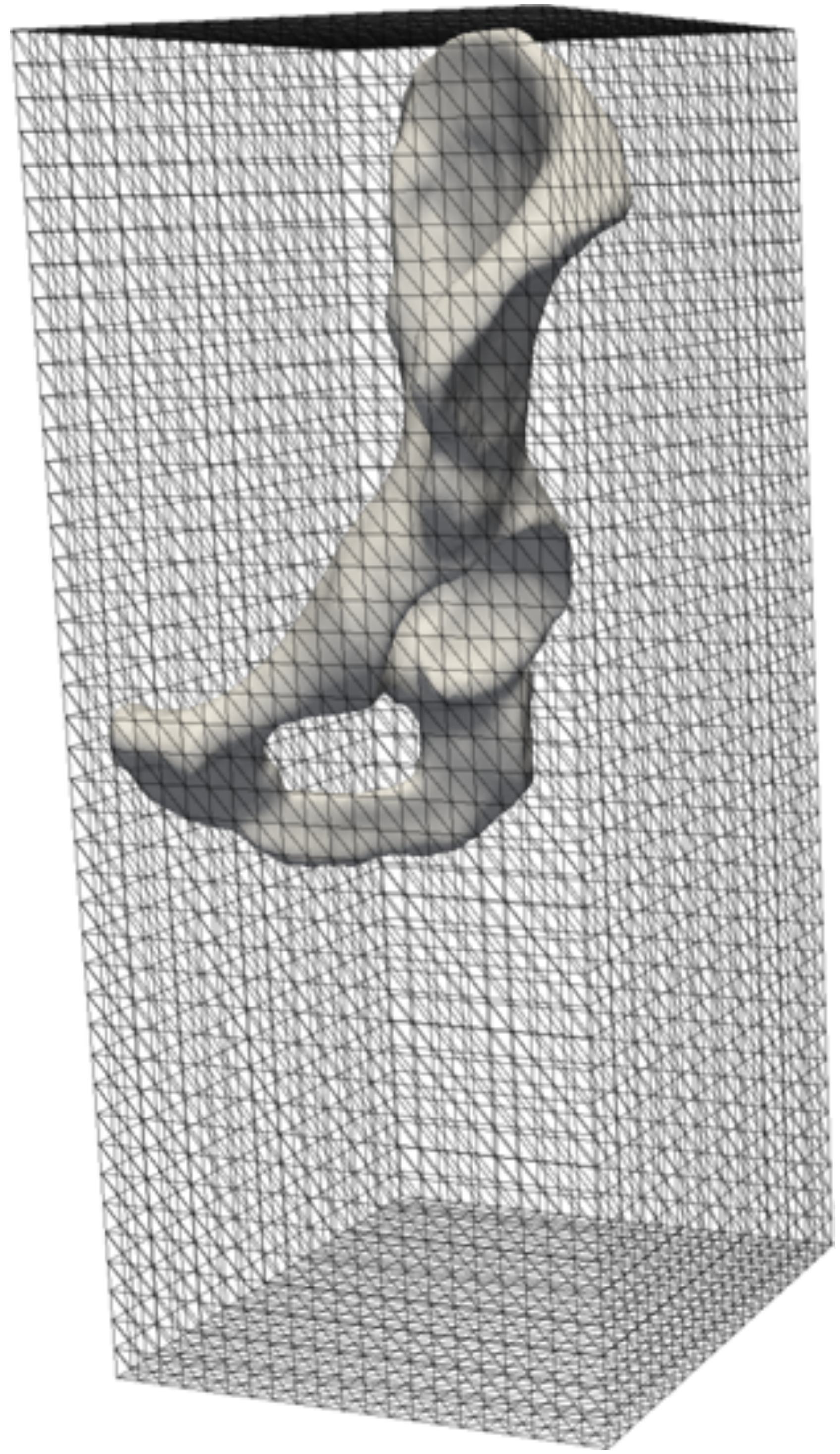


Surface triangulation

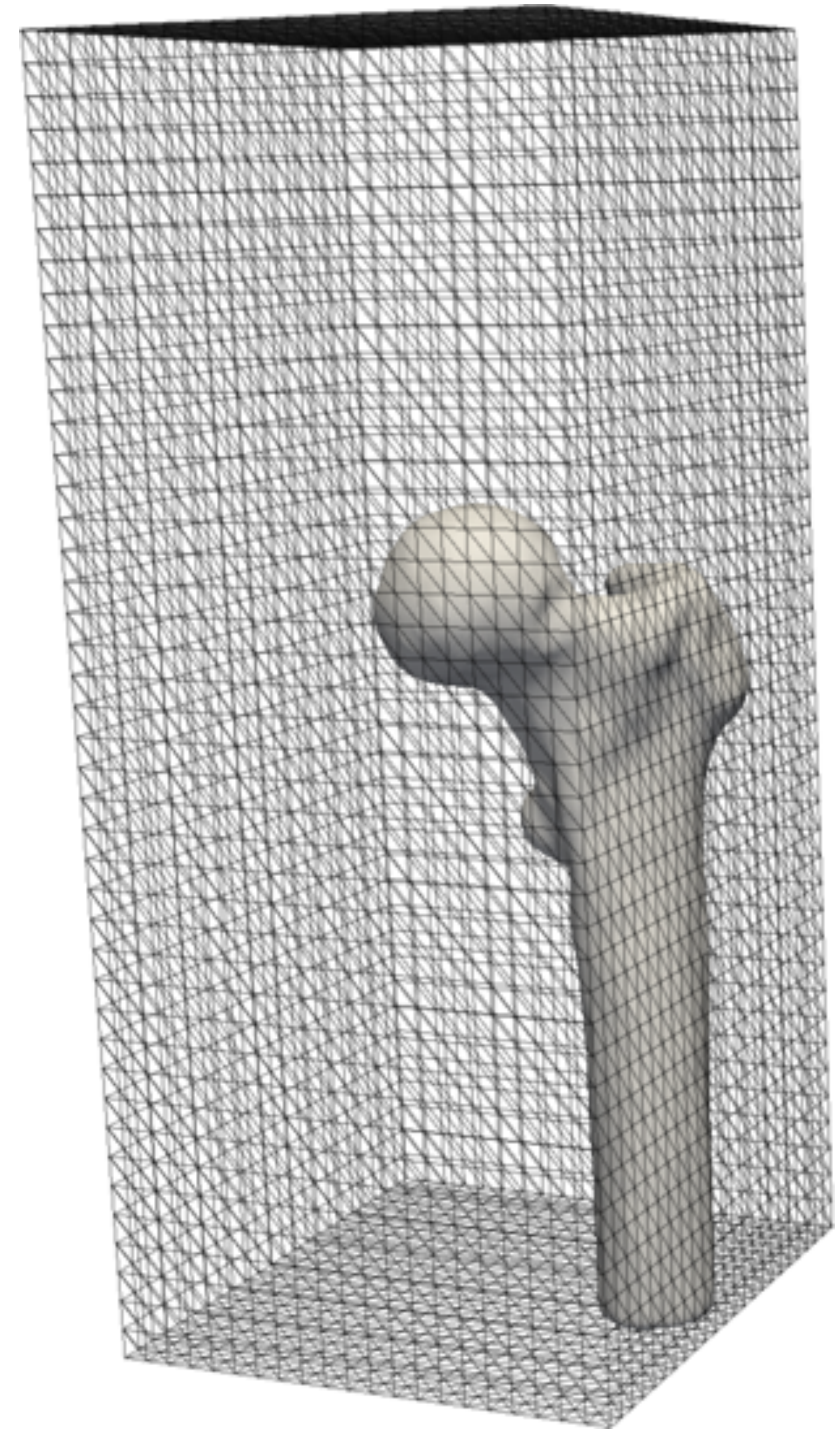


Create Regular Background mesh



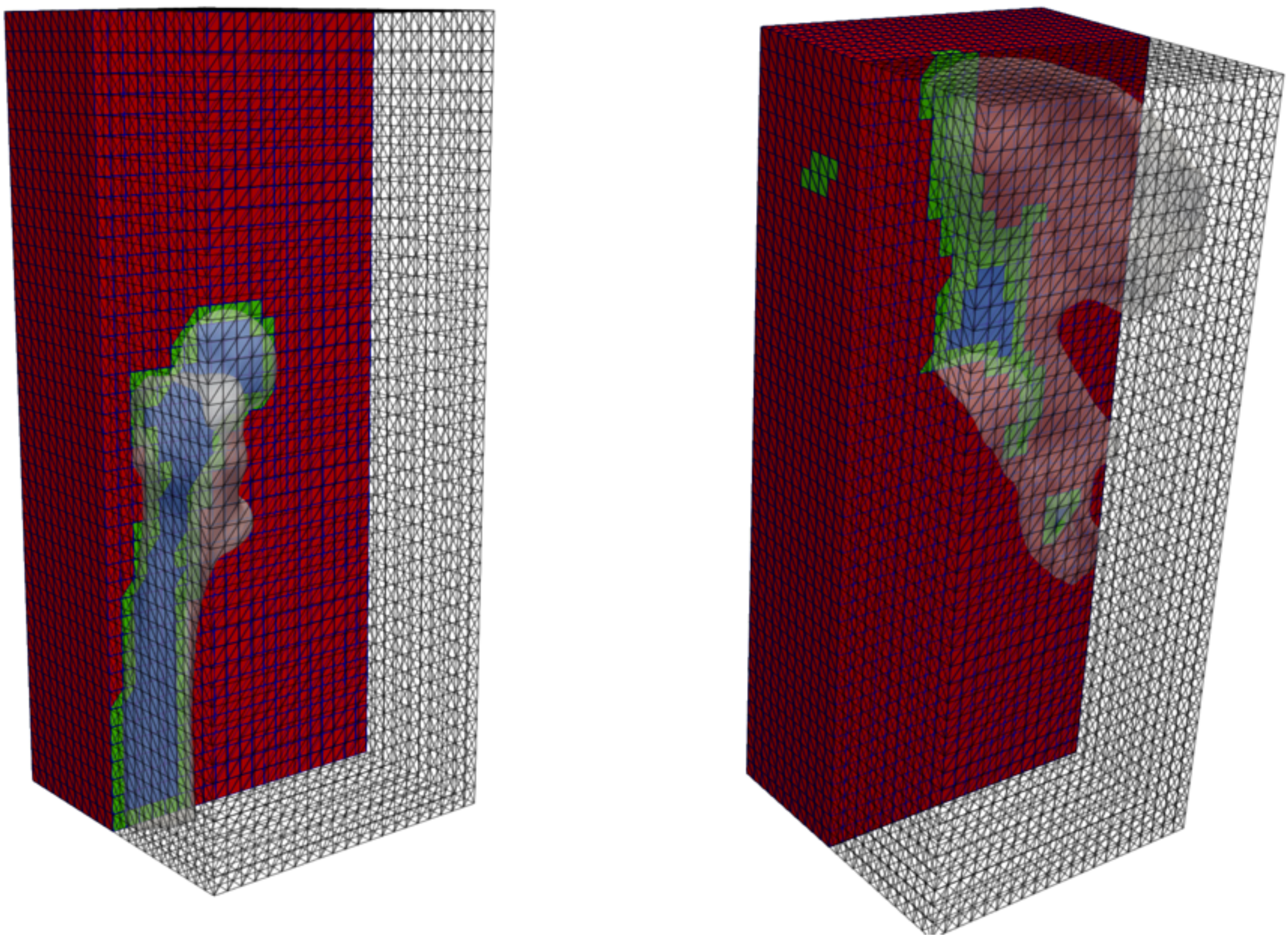


Hip bone surface triangulation embedded in regular background mesh



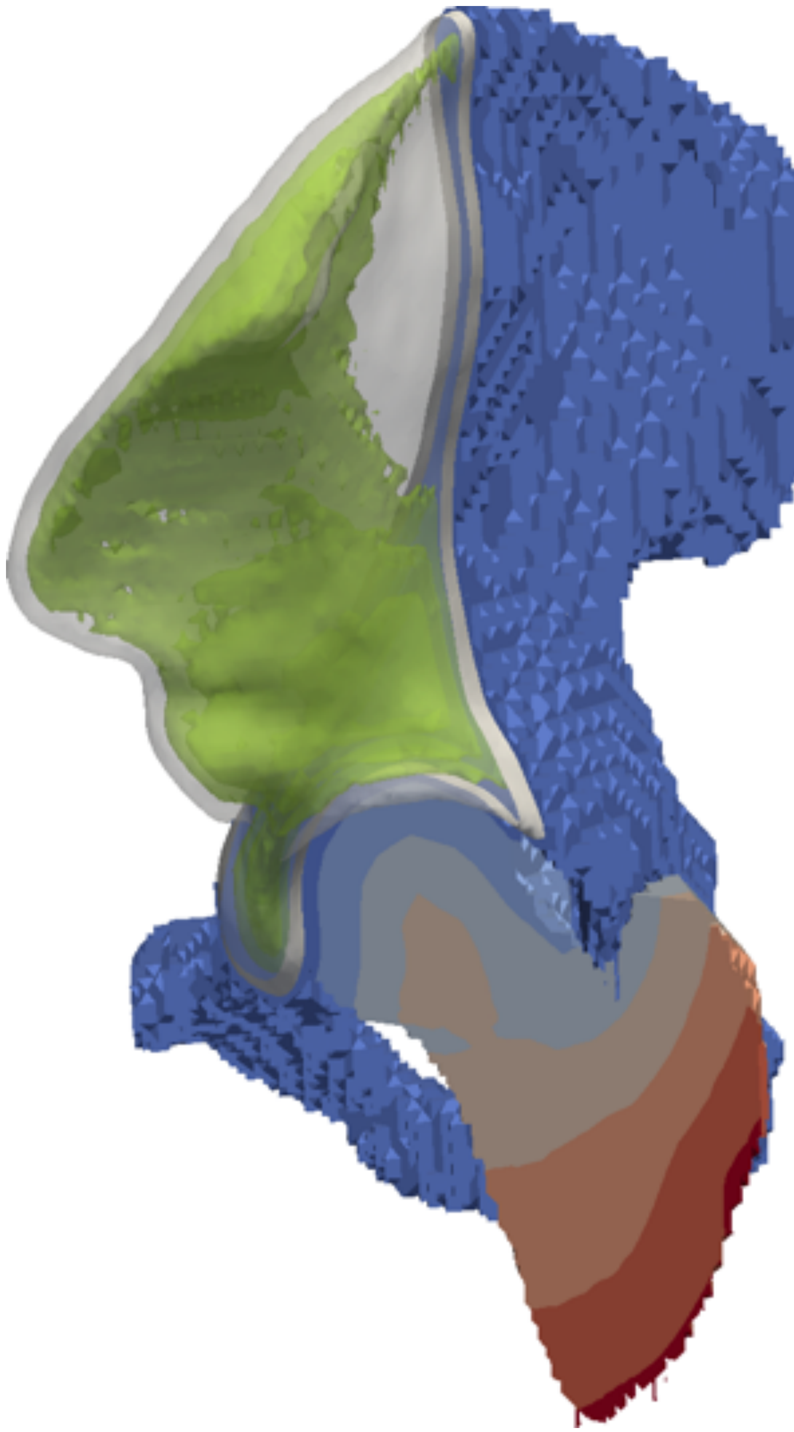
Femur bone surface triangulation embedded in regular background mesh

Determine inside, outside and intersected cells

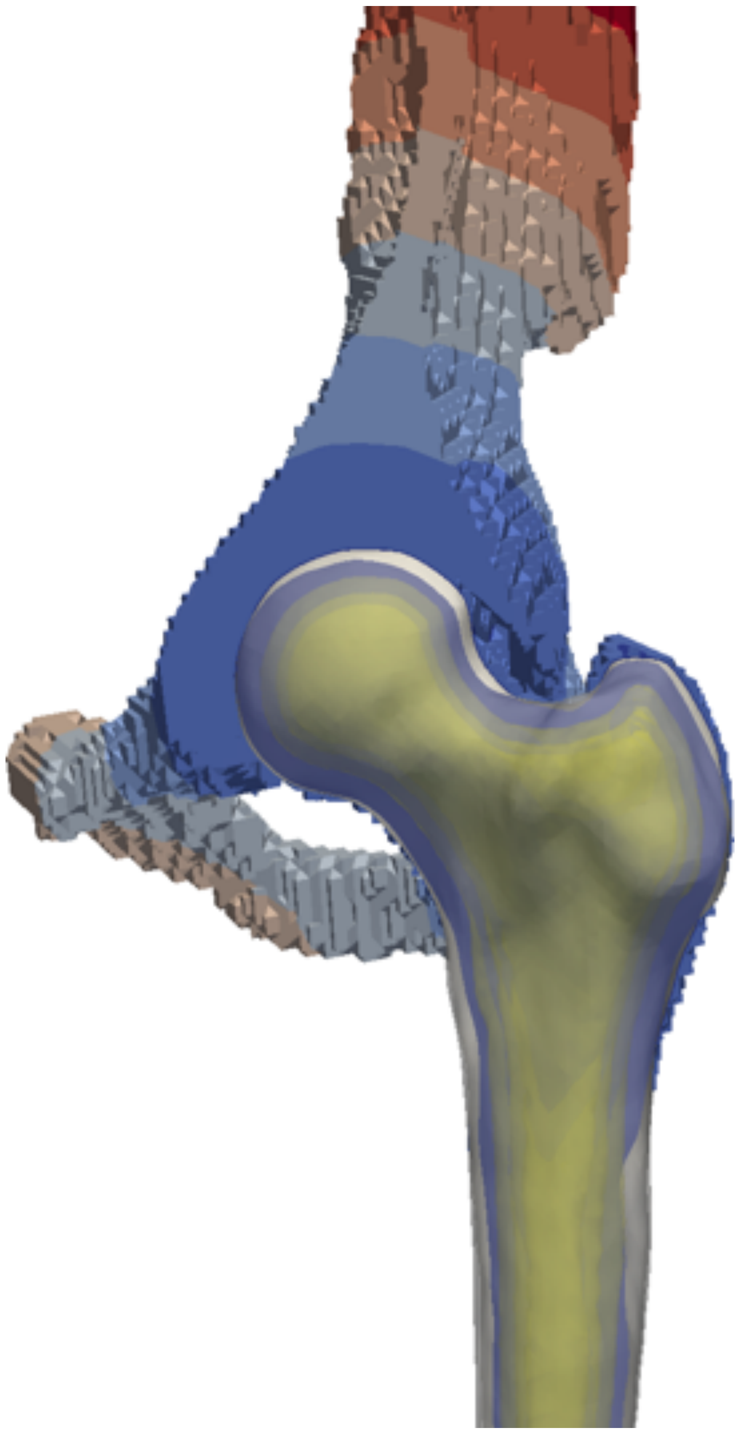


Compute signed distance function for each bone

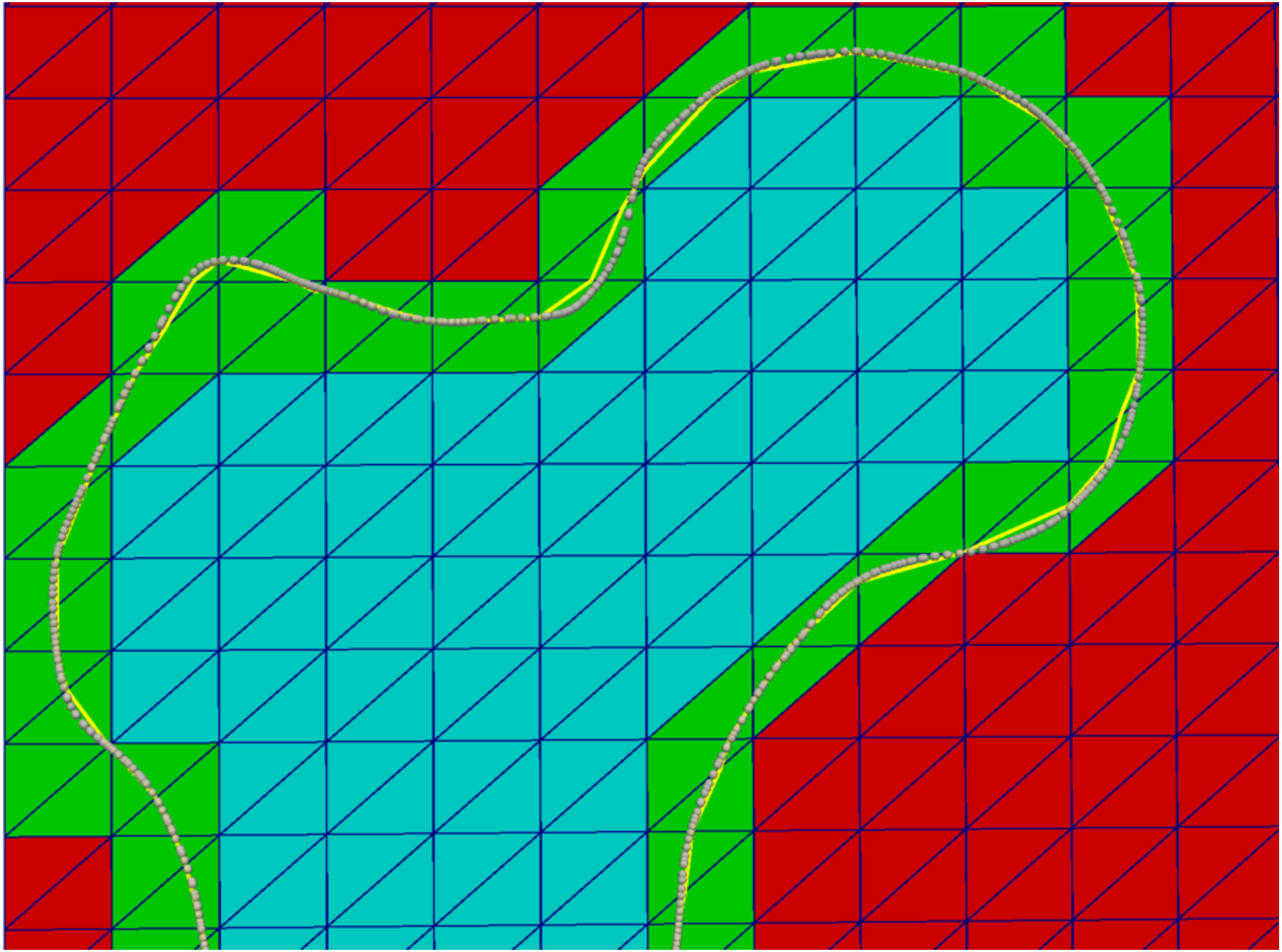
ϕ_1



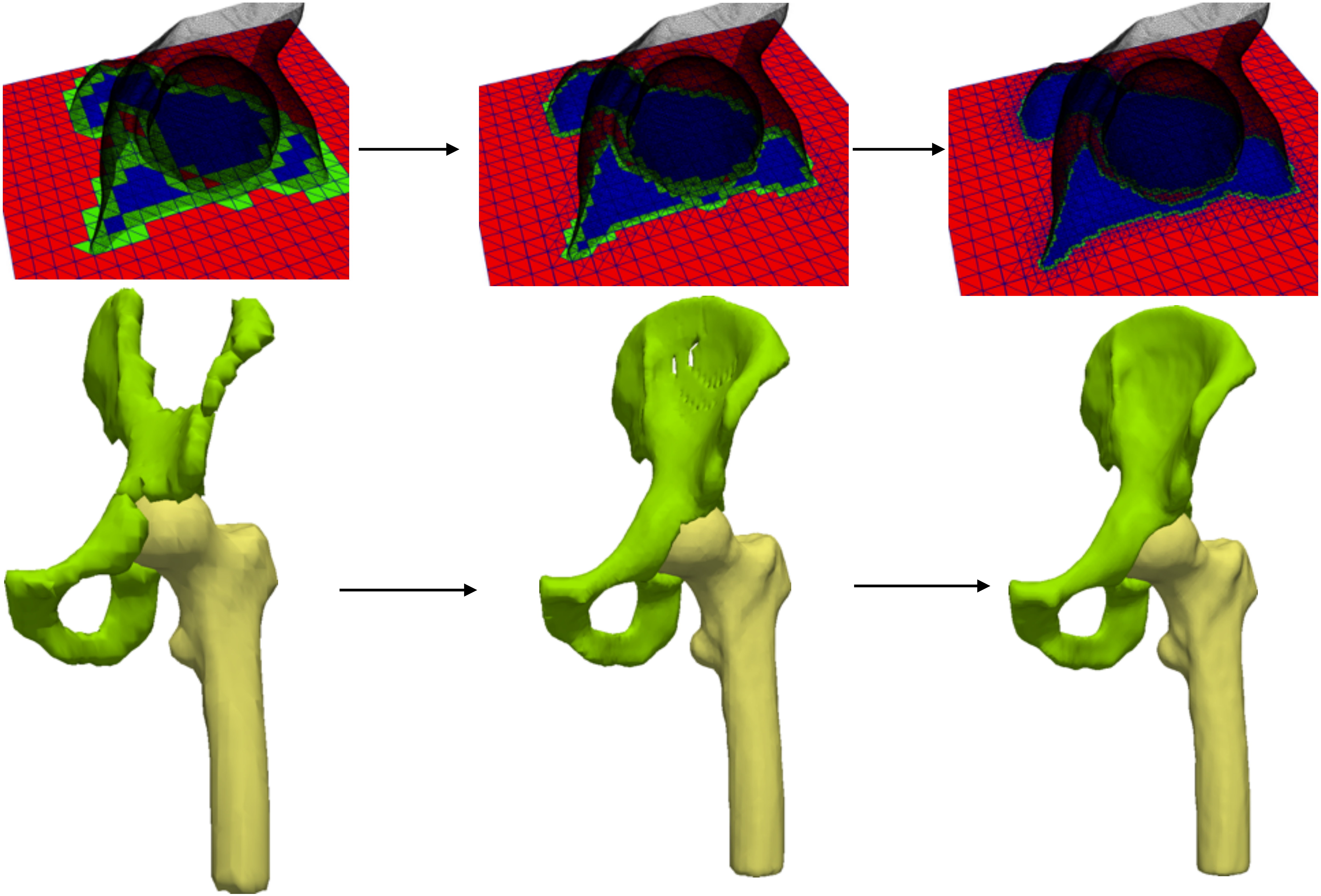
ϕ_2



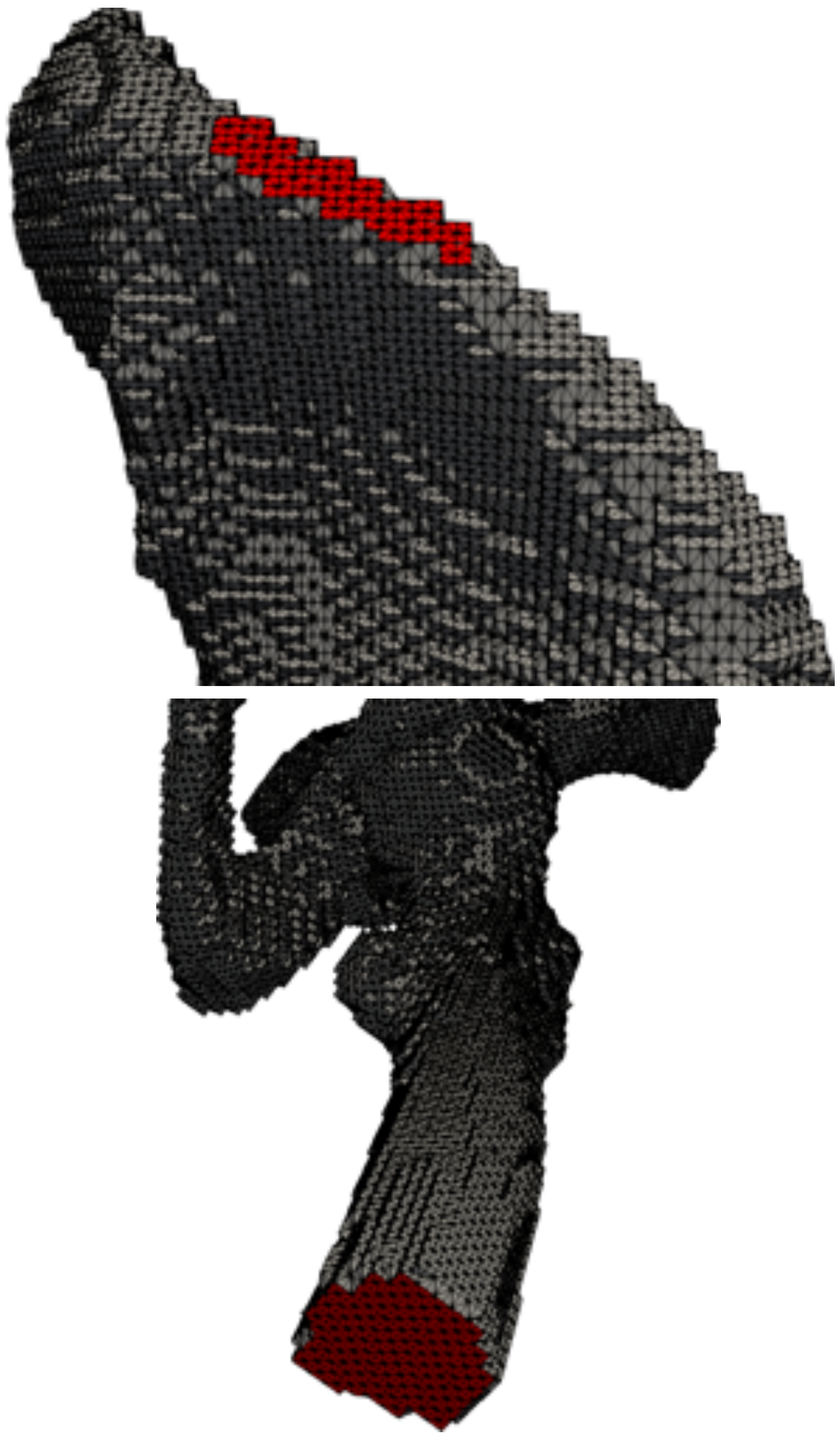
Geometrical Error (Linear Approximation)

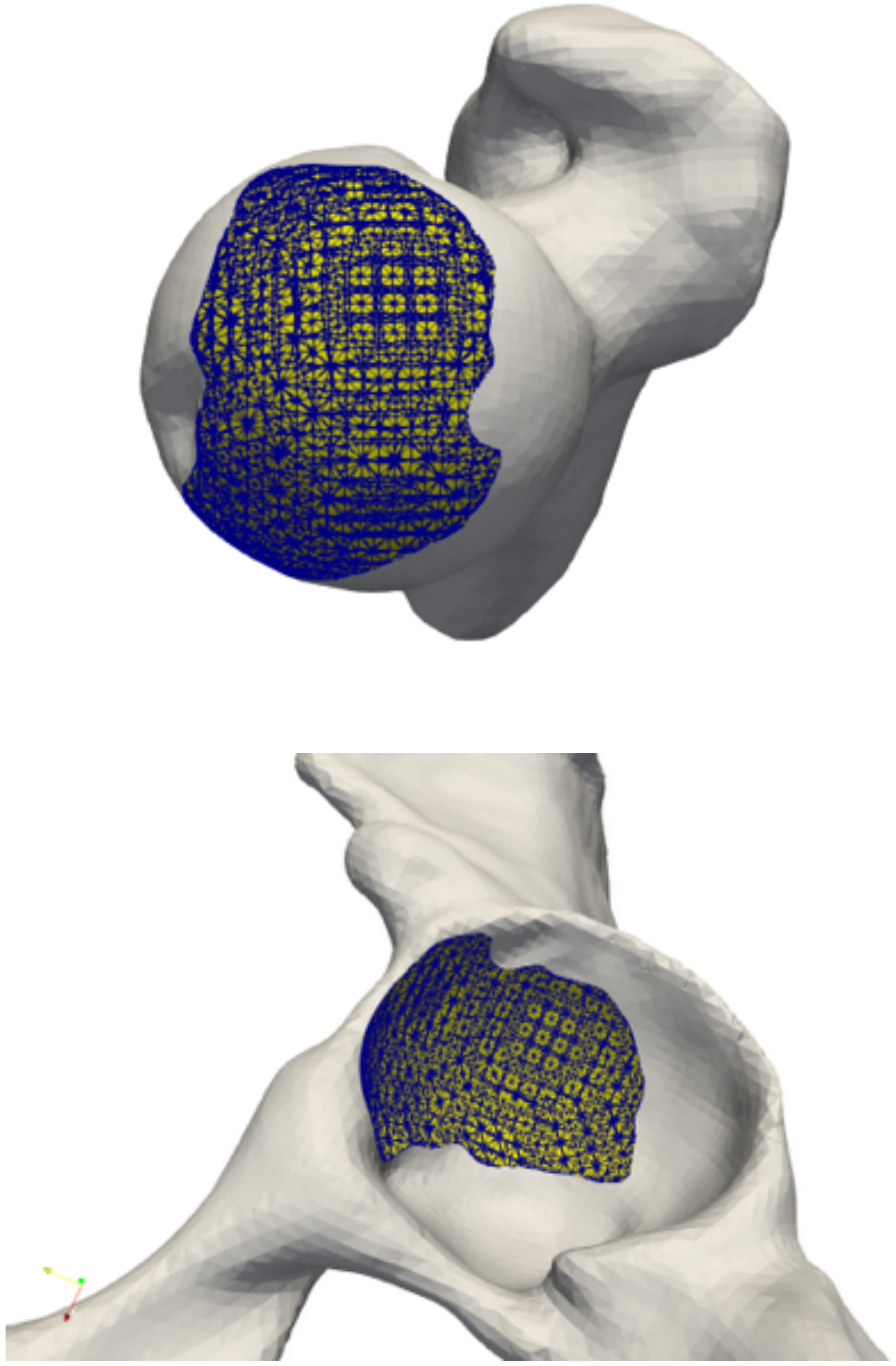


Refine elements that are intersected by surface triangulation

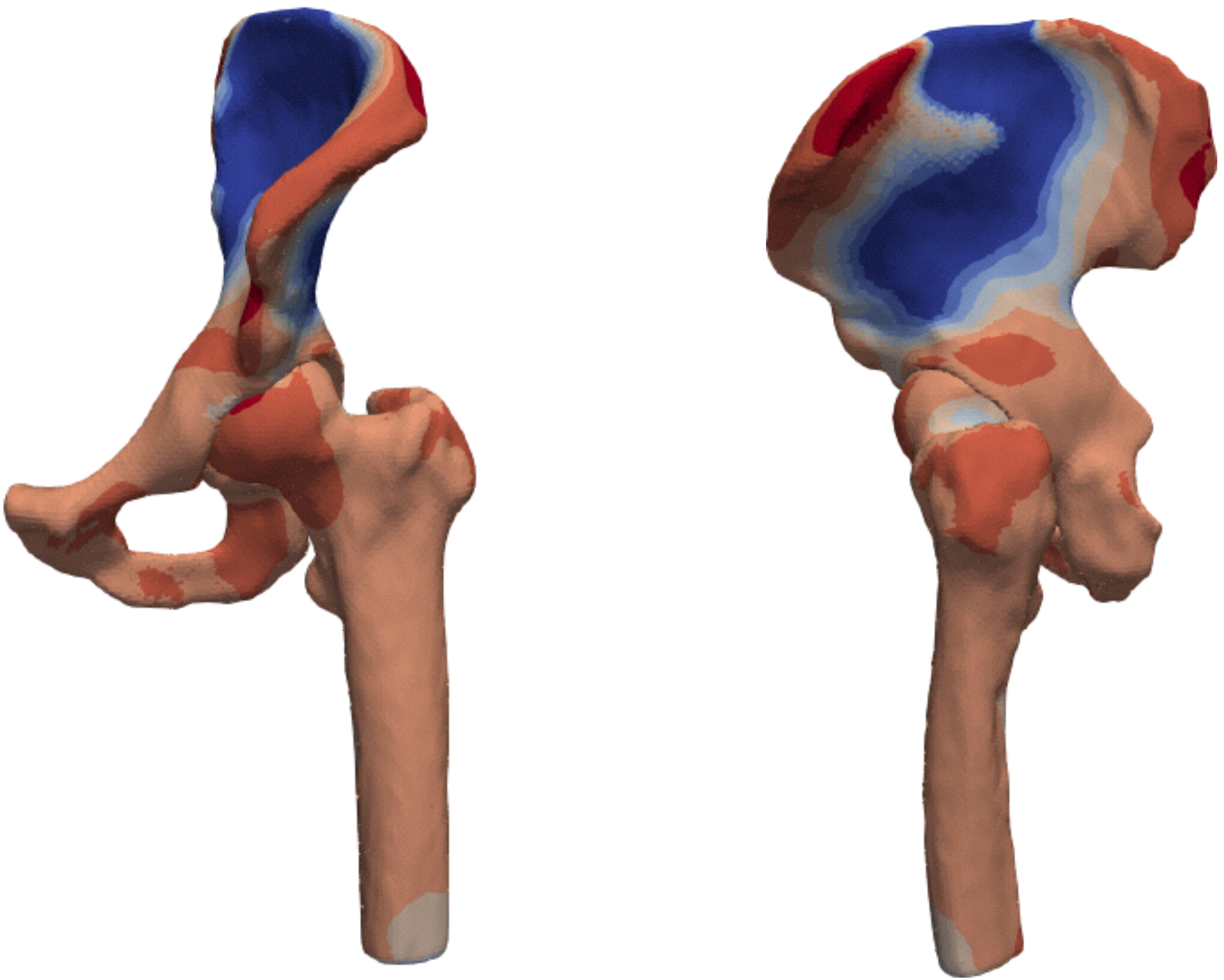


Extract elements and set boundary conditions





Stress Profile σ_{yy}



Stress Profile σ_{yy}