

A Special Purpose Finite-Element Framework for High Temperature Superconductor Applications

Christian Messe

Table of Contents



- **Introduction**
 - Recap: Mixed Formulations
 - Motivation: Why custom codebase?
- **Coding Considerations**
 - Code Architecture & Development Status
 - Memory Management
 - Meshing (example : thin shell with $h-\phi$)
- **Thoughts on Visualization**
- **Other Activities**
- **Summary & Outlook**

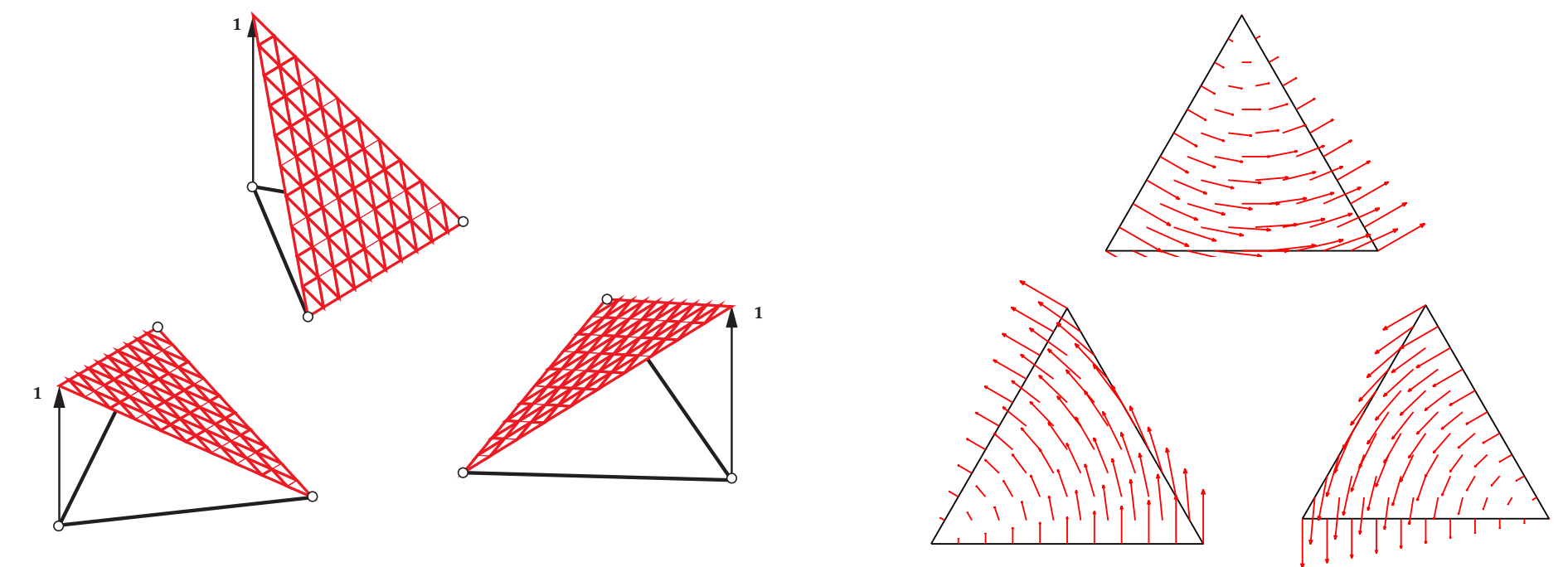
Mixed Formulations

a quick recap

- gaining significant interest over the last five years!
- goal:
 - improve conditioning for system matrix
 - reduce degrees of freedom
→ faster computation
- basic idea
 - governing equations are domain specific
 - domains are connected over interfaces conditions
- challenges
 - user-friendliness
 - mesh generation and memory management
 - iterating over non-linear material models
 - matrices are difficult to solve
(still ill conditioned, non symmetric, non-positive definite)

Example: h- ϕ -a formulation

	Air / Vacuum	Conductor	Ferromagnetic Alloy
Governing Equation	$\nabla \times \mathbf{h} = \mathbf{0}$ Ampère-Maxwell	$\nabla \times \mathbf{h} = \mathbf{j}$ Ampère-Maxwell	$\nabla \times \mathbf{e} = \dot{\mathbf{b}}$ Faraday's Law
Degree of Freedom	$\mathbf{h} = -\nabla \phi$ Magnetic Scalar Potential	\mathbf{h} Magnetic Field	$\mathbf{b} = \nabla \times \mathbf{a}$ Magnetic Vector Potential
Transport Law	none	$\mathbf{e} = \rho \cdot \mathbf{j}$ Ohm's Law	$\mathbf{h} = \mathbf{v} \cdot \mathbf{b}$ Magnetic Law
Comment	minimal number of dofs	need edge elements	simple material law implementation



Lagrange
Elements

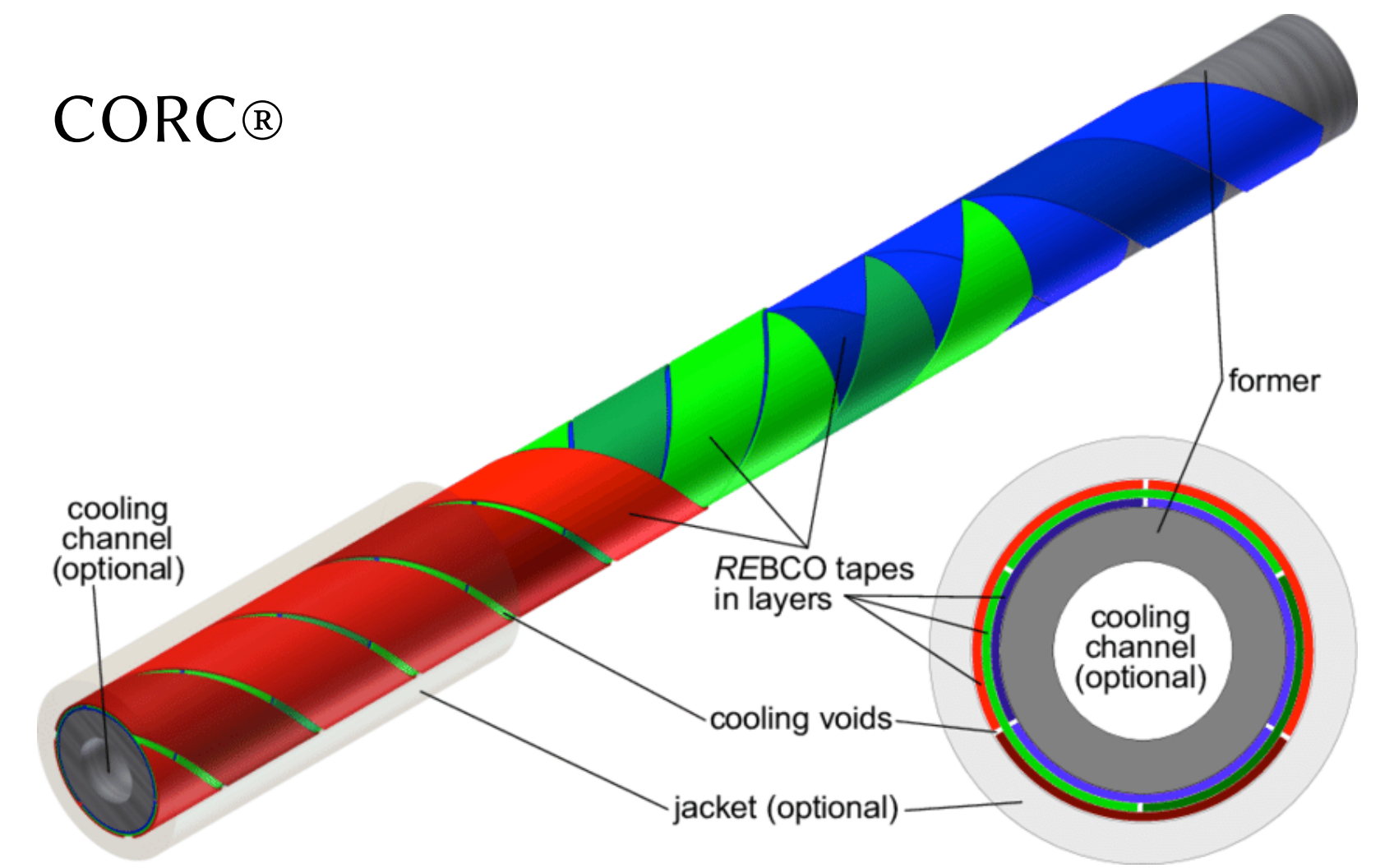
Nédélec
Elements

Motivation

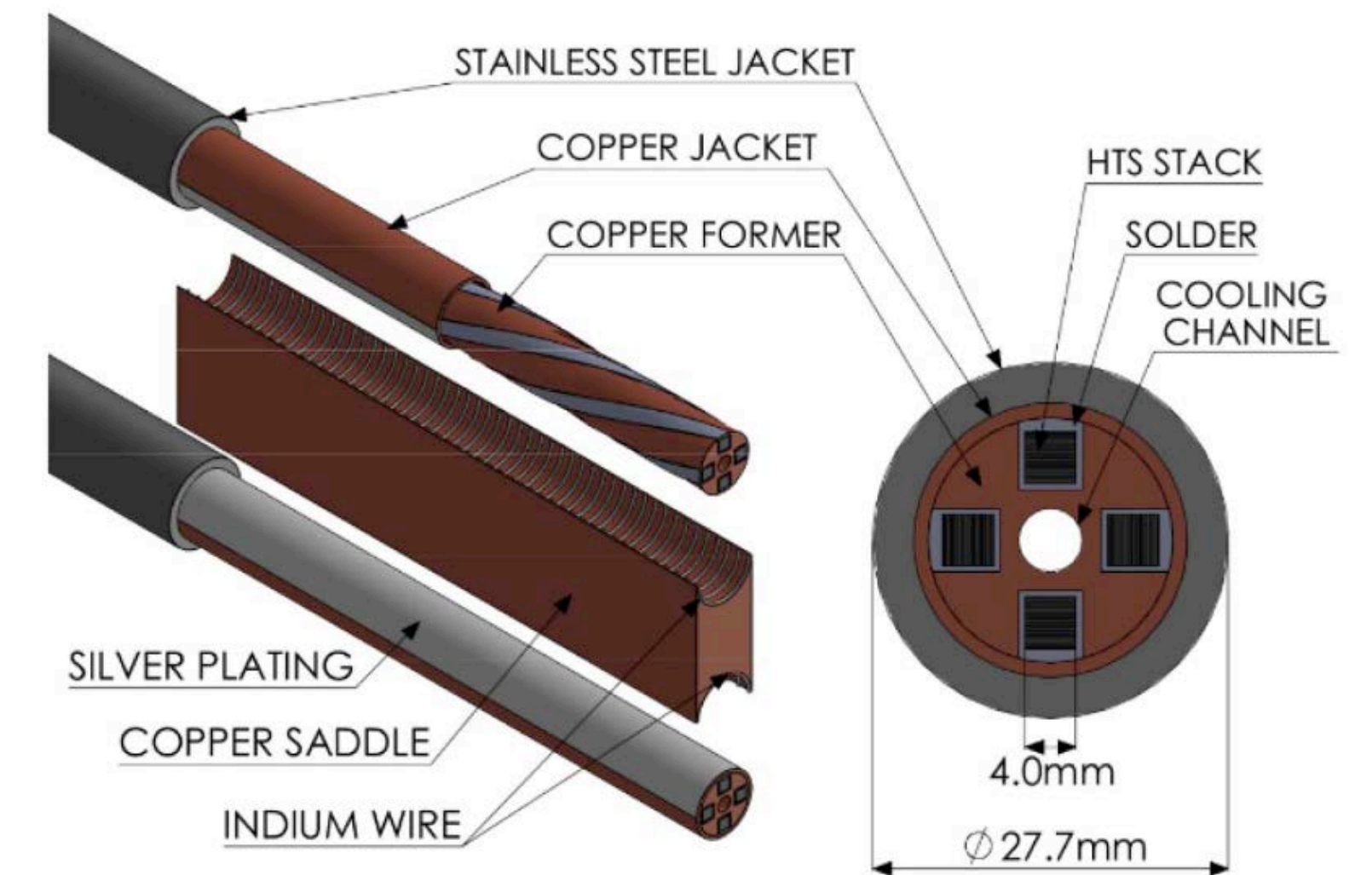
why build a custom codebase?

- **application needs: have thin-shell model that**
 - uses the $h-\phi$ formulation
 - supports thermal conduction and quenching
 - supports current sharing between overlapping tapes
 - supports convective cooling with LN2 and LHe
- **model development needs**
 - have full control over data structure
 - have full knowledge of underlying algorithms
- **utilization of community software**
 - use open source data formats (gmsh, hdf5, exodus-ii)
 - link to popular solver libraries such as PETSc, MKL, ...
- **in-house resource utilization**
 - work with STRUMPACK team to maximize solver performance
<https://portal.nersc.gov/project/sparse/strumpack/>

CORC®



[Source: 10.1088/0953-2048/28/6/065007]

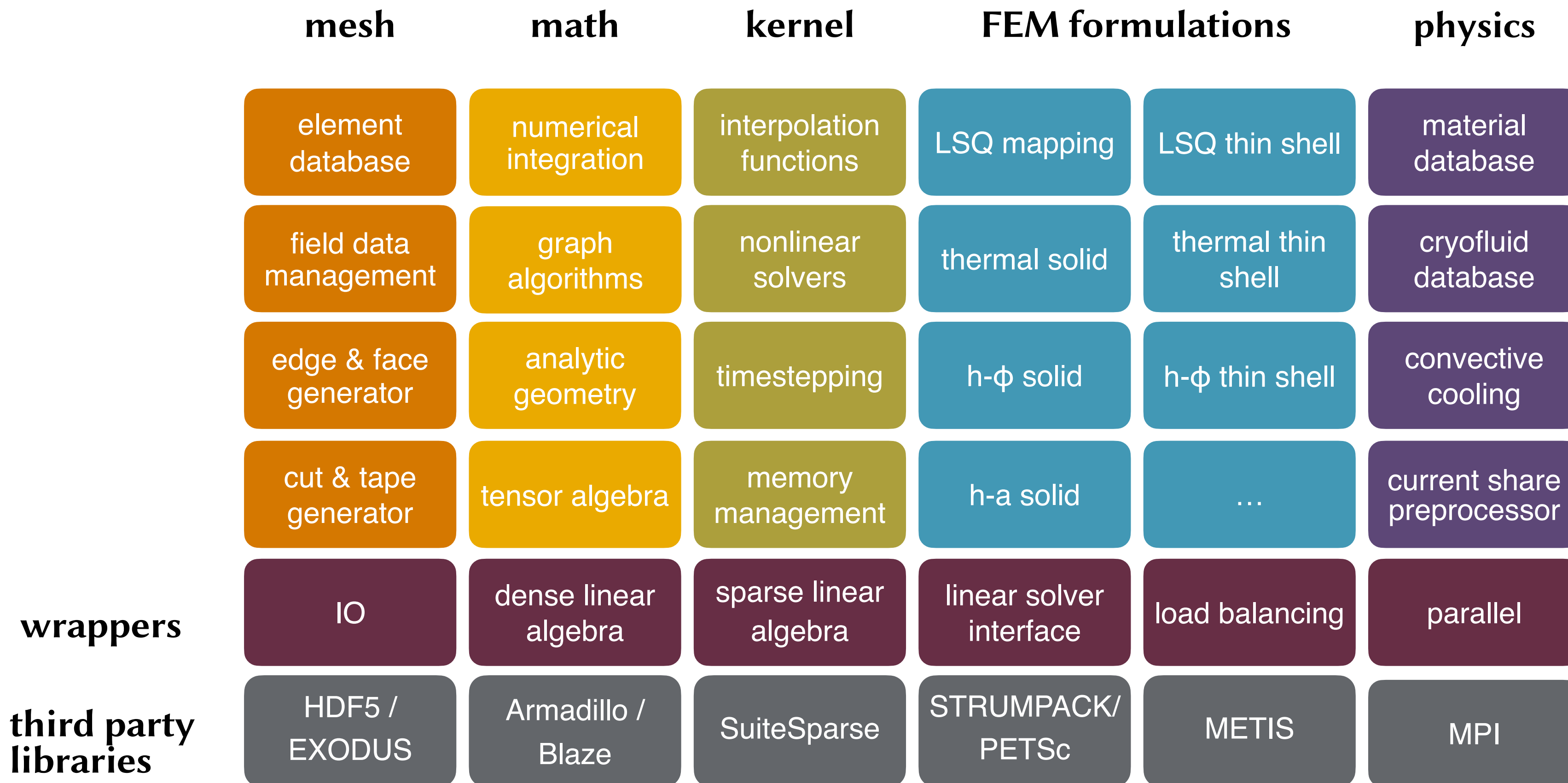


VIPER

[Source: 10.1088/1361-6668/abb8c0]

BELFEM Code Architecture

general overview



philosophy

- consistent naming schemes
→ easier to read for programmers
- highly modular structure
→ easier to extend and maintain
- third party libraries are exclusively accessed through wrapper
→ can interchange when compiling
→ be able to test performance of various third party libraries

BELFEM Code Architecture

current development status

	mesh	math	kernel	FEM formulations	physics
	element database	numerical integration	interpolation functions	LSQ mapping	LSQ thin shell
	field data management	graph algorithms	nonlinear solvers	thermal solid	thermal thin shell
	edge & face generator	analytic geometry	timestepping	h- ϕ solid	h- ϕ thin shell
	cut & tape generator	tensor algebra	memory management	h-a solid	...
wrappers	IO	dense linear algebra	sparse linear algebra	linear solver interface	load balancing
third party libraries	HDF5 / EXODUS	Armadillo / Blaze	SuiteSparse	STRUMPACK/ PETSc	METIS
					MPI

color codes

- pending
- in progress
- complete

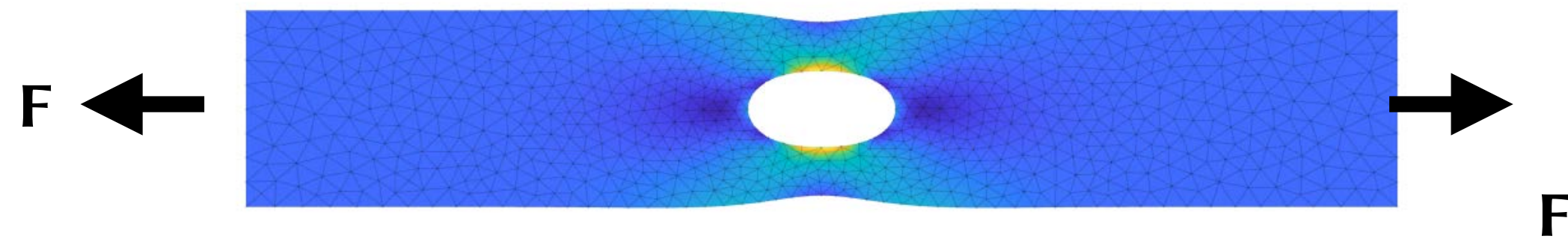
other things to do

- benchmark against COMSOL
- code hardening
- improve CMake setup

Memory Management

Memory Management and Matrix Computation

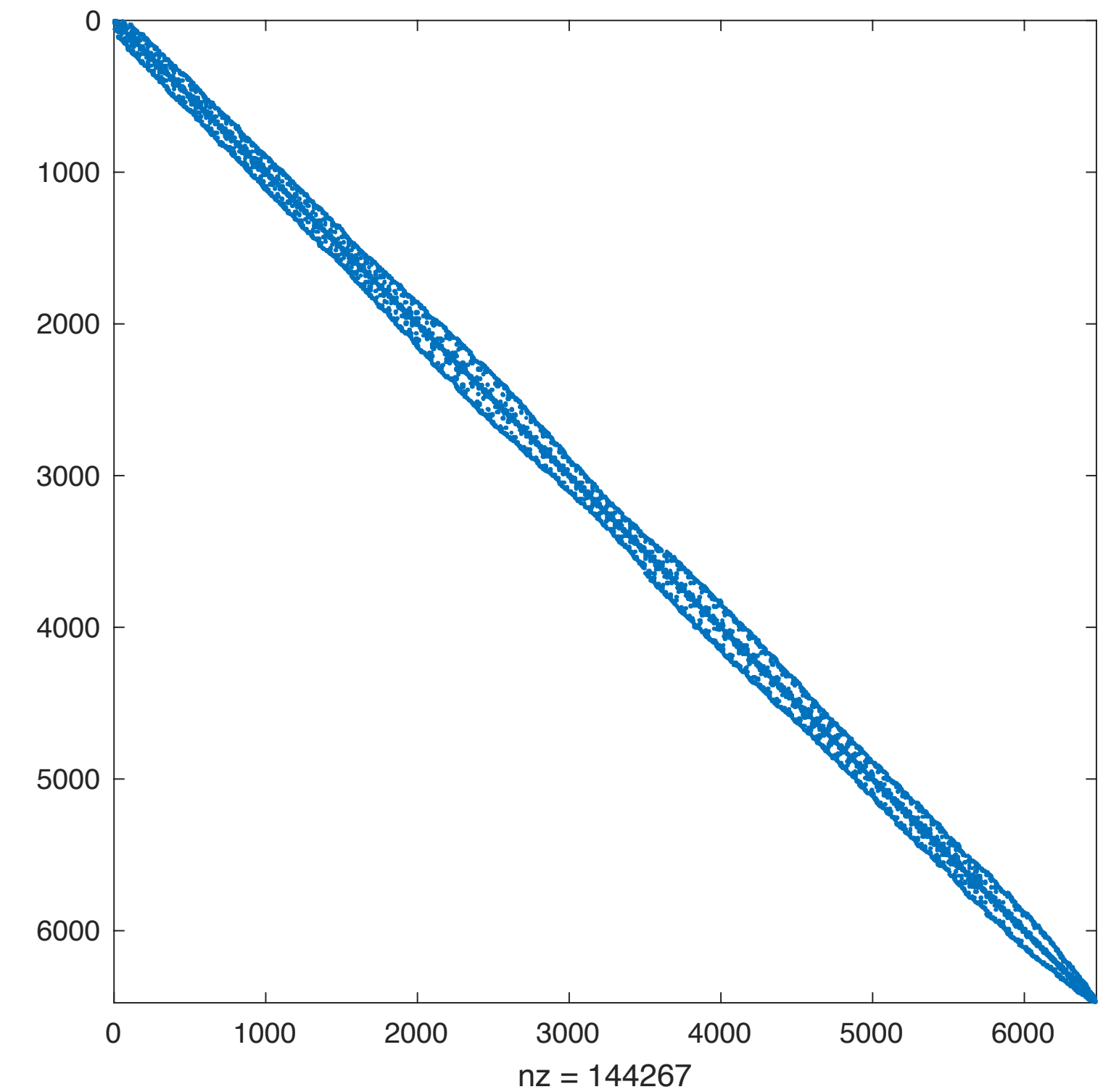
Classic FEM: tensile test example



- one single governing equation
- degrees of freedom are associated with nodes
- simple algorithm for creating adjacency graph

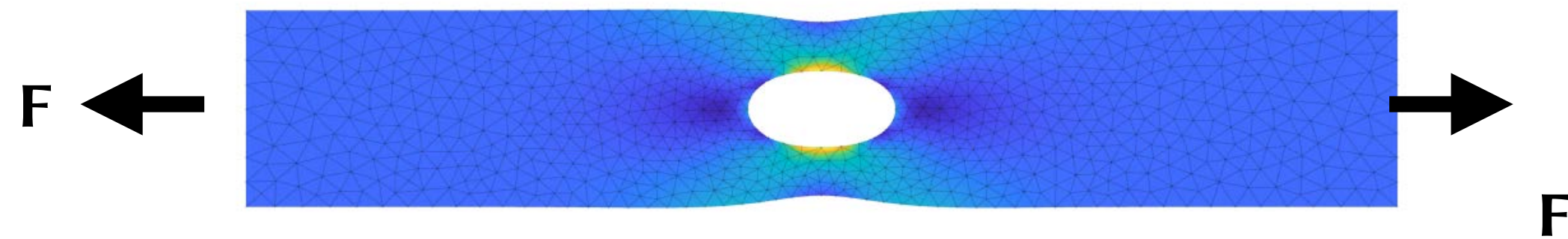
Matrix Sparsity Pattern

only nonzero values are stored in memory



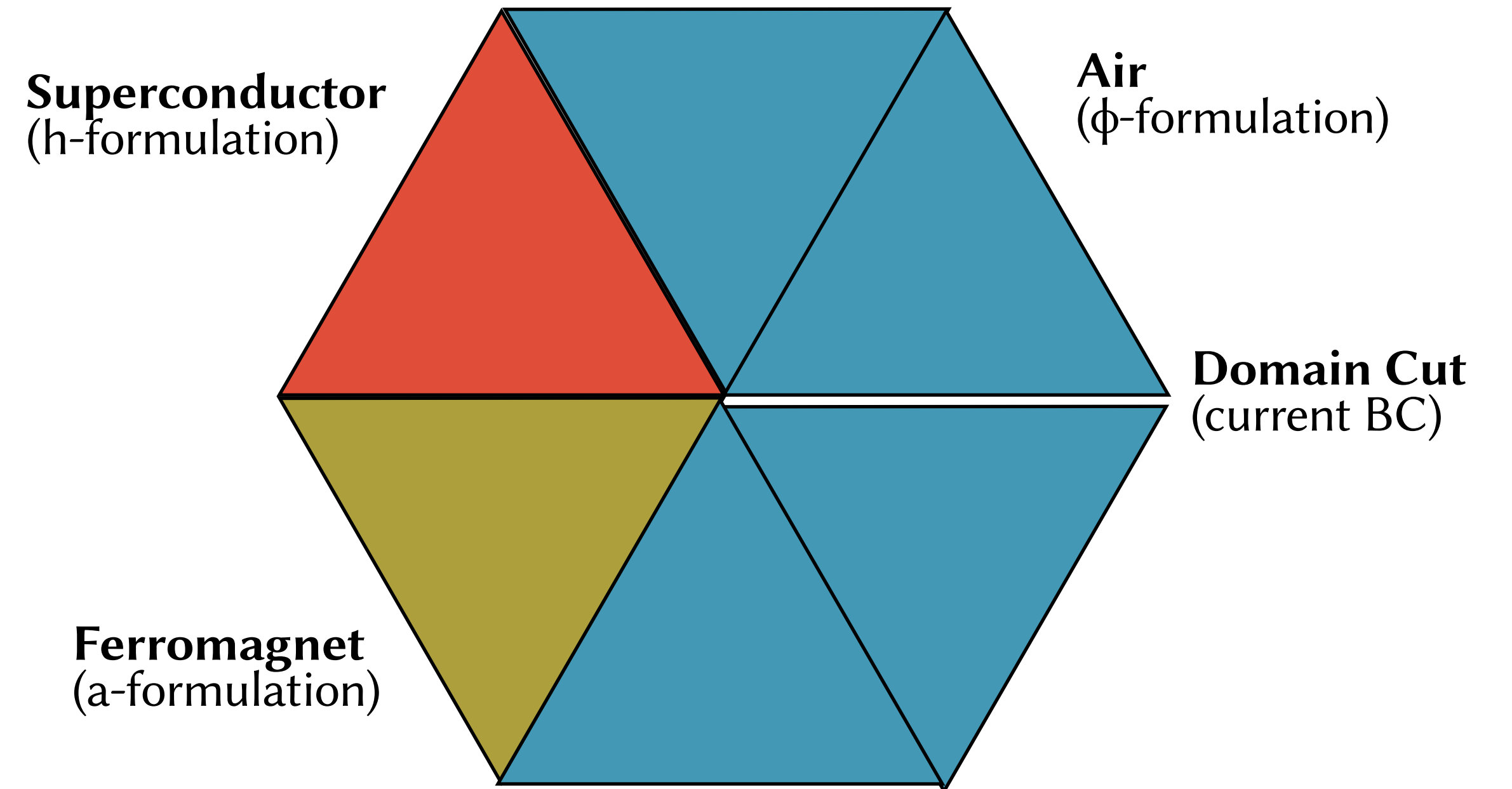
Memory Management and Matrix Computation

Classic FEM: tensile test example



- one single governing equation
- degrees of freedom are associated with nodes
- simple algorithm for creating adjacency graph

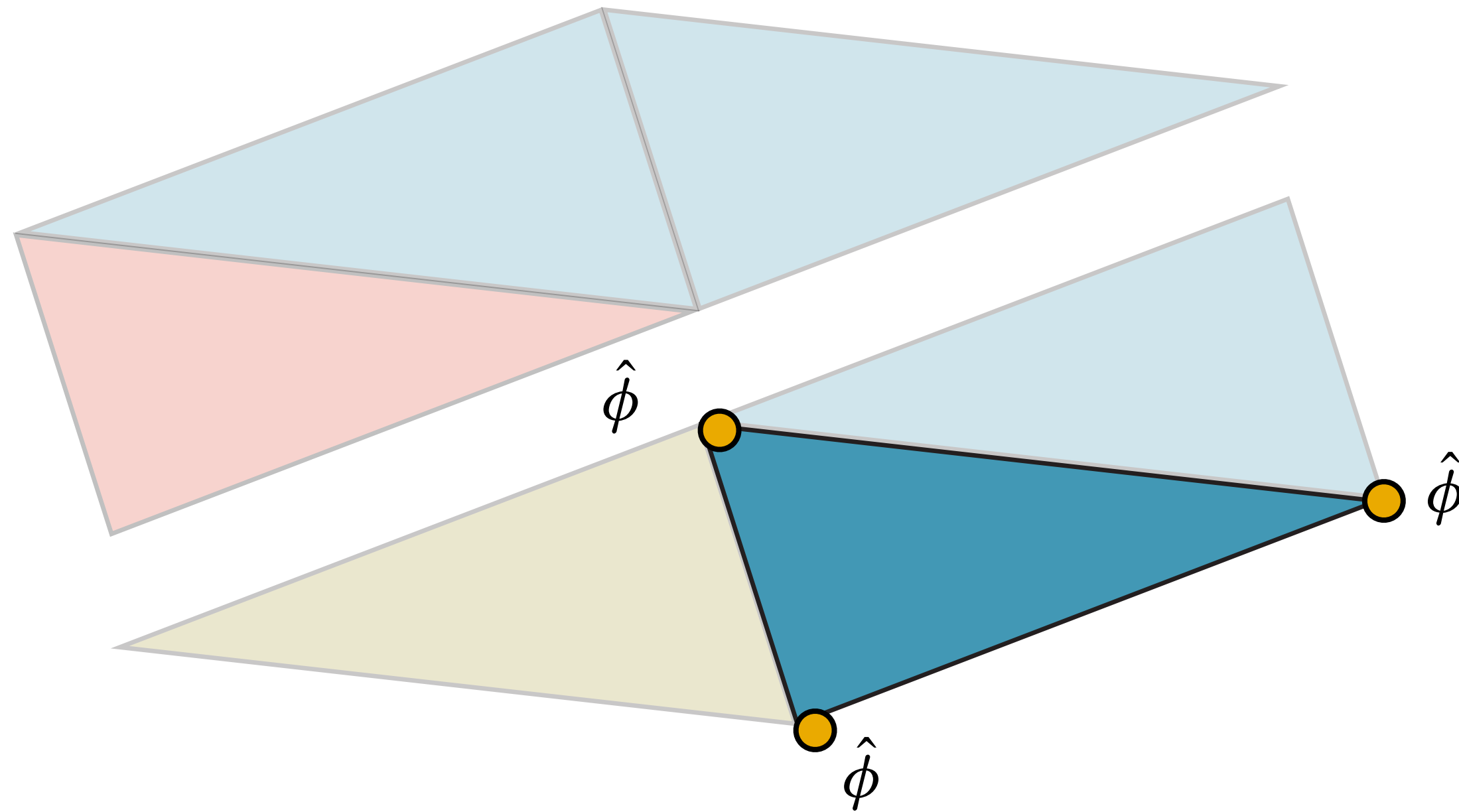
Mixed Formulation:



- block and sideset specific governing equations
- degrees of freedom are associated with nodes, edges or faces
- non-trivial algorithm for creating adjacency graph

Memory Management and Matrix Computation

volume elements



Air Element:

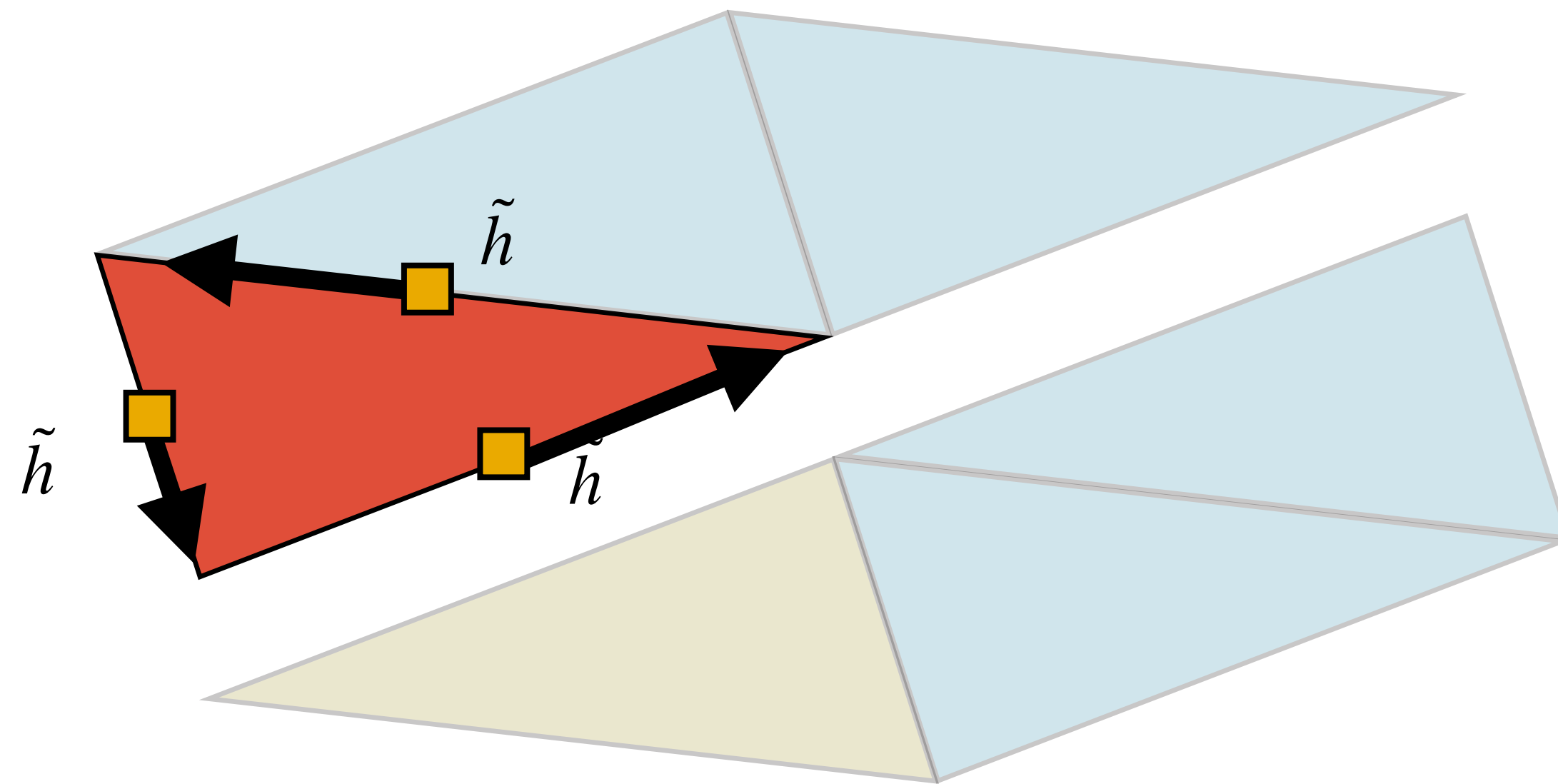
- uses ϕ -formulation
- has node degrees of freedom

$$\mathbf{K} = 0$$

$$\mathbf{M} = \int_{\Omega} \mathbf{B}^T \mu \mathbf{B} dV$$

Memory Management and Matrix Computation

volume elements

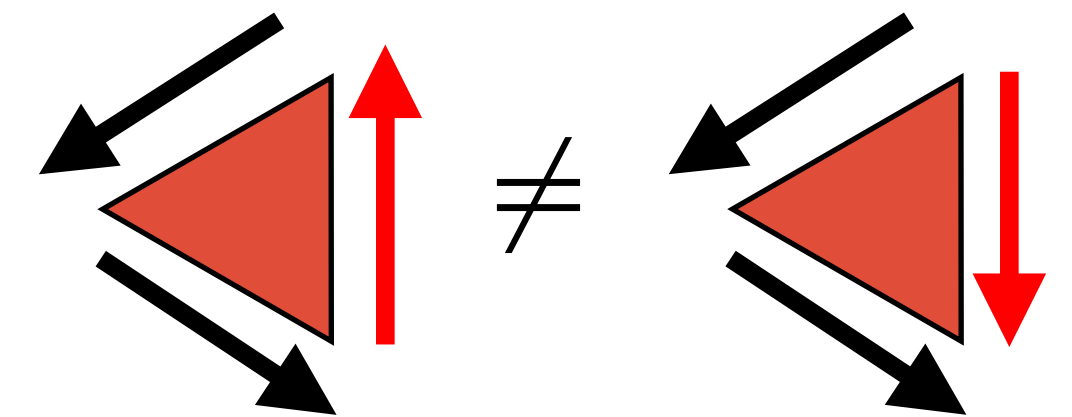


Conducting Element:

- uses h-formulation
- e.g. power law or lookup table for ρ , constant μ
- has edge and face degrees of freedom
- mesh must handle edge and face numbering
- edge direction matters

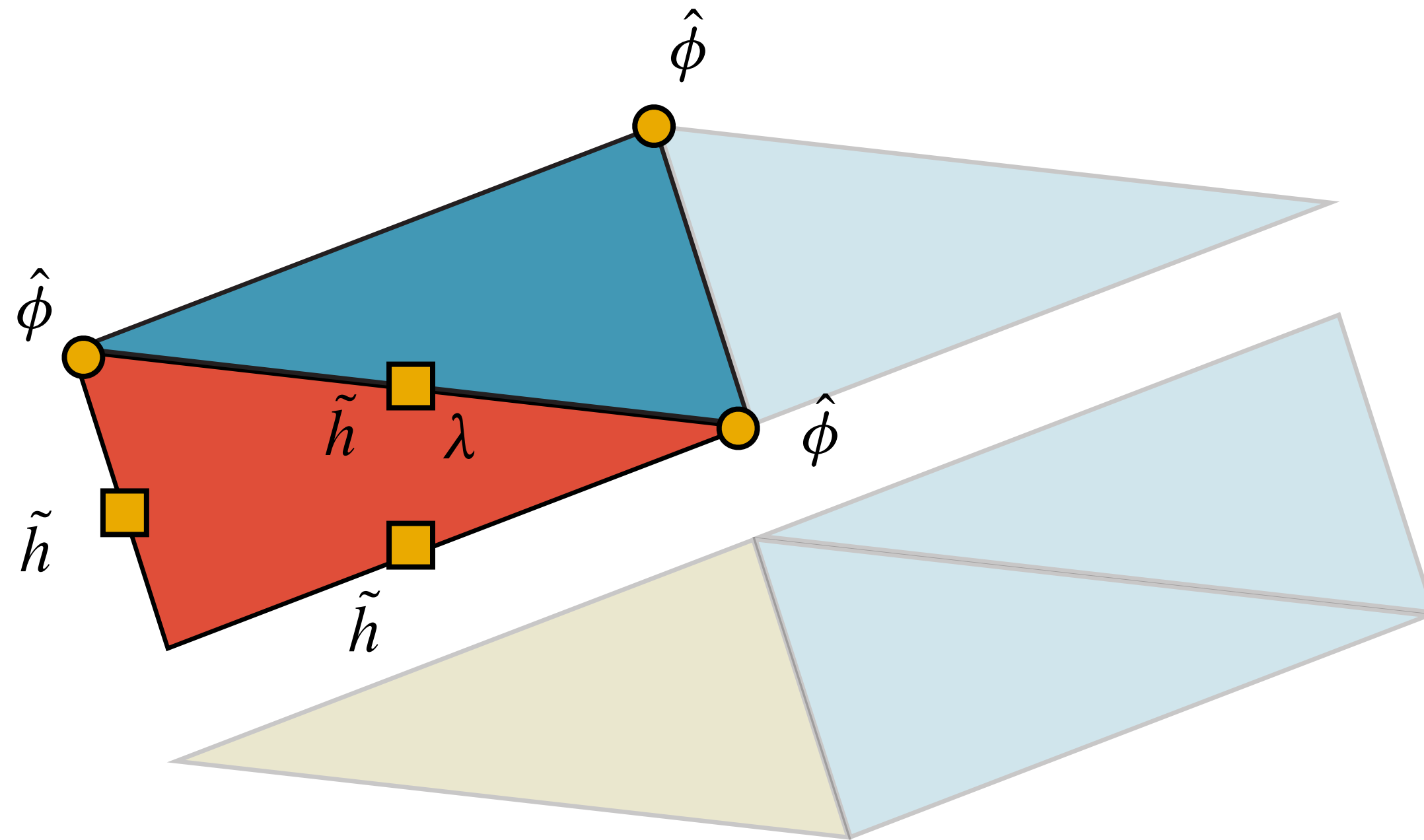
$$\mathbf{K} = \int_{\Omega} \mathbf{C}^T \rho \mathbf{C} dV$$

$$\mathbf{M} = \int_{\Omega} \mathbf{E}^T \mu \mathbf{E} dV$$



Memory Management and Matrix Computation

interface elements between different domain types



Conductor-Air Interface Element

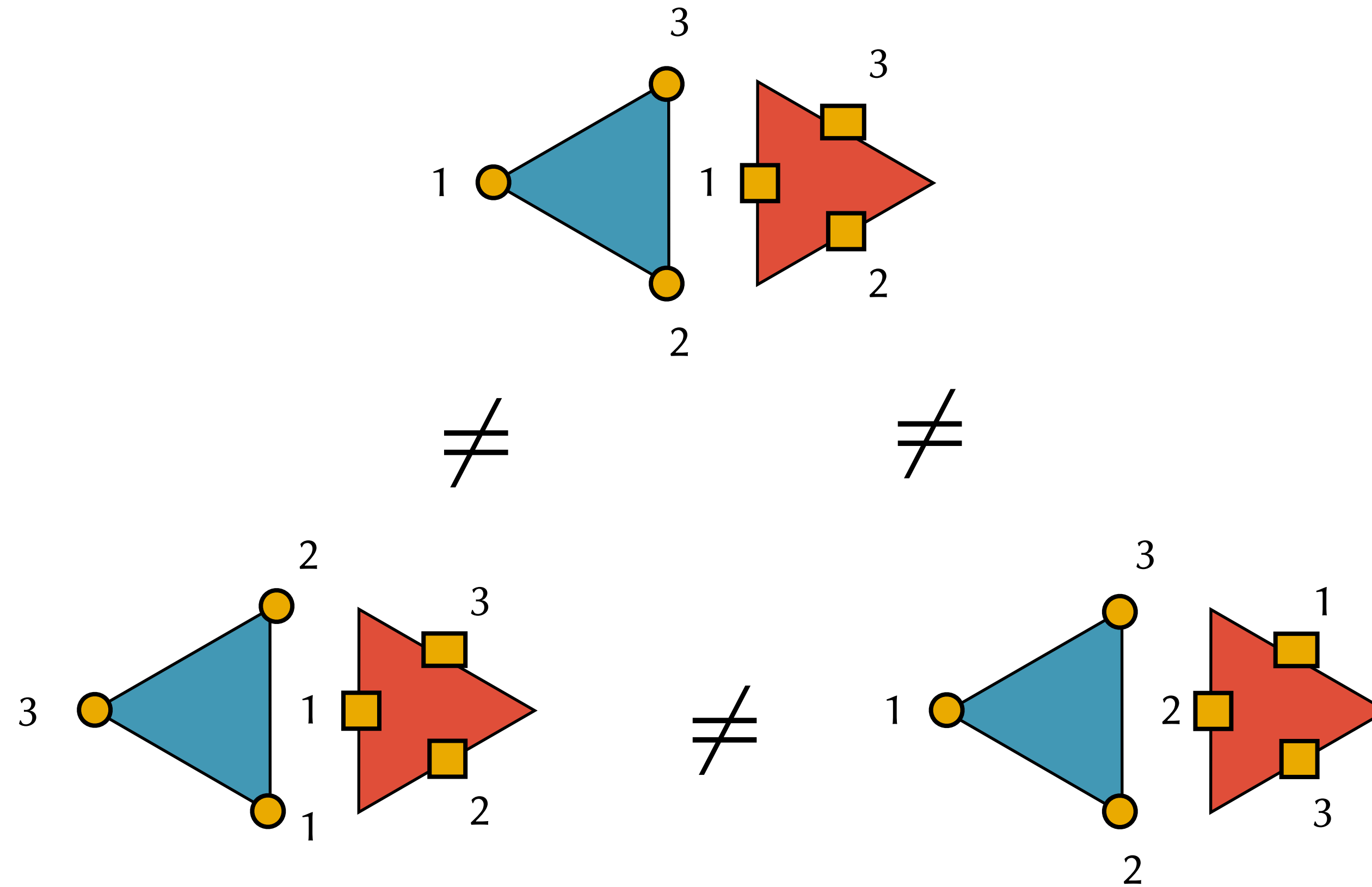
- used for current boundary condition
- uses Lagrange multiplier
- considers orientation of elements
- is non-symmetric!

$$\mathbf{K} = \mu_0 \int_{\partial\Omega} \begin{bmatrix} 0 & 0 & (\mathbf{n}^T \mathbf{E}^T) \\ -\mathbf{N}^T \mathbf{n}^T \mathbf{E} & 0 & (\mathbf{n}^T \mathbf{B}^T) \\ \mathbf{n} \times \mathbf{E} & \mathbf{n} \times \mathbf{B} & 0 \end{bmatrix} dS$$

$$\mathbf{M} = 0$$

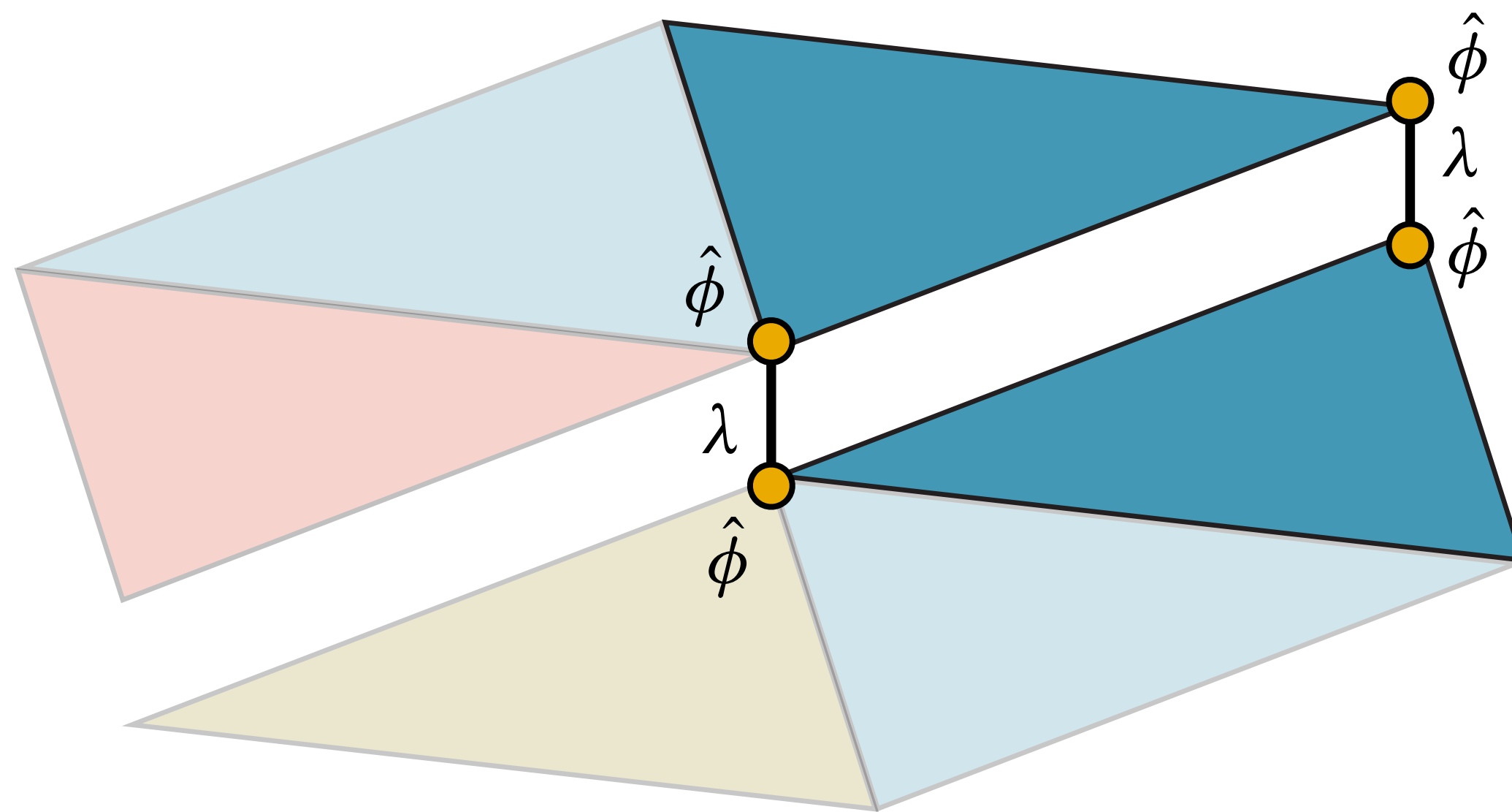
Memory Management and Matrix Computation

element orientation matters!



Memory Management and Matrix Computation

domain cuts



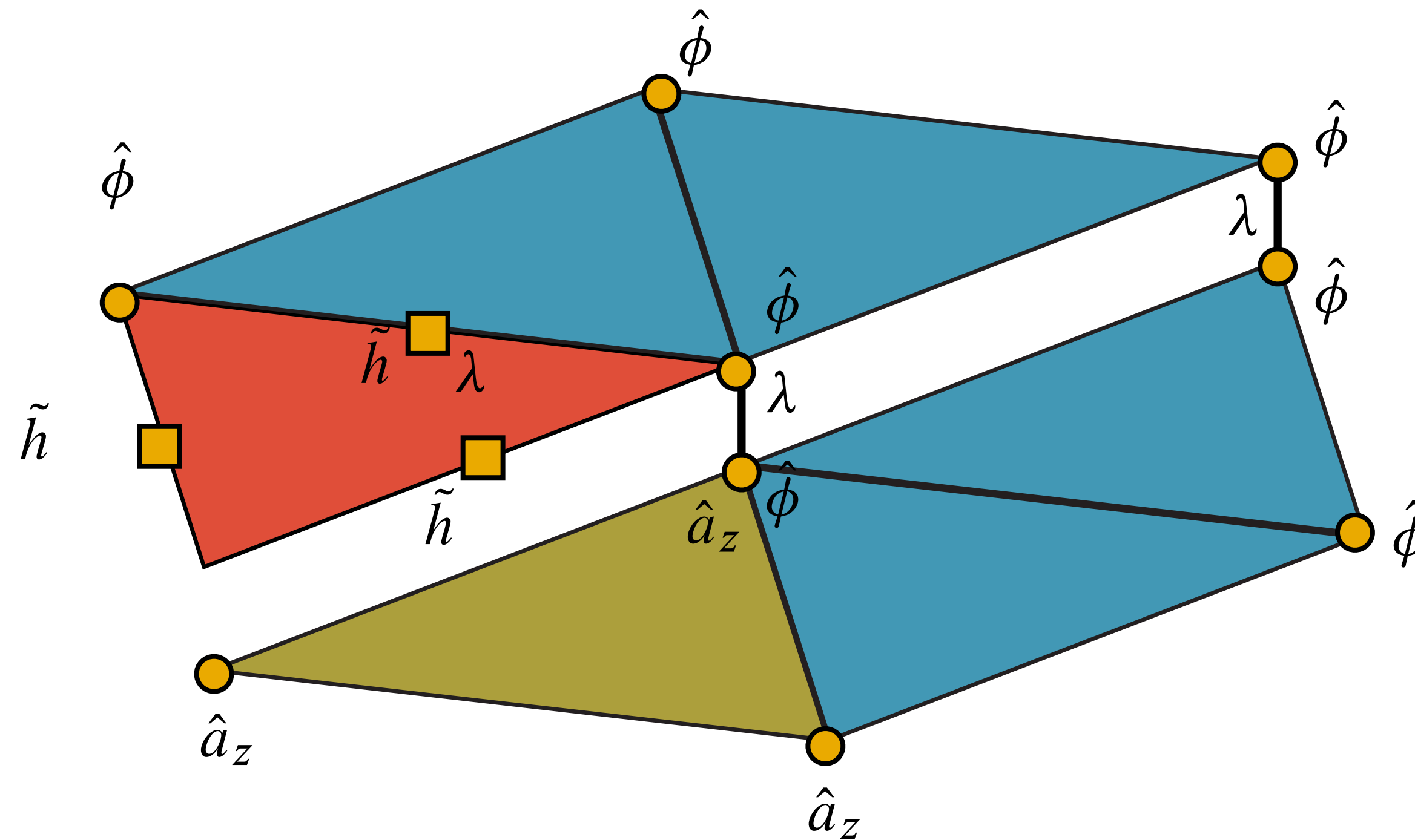
Domain Cut Element:

- uses ϕ -formulation
- has Lagrange multiplier λ as dof
- mesh must handle duplicate nodes

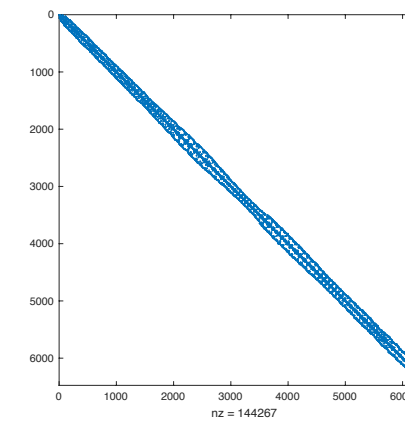
$$\mathbf{K} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & -1 \\ 1 & -1 & 0 \end{bmatrix} \quad \mathbf{f} = \begin{bmatrix} 0 \\ 0 \\ I \end{bmatrix}$$

Memory Management and Matrix Computation

mixed formulation summary



Diversity of DoF-Types \rightarrow non-trivial adjacency graph



h-dofs are connected to edges and faces
a-dofs are connected to nodes
 ϕ -dofs are connected to nodes
 λ -dofs are connected to edges or nodes

Things the code must consider:

- DoF numbering scheme must be unique over all processors
- element orientation matters
- domain cuts require special treatment (direction of cut)
- thin shells require special treatment (element clusters)

\rightarrow mesh needs additional information

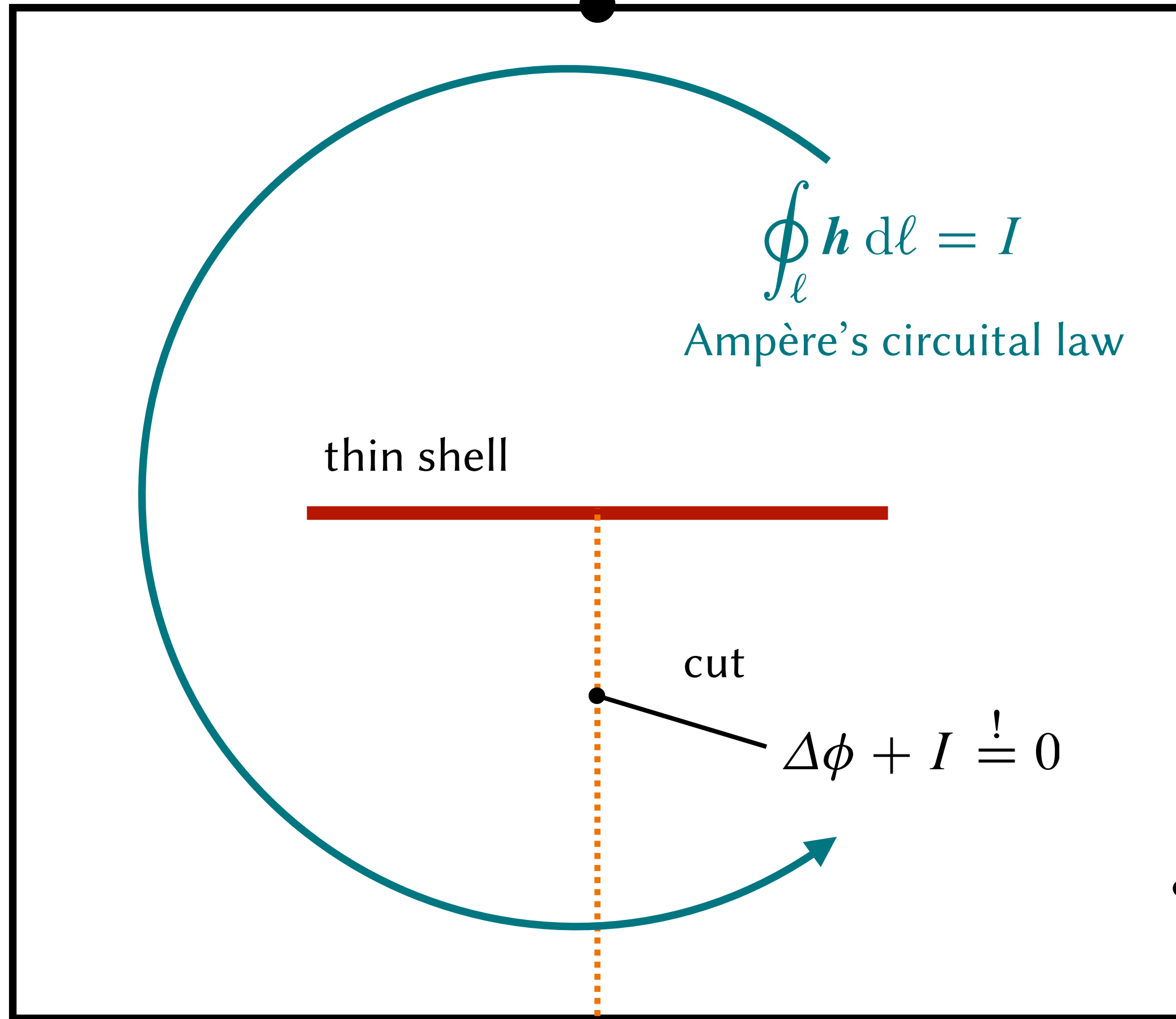
Meshing

[Example: Thin Shells with h - ϕ]

Meshing Example: thin-shell with h - ϕ formulation

problem definition

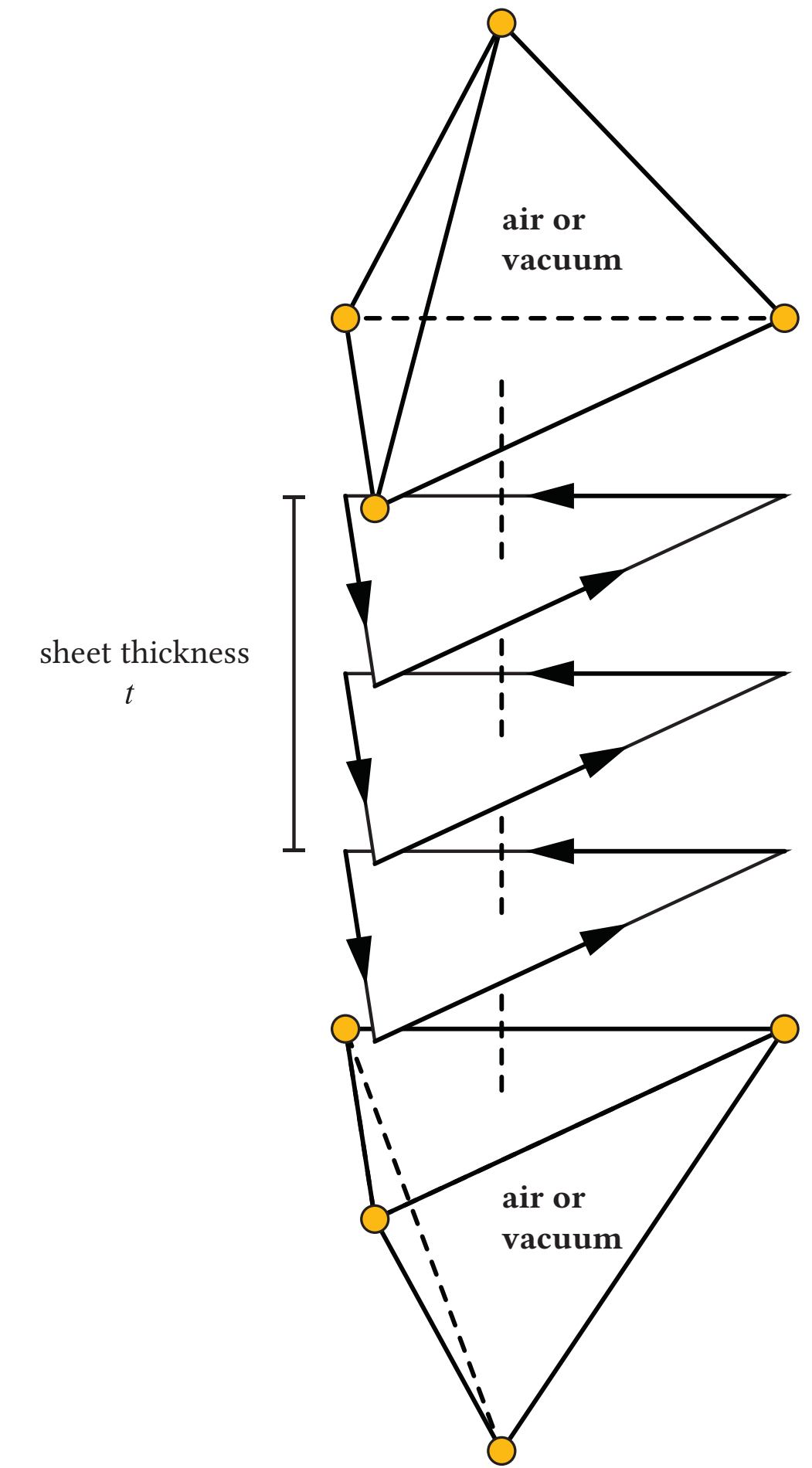
fixed bearing:
strong bc $\phi \stackrel{!}{=} 0$
→ prevent rigid body motion



- for air: $\mathbf{h} = -\nabla\phi$
- imposed integral current: \mathbf{I}
- no background field

magnetic wall:
weak bc $\mathbf{n}^T \nabla\phi = 0$

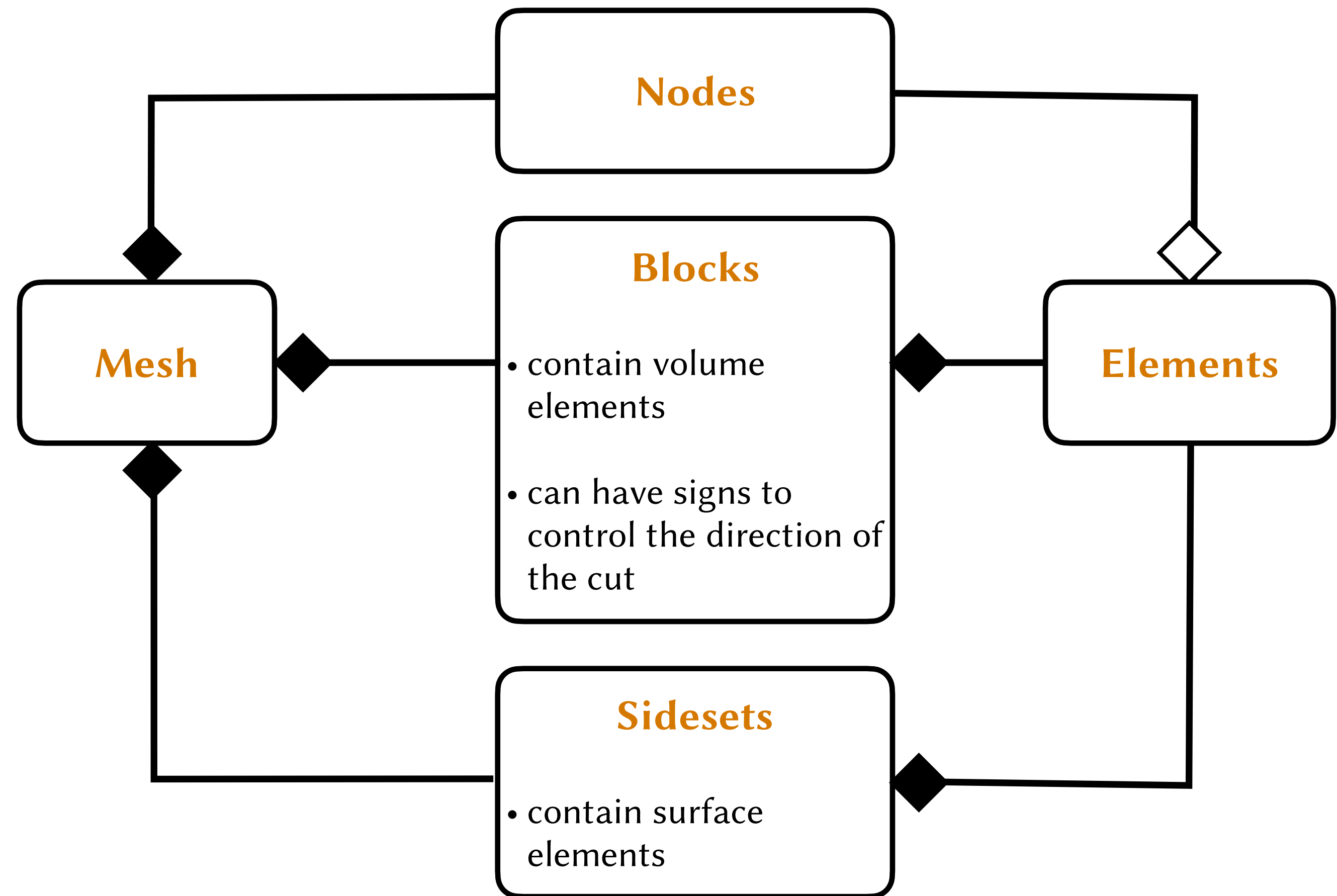
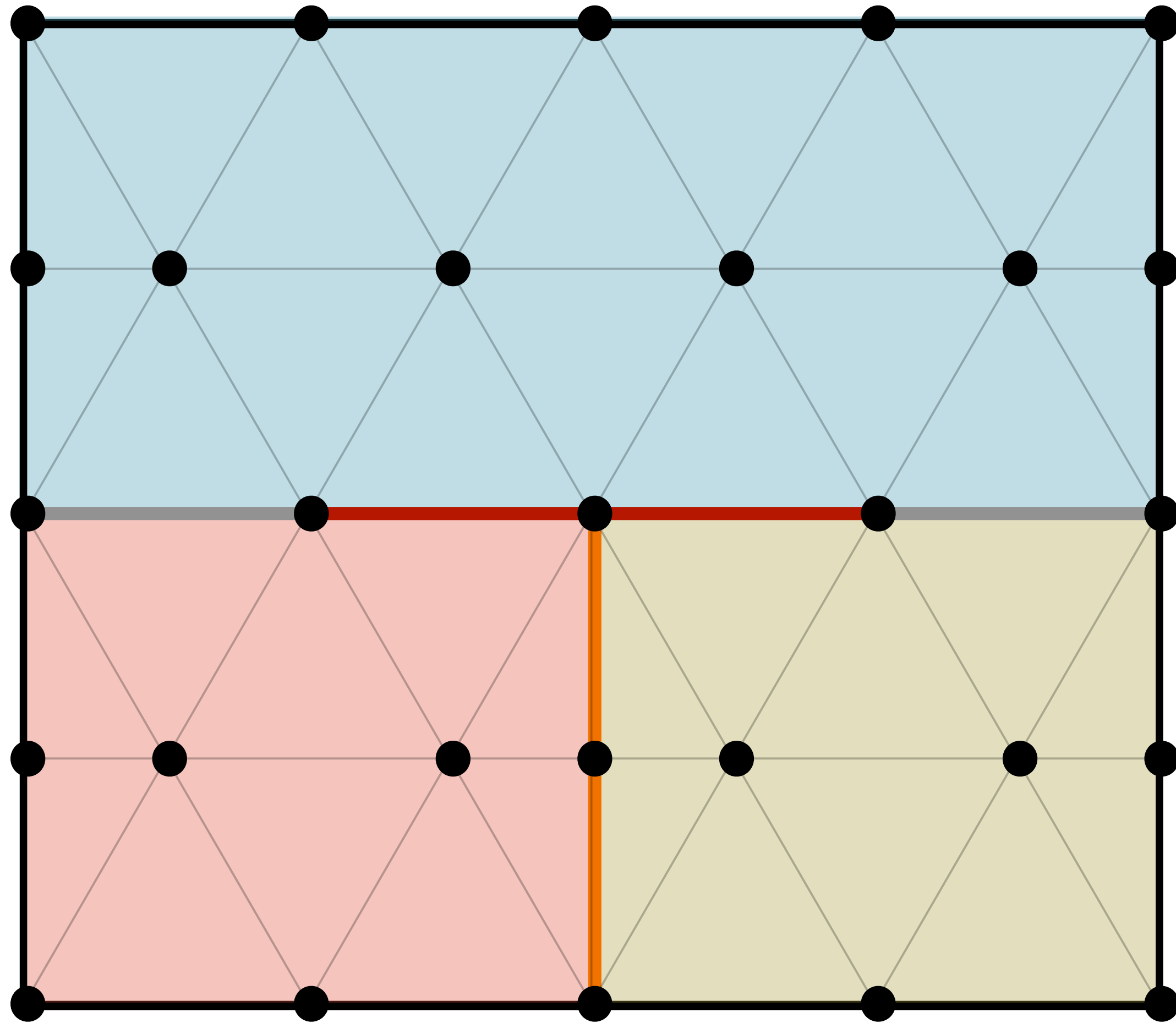
goal: create data structure that contains TS-macroelements



see also: de Sousa Alves et al:
[10.1088/1361-6668/ac3f9e](https://doi.org/10.1088/1361-6668/ac3f9e)
[10.1109/TASC.2022.3143076](https://doi.org/10.1109/TASC.2022.3143076)

Meshing Example: thin-shell with $h-\phi$ formulation

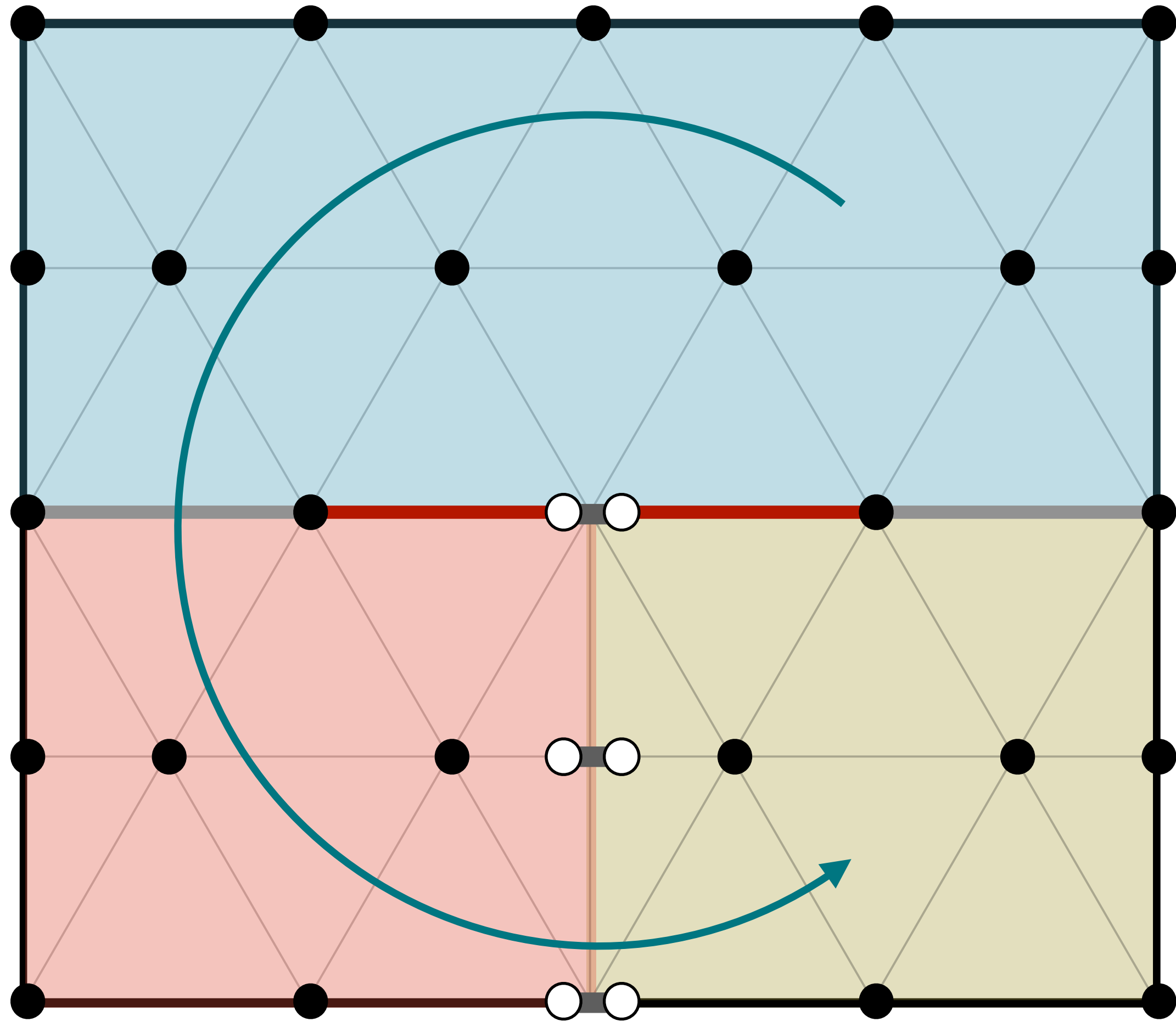
step 0: initial mesh



- most mesh generators create only nodes and elements
- no edges, cuts, sheet elements

Meshing Example: thin-shell with $h-\phi$ formulation

step 1: cutting



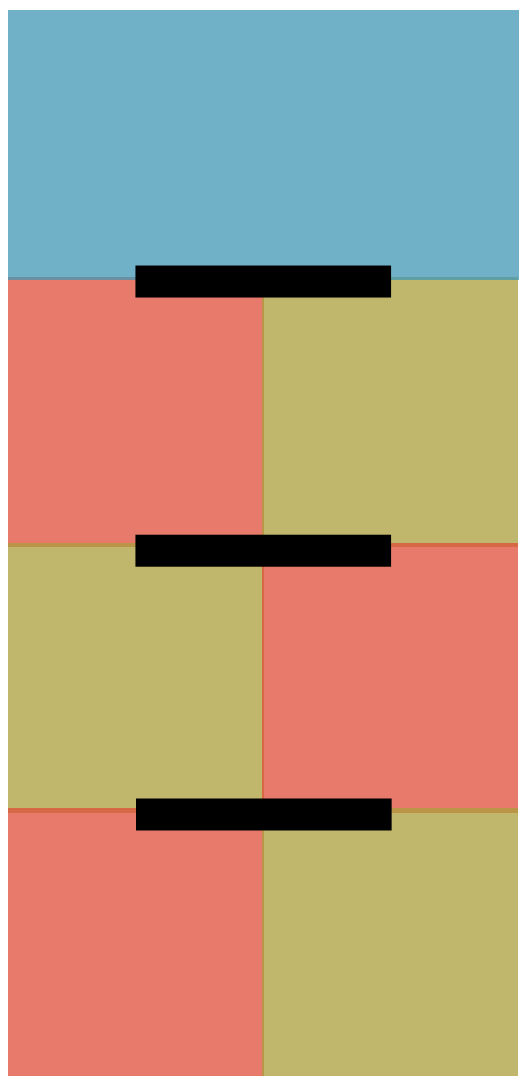
Scissor Object

- duplicates nodes
- reconnects elements with new nodes
- adds connector elements for λ -bc

Mesh

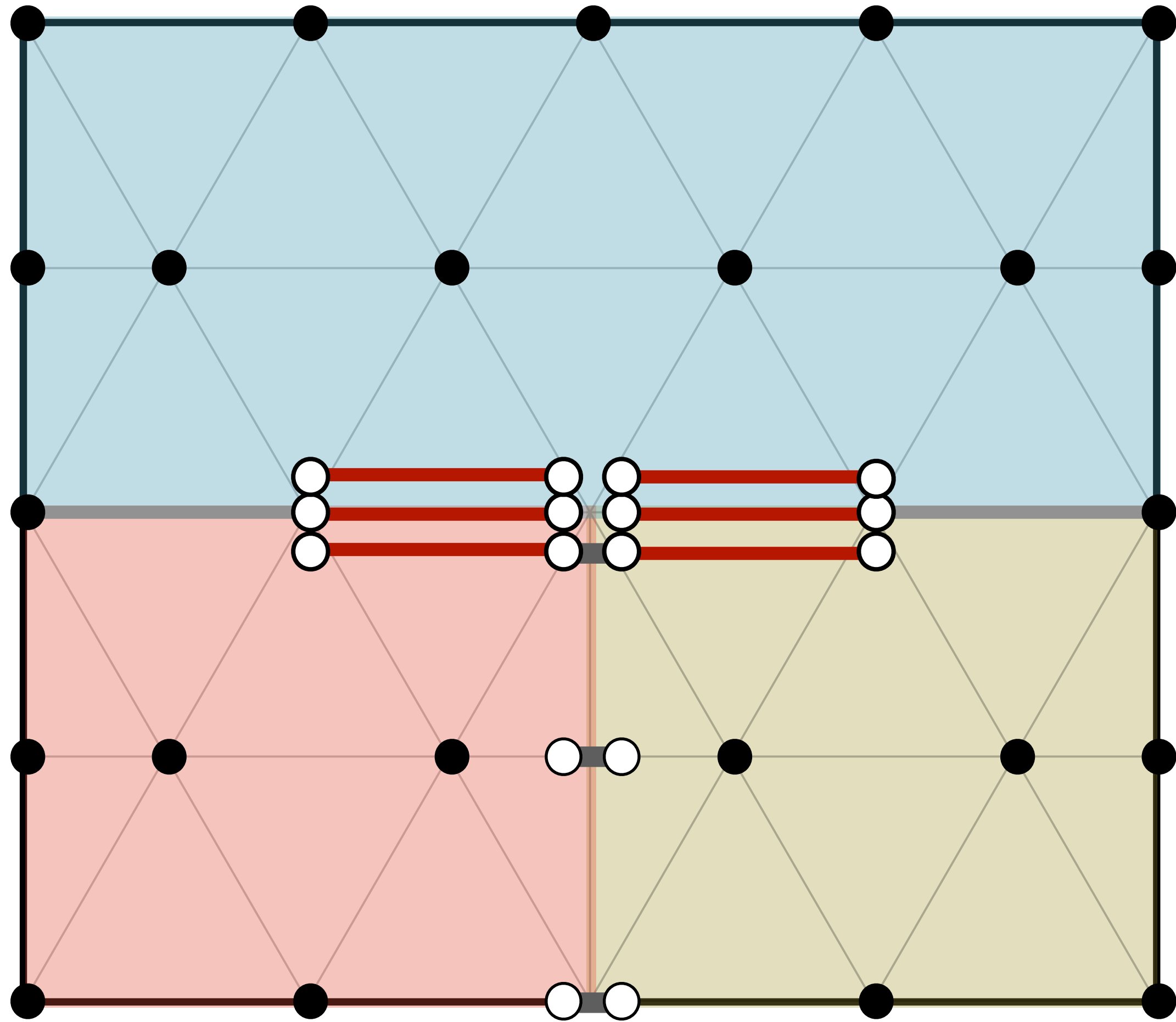
- direction of cut is determined by blocks
 - blue: neutral, no cut here
 - red: negative
 - green: positive
- algorithm is unique if blocks do not share surface with block of same sign (red never touches another red, green never touches another green)

tape stack example



Meshing Example: thin-shell with $h-\phi$ formulation

step 2: taping



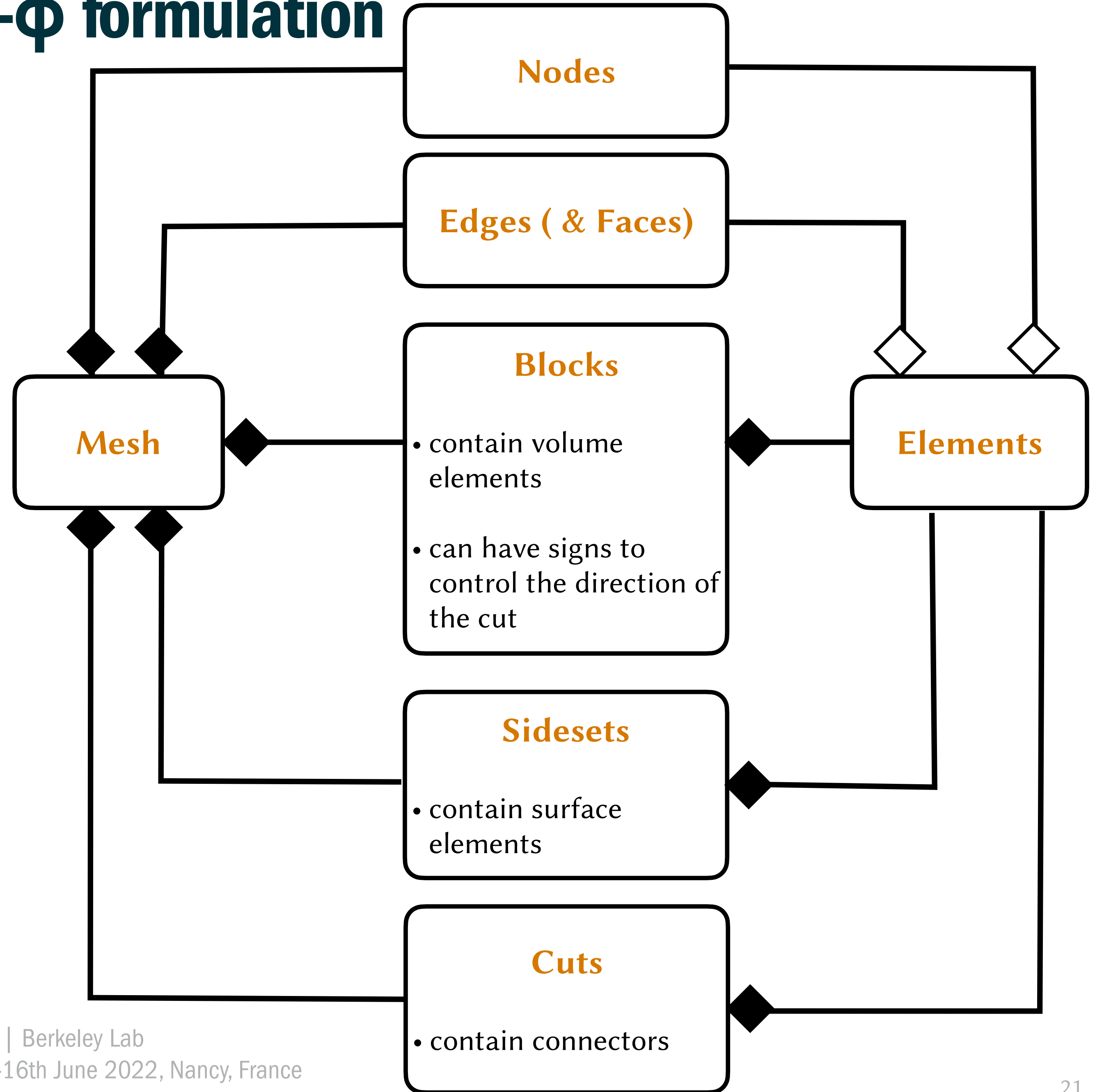
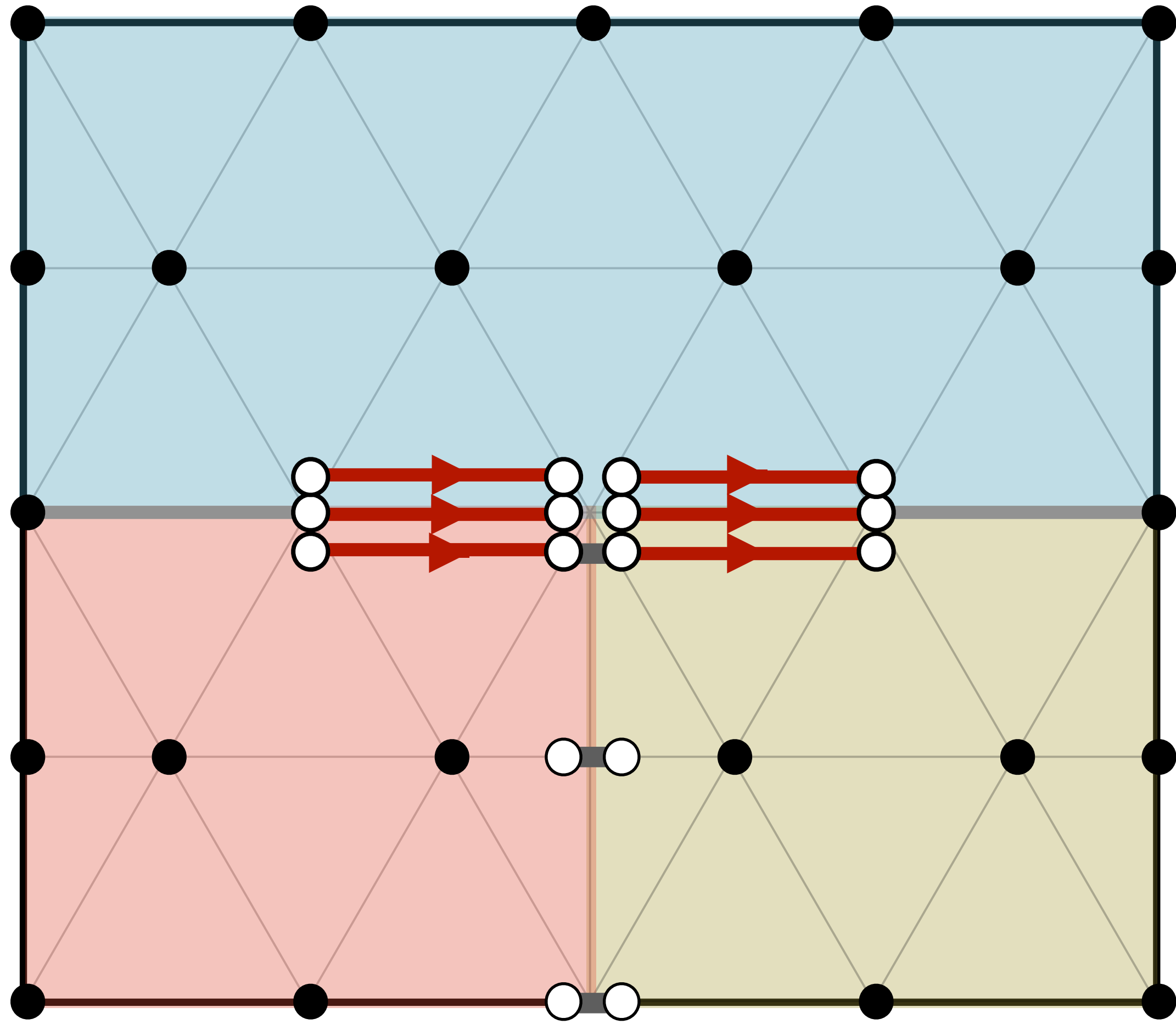
Tape Object

- works similar to scissor method, but adds additional surface elements for tape
- adds one surface per layer in tape

Mesh

Meshing Example: thin-shell with $h-\phi$ formulation

step 3: create edges (and faces)



Thoughts on Visualization

L2-Projection of Current: Projecting the Curl from the h-field

Basic Idea Of L2 Projection: Solve this equation

$$\int_{\Omega} \delta \mathbf{j}^T (\mathbf{j} - \nabla \times \mathbf{h}) dV = 0$$



we want to minimize the error in this bracket

The magnetic field \mathbf{h} is represented on the edges, but we want the current \mathbf{j} . ParaView can't display edge elements, we need to compute the curl of \mathbf{h} and project it on the nodes.

Standard Least Squares

$$\delta \hat{\mathbf{j}}^T \int_{\Omega} \mathbf{N}^T \mathbf{N} dV \hat{\mathbf{j}} = \delta \hat{\mathbf{j}}^T \int_{\Omega} \mathbf{N}^T \mathbf{C} dV \tilde{\mathbf{h}}$$

node values (points to $\hat{\mathbf{j}}$)

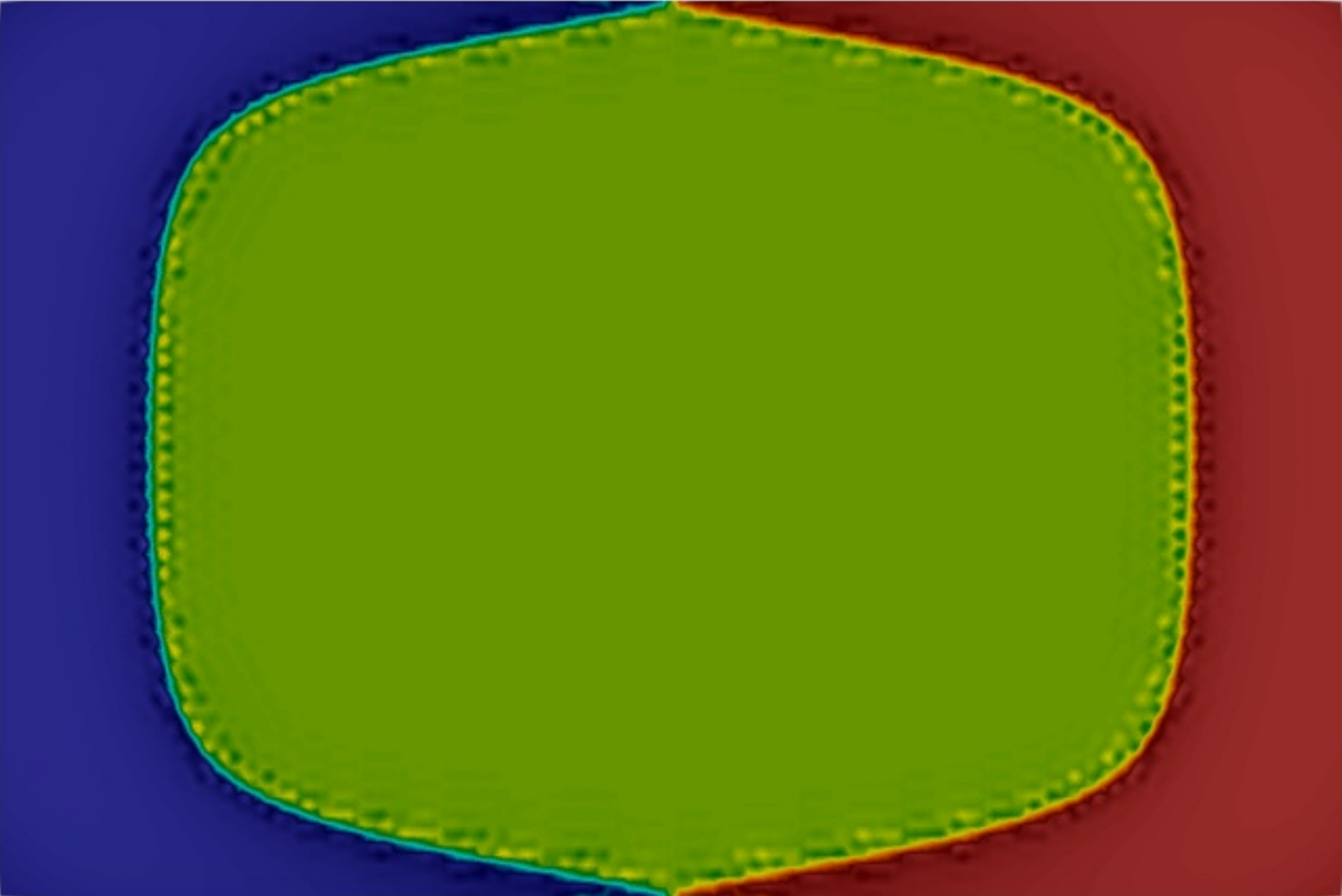
edge dofs (points to $\tilde{\mathbf{h}}$)

curl operator (points to \mathbf{C})

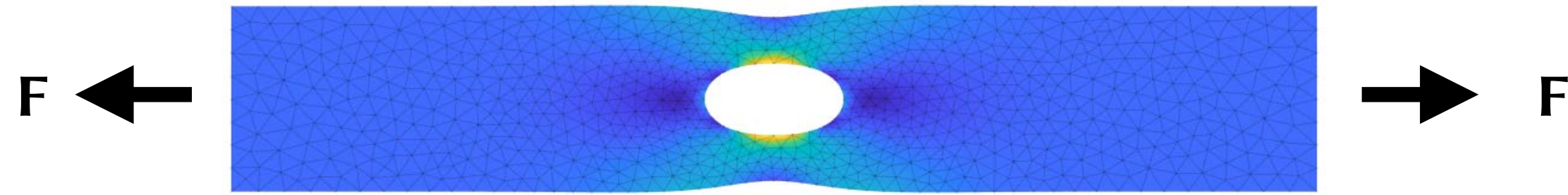
node function (points to \mathbf{N})

"mass matrix" (points to $\int_{\Omega} \mathbf{N}^T \mathbf{N} dV$)

Problem: The projection creates artefacts at the sharp corners



L2-Projection: general thoughts



This problem is very well known:

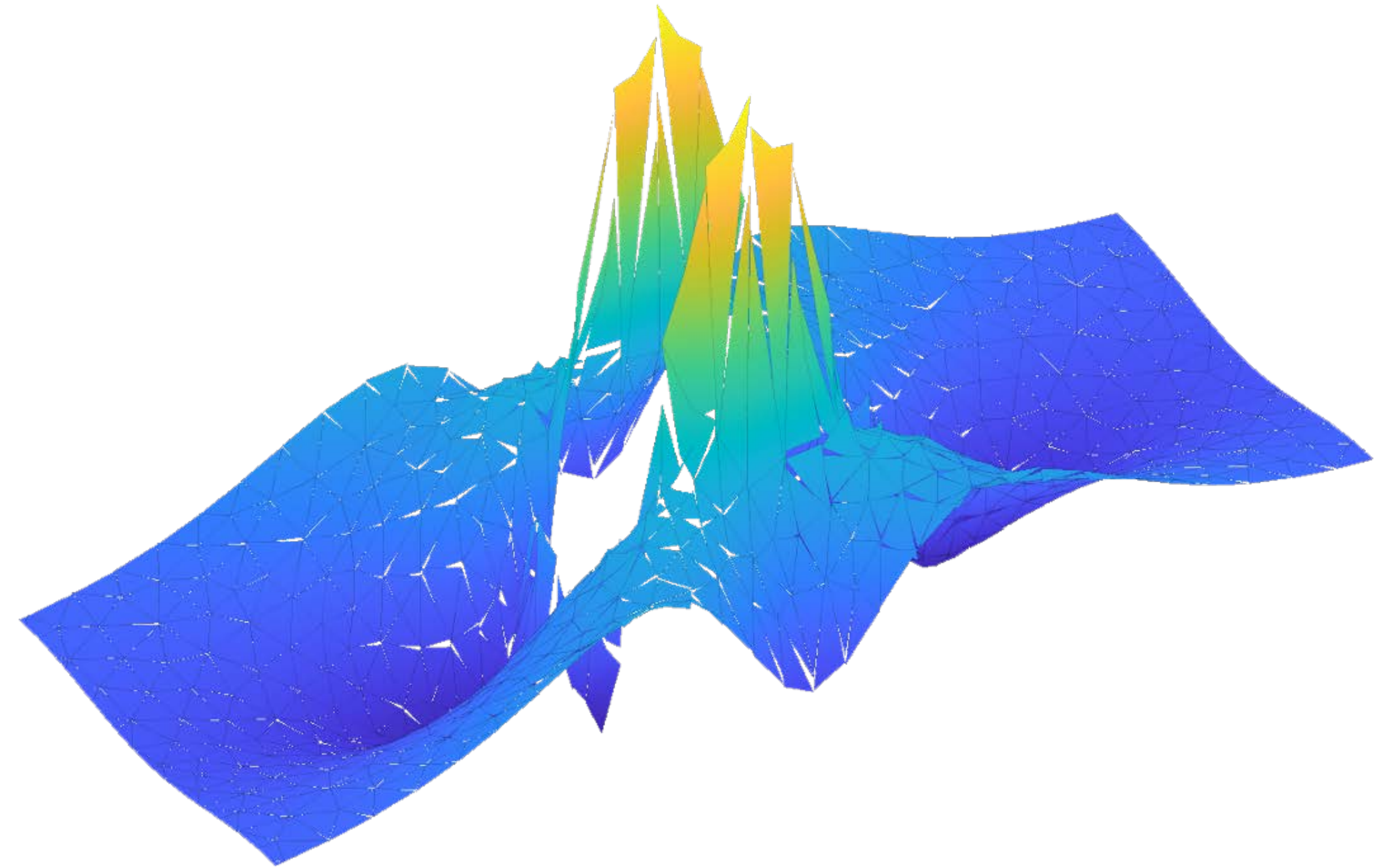
in Lagrangian elements, derivatives are not continuous over element edges!

Hooke's Law

$$\boldsymbol{\sigma} = \boldsymbol{C} : \frac{1}{2} \left[(\nabla \cdot \boldsymbol{u})^T + \nabla \cdot \boldsymbol{u} \right]$$

van Mises stress

$$\sigma_v = \sqrt{\sigma_1^2 + \sigma_2^2 - \sigma_1 \sigma_2 + 3 \tau_{12}^2}$$



L2-Projection of Current: Projecting the Curl from the h-field

Motivation

The magnetic field \mathbf{h} is represented on the edges, but we want the current \mathbf{j} .
 ParaView can't display edge elements, we need to compute the curl of \mathbf{h} and project it on the nodes.

Modifying the equation

$$\int_{\Omega} \delta \mathbf{j}^T (\mathbf{j} + \alpha \nabla \cdot \mathbf{j} - \nabla \times \mathbf{h}) dV = 0$$

↑
 add the gradient and multiply it with a fudge factor. We sacrifice accuracy and obtain smoothness.

Modified Least Squares

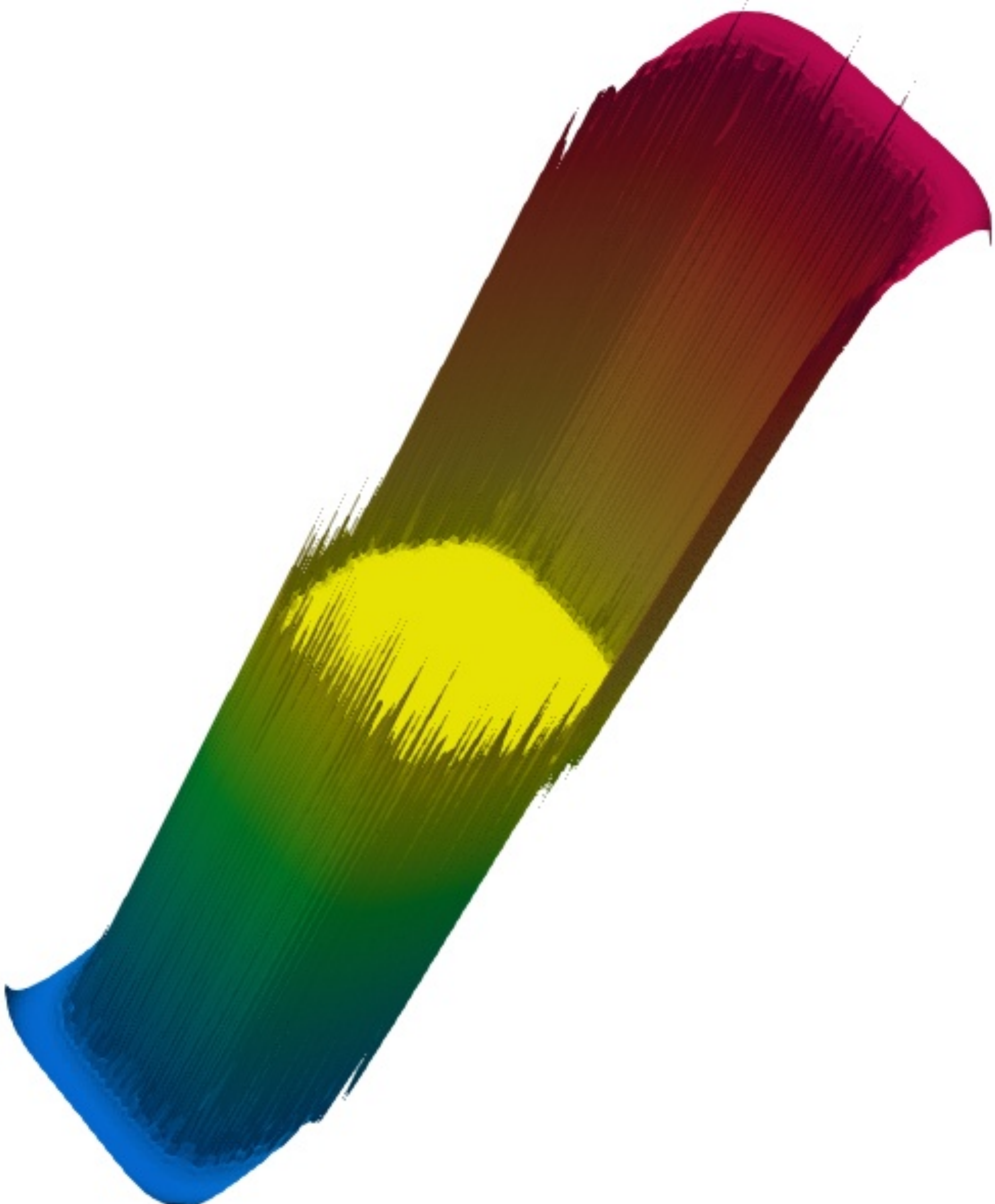
$$\delta \hat{\mathbf{j}}^T \int_{\Omega} (\mathbf{N}^T \mathbf{N} + \alpha \mathbf{B}^T \mathbf{B}) dV \hat{\mathbf{j}} = \delta \hat{\mathbf{j}}^T \int_{\Omega} \mathbf{N}^T \mathbf{C} dV \tilde{\mathbf{h}}$$

"mass matrix" "stiffness matrix" node values edge dofs
 node function curl operator

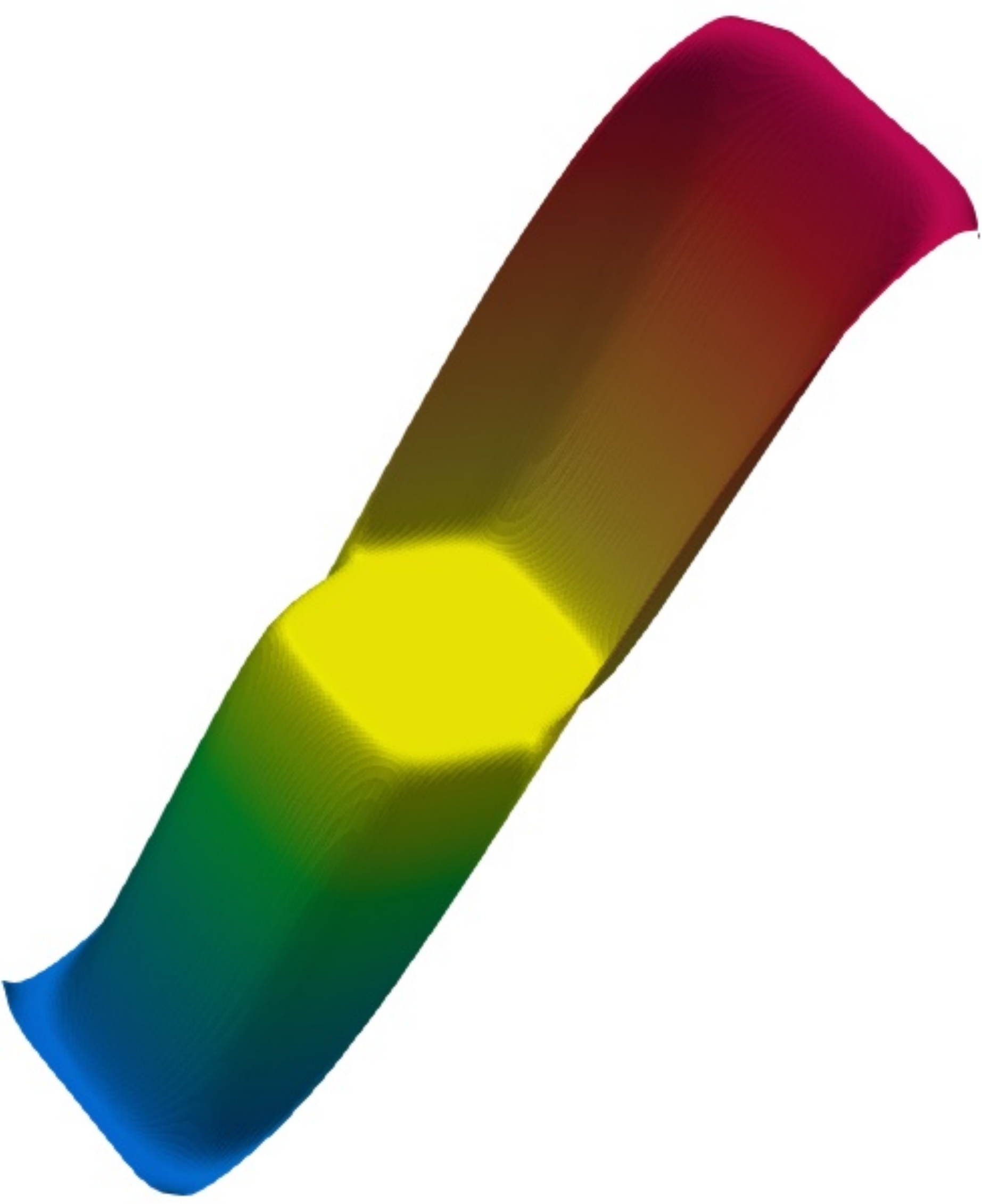


**This Projection blurs out the interface of the current.
 This only impacts the visualization.**

Standard Least Squares



Modified Least Squares



Other Activities

Theory Manual

work in progress



Finite-Element Discretization of the Quasi-Magnetostatic Maxwell Equations

DRAFT

by
Dr.-Ing. Christian Meese

December 2021

Accelerator Technology and Applied Physics Division
Lawrence Berkeley National Laboratory
1 Cyclotron Road, MS4700
Berkeley, CA 94720

1 Element Interpolation Page 10

$$\mathbf{n}_k = \begin{cases} +1 & \text{if the edge } k \text{ has the same direction on } \epsilon \text{ as the owner of the edge} \\ -1 & \text{else.} \end{cases} \quad (1.30)$$

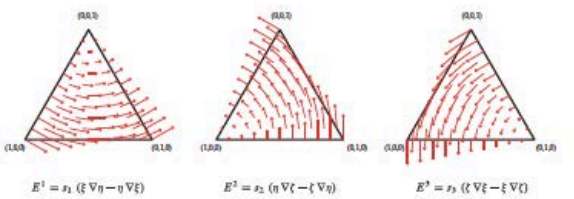


Figure 1.5: Shape function of first order Nédélec triangle

1.5.2 Curl Operator

The entries in the C-operator matrix in Eq. (1.12) are found by first computing the derivatives in the parameter space and second transforming them into the geometry space. The components of Eqs. (1.25) and Eq. (1.26) must first be derived to ξ and η .

$$\begin{bmatrix} E_{1,1}^1 & E_{1,2}^1 & E_{1,3}^1 \\ E_{2,1}^1 & E_{2,2}^1 & E_{2,3}^1 \\ E_{3,1}^1 & E_{3,2}^1 & E_{3,3}^1 \end{bmatrix} = \mathbf{J}^T \begin{bmatrix} E_{1,1}^2 & E_{1,2}^2 & E_{1,3}^2 \\ E_{2,1}^2 & E_{2,2}^2 & E_{2,3}^2 \\ E_{3,1}^2 & E_{3,2}^2 & E_{3,3}^2 \end{bmatrix} \quad (1.31)$$

and

$$\begin{bmatrix} E_{1,1}^2 & E_{1,2}^2 & E_{1,3}^2 \\ E_{2,1}^2 & E_{2,2}^2 & E_{2,3}^2 \\ E_{3,1}^2 & E_{3,2}^2 & E_{3,3}^2 \end{bmatrix} = \mathbf{J}^T \begin{bmatrix} E_{1,1}^3 & E_{1,2}^3 & E_{1,3}^3 \\ E_{2,1}^3 & E_{2,2}^3 & E_{2,3}^3 \\ E_{3,1}^3 & E_{3,2}^3 & E_{3,3}^3 \end{bmatrix} \quad (1.32)$$

where $E_{1,1}^2$ and $E_{1,1}^3$ become zero. After a few rearrangements, the following operator is found

$$\mathbf{C} = \frac{2}{\det \mathbf{J}} \begin{bmatrix} s_1 & s_2 & s_3 \end{bmatrix} \quad (1.33)$$

2 Weak Form Page 22

2.6 Magnetic Field Conform Formulation (h-conform)

The fundamental lemma, Eq. (2.3) is applied to Faraday's law of induction, (2.25). The magnetic field density \mathbf{h} is identified as degree of freedom. Hence, the test function is named $\delta \mathbf{h}$

$$\int_{\Omega} \delta \mathbf{h}^T (\mathbf{h}_j + \text{curl } \epsilon) dV = 0 \quad (2.36)$$

For the second expression, the inverse form of Eq. (2.29) is used

$$\int_{\Omega} \delta \mathbf{h}^T \mathbf{h}_j dV = \int_{\Omega} \delta \mathbf{h}^T \mu_j \mathbf{h}_j dV + \int_{\Omega} \delta \mathbf{h}^T \mu_j \mathbf{h} dV \quad (2.37)$$

Stokes' theorem, Eq. (2.5), is applied to the second expression in Eq. (2.36)

$$\int_{\Omega} \delta \mathbf{h}^T \text{curl } \epsilon dV = \int_{\Omega} (\text{curl } \delta \mathbf{h})^T \epsilon dV + \int_{\partial \Omega} \delta \mathbf{h}^T (\mathbf{n} \times \epsilon) dS \quad (2.38)$$

The boundary integral in Eq. (2.38) is of special interest. If a coupling with the \mathbf{h} -conform formulation is desired, the electric field ϵ can be expressed by means of the voltage v (if it exists) and the vector potential \mathbf{a} . On the other hand, expressing ϵ by means of the current density \mathbf{j} allows current coupling

$$\int_{\partial \Omega} \delta \mathbf{h}^T (\mathbf{n} \times \epsilon) dS = \int_{\partial \Omega} \delta \mathbf{h}^T [(\text{grad } v + \mathbf{a}_j) \times \mathbf{n}] dS = \int_{\partial \Omega} \delta \mathbf{h}^T [\mathbf{n} \times (\rho \mathbf{j})] dS \quad (2.39)$$

In conformity with Eq. (2.35), assuming $v = 0$, the weak form for the \mathbf{h} -conform formulation can be expressed as

$$\int_{\Omega} \delta \mathbf{h}^T \mu_j \mathbf{h}_j dV + \int_{\Omega} \delta \mathbf{h}^T \mu_j \mathbf{h} dV + \int_{\Omega} (\text{curl } \delta \mathbf{h})^T \rho \text{curl } \mathbf{h} dV - \int_{\partial \Omega} \delta \mathbf{h}^T (\mathbf{n} \times \mathbf{a}_j) dS = 0 \quad (2.40)$$

A closer look at Eq. (2.40) reveals that the \mathbf{h} -formulation might be better suited for discretizing a superconducting domain, while the expression μ_j can become very unhandy to compute if μ is not constant. In those cases, the \mathbf{h} -conform formulation might be used to avoid computing this term.

3 Discretizations Page 31

Using these discretizations, the mass Matrix reads

$$\mathbf{M} = \int_{\Omega} \mathbf{B}^T \mu \mathbf{B} dV \quad (3.42)$$

the stiffness matrix only exists for the case $\mu \neq \text{const}$.

$$\mathbf{K} = \int_{\Omega} \mathbf{B}^T \mu_j \mathbf{B} dV \quad (3.43)$$

3.2.3 \mathbf{h} - ϕ Domain Interface

The interface element looks similar as for the \mathbf{h} - \mathbf{a} interface. An example is shown in Fig. 3.3.

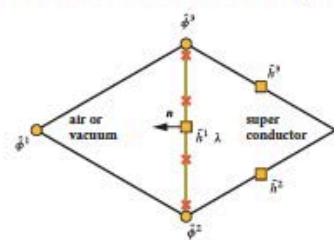


Figure 3.3: TR13 Interface Element for \mathbf{h} - ϕ coupling

The normal and the tangential components of \mathbf{h} along the domain interface must be coupled separately.¹⁴ The components within the contact plane are coupled using a Lagrangian multiplier. The functional reads

$$\Pi = \int_{\Omega} \lambda^T \mathbf{n} \times (\mathbf{h} - \mathbf{h}) dS = 0 \quad (3.44)$$

Using the interpolations $\mathbf{h} \approx \mathbf{E} \cdot \hat{\mathbf{h}}$ for the superconducting domain, and $\mathbf{h} \approx -\mathbf{B} \cdot \hat{\phi}$ for the non-conducting domain, we obtain the following coupling matrix

$$\delta \Pi \approx \begin{bmatrix} \delta \hat{\mathbf{h}}^T & \delta \hat{\phi}^T & \delta \lambda^T \end{bmatrix} \int_{\partial \Omega} \begin{bmatrix} \mathbf{0} & \mathbf{0} & (\mathbf{n} \times \mathbf{E})^T \\ \mathbf{0} & \mathbf{0} & (\mathbf{n} \times \mathbf{B})^T \\ \mathbf{n} \times \mathbf{E} & \mathbf{n} \times \mathbf{B} & \mathbf{0} \end{bmatrix} dS \cdot \begin{bmatrix} \hat{\mathbf{h}} \\ \hat{\phi} \\ \lambda \end{bmatrix} = 0 \quad (3.45)$$

3 Discretizations Page 34

$$\mathbf{M} = \int_{\partial \Omega} \mathbf{N}^T (\mathbf{N} \times \mathbf{n}) dS \quad (3.52)$$

$$\mathbf{K} = \int_{\partial \Omega} \mathbf{N}^T (\mathbf{B} \times \mathbf{n}) dS = -\mathbf{M}^T \quad (3.53)$$

with \mathbf{n} pointing outwards from the ferromagnetic domain.

3.2.5 \mathbf{h} - \mathbf{a} Domain Interface

The \mathbf{h} - \mathbf{a} domain interface between the superconducting domain and the ferromagnetic domain is the same as the one described in Sec. 3.1.4.

3.2.6 Cuts

The integral form Of Ampere's law states that in the quasi-magnetostatic case, the integral over an arbitrary loop L in the domain Ω must be equal to the current I inside the loop.

$$\oint_L \mathbf{h} d\mathbf{l} = I \quad (3.54)$$

This can not be guaranteed if the domain is not simply connected.¹⁵ A remedy is found by applying sufficient cuts in the domain to make the non-superconducting part simply connected, see Fig. 3.5.

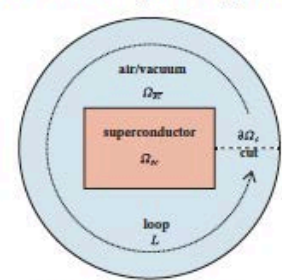


Figure 3.5: Making the domain simply connected, after Dalaz.¹

Along the cut, each node is associated with two but one degrees of freedom for the scalar potential ϕ . Following the notation of Dalaz,^{16,17} we use the notation ϕ_i and ϕ_{i+} . At each node k along the cut, the following condition must be fulfilled, see Fig. 3.6.

References Page 41

[1] Kuczmarski, M. and Iványi, A., *The Finite Element Method in Magnetics*, Akadémia Kiadó, Budapest, Hungary, 2008.

[2] Meiner, G., editor, *The Finite Element Method for Electromagnetic Modeling*, John Wiley & Sons, Inc., Hoboken, NJ, 2008.

[3] Dalaz, J., Geuzaine, C., and Vanderheyden, B., "Finite-Element Formulations for Systems With High-Temperature Superconductors," *IEEE Transactions on Applied Superconductivity*, Vol. 30, No. 3, pp. 1-13, 2019, doi: 10.1109/tasc.2019.2935429.

[4] Zienkiewicz, O. C., Taylor, R. L., and Fox, D. D., *The Finite Element Method for Solid and Structural Mechanics*, Butterworth-Heinemann, Oxford, UK, 7th ed., 2013.

[5] Gupta, K. K. and Meek, J. L., *Finite Element Multidisciplinary Analysis*, AIAA Education Series, American Institute of Aeronautics and Astronautics, Reston, Virginia, 2nd ed., 2003.

[6] Bronshteyn, I. N., Semendyayev, K. A., Musiol, G., and Mühlbig, H., *Handbook of Mathematics*, chap. Newton's Method, Eq. (19.62), Springer-Verlag, Heidelberg, Germany, 6th ed., 2015.

[7] Nedelec, J. C., "Mixed finite elements in \mathbb{R}^3 ," *Numerische Mathematik*, Vol. 35, No. 3, pp. 315-341, 1980, doi: 10.1007/bf01196415.

[8] Schoof, L. A. and Yarberry, V. R., "EXODUS II-A Finite Element Data Model," Tech. Rep. SAND-92-2137, Sandia National Laboratories, 1994.

[9] Nédélec, J. C., "A new family of mixed finite elements in \mathbb{R}^3 ," *Numerische Mathematik*, Vol. 50, No. 1, pp. 57-81, 1986, doi: 10.1007/bf01189668.

[10] Zaglmayr, S., *High Order Finite Element Methods for Electromagnetic Field Computation*, Ph.D. thesis, Johannes Kepler Universität, Linz, Austria, 2006.

[11] Brambilla, R., Grilli, F., Martini, L., Bocchi, M., and Angeli, G., "A Finite-Element Method Framework for Modeling Rotating Machines With Superconducting Windings," *IEEE Transactions on Applied Superconductivity*, Vol. 28, No. 5, pp. 1-11, 2018, doi: 10.1109/tasc.2018.2812884.

[12] Dalaz, J., Harutyunyan, M., Bortot, L., Schops, S., Vanderheyden, B., and Geuzaine, C., "On the Stability of Mixed Finite-Element Formulations for High-Temperature Superconductors," *IEEE Transactions on Applied Superconductivity*, Vol. 31, No. 6, pp. 1-12, 2021, doi: 10.1109/tasc.2021.3098724.

[13] Meiner, A., *Numerik linearer Gleichungssysteme*, Viewig & Sohn Verlag, Wiesbaden, 3rd ed., 2008.

[14] Arsenault, A., Sirois, F., and Grilli, F., "Implementation of the H-phi Formulation in COMSOL Multiphysics for Simulating the Magnetization of Bulk Superconductors and Comparison With the H-Formulation," *IEEE Transactions on Applied Superconductivity*, Vol. 31, No. 2, pp. 1-11, 2021-3, doi: 10.1109/tasc.2020.3033998.

- detailed derivation of modern \mathbf{h} - \mathbf{a} and \mathbf{h} - ϕ weak forms
- contains full description of element formulation
- explains algorithms such as edge- and face generation
- all equations in standardized finite-element notation (as in Zienkiewicz, Belytschko, Bathe, ...)
- textbook-like!

Other Activities

Berkeley Lab Scalable Solvers Group

- work with STRUMPACK development team on optimized solver interface



Commonwealth Fusion Systems & MIT PSFC

- development of material properties library
- be part of BELFEM
- also stand-alone, being linked by other codes such as SparseLizard

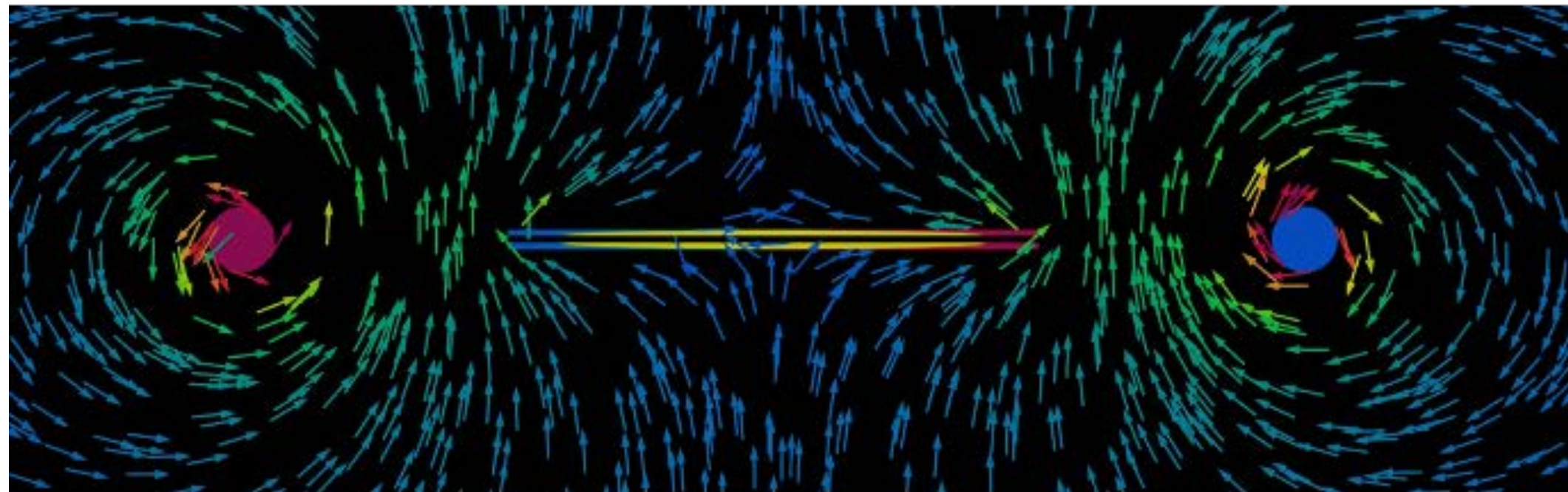
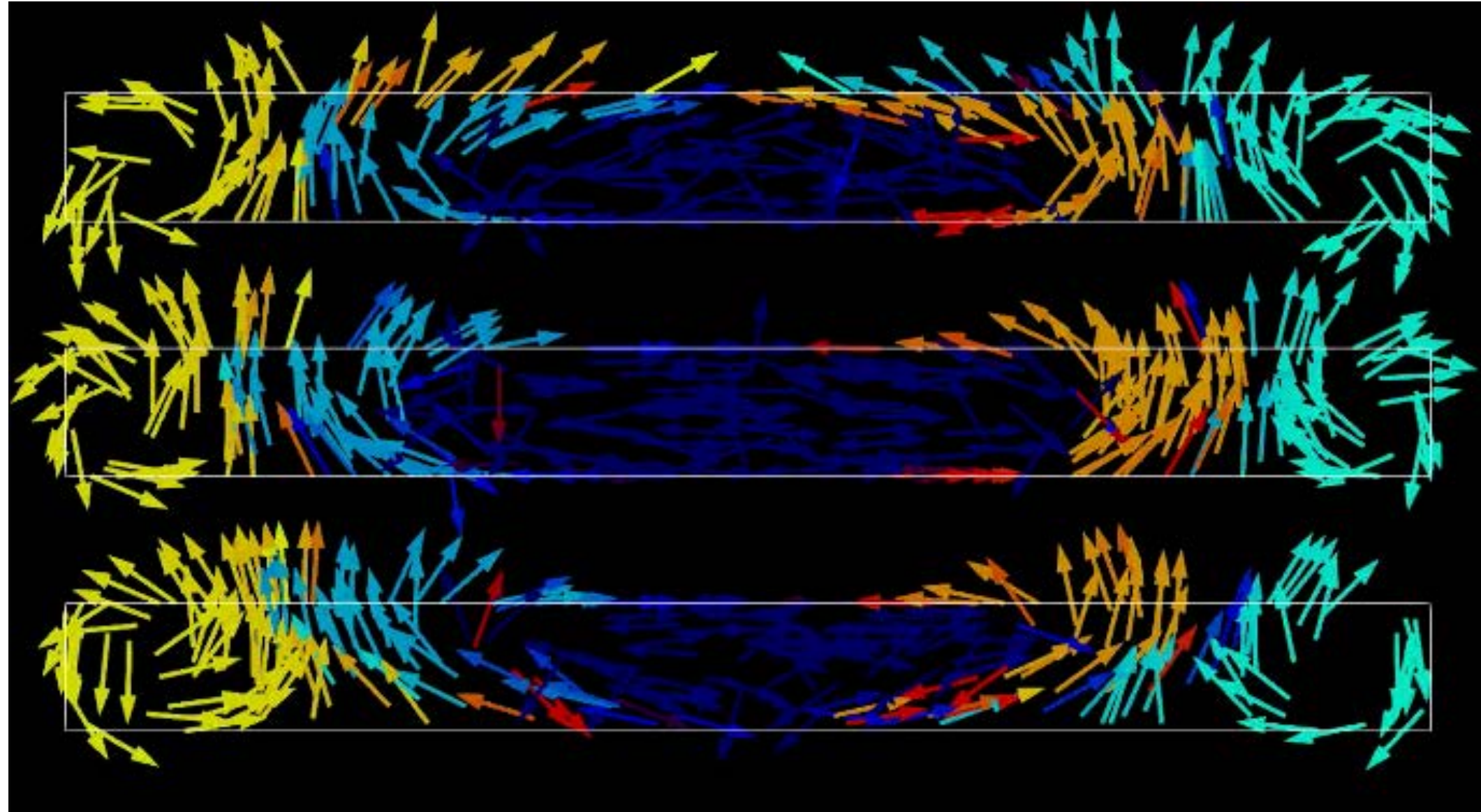


Politecnico di Torino, DNERG (starting this summer)

- TS current sharing models & benchmarking against COMSOL



Summary



- building finite-element framework tailored to HTS cable & magnet development needs
- support modern mixed formulations such as h - a and h - ϕ
- support thin shells & multi-physics (work in progress)
- code designed to run in parallel on HPC node
- work on textbook-like theory manual
- work on a stand alone material database

Outlook



during the next months:

- complete thin shell implementation for and $h-\phi$
- complete work on TS-quenching & current sharing model (2D) continue work on STRUMPACK interface

during the next year:

- work on geometry-preprocessor for 3D current sharing
- streamline workflow for real world applications
- work on convective cooling module
- publish code under BSD-3-like license