Finite-Element-Modeling of HTS Cables: Progress Update

Christian Messe

BERKELEY LAB

U.S. MAGNET DEVELOPMENT PROGRAM

07/20/2021

Overview

- About Me
- Project Goals
- Finite Element Code Development
  - Motivation
  - Current Structure and Previous Examples
  - Special things on Magnetic Formulations
- Near- and Longterm Roadmaps
- Possible Cooperations

Christian Messe

About Me

2017

PhD of Aerospace Engineering
University of Stuttgart, Germany

- analysis of hypersonic vehicles
- finite element modeling
- scientific code development

2018

Postdoc at University of Colorado
DARPA Transformative Design Program:
Topology Optimization

- B-Spline based hierarchical meshes
- finite element kernel development

2019
-2020

DLR

Scientist at German Aerospace Center

- thermal analysis of spacecraft reentry

June
2021

Postdoc at Berkeley Lab
Accelerator Technology and Applied Physics Division

Goals Within the MPD Project

short term:

support conceptual design of HTS cables and magnets

- magnetization
- current distribution
- quenching

long term:

- development of new methods for HTS modeling (adaptive meshing, XFEM ...)
- creation of a hierarchical, multiphysics toolbox for HTS cable modeling

general idea:
modify my pre-existing finite element code for this application

General Thoughts on Finite-Element-Modeling

understanding the difference to common problems

Finite Element Code Development

Motivation: Why use a custom code?

Reasons for developing a custom code

- full control over modeling approach
- forced to fully understand modeling
- maximum flexibility of data structure
- free choice of pre- and post processing tools
- no license fees

Tradeoffs

- more developing overhead
- limited human resources

Finite Element Code Development

Design Requirements for a Custom C++ Framework

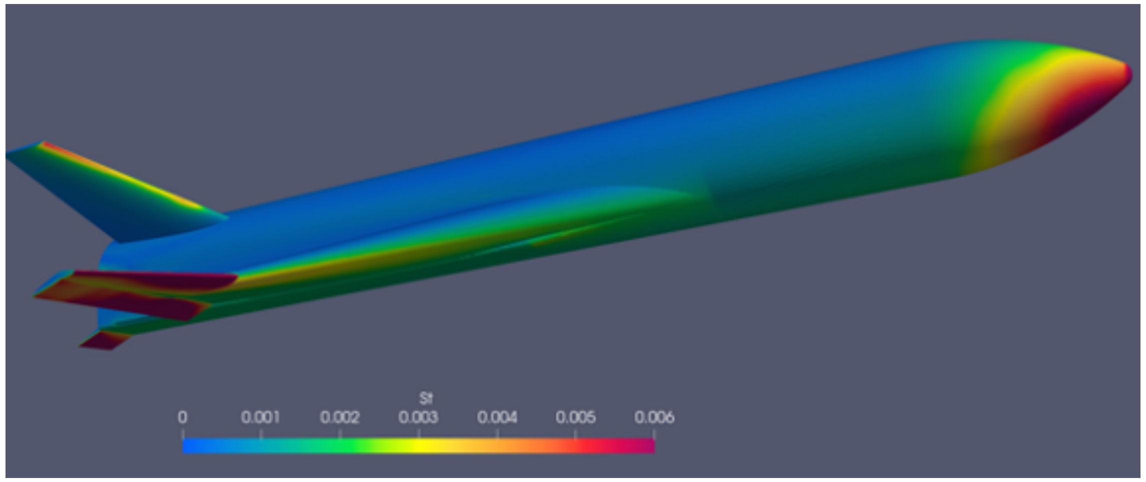
- maintainable and flexible code structure
- portability (gcc, clang, icc, linux / darwin)
- parallel support (MPI)
- simple MATLAB-like dense linear algebra (Armadillo, Blaze)
- interface to standard open source solver libraries (MUMPS / PETSc)

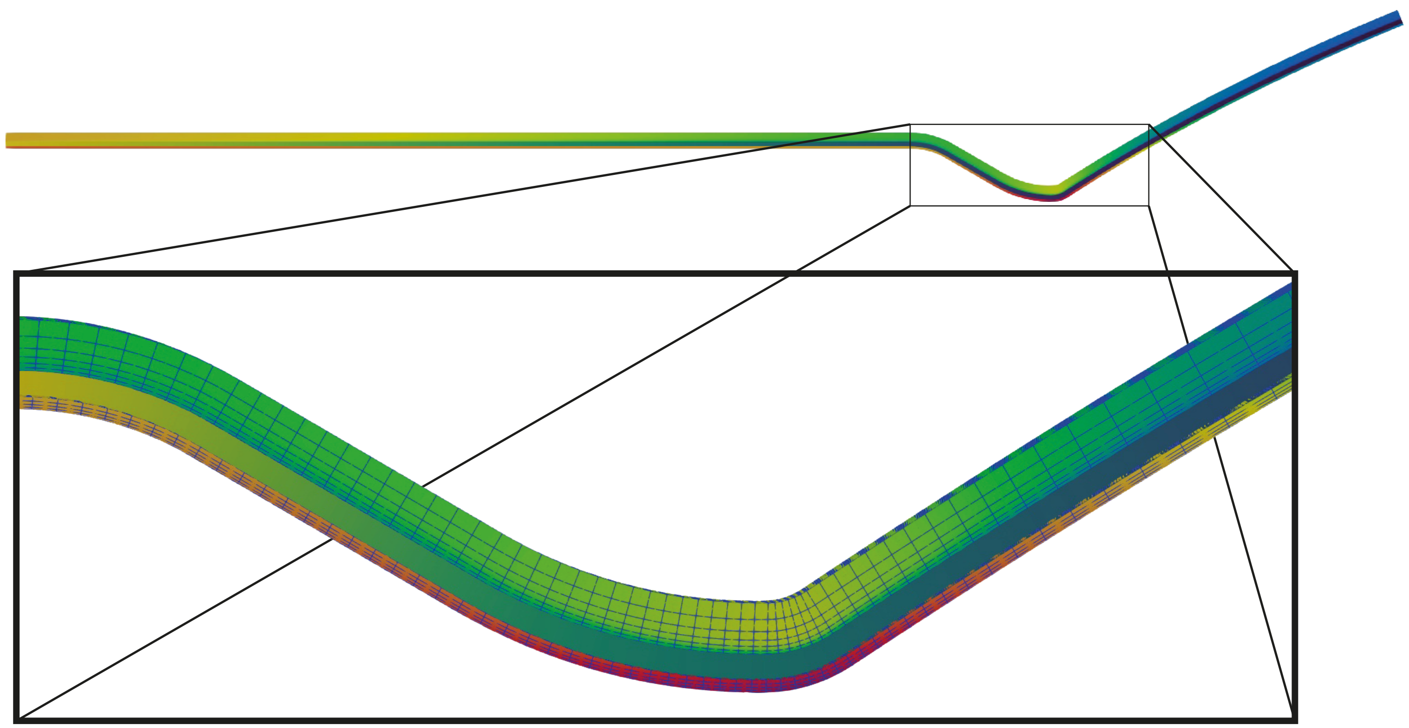
- data-IO support for common open-source data formats:
  - Preprocess: GMSH (*.msh)
  - Postprocess: ParaView (*.vtk/*.exo)
  - Special Purpose: Python (*.xml/*.hdf5)

Finite Element Code Development

Development History

- started coding in December 2018 as personal project under open-source license
- Spring 2019: Involvement in DLR TRANSIENT project
- Winter 2020: involvement in industry project on cryogenic rocket engine cooling





Finite Element Code Development

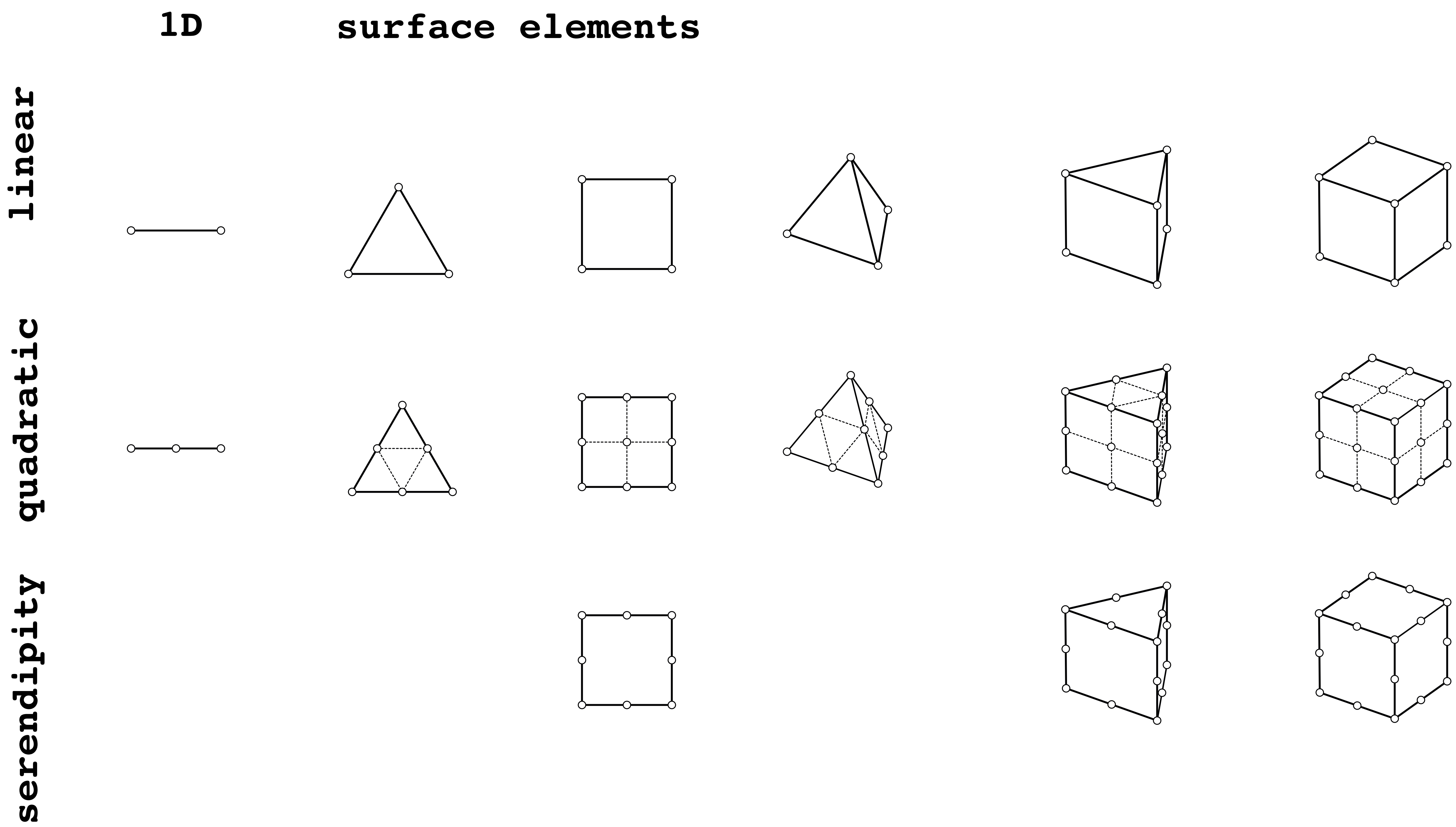
Code Architecture

	mesh	FEM		math	physics	
	Element Database	Element Database	Timestepping	Integration	LSQ mapping	fluid database
	Blocks & Sets	Blocks & Sets	Newton-Raphson	graph algorithms	thermal	gasdynamics
	Field Data	Post Process	B-Spline Lookup Tables	analytic geometry	linear elastics	chemistry
	Parallel Distribution	DOF Management	Solver Interface	tensor algebra	electro-magnetics	material database
wrappers	IO	Comm	Partitioning	LINALG	Sparse	Solver
third party libraries	HDF5 / EXODUS	MPI	METIS	Armadillo / Blaze	SuiteSparse	MUMPS/ PETSc

code size: ca. 175 kloc

Finite Element Code Development

Current Finite Element Database



Example of Previous Application

demonstrate maturity of codebase

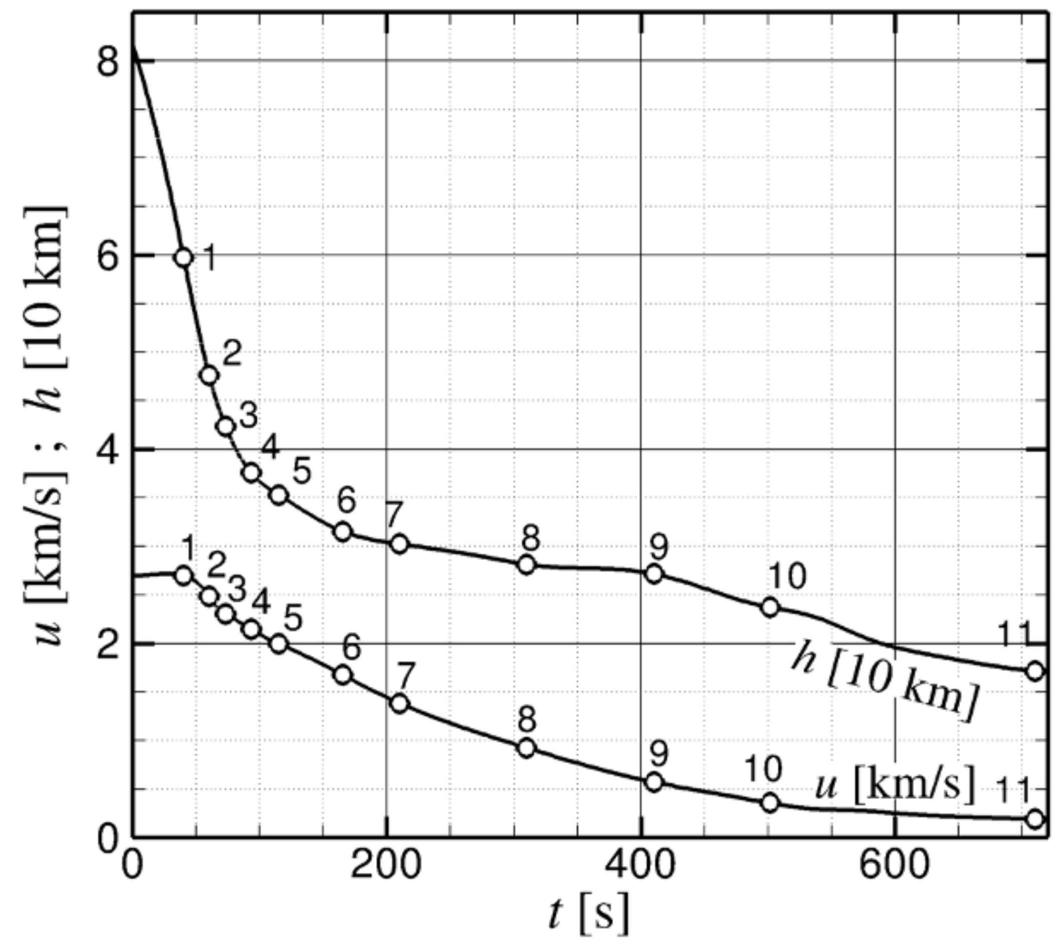
Finite Element Code Development

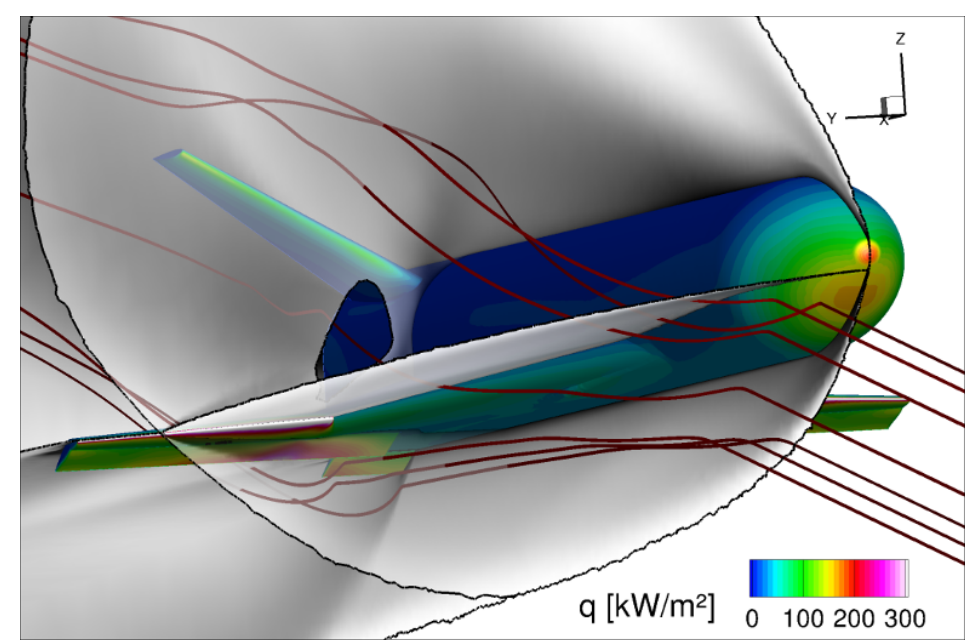
Example of Previous application

Goal: compute surface temperatures during spacecraft reentry and find appropriate thermal shielding and insulation

- precomputed reentry trajectory

- computed wall temperature-dependent heat load using the DLR Tau code for Tw=300K, 600K, 900K, 1200K





Karl et. al: Aerothermal Databases and Heat-Load Prediction for Re-Usable Launch Vehicle Configurations, HiSST: 2nd International Conference on High-Speed Vehicle Science Technology, 2020, Burges, Belgium

Classical Workflow

geometry generator [Python / GMSH]

- specifies geometric shape
- specifies blocks and sidesets
  (e.g. groups with same material)

↓

mesh generator

- generates nodes and elements
- provides node-to-element connectivity

↓

kernel [FEM code]

- handles list of equations to be solved
- distributes data among processors

↓

DOF manager

- creates degrees of freedom based on mesh
- builds the system of equations

↓

Solver [PETSc]

- creates degrees of freedom based on mesh
- creates finite elements based on mesh elements

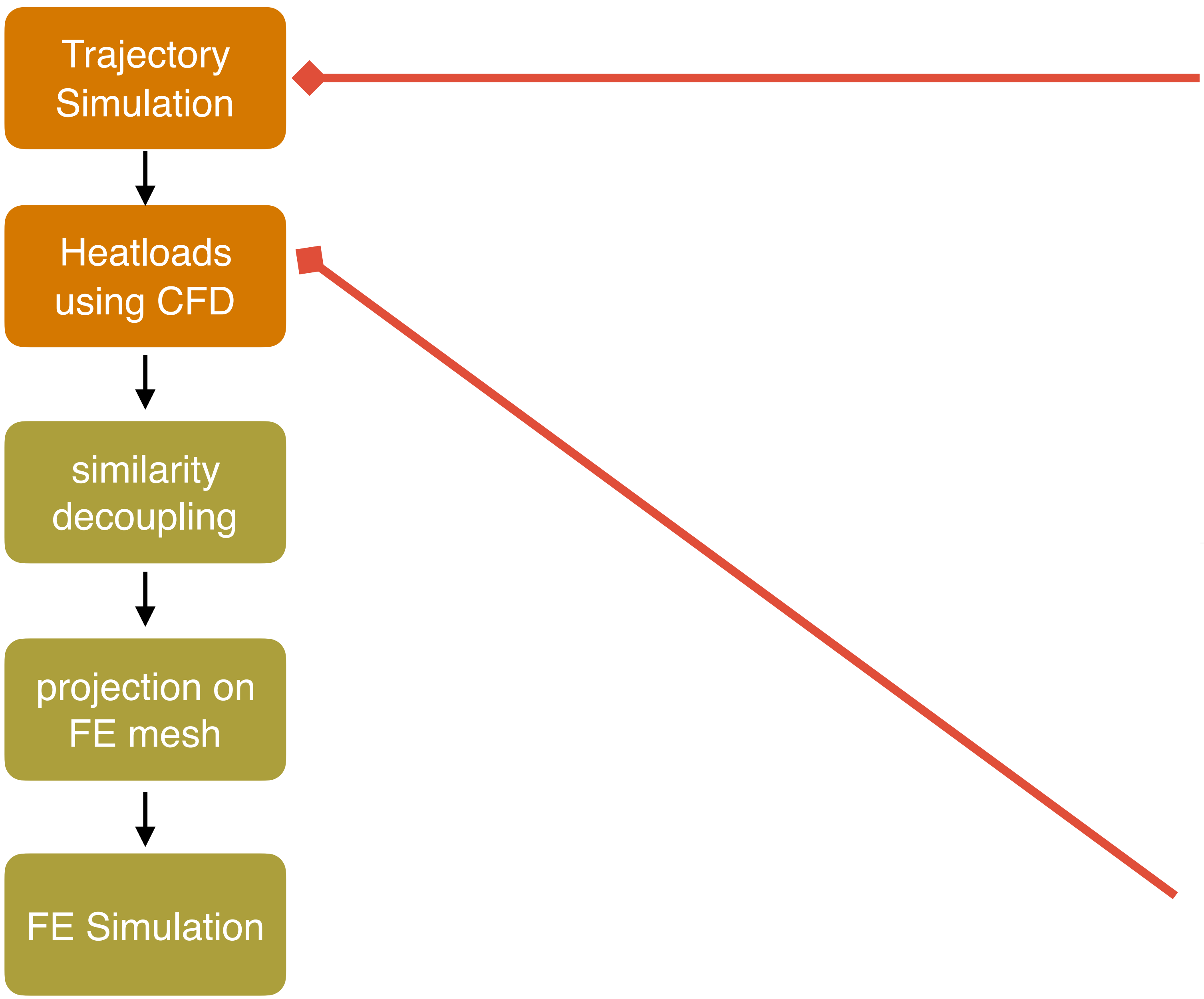
↓

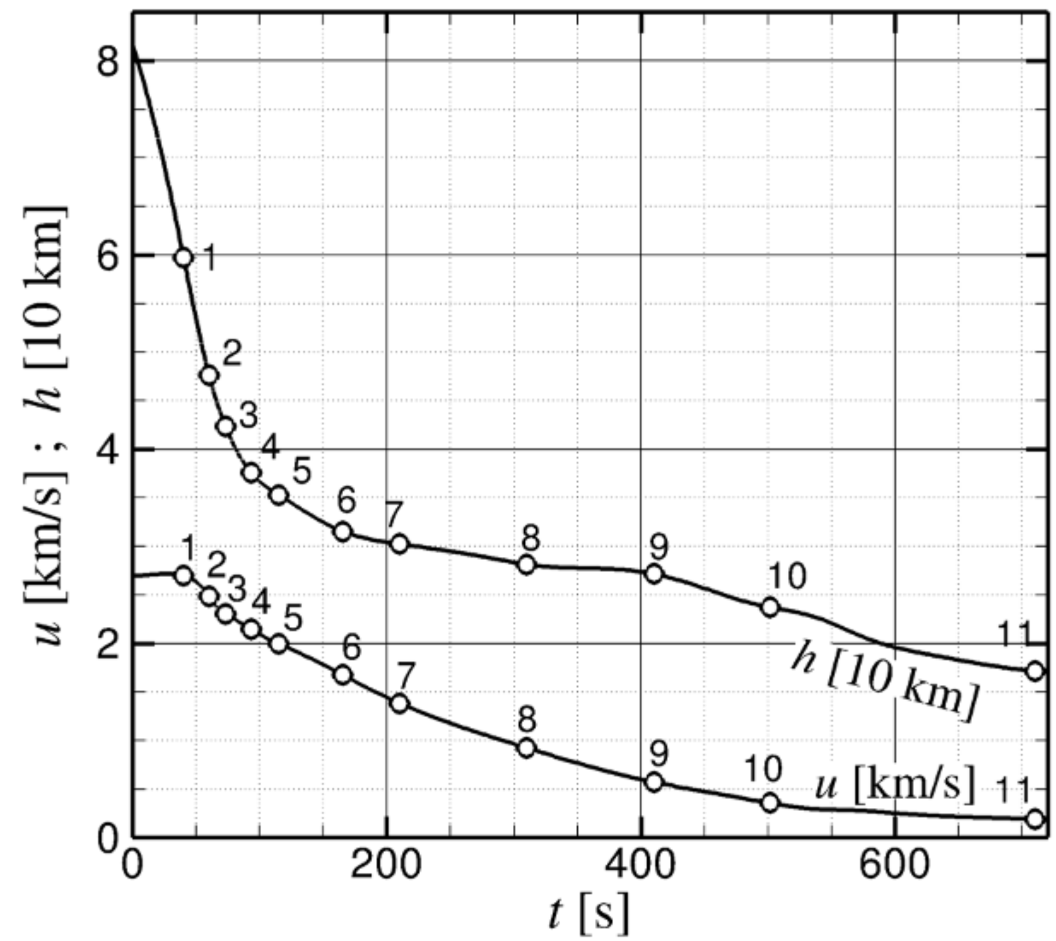
Post Processor [ParaView]

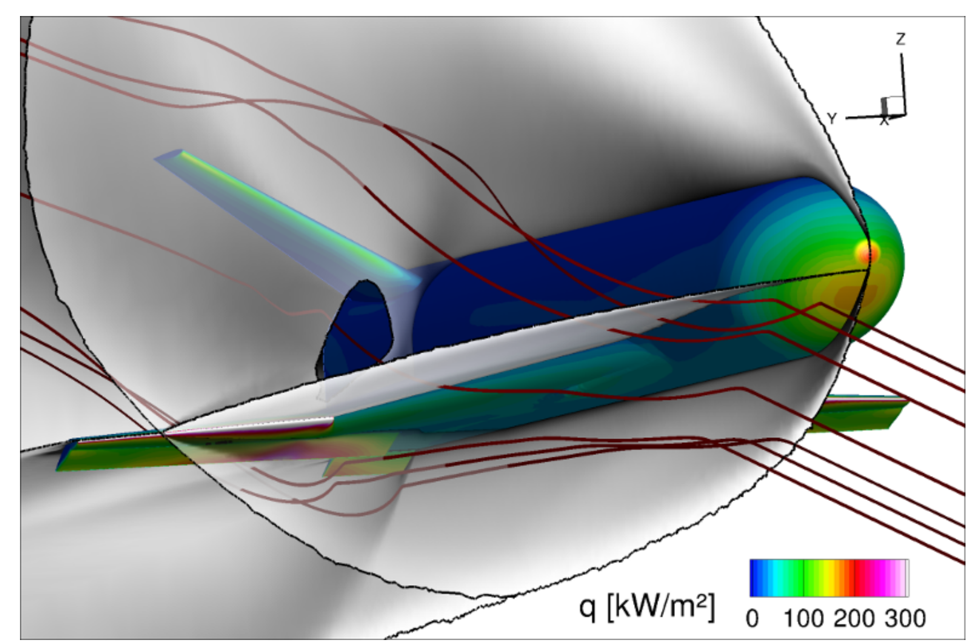
- visualizes the data

Finite Element Code Development

Example of Previous application

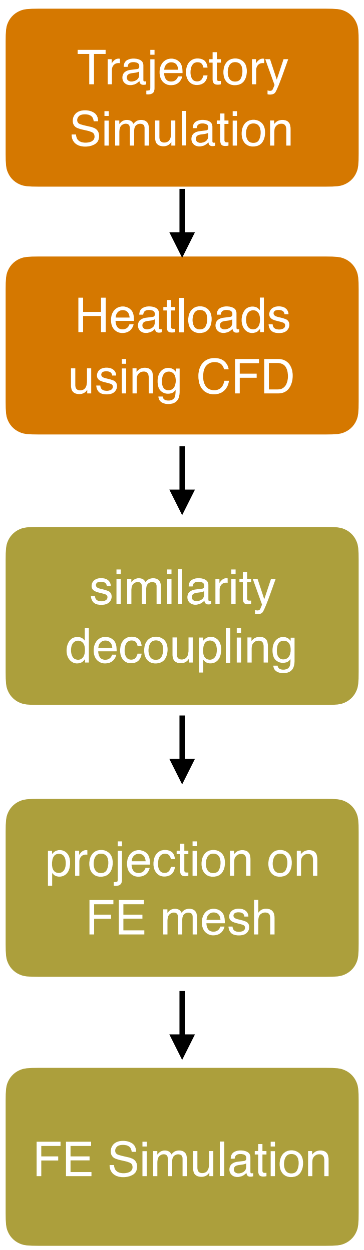






Finite Element Code Development

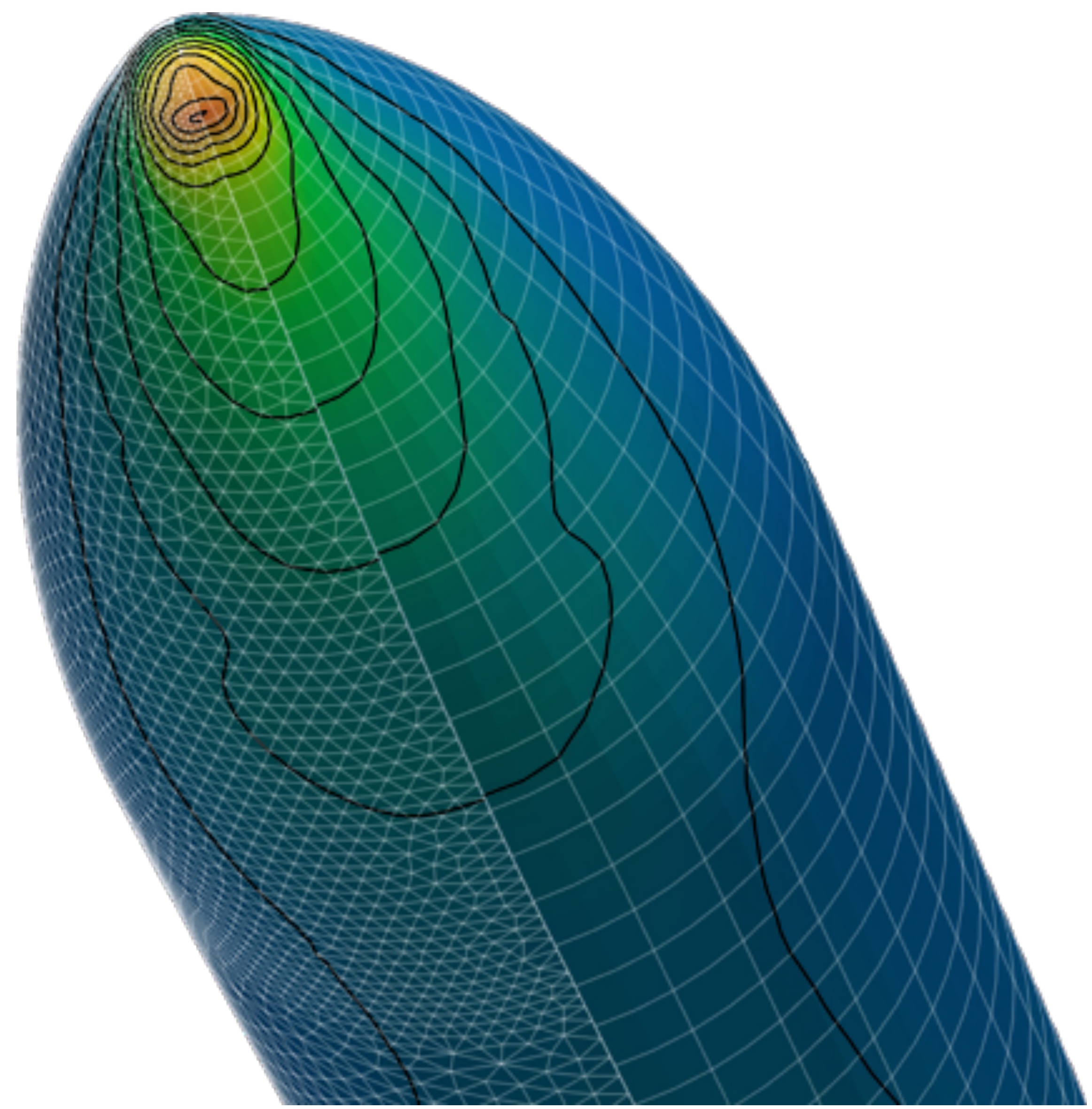
Example of Previous application



$$St_\infty = \frac{\dot{q}}{\rho_\infty \, u_\infty \, (h_r - h_w)}$$

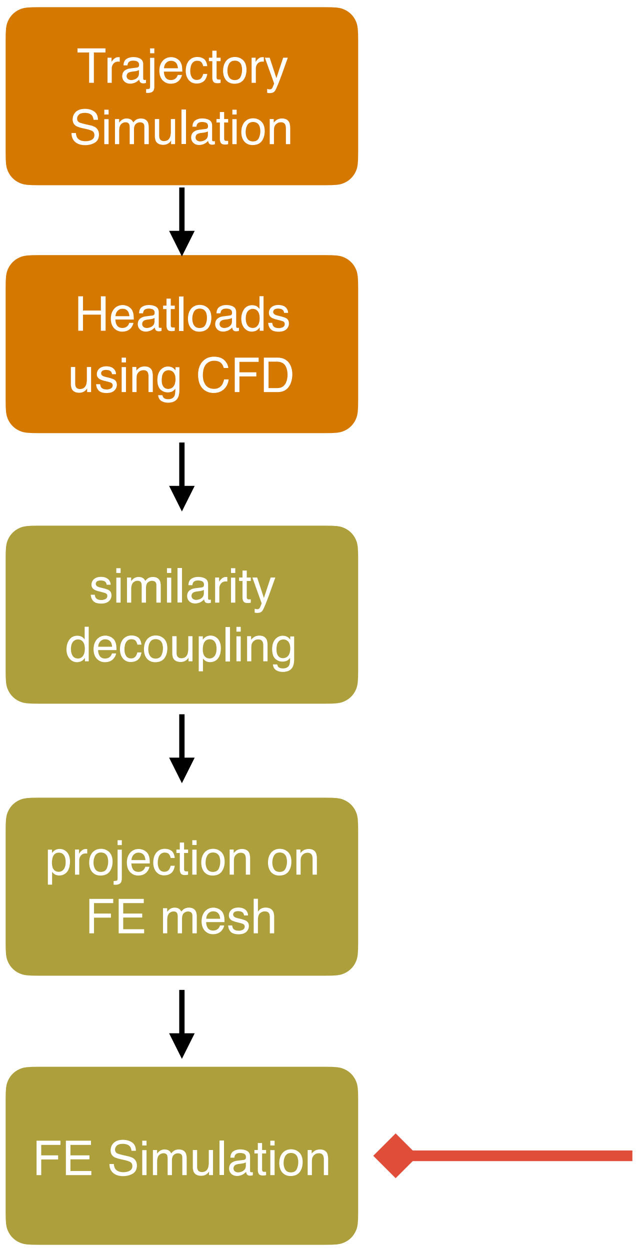
Galerkin Least Squares projection

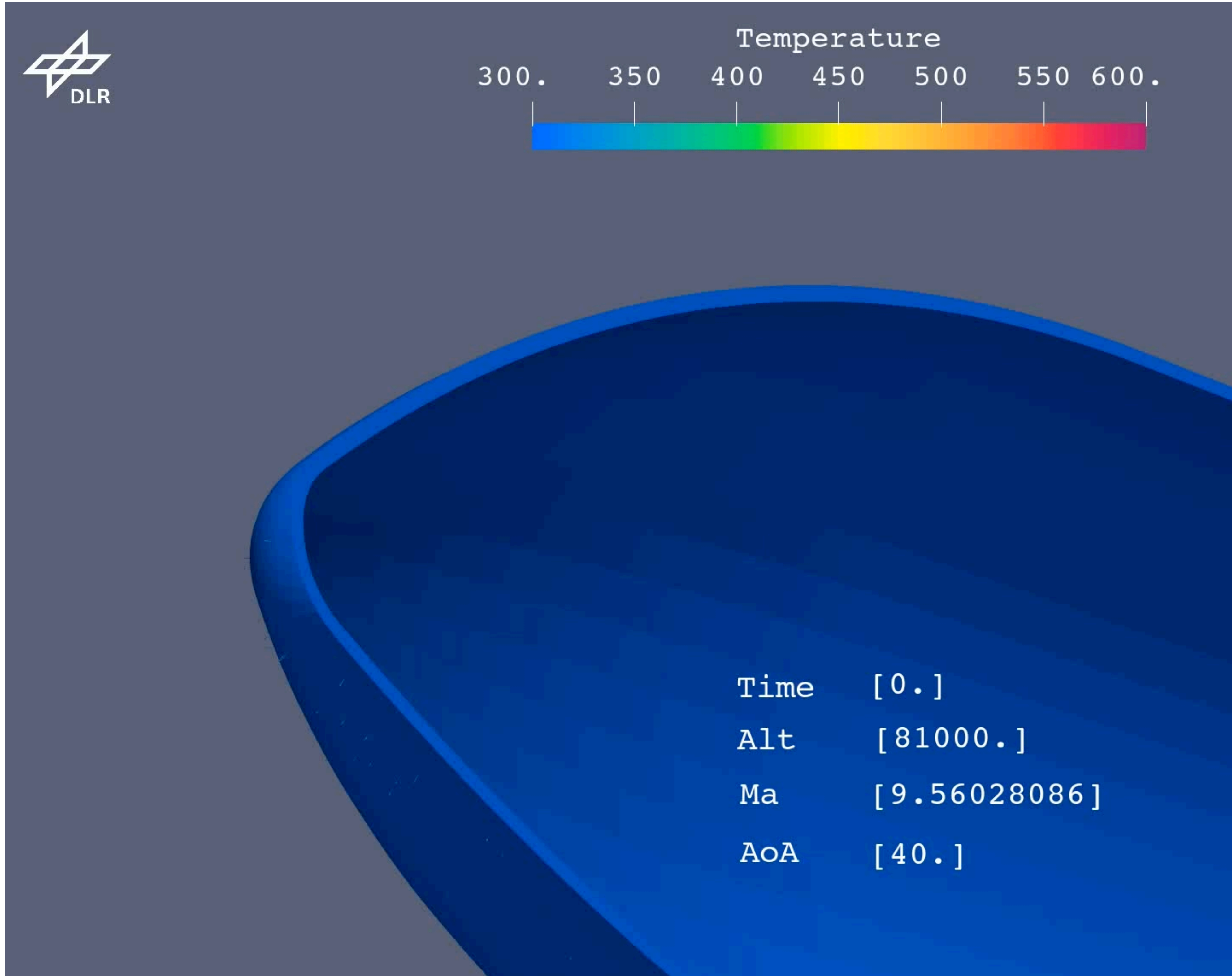
$$\int_\Omega \boldsymbol{\delta\phi}^{\mathbf{T}} \left[\boldsymbol{N}^{\mathbf{T}} \boldsymbol{N} \hat{\boldsymbol{\phi}} - \boldsymbol{N}^{\mathbf{T}} \phi \right] \mathrm{d}V = 0$$



Finite Element Code Development

Example of Previous application





Finite-Element Modeling of Electromagnetics

first thoughts, mathematical approach and curiosities

Finite-Element Modeling of Electromagnetics

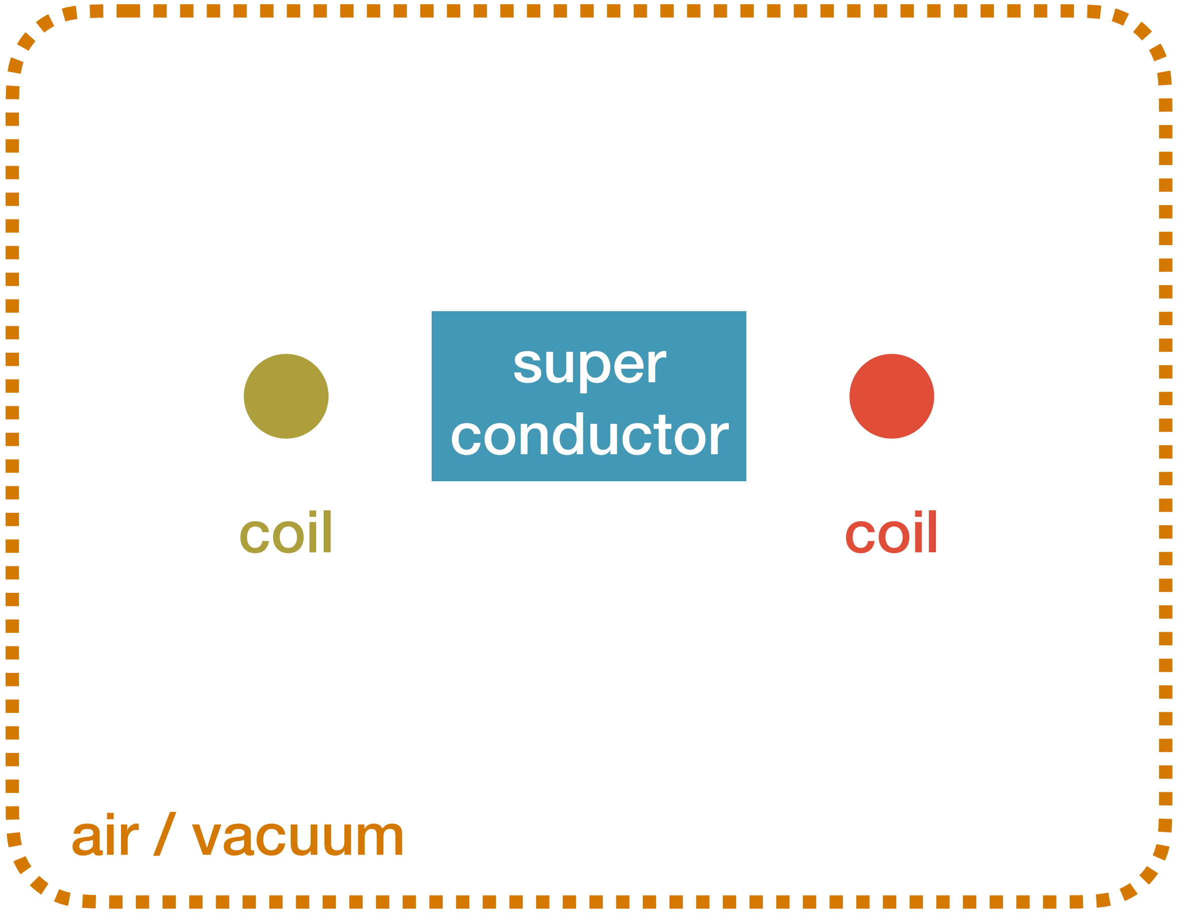
first thoughts, mathematical approach and curiosities

first steps to be taken:

- understand state-of-the-art modeling approaches
- unique aspects of electromagnetics compared to other FE problems
- implement relevant benchmark problems
- identify challenges and possible solutions

Finite-Element Blocks

Example of use: Superconductor in electric field



- quasistatic approach
  (neglecting displacement currents)

$$\text{div}\, \boldsymbol{d} = \rho_v \rightarrow 0$$

$$\text{div}\, \boldsymbol{b} = 0$$

$$\text{curl}\, \boldsymbol{e} = -\frac{\partial}{\partial t}\boldsymbol{b}$$

$$\text{curl}\, \boldsymbol{h} = \boldsymbol{j} + \frac{\partial}{\partial t}\boldsymbol{d} \rightarrow 0$$

The Formulation Problem

b-conform formulation (aka a-v)

$$\int_{\Omega} \boldsymbol{\delta a}^{\mathbf{T}} \boldsymbol{\sigma} \frac{\partial}{\partial t} \boldsymbol{a} \, \mathrm{d}V + \int_{\Omega} (\mathrm{curl}\, \boldsymbol{\delta a})^{\mathbf{T}} \boldsymbol{\nu} \, \mathrm{curl}\, \boldsymbol{a} \, \mathrm{d}V + \int_{\Omega} \boldsymbol{\delta a}^{\mathbf{T}} \boldsymbol{\sigma} \, \mathrm{grad}\, v \, \mathrm{d}V = \int_{\partial\Omega} \boldsymbol{\delta a}^{\mathbf{T}} (\boldsymbol{h} \times \boldsymbol{n}) \, \mathrm{d}S.$$

$$\int_{\Omega} (\mathrm{grad}\, \delta v)^{\mathbf{T}} \boldsymbol{\sigma} \frac{\partial}{\partial t} \boldsymbol{a} \, \mathrm{d}V + \int_{\Omega} (\mathrm{grad}\, \delta v)^{\mathbf{T}} \boldsymbol{\sigma} \, \mathrm{grad}\, v \, \mathrm{d}V = \int_{\Omega} (\mathrm{grad}\, \delta v)^{\mathbf{T}} \boldsymbol{j} \, \mathrm{d}V,$$

dof: magnetic vector potential **a** and electric voltage **v**

problem: infinite electric conductivity of super conductor $\sigma \rightarrow \infty$

h-conform formulation (aka h-ϕ)

$$\int_{\Omega} \boldsymbol{\delta h}^{\mathbf{T}} \frac{\partial}{\partial t} (\boldsymbol{\mu}\, \boldsymbol{h}) \, \mathrm{d}V + \int_{\Omega} (\mathrm{curl}\, \boldsymbol{\delta h})^{\mathbf{T}} \boldsymbol{\rho} \, \mathrm{curl}\, \boldsymbol{h} \, \mathrm{d}V = \int_{\partial\Omega} \boldsymbol{\delta h}^{\mathbf{T}} \boldsymbol{e} \times \boldsymbol{n} \, \mathrm{d}S$$

dof: magnetic flux density **h**

problem: infinite electric resisitvity of air $\rho \rightarrow \infty$

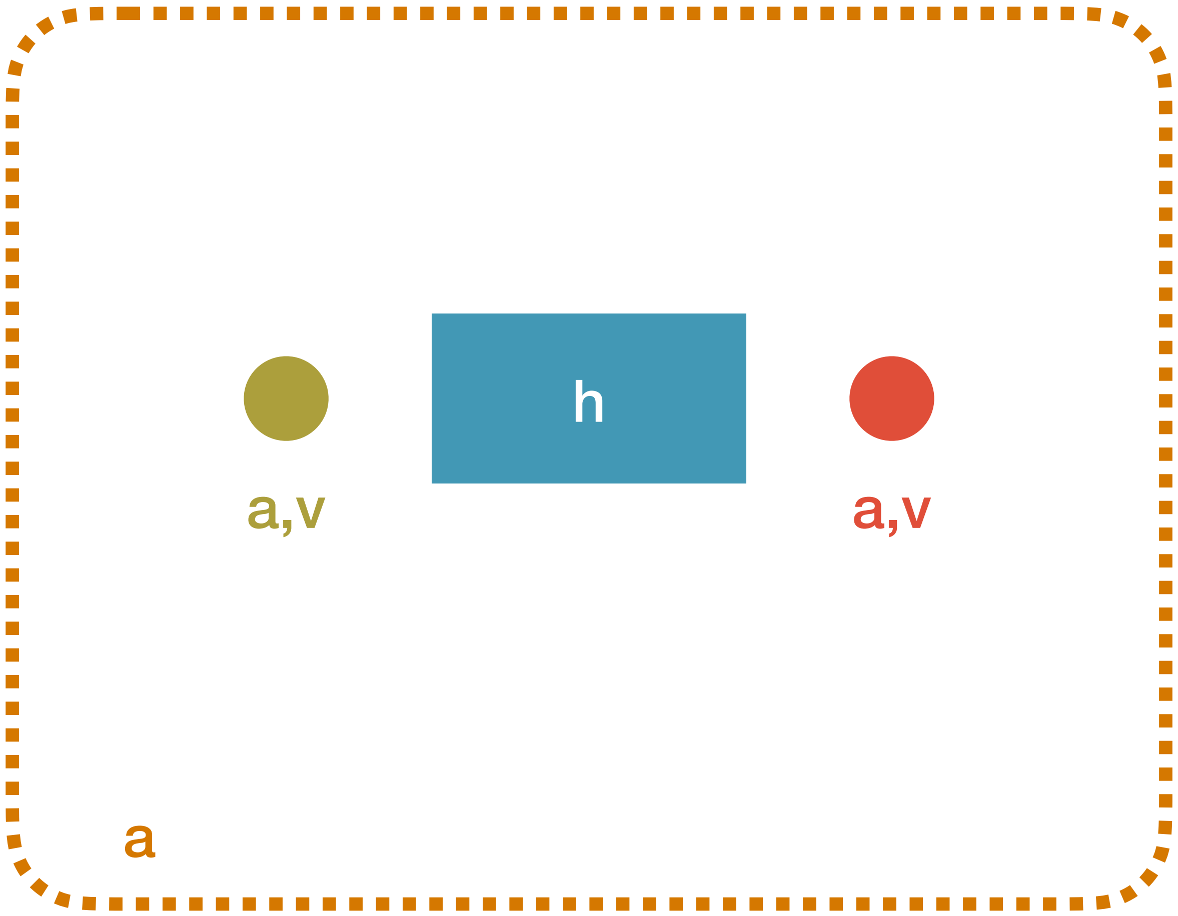
$$\boldsymbol{b} = \mathrm{curl}\, \boldsymbol{a}$$

$$\nu = \mu^{-1}$$

$$\sigma = \rho^{-1}$$

Way Out

Block-Wise Selection of Formulation



basic idea:

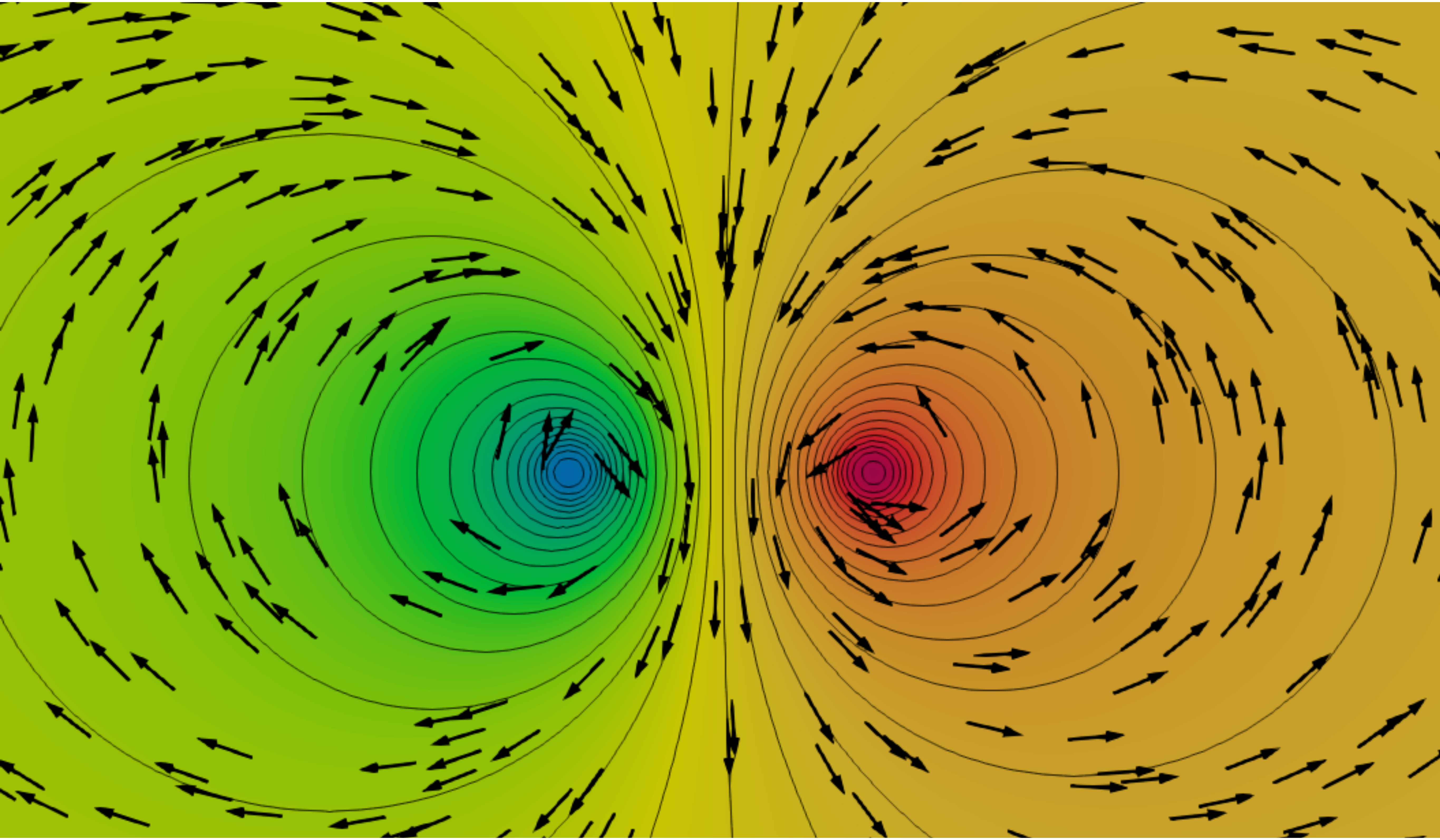
- use a-v formulation in air and h-ϕ formulation in superconductor
- use appropriate coupling at interfaces

→ different blocks have different equations and degrees of freedom

→ requires overhaul of DOF-managing module in code (in progress)

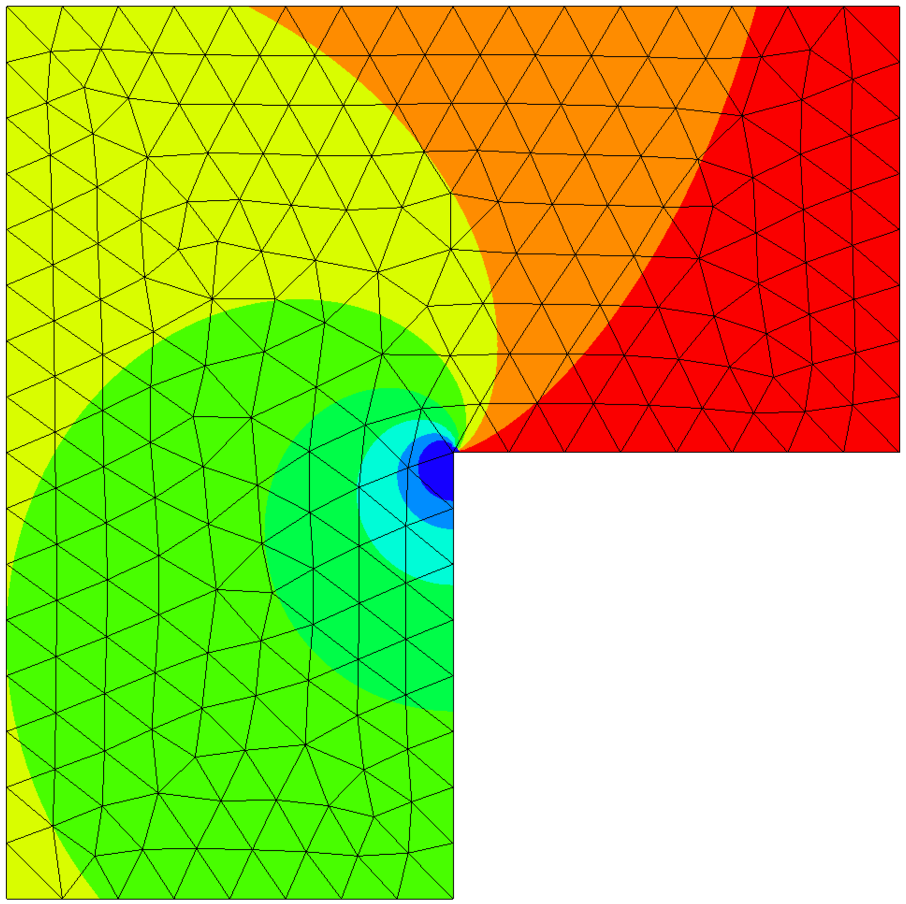
Proof of Concept

Dipole-Problem with current finite element code

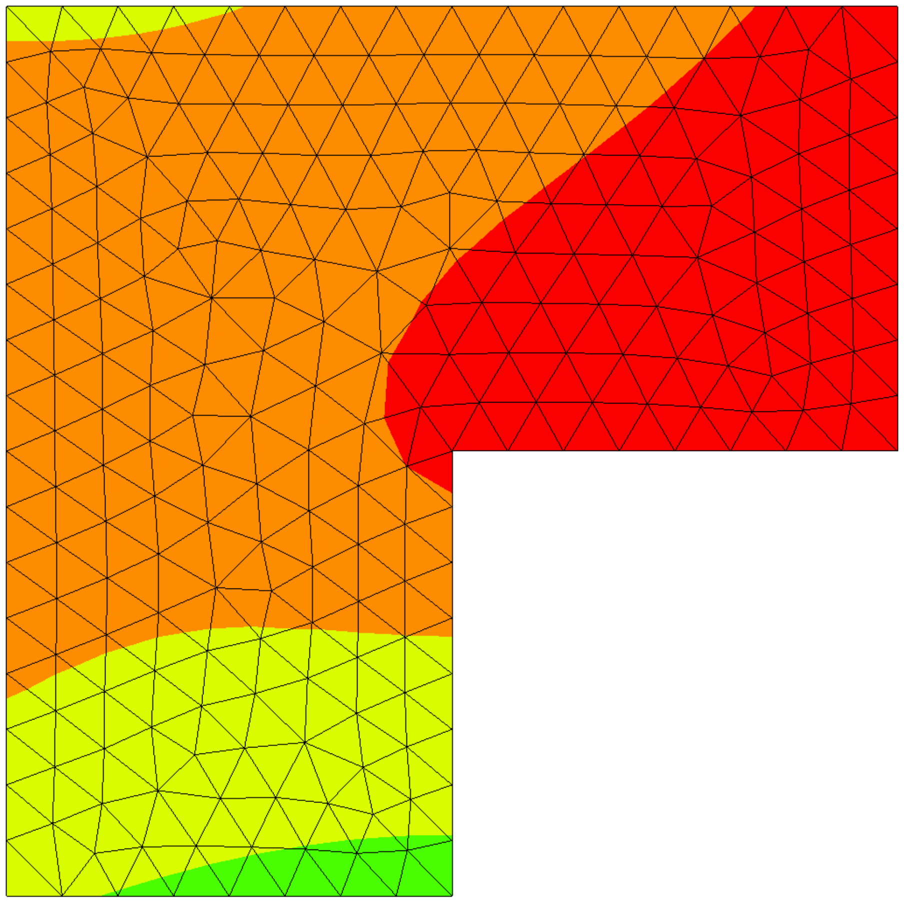


The curl conformity problem

exact solution of an electric field



FEM solution using Lagrangian Elements (node-based)

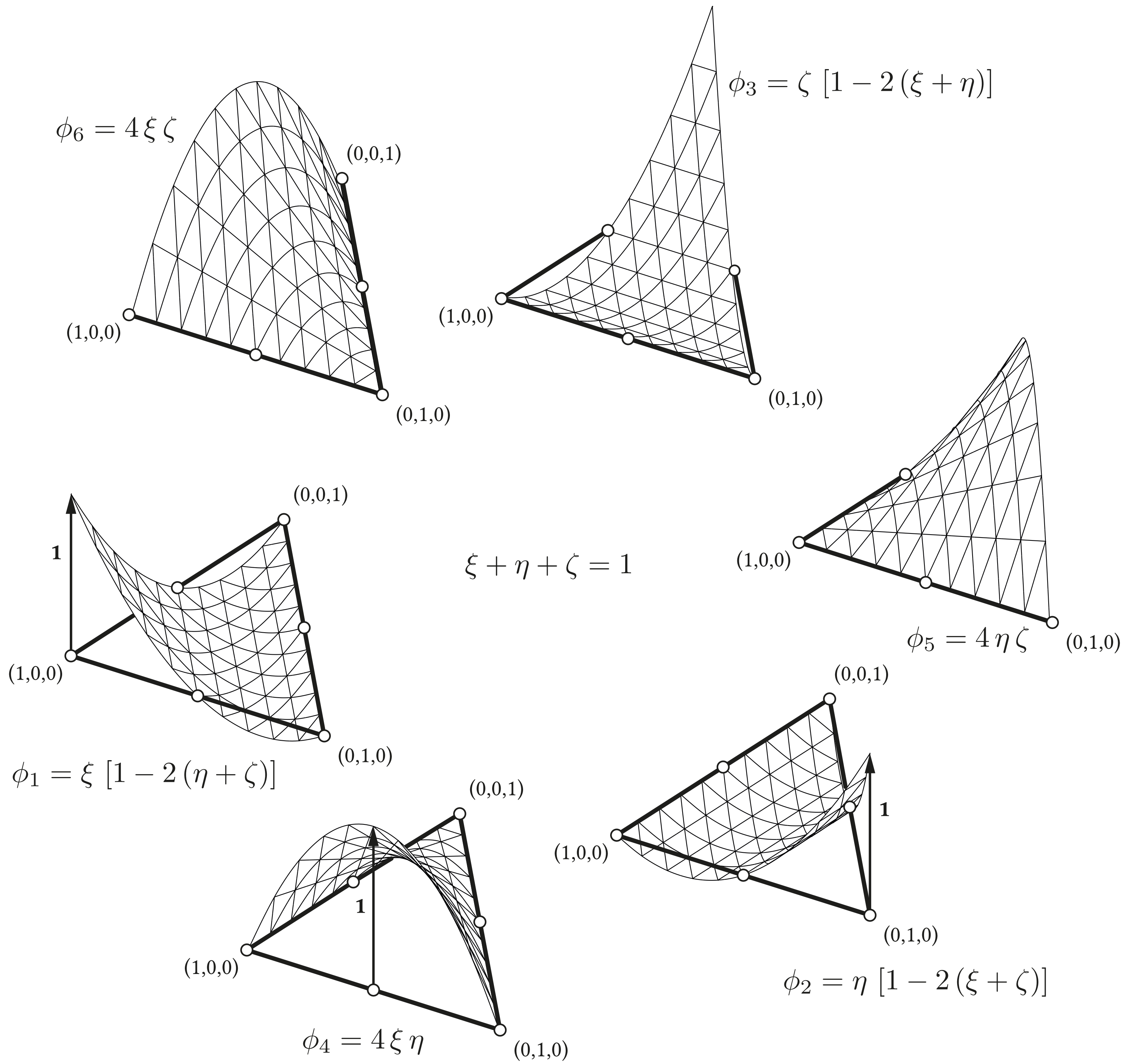


source: http://web.pdx.edu/~gjay/

→ for the Maxwell-equations, classical finite element formulations can fail due to not being curl-conform across element edges

Way Out

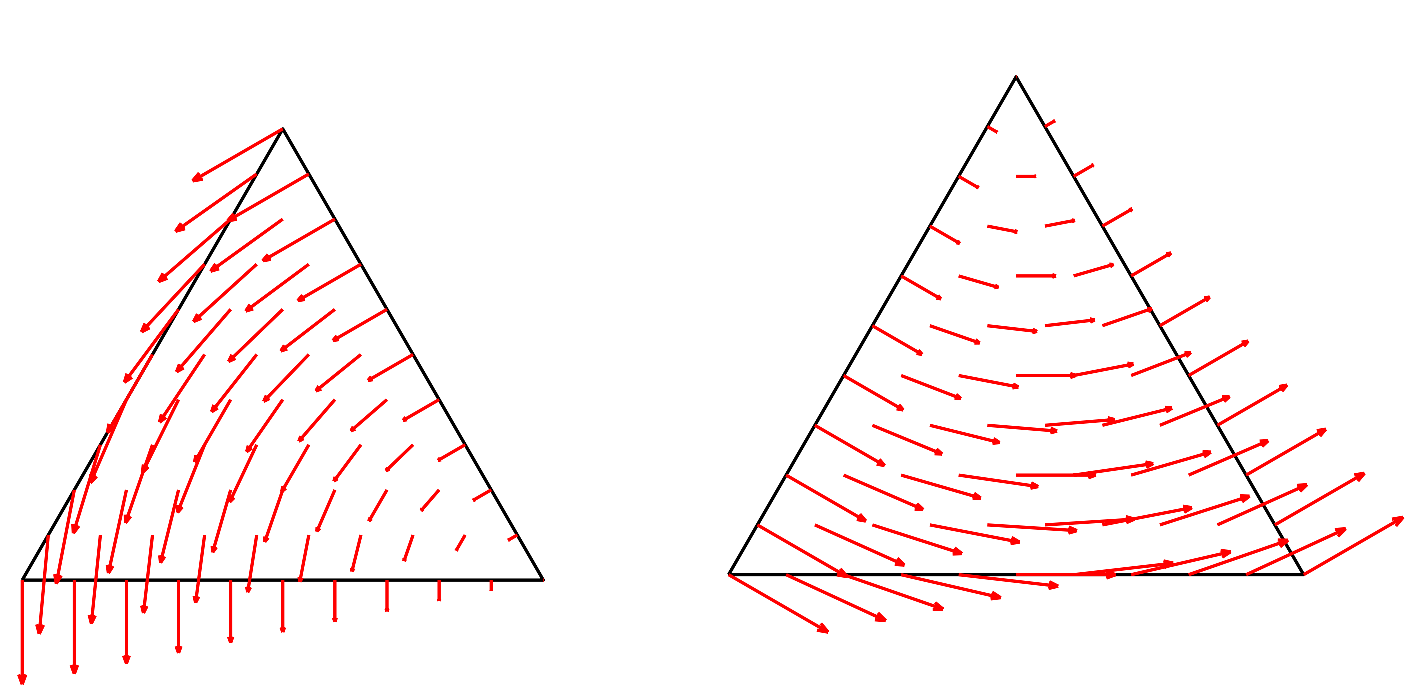
Taking a look at how element degrees of freedom work



- in classical Lagrange elements, the degrees of freedom sit on the nodes
- the sum of the flux over all edges of an element must be zero, which is not guaranteed in this formulation!

Way Out

Introduction of edge-based degrees of freedom (aka Nédélec elements)



- in Nédélec elements, the degrees of freedom sit on the edges (and faces for higher order)

→ mesh-module of code must generate edges and keep track of edge directions (complete)

→ DOF-managing module must be able to handle both node- and edge-based dofs (in progress)

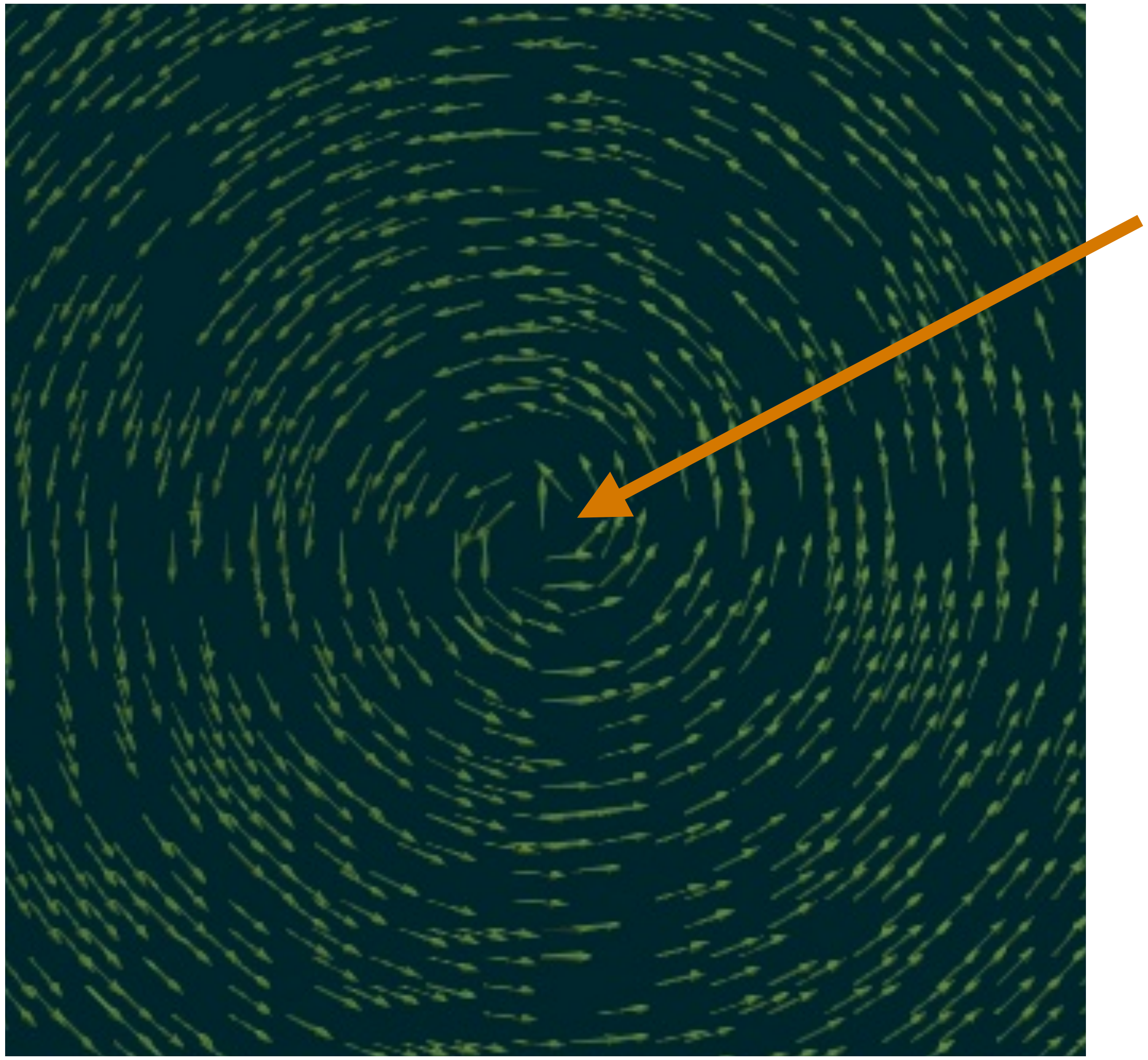
→ edge-dofs can't be visualized with ParaView: need L2 projection on nodes

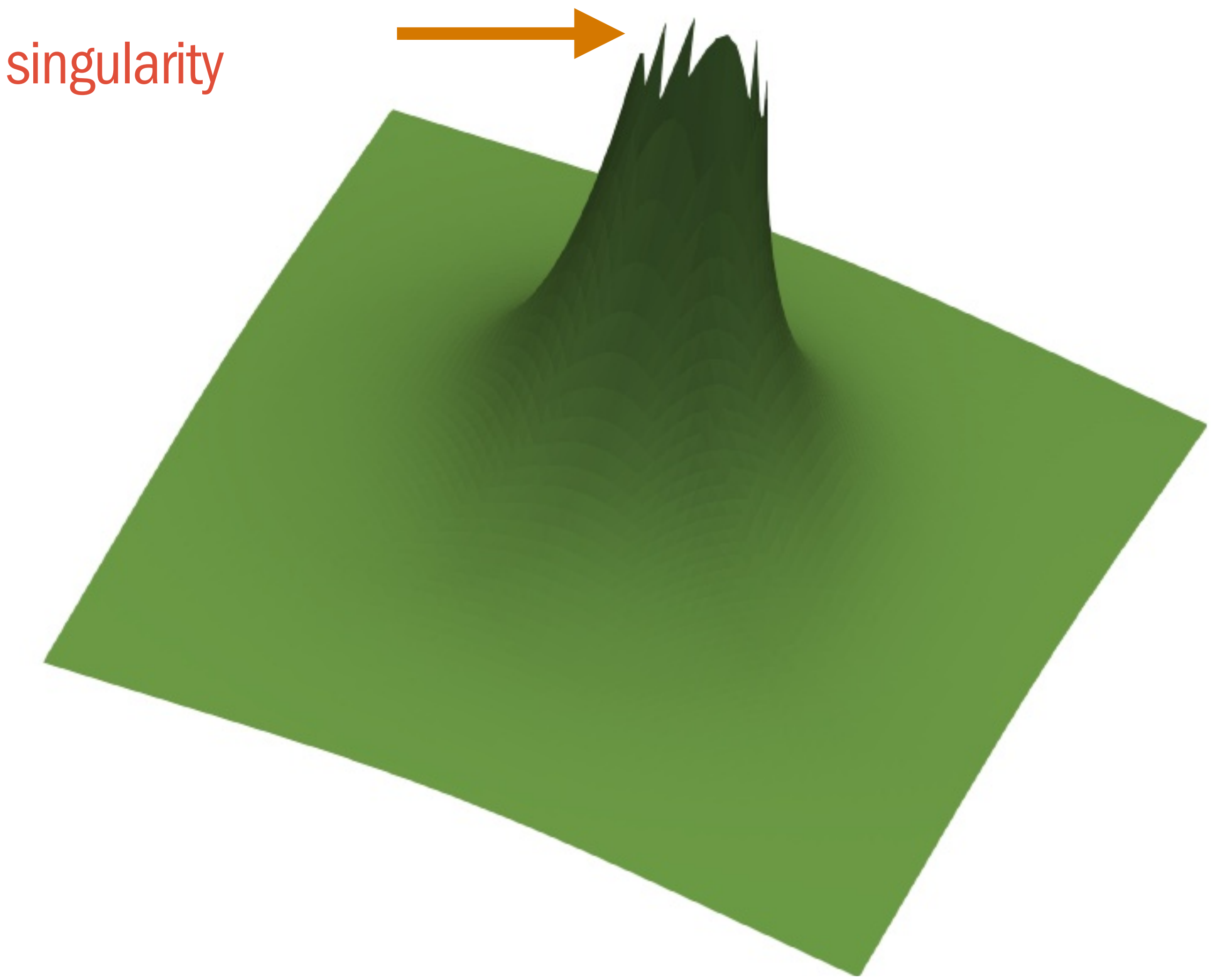
Edge Elements

Proof of Concept: simple analytic field with known solution for curl

$$\vec{a} = \frac{1}{\sqrt{x^2 + y^2}} \begin{bmatrix} x \\ y \end{bmatrix}$$

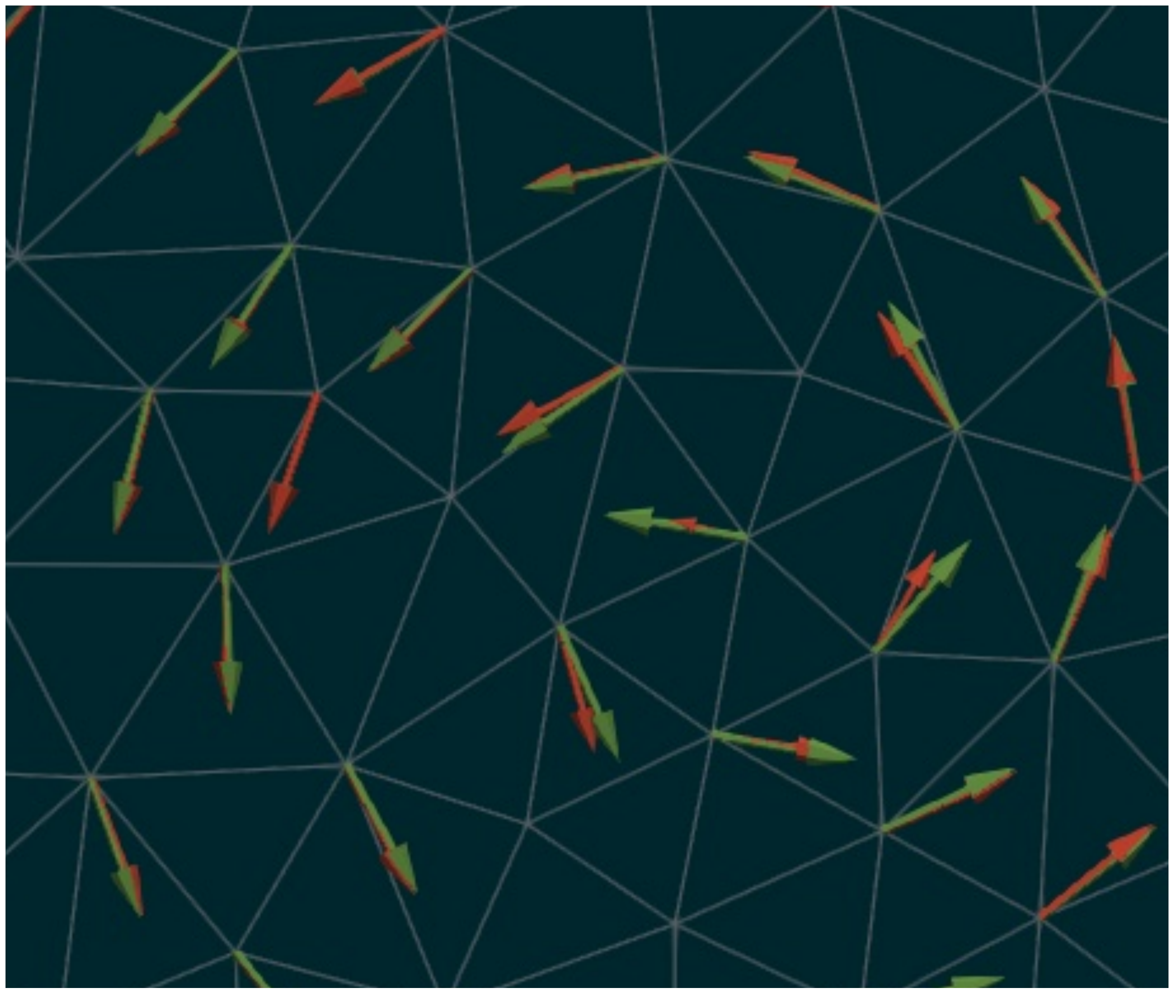
$$\nabla \times \vec{a} = \frac{\partial a_y}{\partial x} - \frac{\partial a_x}{\partial y}$$

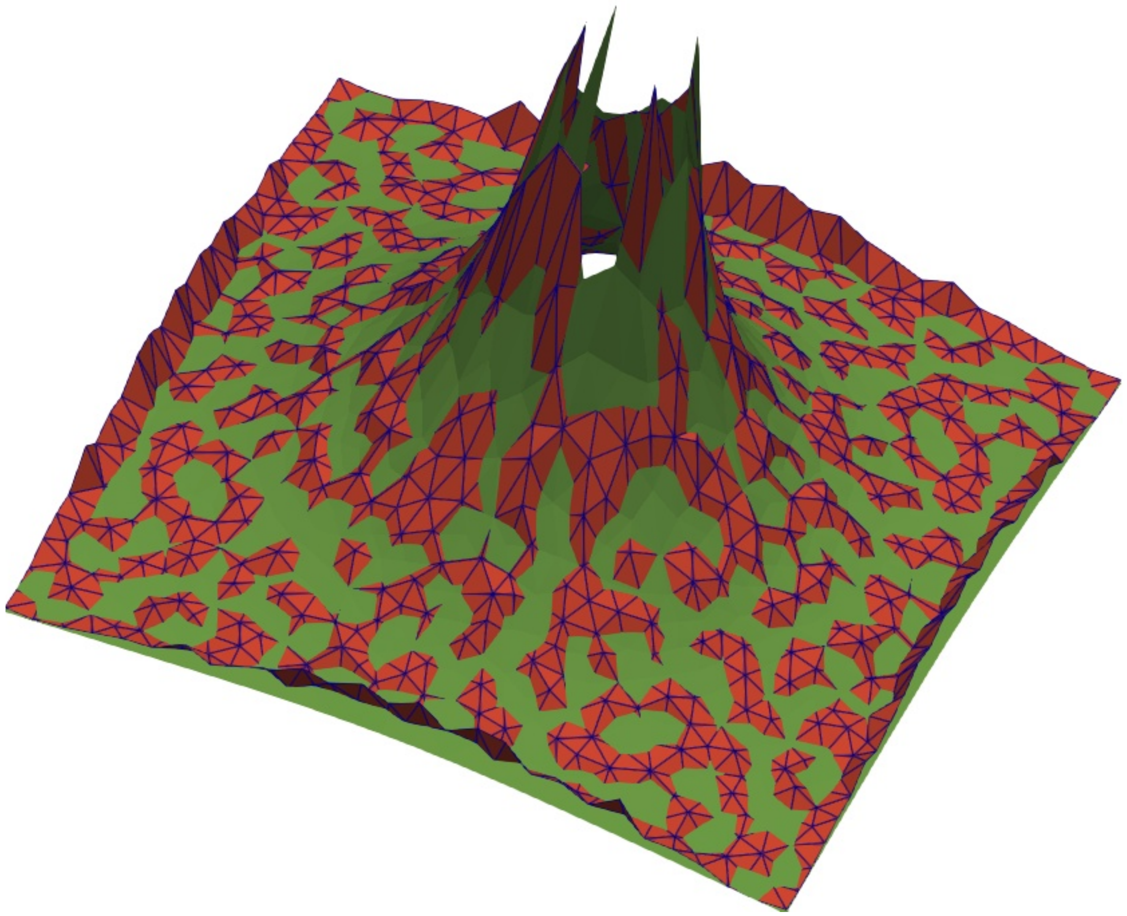




Edge Elements

Proof of Concept: L2-projection of Edge Field on Nodes for Visualization





- green: exact solution
- red: approximation using L2-projection

Intermediate Summary

- need to use a-formulation in air and h-formulation in superconductor
  - code must be able to handle block-specific equations and dofs
- node-based element approach can fail in some circumstances
  - need to use edge-elements in superconducting area
  - need L2-Projection for visualization
  - need to consider impact on conditioning of system matrix

Next Steps ...

proof of concept and code validation

Near Term Roadmap

superconducting block problem

- finish new DOF management system
- simple transient problem with changing current
- investigate behaviour of quenching model
  - investigate suitable formulations (h, a-v, Nédélec ...)
  - impact of non-linear material laws → code stability, XFEM?
  - identify suitable Krylov solvers and preconditioners
  - code hardening
- extension to 3D and axisymmetric problems

benchmarks

- compare code against ANSYS User Elements SCPack
- validate code against selected benchmark problems from HTS Modeling Workgroup

Long Term Roadmap

comparison with experiments

- investigate magnetization and current distribution in tapes and cables

  →compare with ATAP and MDP experiments
- support development of better geometries for CORC cables

possible cooperations

- LBL Computational Research Division

  develop better preconditioning techniques
- knowledge and data exchange within the MDP program
- University of Colorado at Boulder:

  XFEM and adaptive meshing techniques