TAILSIT

Recent Developments of the EM-Module in LS-DYNA

Lars Kielhorn, Thomas Rüberg, & Jürgen Zechner

Contents

Things we do. . .

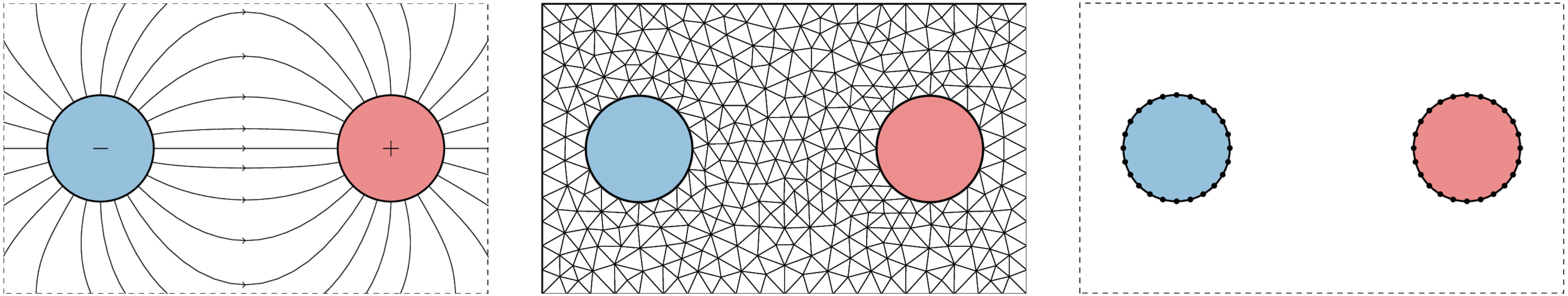
Things we do for ANSYS/DYNA

Conclusion & Outlook

BEM Characteristics

TAILSIT

$$\Delta u = 0 \text{ in } \Omega \quad \text{and} \quad u = g \text{ on } \Gamma \quad \Rightarrow \quad \int_\Gamma U(x,y)\lambda(y)\,\mathrm{d}s_y = g(x) \quad \Rightarrow \quad \lambda = DtN(g)$$



$\Rightarrow$ Dirichlet-to-Neumann map: $\boldsymbol{\lambda} = DtN(\boldsymbol{A}_\Gamma)$

- \+ Only surface discretisation needed
- \+ Representation of infinite domain
- \+ Accurate surface data

- \- Only useful for linear problems
- \- Singular integrals
- \- Fully populated system matrix

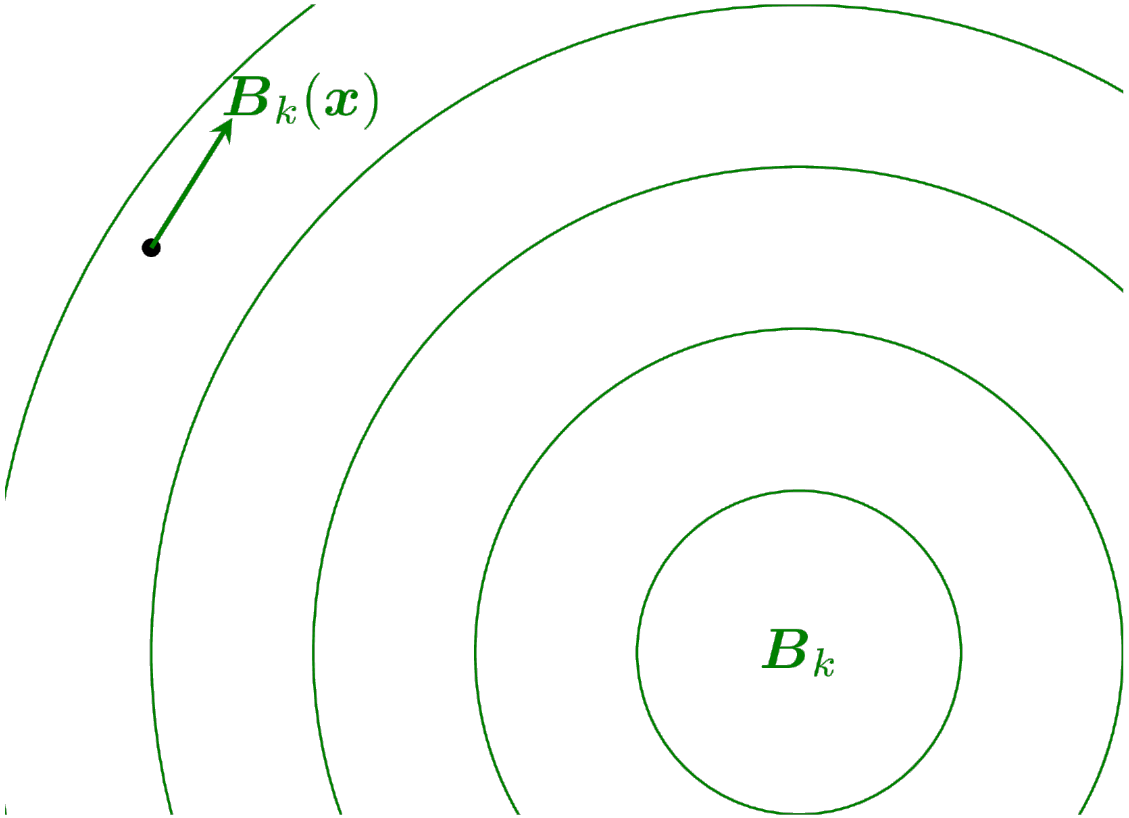
Electromagnetic Navigation

TAILSIT

Where is point $\boldsymbol{x}$ if it cannot be seen?

Given:

- n_k known source fields $\boldsymbol{B}_k$ with frequency ω_k
- measurements at point $\boldsymbol{x}$
- iterative algorithm $\{\boldsymbol{B}_k(\boldsymbol{x}^{(\ell)})\} \to \boldsymbol{x}$



TAILSIT

Where is point $\boldsymbol{x}$ if it cannot be seen?

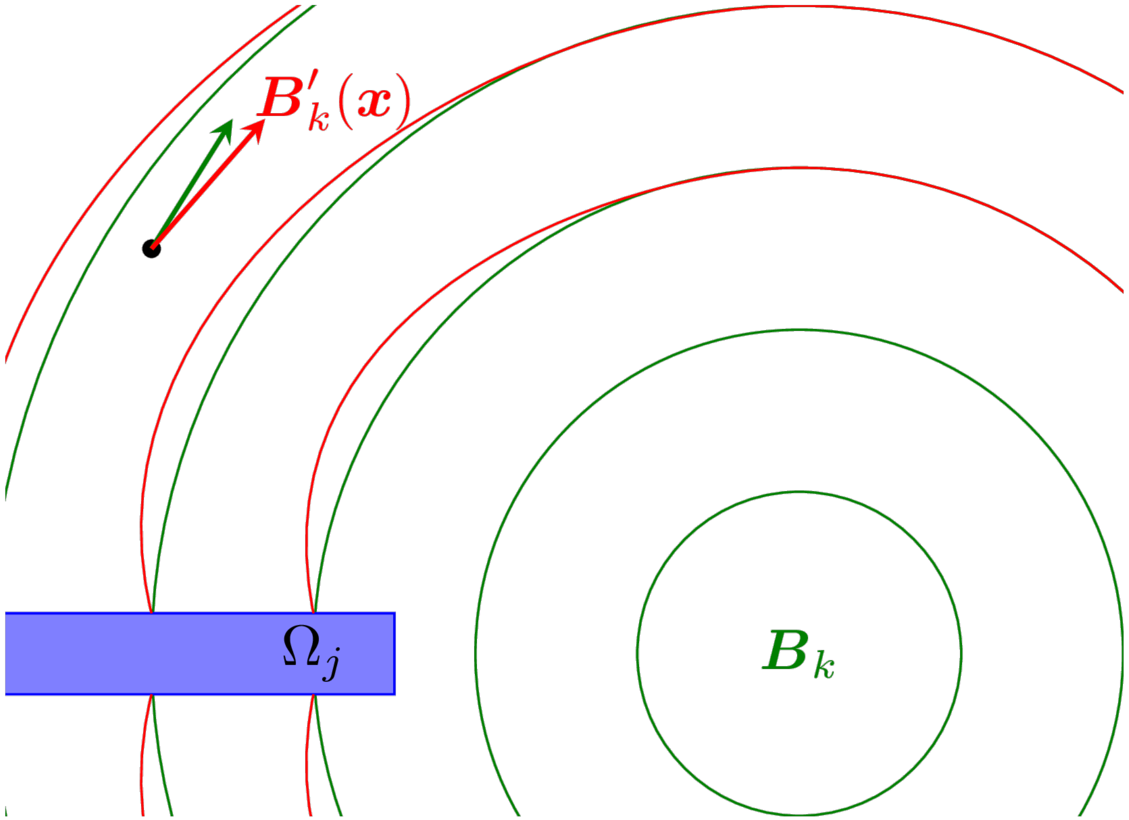
Given:

- n_k known source fields $\boldsymbol{B}_k$ with frequency ω_k
- measurements at point $\boldsymbol{x}$
- iterative algorithm $\{\boldsymbol{B}'_k(\boldsymbol{x}^{(\ell)})\} \to \boldsymbol{x}$

Problem:

- conductors Ω_j enter the scene
- disturbed field is measured

$$\boldsymbol{B}'_k(\boldsymbol{x}) = \boldsymbol{B}_k(\boldsymbol{x}) + \Delta \boldsymbol{B}_k(\boldsymbol{x})$$



Changes of constellation and real-time demand

Single-domain Case

Discretised integral equations

$$\begin{pmatrix} N^+ + N_j^k & B^+ + B_j^k \\ C^+ + C_j^k & A^+ + A_j^k \end{pmatrix} \begin{pmatrix} e_j^k \\ b_j^k \end{pmatrix} = \begin{pmatrix} f_j(\boldsymbol{A}_k^0, \boldsymbol{B}_k^0) \\ g_j(\boldsymbol{A}_k^0, \boldsymbol{B}_k^0) \end{pmatrix} \quad \oplus \quad \text{evaluation } \Delta\boldsymbol{B}_k(\boldsymbol{x}^{(\ell)}) = \dots$$

Notation

- N, A: hypersingular and single layer operators
- B, C: adjoint double layer and double layer operators

Shorthand

$$\boxed{M_{jj}^k z_j^k = h_j^k}$$

Approach

- for every possible conductor Ω_j
- precompute the inverse of M_{jj}^k for every background field k (and save to disc)
- (load system from disc and) solve for $\{e_j^k, b_j^k\}$
- evaluate $\Delta\boldsymbol{B}_k$ at current iterate $\boldsymbol{x}^{(\ell)}$

TAILSIT

Many conduction domains

$$\begin{pmatrix} M_{11}^k & M_{12}^0 & \cdots \\ M_{21}^0 & M_{22}^k & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix} \begin{pmatrix} z_1^k \\ z_2^k \\ \vdots \end{pmatrix} = \begin{pmatrix} h_1^k \\ h_2^k \\ \vdots \end{pmatrix}$$

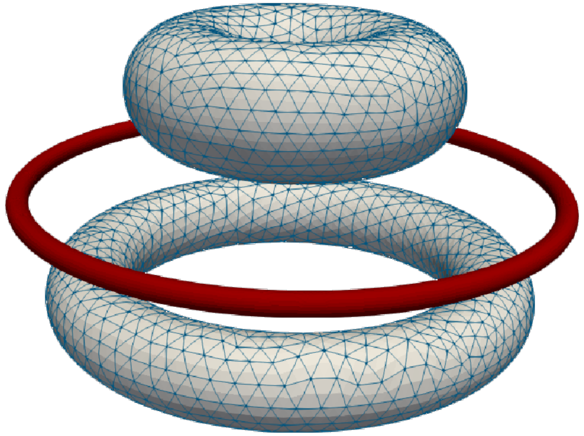
Challenges

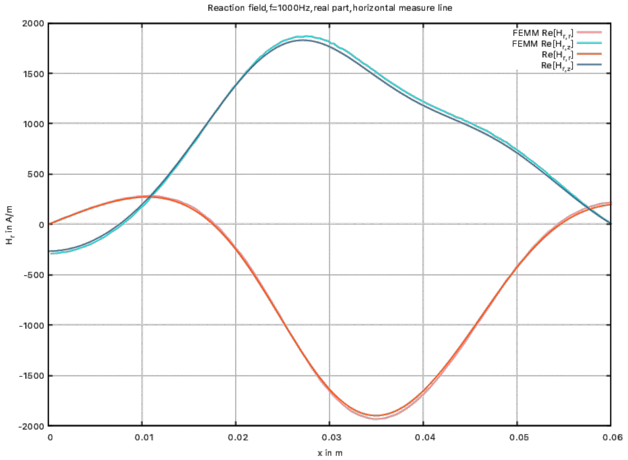
- Interaction terms M_{ij}^0 $(i \neq j)$ cannot be pre-computed
- We cannot have the system's inverse matrix on disc

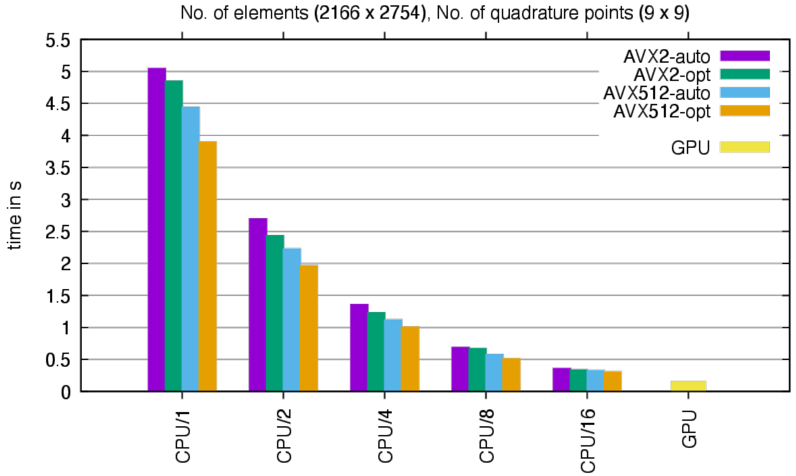
Procedure

- Precompute inverses of M_{ii}^k independent of constellation. Store them in database (HDF5)
- In each frame, compute interactions M_{ij}^0 as fast as possible
- Solve system with preconditioned GMRES

Results







- CPU: `Intel(R) Xeon(R) Gold 6154 CPU @ 3.00GHz (Q4 2017)`
- GPU: `AMD/ATI Baffin Radeon RX 550 640SP (2GB, Apr 2017)`
- Verification example: two conducting tori, using FEMM as reference

$$f = 1\,000\mathrm{Hz}\,, \quad \sigma = 2 \cdot 10^6 \mathrm{S\,m}^{-1}\,, \quad \mu_r = 100$$

- Threading + SIMD and GPU performance

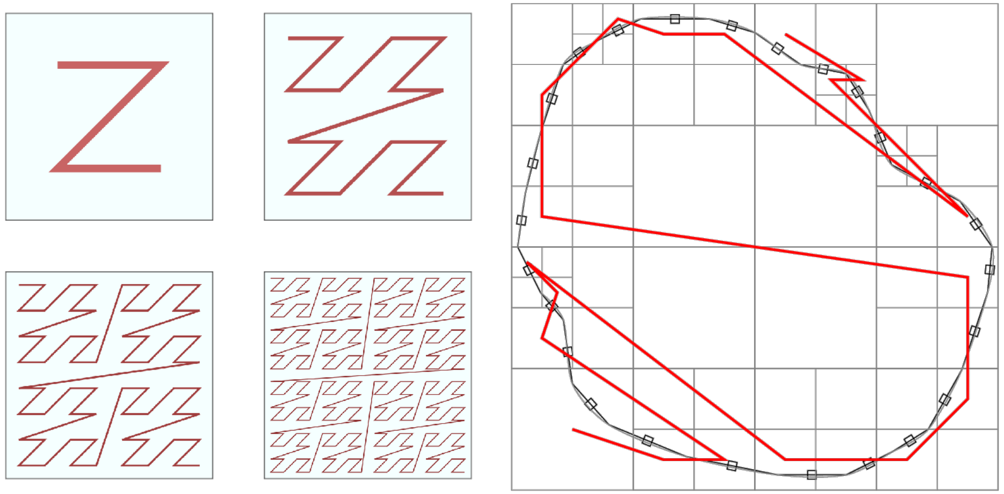
MPP-FMM: Distributed Memory Parallelization

TAILSIT

- Message Parsing Interface (MPI)
- Parallelization challanging due to unbalanced tree structure

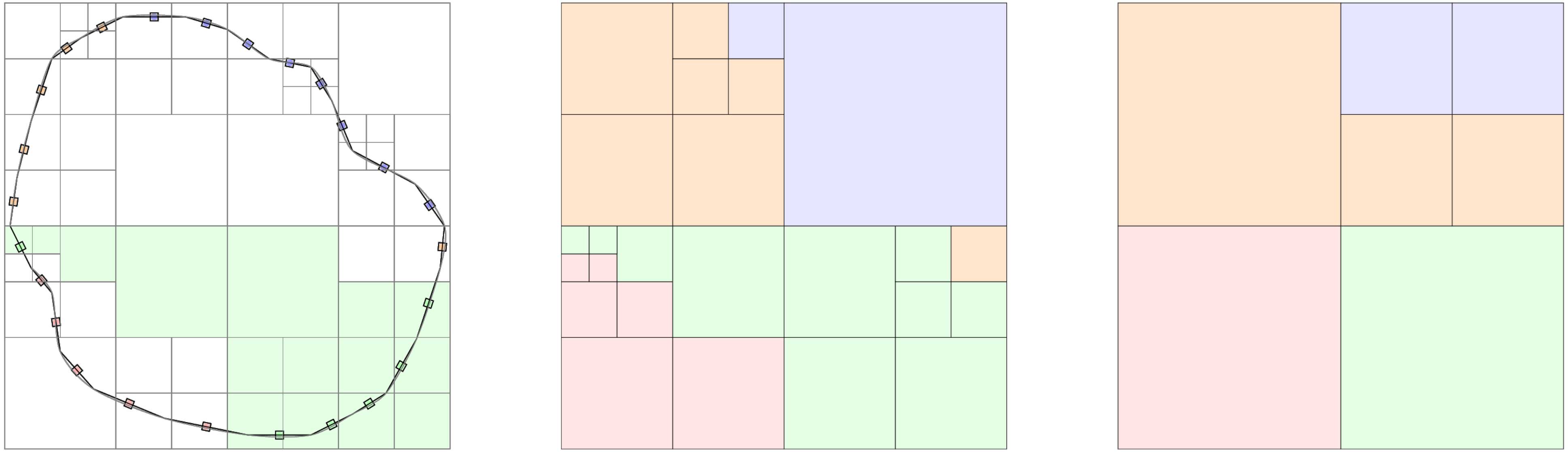
One-time preparation steps

- Linear tree: Sort boxes w.r.t. space-filling curve (Z-curve, Morton ID)
- Task-splitting/load balancing based on Morton ID
- “Local Essential Tree”



MPP-FMM: Partition & Load Balancing

TAILSIT



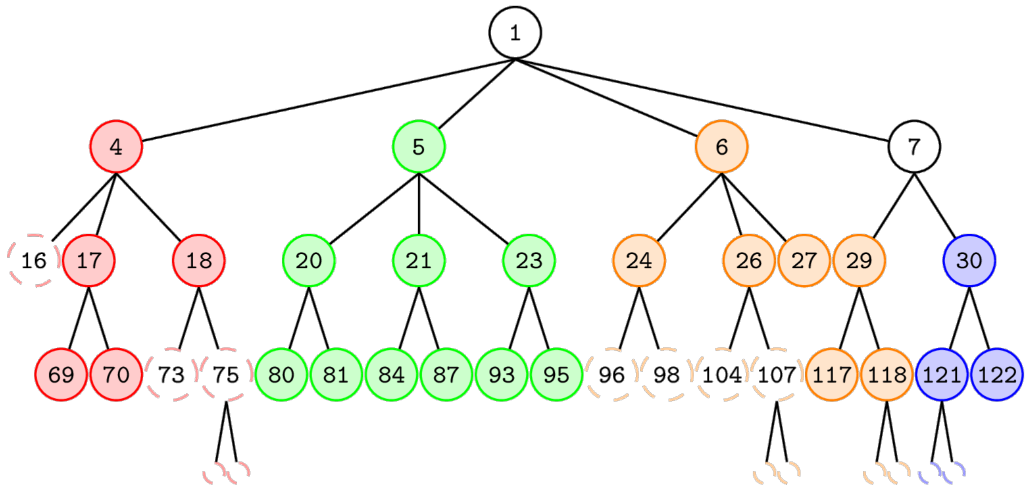
- Each process reads a part of the mesh: elements $\rightarrow$ Morton ID
- Globally sort elements w.r.t. Morton ID and redistribution
- Coarse blocks $\rightarrow$ global partition & load balancing
- Distribute elements according to the partition

MPP-FMM: Parallel FMM-Algorithm

TAILSIT

One-time preparation steps

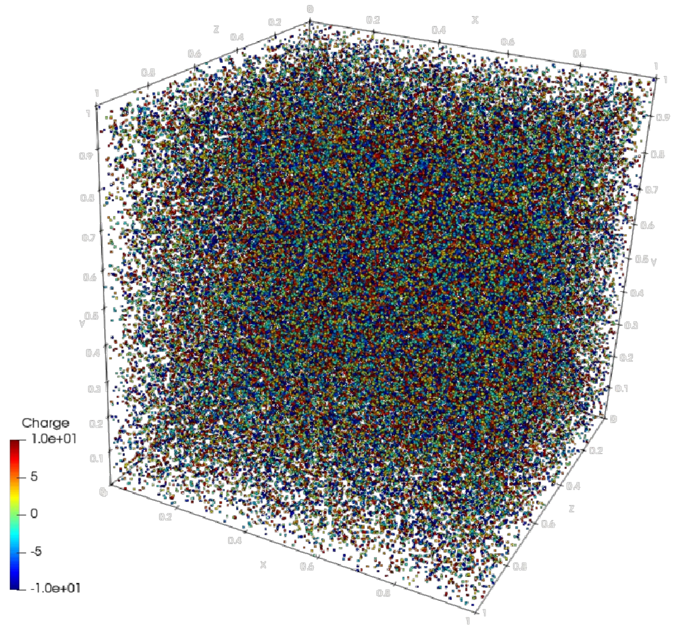
- Construction of the LET
- Set up and store transfer operators
- Create near-field matrix

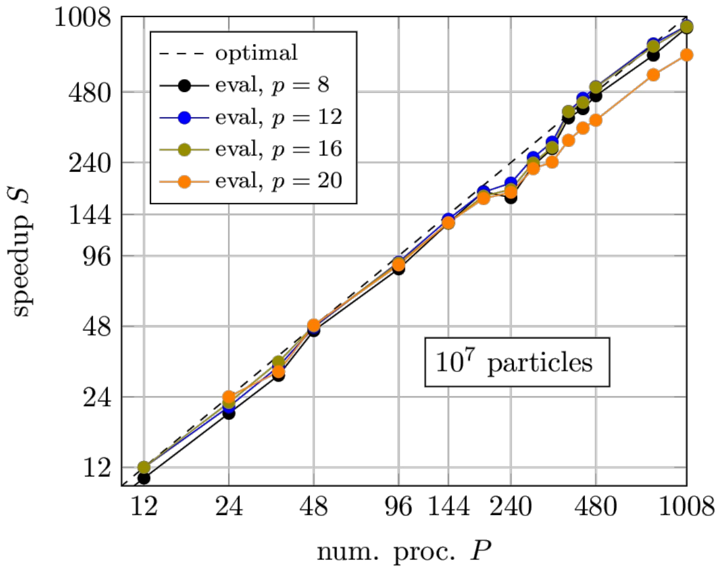


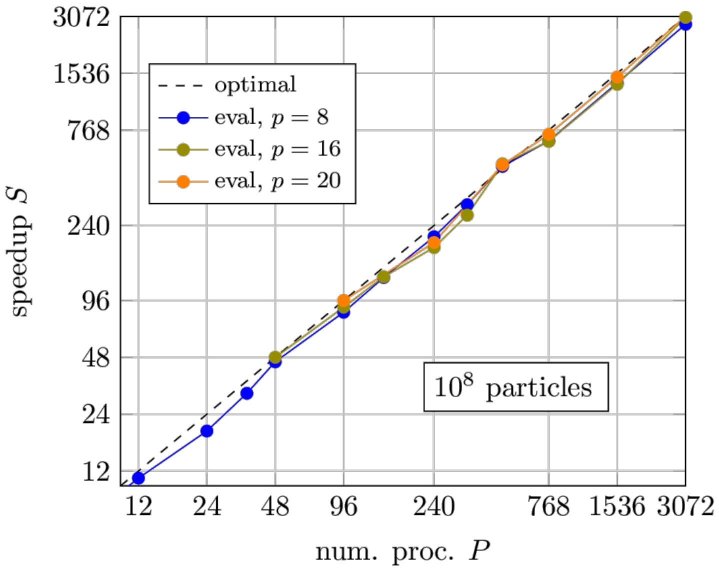
Matrix-vector product

- Parallel sparse *Matrix-Vector product* for near-field interaction
- Upward pass in the LET on each process: P2M, M2M
- Communicate moments to halo boxes on other processes
- M2L on the LET of each process
- Downward pass and evaluation: L2L, L2P

N-body Simulation





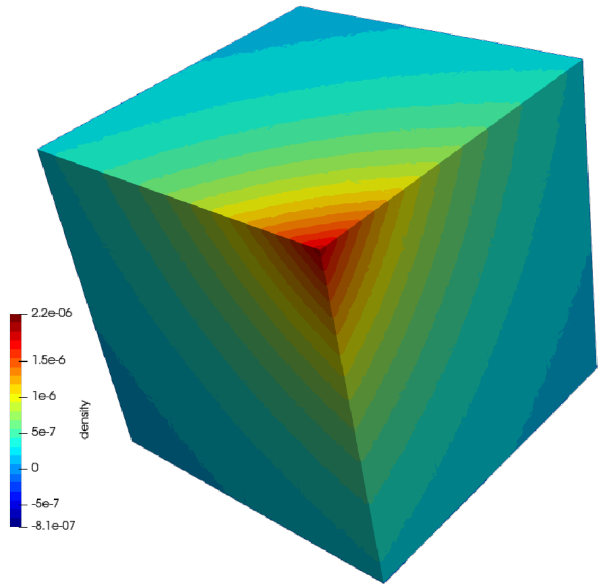


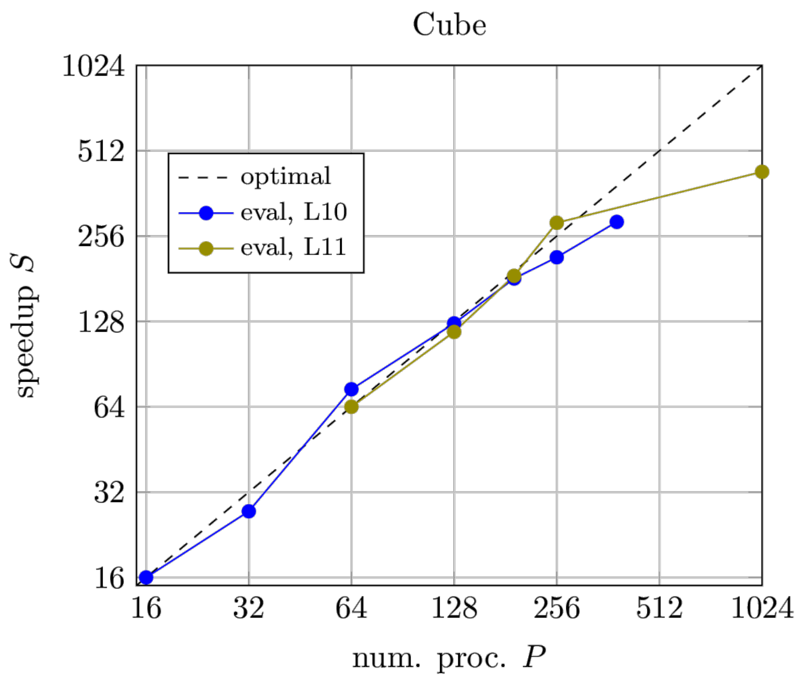
- Cooperation with Vienna Scientific Cluster (VSC)
- Particle simulation for performance analysis
- Optimal scaling for ∼3K processes

MPP-FMM: Benchmark Electrostatics

TAILSIT

- Check accuracy of FMM approximation
- Iterative Solution ($\approx$ 20 iterations, $\varepsilon_r = 10^{-5}$)
- Simulation with $n \approx$ 25 mio. (L10) and $n \approx$ 100 mio. (L11) *surface* elements, $p = 7$





[s]	P	t_{setup}	t_{solve}
L10	128	521.18	97.9
	256	271.9	59.1
L11	256	1065.8	215.5
	1024	271.0	142.3

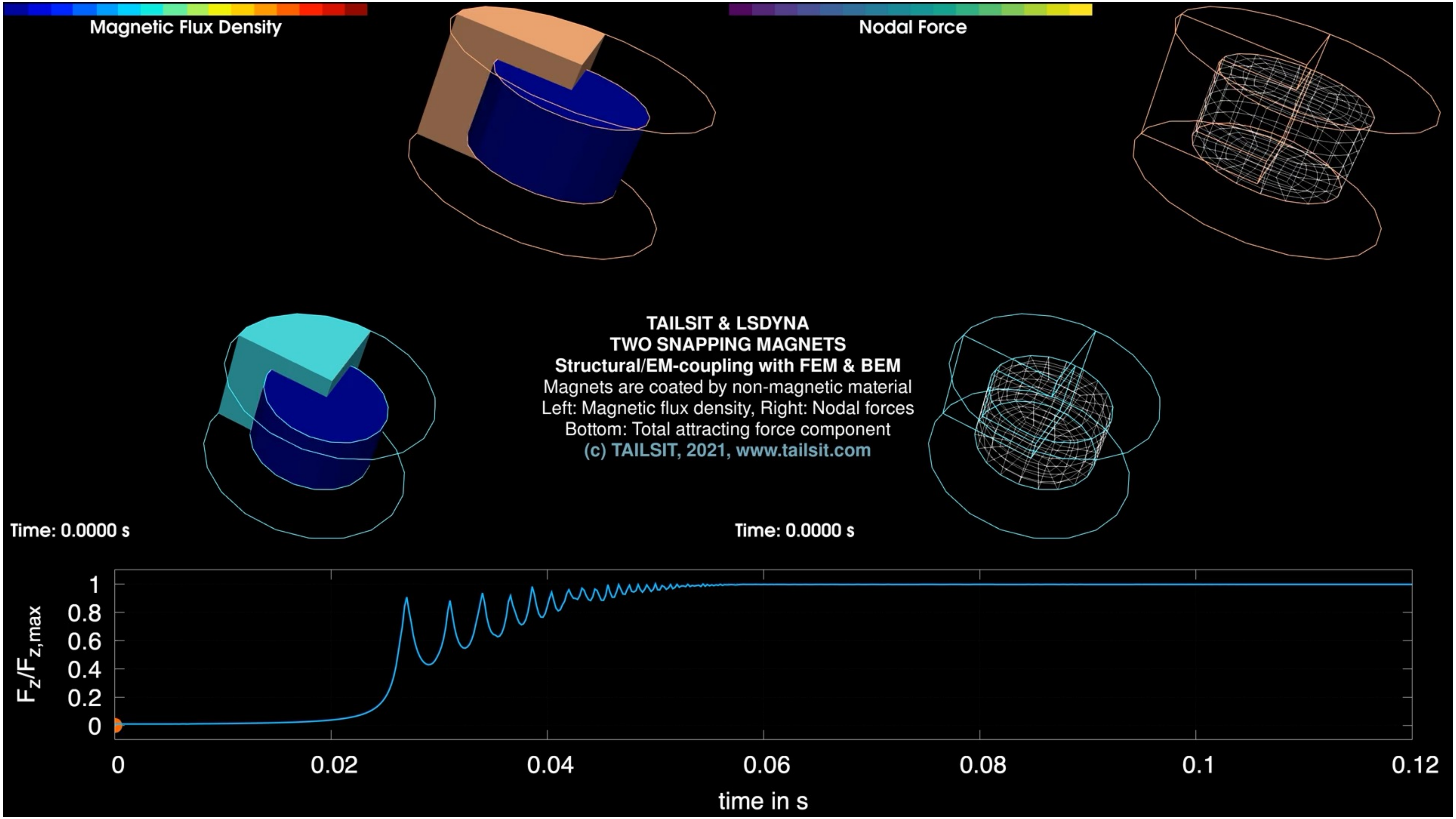
Things we do…

Things we do for ANSYS/DYNA

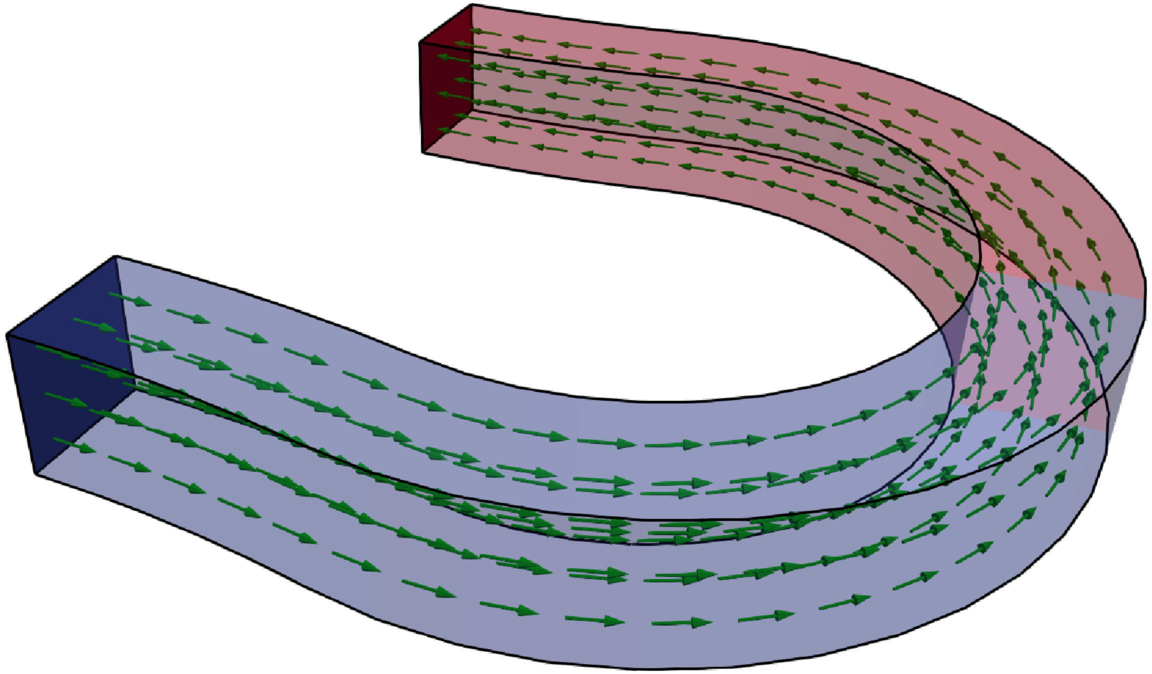
Conclusion & Outlook

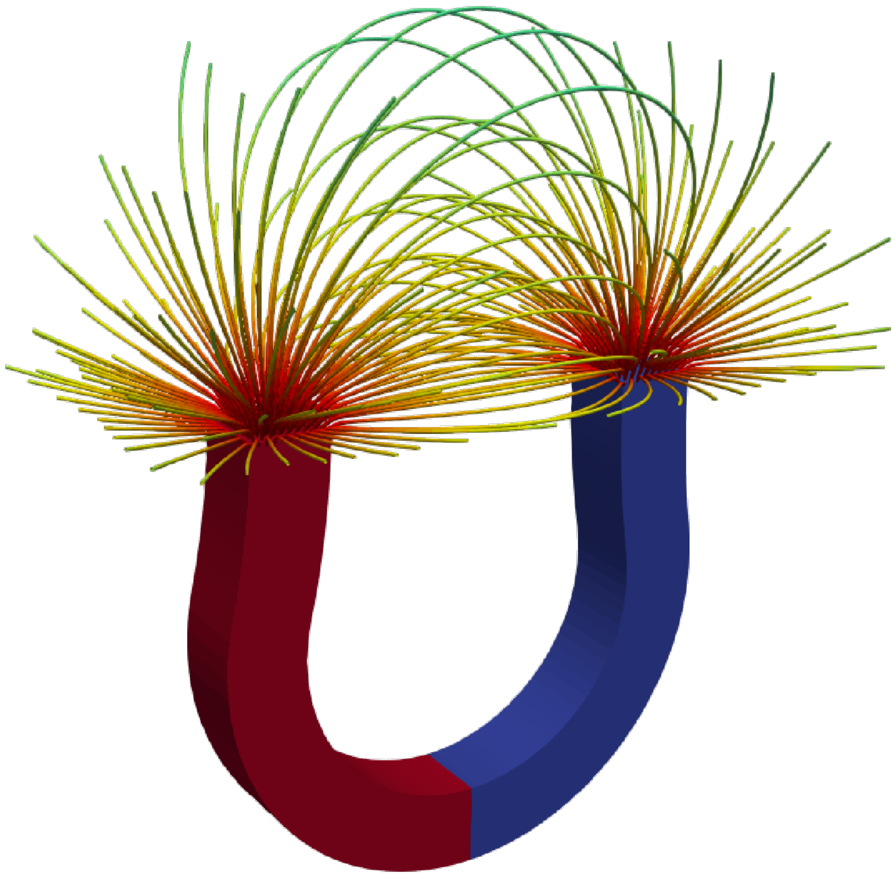
Snapping magnets

TAILSIT



Permanent magnets





- **+** Simple implementation $(\boldsymbol{M}, \mathbf{curl}\,\boldsymbol{A}')_\Omega$
- **-** Typically modeled as zero conductivity region (Raises solvability questions)

Auxilary Maxwell Space (AMS) Preconditioning

TAILSIT

- Simple magnetostatics (FEM only)

$$(\mu^{-1}\,\mathbf{curl}\,\boldsymbol{A}, \mathbf{curl}\,\boldsymbol{A}')_\Omega = (\boldsymbol{M}, \mathbf{curl}\,\boldsymbol{A}')_\Omega \quad \Longrightarrow \quad P^{-1}Ax = P^{-1}b$$

- Vector potential $\boldsymbol{A}$ is not unique. Any $\widetilde{\boldsymbol{A}} := \boldsymbol{A} + \mathbf{grad}\,\varphi$ is a solution. Thus, A is singular and we can't choose $P^{-1} \leftarrow A^{-1}$.
- AMS Preconditioning[1] provides an alternative

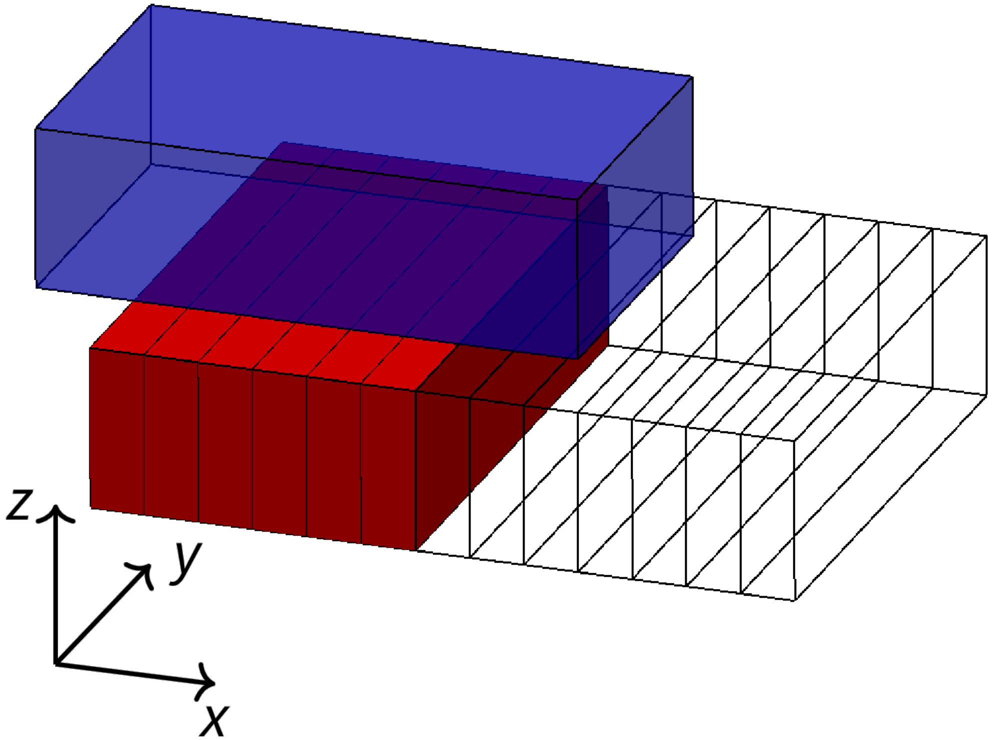
$$P^{-1} \leftarrow \Lambda^{-1} + \Pi L^{-1} \Pi^\top$$

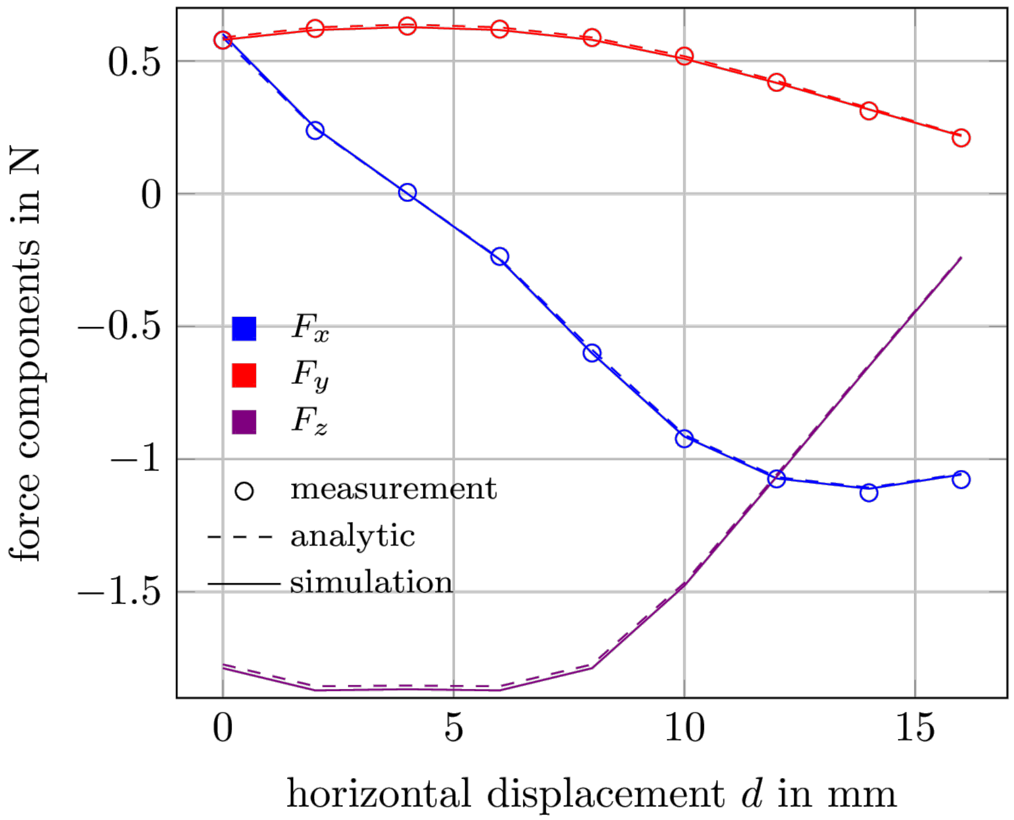
  - Λ: Smoother (we use simply a Jacobi smoother)
  - Π: Interpolation matrix $\Pi : [H^1(\Omega)]^3 \to H(\mathbf{curl}, \Omega)$
  - L: Stiffness matrix corresponding to the vector Laplacian (better: use AMG to approximate L^{-1})

[1] HIPTMAIR, R.; XU, J.. Nodal auxiliary space preconditioning in H (curl) and H (div) spaces. SIAM Journal on Numerical Analysis, 2007,45(6),pp.2483-2509

Permanent Magnets & EM-Forces

TAILSIT

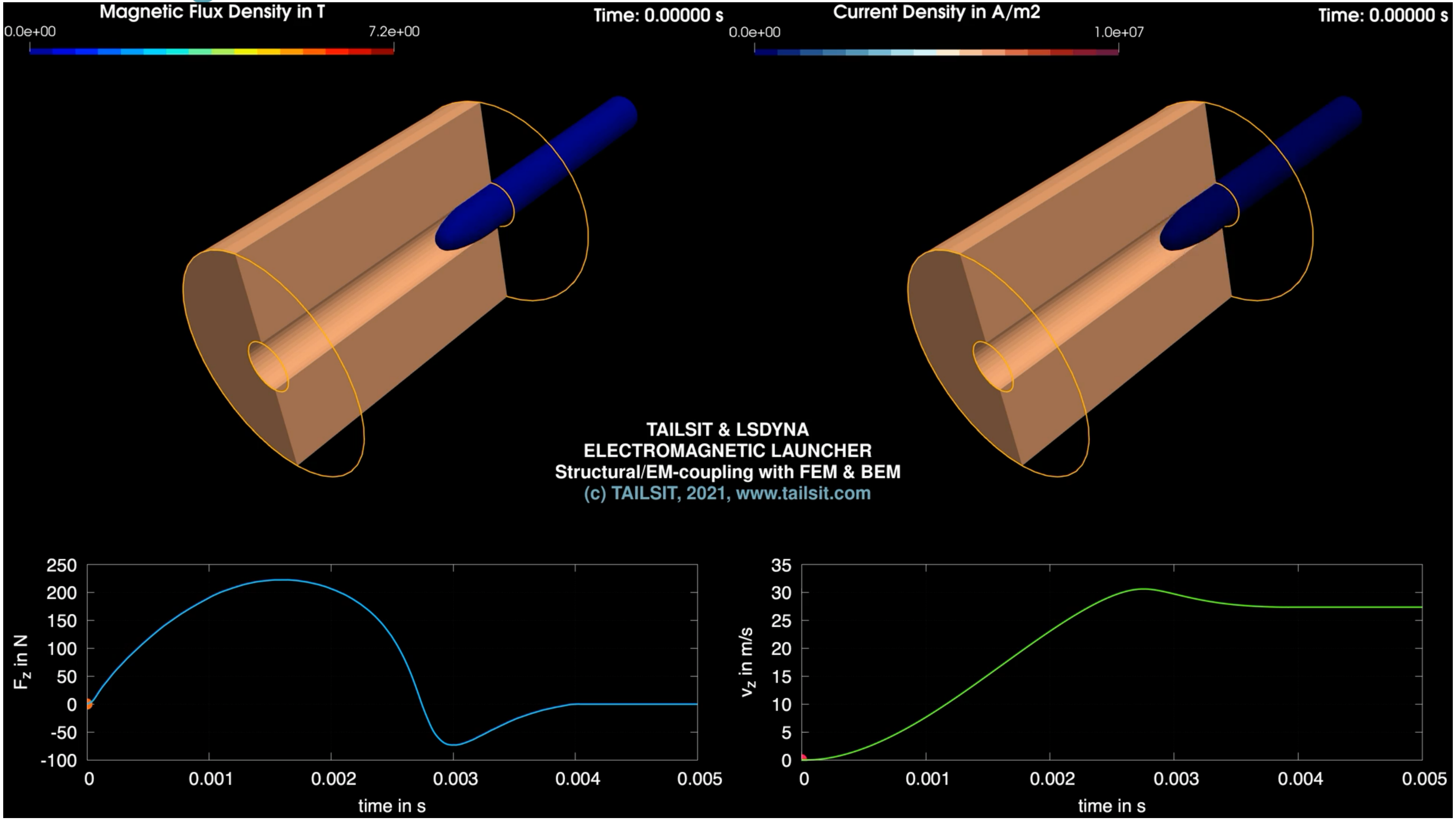




- AKOUN G.; Yonnet J.-P.. 3d analytical calculation of the forces exerted between two cuboidal magnets. IEEE Transactions on magnetics, 1984, 20(5), pp.1962–1964

Ferromagnetic material & EM-Forces

TAILSIT

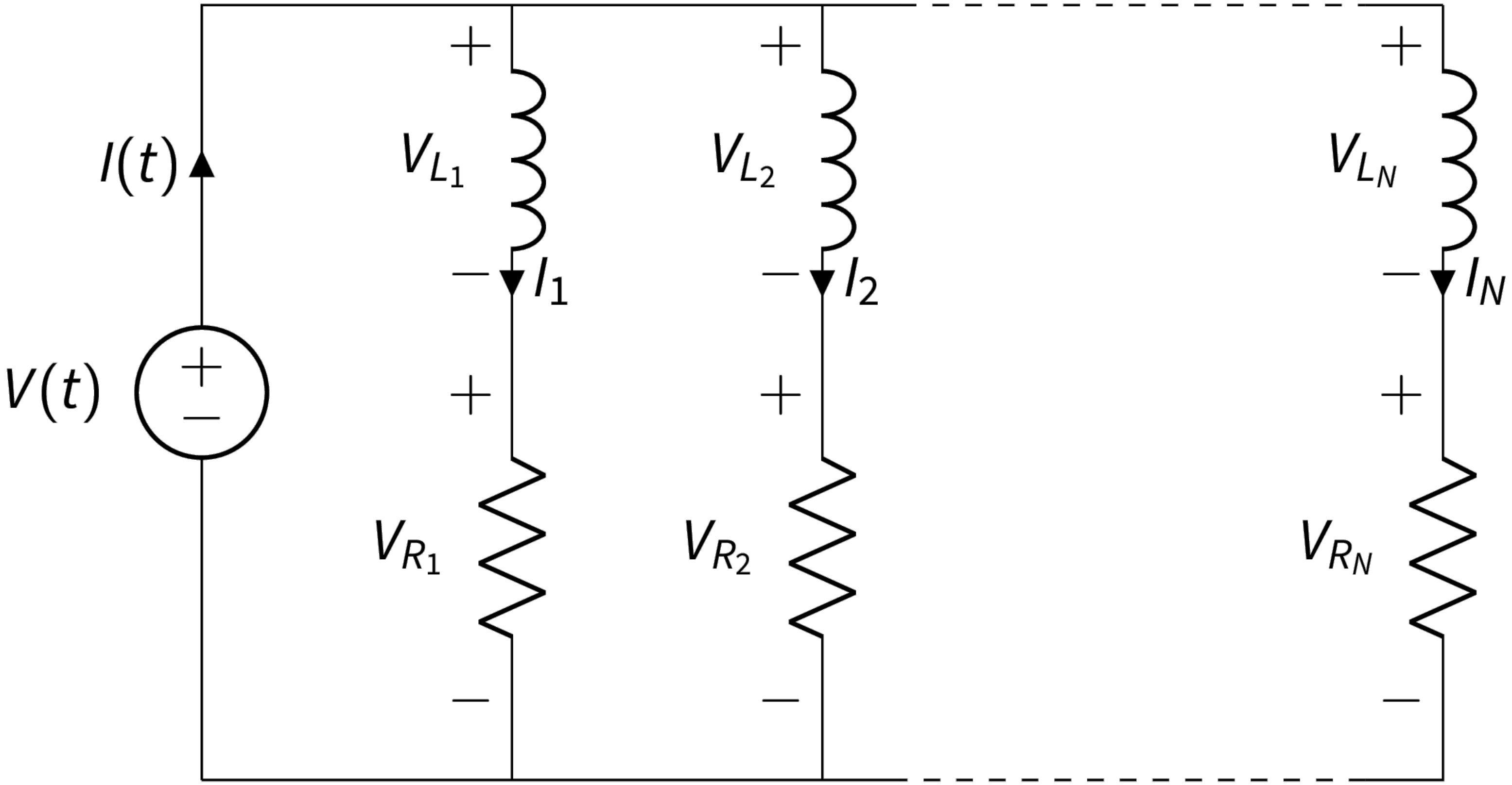


Voltage driven coils

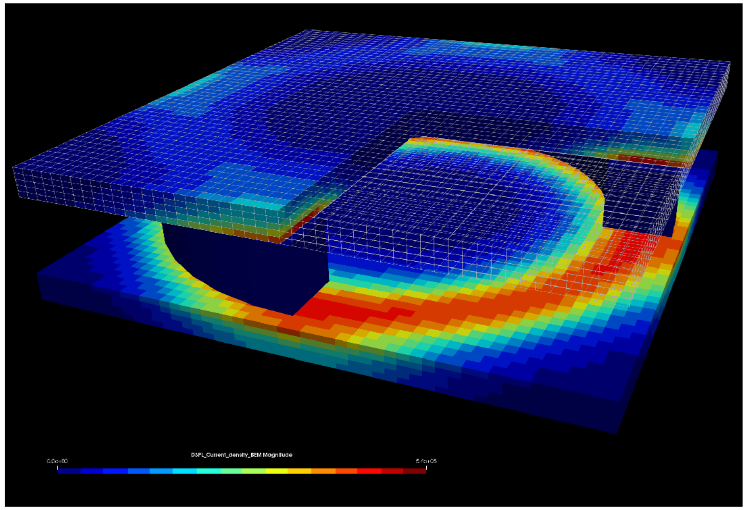
- Enhanced FEM-BEM system / Solve via Schur-complement

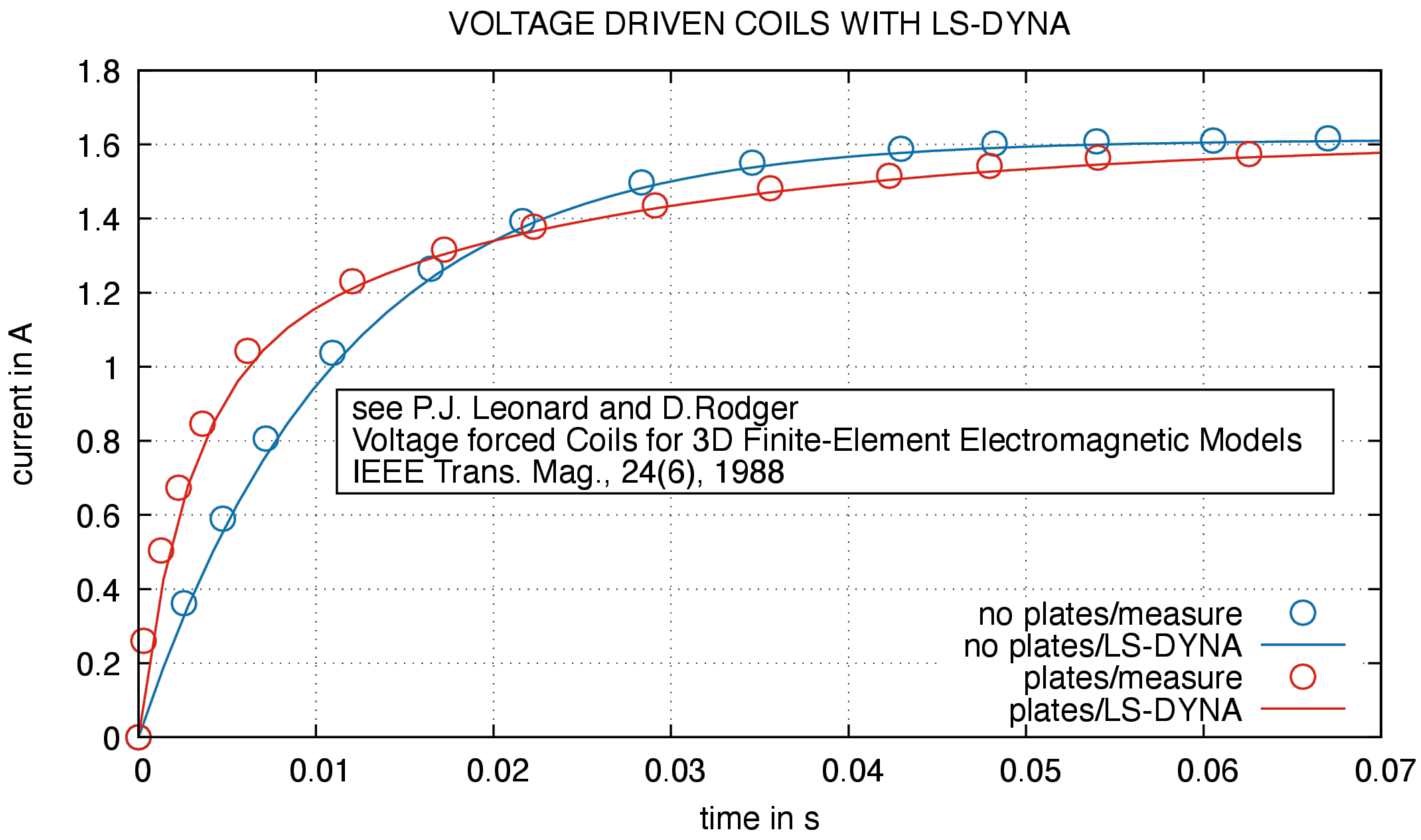
$$Sx = f(i) \implies \begin{bmatrix} S & -T^\top \\ -T & -R \end{bmatrix} \cdot \begin{bmatrix} x \\ i \end{bmatrix} = \begin{bmatrix} 0 \\ g(V) \end{bmatrix} \implies (S+T^\top R^{-1} T)x = -T^\top R^{-1} g$$

- Network model



Voltage driven coils





Things we do…

Things we do for ANSYS/DYNA

Conclusion & Outlook

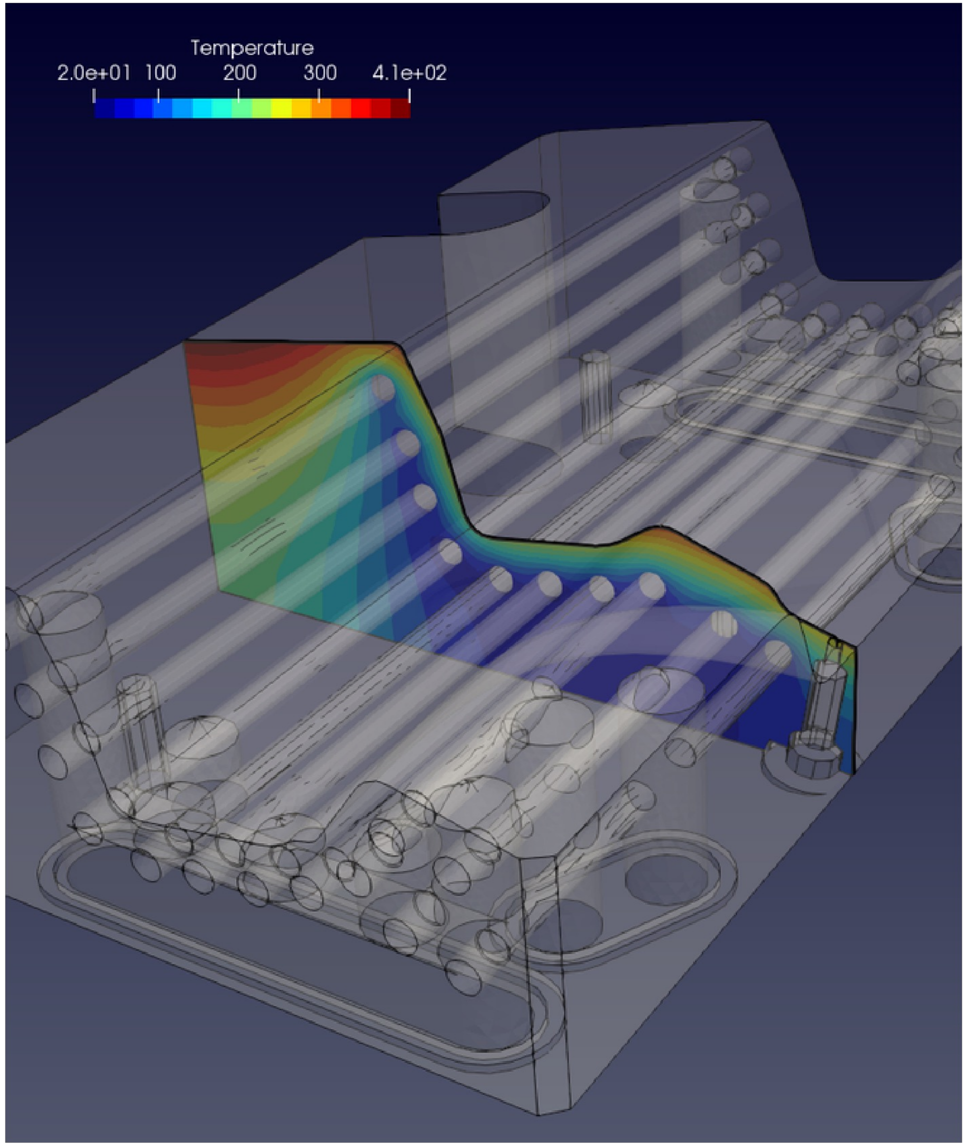
Conclusion & Outlook

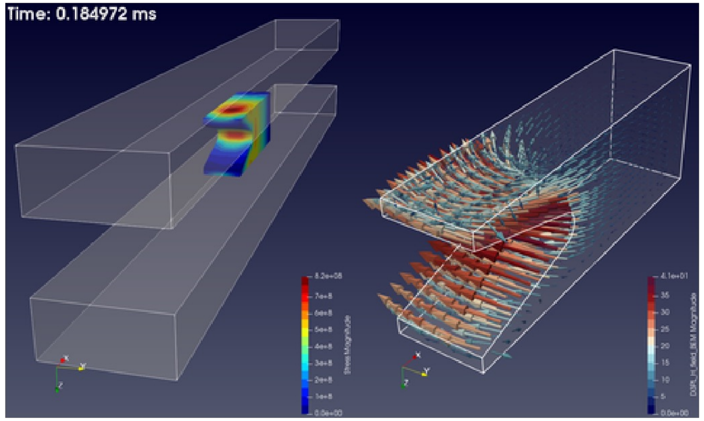
Current state

- Monolithic, symmetric, robust FEM-BEM coupling
- Accurate results
- Ferromagnetic materials
- Computation of Nodal forces
- Permanent magnets
- AMS Preconditioner
- Voltage/Current driven coils

Things to do...

- Testing, bugfixing and release
- Improve the preconditioner(s)
    - Possibly propagate the AMS preconditioner to pure FEM modules of ANSYS and/or LS-DYNA
    - Direct solvers are used to tackle the FEM stiffness matrices in P. What about AMG?
    - Use better preconditioner for the BEM part.
- Frequency domain simulations.
- Higher order schemes?
- BEM with optimal complexity?





D3PLOT

Pictures created with ParaView using D3PLOT2HDF5 by TAILSIT (www.tailsit.com)