Finite Element Maxwell Solvers : Theory and Applications

Joachim Schöberl

Center for Computational Engineering Sciences (CCES)
RWTH Aachen University

S. Zaglmayr

FWF Start Project Y-192
"3D hp-Finite Elements: Fast Solvers and Adaptivity"

Markus Wabro

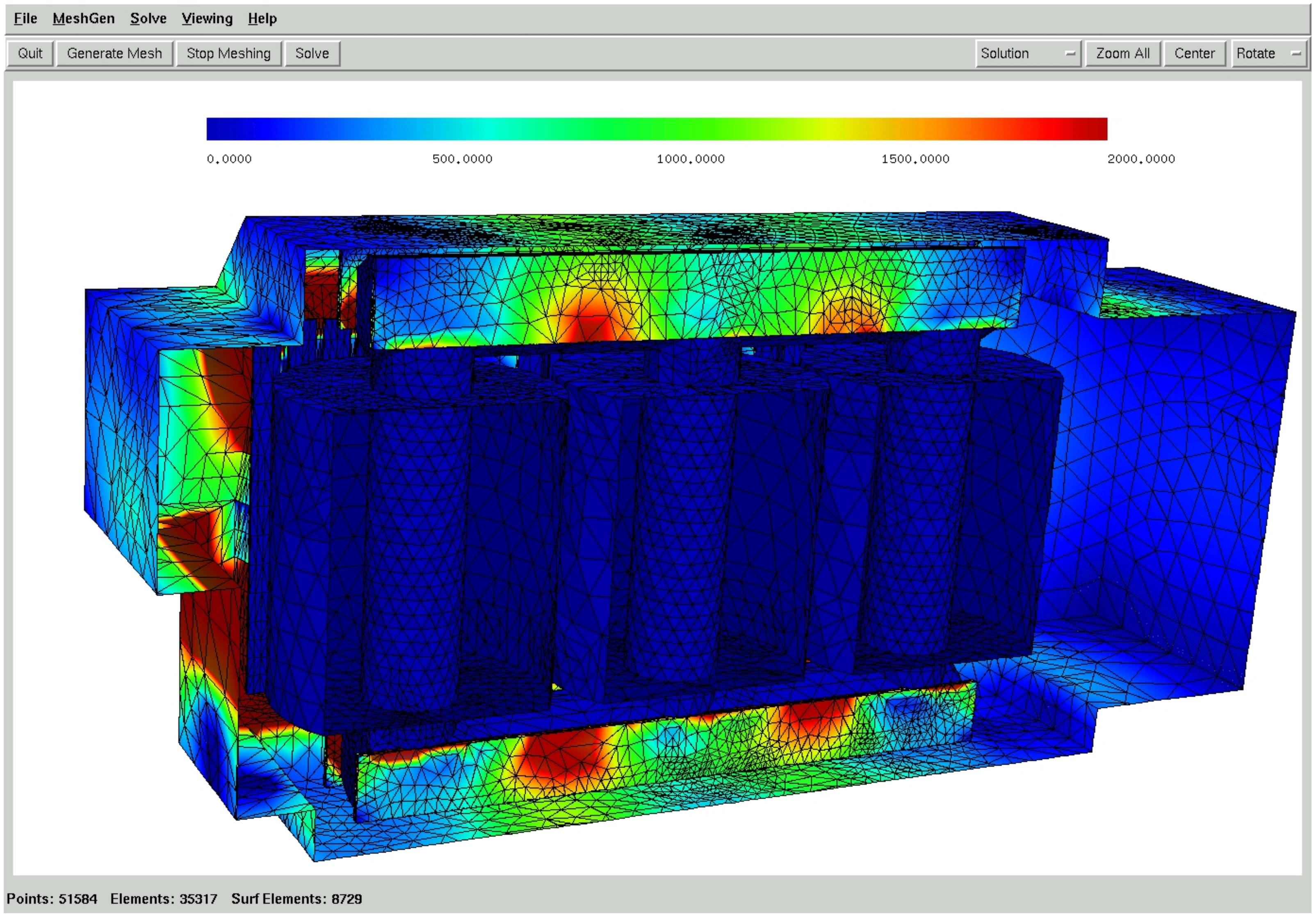
FEMworks - Finite Element Software and Consulting GmbH

ASIM Workshop, Aachen, Feb 28, 2007

Contents:

- Maxwell equations
- Finite Elements for Maxwell equations
- Iterative Equation Solvers
- Applications: Transformer, Bore-hole EM
- Netgen/NgSolve Software

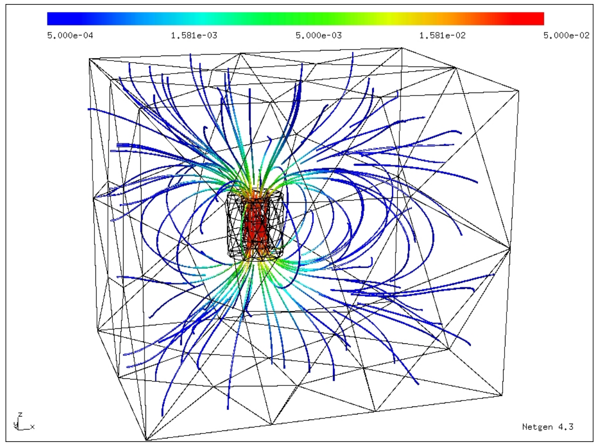
Loss density in a Power Transformer



Transformers built by Siemens / EBG Transformatorenbau, Linz

Simulation with Netgen/NgSolve

Equations of Magnetostatics



Given:

j .. current density s.t. $\text{div } j = 0$

Compute:

B .. magnetic flux density
H .. magnetic field intensity

such that

$$B = \mu H \qquad \text{div } B = 0 \qquad \text{curl } H = j$$

with the boundary conditions

$$\text{either} \quad B \cdot n = 0 \quad \text{or} \quad H \times n = 0$$

Vector potential formulation

Since $\text{div } B = 0$ (plus compatibility conditions), there exists a vector potential A such that

$$B = \text{curl } A$$

Combining the equations above gives us

$$\text{curl } \mu^{-1} \, \text{curl} \, A = j$$

with boundary conditions

$$\text{either} \quad A \times n = 0 \quad \text{or} \quad (\mu^{-1} \, \text{curl } A) \times n = 0$$

Vector potential formulation

Since $\mathrm{div}\ B = 0$ (plus compatibility conditions), there exists a vector potential A such that

$$B = \mathrm{curl}\ A$$

Combining the equations above gives us

$$\mathrm{curl}\ \mu^{-1}\,\mathrm{curl}\,A = j$$

with boundary conditions

$$\text{either} \quad A \times n = 0 \quad \text{or} \quad (\mu^{-1}\,\mathrm{curl}\ A) \times n = 0$$

The weak formulation is to find $A \in V := H_D(\mathrm{curl})$ such that

$$\int \mu^{-1}\,\mathrm{curl}\ A\ \mathrm{curl}\,v\,dx = \int j v\,dx \qquad \forall\, v \in V.$$

Problem: A is defined up to a $\nabla\varphi$!

Gauging possibilities

1. Do not gauge, work on factor space $H(\text{curl})/\nabla H^1$.

   Fine, if error estimates etc. only depend on $\operatorname{curl} A$

2. Gauging by regularization. Add small L_2-term:

$$\int \mu^{-1} \operatorname{curl}\ A\ \operatorname{curl} v\, dx + \varepsilon \int Av\, dx = \int jv\, dx \qquad \forall\, v \in V.$$

   Fine, if error estimates etc. do not depend on ε.

3. Gauging by explicit constraints, i.e., solve the mixed problem:

$$\begin{array}{ccccc} \int \mu^{-1} \operatorname{curl} A \operatorname{curl} v\, dx & + & \int v \nabla \varphi\, dx & = & \int jv\, dx & \forall\, v \in H(\text{curl}) \\ \int A \nabla \psi\, dx & & & = & 0 & \forall\, \psi \in H^1 \end{array}$$

   Space for Lagrange parameter is H^1.

Full Maxwell equations

Time harmonic setting:

$$\begin{aligned}\text{curl } H &= j_i + \sigma E + i\omega\varepsilon E,\\ \text{curl } E &= -i\omega\mu H.\end{aligned}$$

By introducing the magnetic vector potential $A = \frac{-1}{i\omega}E$, there follows

$$H = \frac{-1}{i\omega\mu}\text{curl } E = \mu^{-1}\text{ curl } A$$

Strong vector potential formulation:

$$\text{curl } \mu^{-1}\text{curl } A + i\omega\sigma A - \omega^2\varepsilon A = j_i$$

with boundary conditions:

$$A \times n = 0, \qquad \text{or} \qquad (\mu^{-1}\text{curl } A) \times n = j_s, \qquad \text{or} \qquad (\mu^{-1}\text{curl } A) \times n = \kappa(A \times n)$$

Variational problems in $H(\mathrm{curl})$

Function space

$$H(\mathrm{curl}) := \{u \in [L_2]^3 : \mathrm{curl}\; u \in [L_2]^3\}$$

Magnetostatic/Eddy-current problem in weak form:

Find vector potential $A \in H(\mathrm{curl})$ such that

$$\int_\Omega \mu^{-1}\, \mathrm{curl}\; A \cdot \mathrm{curl}\; v \, dx + \int_\Omega i\omega\sigma\, A \cdot v \, dx = \int_\Omega j \cdot v \, dx \qquad \forall\, v \in H(\mathrm{curl}),$$

Gauging by regularization in insulators

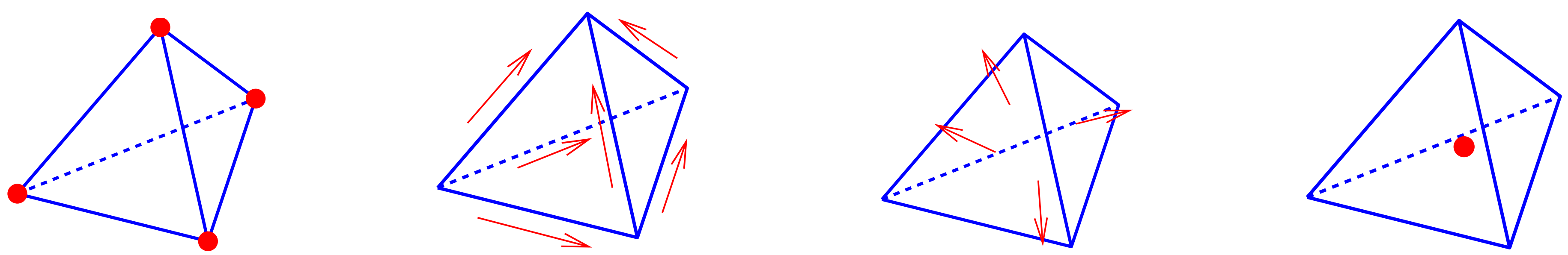
Maxwell eigenvalue problem:

Find eigenfrequencies $\omega \in \mathbb{R}_+$ and $E \in H(\mathrm{curl})$ such that

$$\int_\Omega \mu^{-1}\; \mathrm{curl}\; E \cdot \mathrm{curl}\; v \, dx = \omega^2 \int_\Omega \varepsilon\, E \cdot v \, dx \qquad \forall\, v \in H(\mathrm{curl})$$

The de Rham Complex

$$
\begin{array}{ccccccc}
H^1 & \xrightarrow{\nabla} & H(\text{curl}) & \xrightarrow{\text{curl}} & H(\text{div}) & \xrightarrow{\text{div}} & L^2 \\
\bigcup & & \bigcup & & \bigcup & & \bigcup \\
W_h & \xrightarrow{\nabla} & V_h & \xrightarrow{\text{curl}} & Q_h & \xrightarrow{\text{div}} & S_h
\end{array}
$$



satisfies the **complete sequence property**

$$
\begin{aligned}
\text{range}(\nabla) &= \text{ker}(\text{curl}) \\
\text{range}(\text{curl}) &= \text{ker}(\text{div})
\end{aligned}
$$

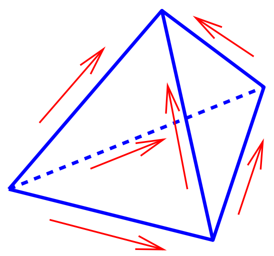
on the continuous and the discrete level.

Important for stability, error estimates, preconditioning, ...

Low-order $H(\text{curl})$ finite elements

First order Nédélec I elements:

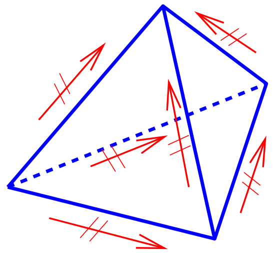
$$V_h = \{v \in H(\text{curl}) : v|_T = a_T + b_T \times x\}$$



first order approximation for A-field and B-field

First order Nédélec II elements:

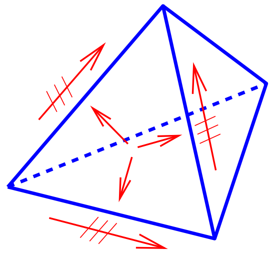
$$V_h = \{v \in H(\text{curl}) : v|_T \in [P^1]^3\}$$



second order for A-field, first order for B-field

Second order Nédélec II elements:

$$V_h = \{v \in H(\text{curl}) : v|_T \in [P^2]^3\}$$



third order for A-field, second order for B-field

If comparing resources vs. accuracy, second order elements are more efficient.

On the construction of high order $H(\mathrm{curl})$ finite elements

- [Dubiner, Karniadakis+Sherwin] H^1-conforming shape functions in tensor product structure
  $\longrightarrow$ allows fast summation techniques

- [Webb] $H(\mathrm{curl})$ hierarchical shape functions with local complete sequence property convenient to implement up to order 4

- [Demkowicz+Monk] Based on global complete sequence property construction of Nédélec elements of variable order (with constraints on order distribution) for hexahedra

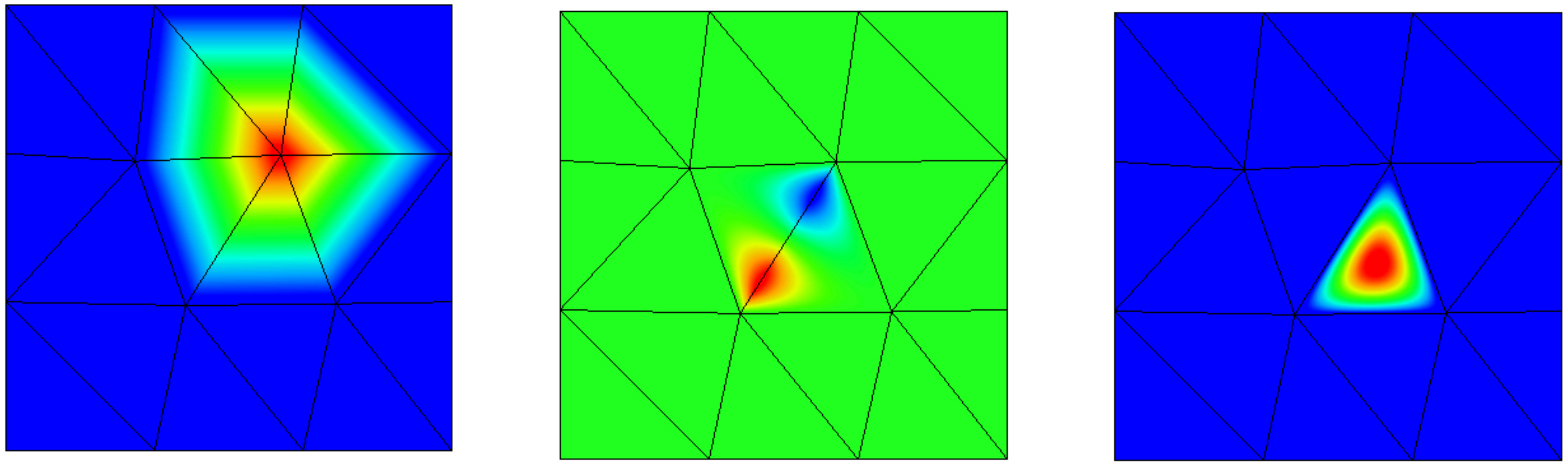
- [Ainsworth+Coyle] Systematic construction of $H(\mathrm{curl})$-conforming elements of arbitrarily high order for tetrahedra

- [Schöberl+Zaglmayr] Based on **local complete sequence property** and by using **tensor-product structure** we achieve a **systematic strategy** for the construction of $H(\mathrm{curl})$-conforming hierarchical shape functions of **arbitrary** and **variable order for common element geometries** (segments, quadrilaterals, triangles, hexahedra, tetrahedra, prisms).
  [COMPEL, 2005], PhD-Thesis Zaglmayr 2006

Hierarchical $V - E - F - C$ basis for H^1-conforming Finite Elements

The high order elements have basis functions connected with the vertices, edges, (faces,) and cell of the mesh:

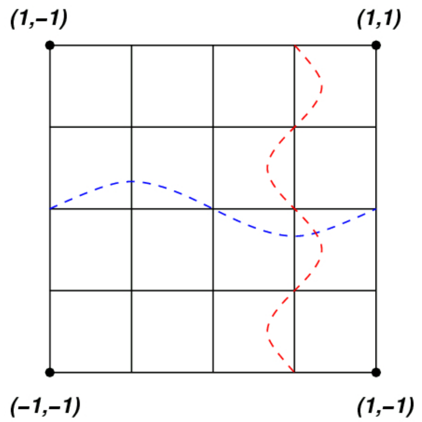
Vertex basis function — Edge basis function p=3 — Inner basis function p=3



This allows an individual polynomial order for each edge, face, and cell..

High-order H^1-conforming shape functions in tensor product structure

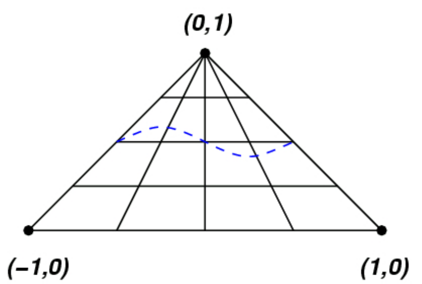
Exploit the tensor product structure of quadrilateral elements to build edge and face shapes



Family of orthogonal polynomials $(\,P_k^0[-1,1]\,)_{2\le k\le p}$ vanishing in ± 1.

$$\begin{aligned} \varphi^F_{i\,j}(x,y) &= P_i^0(x)\,P_j^0(y), \\ \varphi^{E_1}_i(x,y) &= P_i^0(x)\,\tfrac{1-y}{2}. \end{aligned}$$

Tensor-product structure for triangle [Dubiner, Karniadakis+Sherwin]:



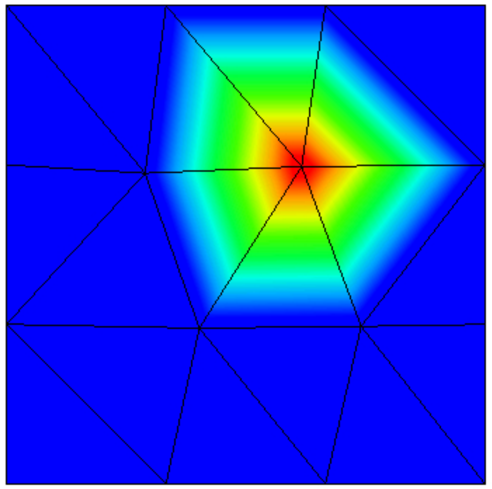
Collapse the quadrilateral to the triangle by $x \to (1-y)x$

$$\begin{aligned} \varphi^{E_1}_i(x,y) &= P_i^0(\tfrac{x}{1-y})\,(1-y)^i \\ \varphi^F_{i\,j}(x,y) &= \underbrace{P_i^0(\frac{x}{1-y})(1-y)^i}_{u_i(x,y)}\,\underbrace{P_j(2y-1)y}_{v_j(y)} \end{aligned}$$

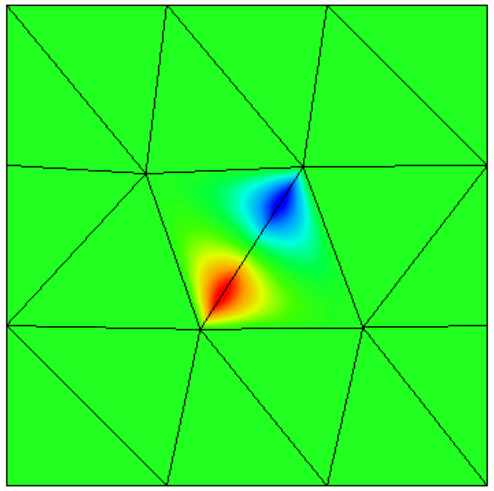
Remark: Implementation is free of divisions

The deRham Complex tells us that $\nabla H^1 \subset H(\mathrm{curl})$, as well for discrete spaces $\nabla W^{p+1} \subset V^p$.

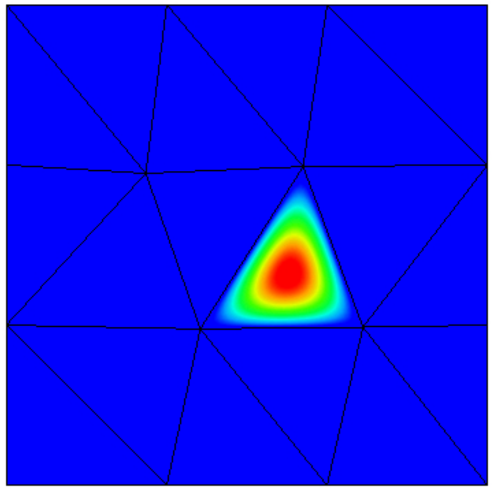
Vertex basis function



Edge basis function p=3

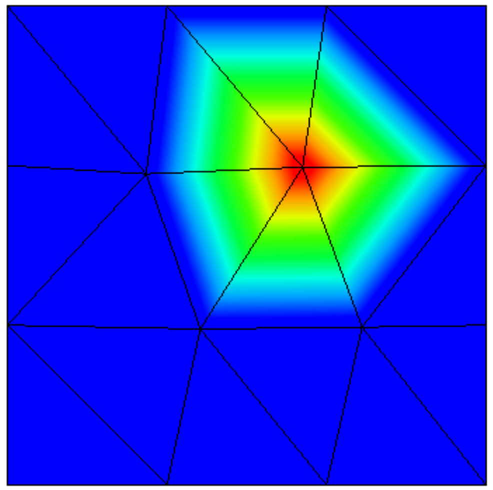


Inner basis function p=3

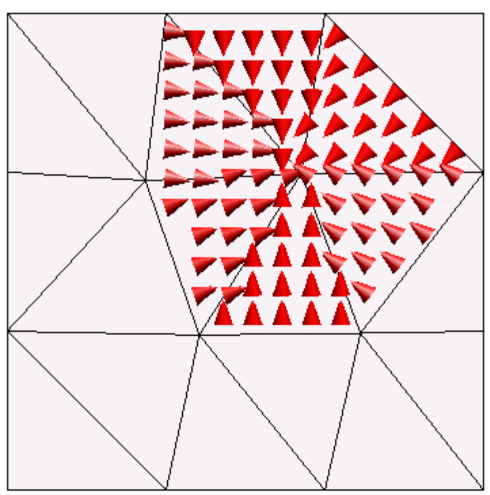


The deRham Complex tells us that $\nabla H^1 \subset H(\mathrm{curl})$, as well for discrete spaces $\nabla W^{p+1} \subset V^p$.

Vertex basis function

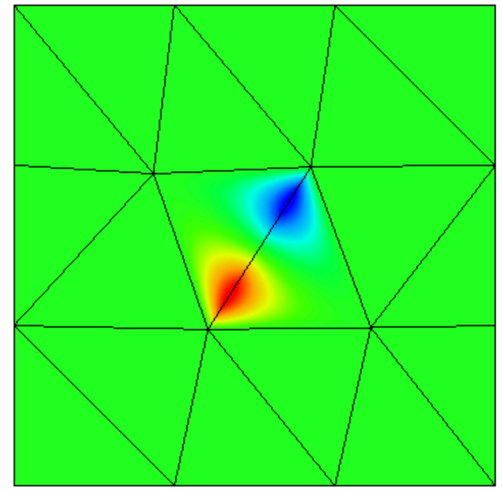


$\downarrow \nabla$

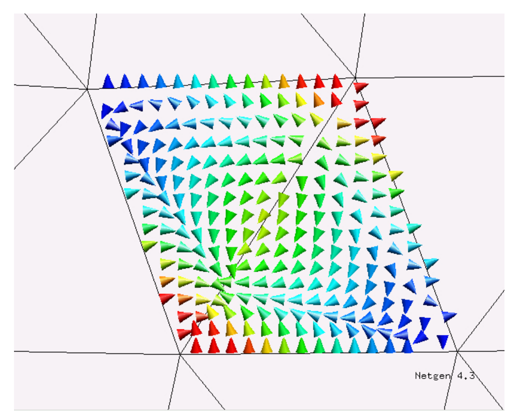


$$\nabla W_{V_i} \subset V_{\mathcal{N}_0}$$

Edge basis function p=3

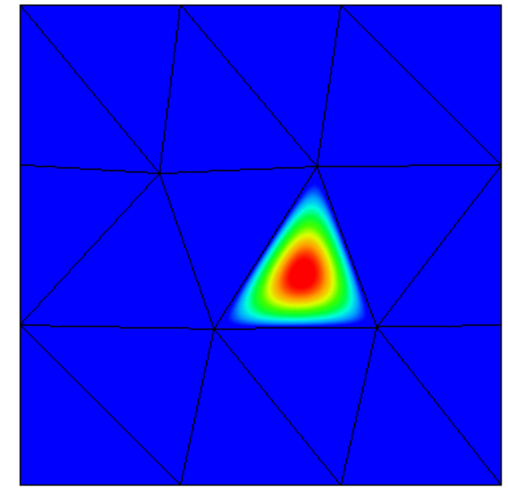


$\downarrow \nabla$

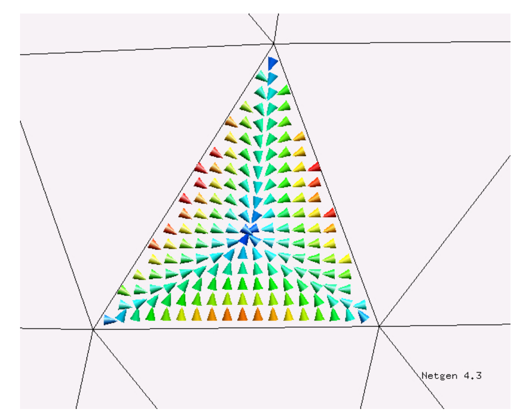


$$\nabla W_{E_k}^{p+1} = V_{E_k}^{p}$$

Inner basis function p=3



$\downarrow \nabla$



$$\nabla W_{F_k}^{p+1} \subset V_{F_k}^{p}$$

$H(\mathrm{curl})$-conforming face shape functions with $\nabla W_F^{p+1} \subset V_F^p$

We use inner H^1-shape functions spanning $W_F^{p+1} \subset H^1$ of the structure

$$\varphi_{i,j}^{F,\nabla} = u_i(x,y)\, v_j(y).$$

We suggest the following $H(\mathrm{curl})$ face shape functions consisting of 3 types:

- **Type 1: Gradient-fields**

$$\varphi_{1,\,i,j}^{F,curl} = \nabla\, \varphi_{i,j}^{F,\nabla} = \nabla(u_i\, v_j) = u_i\, \nabla v_j + v_j\, \nabla u_i$$

- **Type 2: other combination**

$$\varphi_{2,\,i,j}^{F,\mathrm{curl}} = u_i\, \nabla v_j - v_j\, \nabla u_i$$

- **Type 3:** to achieve a base spanning V_F $(p-1)$ lin. independent functions are missing

$$\varphi_{3,\,j}^{F,\mathrm{curl}} = \mathcal{N}_0(x,y)\, v_j(y).$$

Localized complete sequence property

We have constructed **V**ertex-**E**dge-**F**ace-**C**ell shape functions satisfying

$$
\begin{array}{lllllll}
W^V_{h,\,p+1=1} & \xrightarrow{\nabla} & V^{\mathcal{N}_0}_h & \xrightarrow{\text{curl}} & Q^{\mathcal{RT}_0}_h & \xrightarrow{\text{div}} & S_{h,\,0} \\
W^E_{p_E+1} & \xrightarrow{\nabla} & V^E_{p_E} & & & & \\
W^F_{p_F+1} & \xrightarrow{\nabla} & V^F_{p_F} & \xrightarrow{\text{curl}} & Q^F_{p_F-1} & & \\
W^C_{p_C+1} & \xrightarrow{\nabla} & V^C_{p_C} & \xrightarrow{\text{curl}} & Q^C_{p_C-1} & \xrightarrow{\text{div}} & S^C_{p_C-2}.
\end{array}
$$

Advantages are

- allows arbitrary and variable polynomial order on each edge, face and cell
- Additive Schwarz Preconditioning with cheap $\mathcal{N}_0 - E - F - C$ blocks gets efficient
- Reduced-basis gauging by skipping higher-order gradient bases functions
- discrete differential operators B_∇, B_{curl}, B_{div} are trivial

Iterative Equation Solvers

Since the systems of equations get huge, the goal is to apply iterative methods.

The performance of Krylov space methods as CG, QMR, GMRES heavily depends on the applied *preconditioner*. Classical examples for preconditioners are the Jacobi preconditioner ($C = \text{diag}\, A$), Block-Jacobi, SSOR, ADI, Incomplete Cholesky, ...

Concept of Multigrid

To solve the problem on the finest mesh, take additional, coarser meshes.

In the best case, you have a hierarchy of meshes.

Next, pick a cheap iteration on each level (e.g. Jacobi, Gauss-Seidel, ...). This iteration is called smoother.

Multigrid is a strategy to combine these cheap (but inefficient) preconditioners, to one, cheap and efficient preconditioner.

Each component is responsible for certain components.

Algebraic Multigrid (AMG): Only the fine grid matrix is given. Compute an artificial hierarchy.

S. Reitzinger + J.S.: AMG for Maxwell problems

Additive Schwarz preconditioning for parameter-dependent systems

Let

$$A = K + \kappa\, M \qquad \text{with} \qquad V_0 = \text{kern}\ \{K\}$$

The ASM-Lemma gives an explicit representation for the preconditioner C:

$$u^T C u = \inf_{\substack{u_i \in V_i \\ u = \sum u_i}} \sum u_i^T A u_i$$

.

Theorem: The AS-preconditioner is robust in $\kappa \in (0, 1]$, i.e.,

$$u^T C u \approx u^T A u$$

if and only if

$$V_0 = \sum_{i=1}^{m} V_i \cap V_0.$$

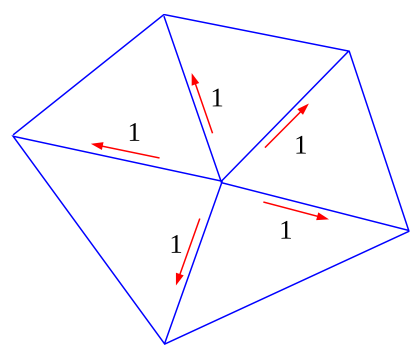
Hiptmair / Arnold-Falk-Winther smoothers for $H(\mathrm{curl})$ problems

There is $V_0 = \nabla W_h$. Use a block smoother with blocks V_i such that

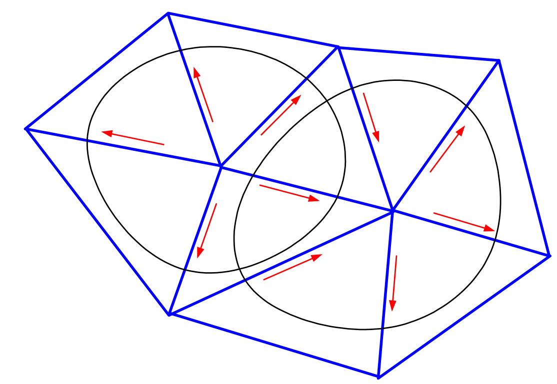
$$\forall\, j\, \exists i : \qquad \nabla W_j \subset V_i,$$

where the scalar space is decomposed as $W_h = \sum W_j$.

Gradient of vertex shape function: Hiptmair blocks



Arnold-Falk-Winther: Use large blocks:

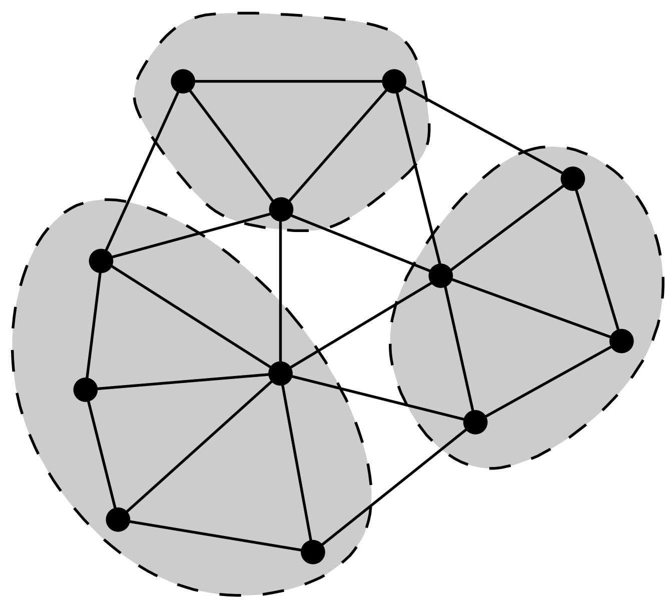


Hiptmair: one iteration is cheaper, less memory

Arnold-Falk-Winther: less iterations, simpler implementation

Algebraic coarsening based on Agglomeration

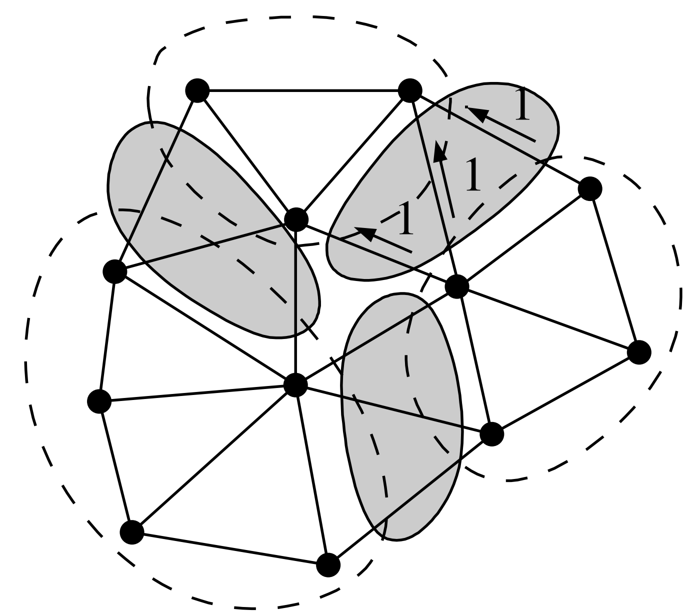
[S. Reitzinger + J. S., 2002]



Coarse grid vertices are defined by the mapping

Ind(.) : Vertex $\rightarrow$ Cluster

Allows to define the full coarse grid topology:
E_{IJ} is a coarse grid edge if and only if $I \neq J$, and there are fine grid vertices i and j s.t.:



$$I = Ind(i), \quad J = Ind(j), \qquad E_{ij} \text{ is a fine grid edge}$$

The 2-Level de Rham diagram:

$$
\begin{array}{ccccccc}
V^v & \xrightarrow{B_\nabla} & V^e & \xrightarrow{B_{\rm curl}} & V^f & \xrightarrow{B_{\rm div}} & V^c \\
\downarrow \Pi^W & & \downarrow \Pi^V & & \downarrow \Pi^Q & & \downarrow \Pi^S \\
V^v_{coarse} & \xrightarrow{B_\nabla} & V^e_{coarse} & \xrightarrow{B_{\rm curl}} & V^f_{coarse} & \xrightarrow{B_{\rm div}} & V^c_{coarse}
\end{array}
$$

- The algebraically constructed coarse spaces form a complete sequence. Thus, Hiptmair / Arnold+Falk+Winther smoothers can be applied.
- There are commuting interpolation operators. This is essential for the two-level analysis.

Model Problem

$\Omega = (0,1)^3$, $V = H_0(\text{curl})$, $f = (1,0,0)$,

Variational form:

$$\int \text{curl}\ u \,\text{curl}\ v\, dx + 10^{-3} \int uv\, dx = \int fv\, dx$$

Variable V cycle:

N_h^e	setup (sec)	solver (sec)	iteration
4184	0.15	0.31	11
31024	1.32	6.21	15
238688	11.39	63.93	17

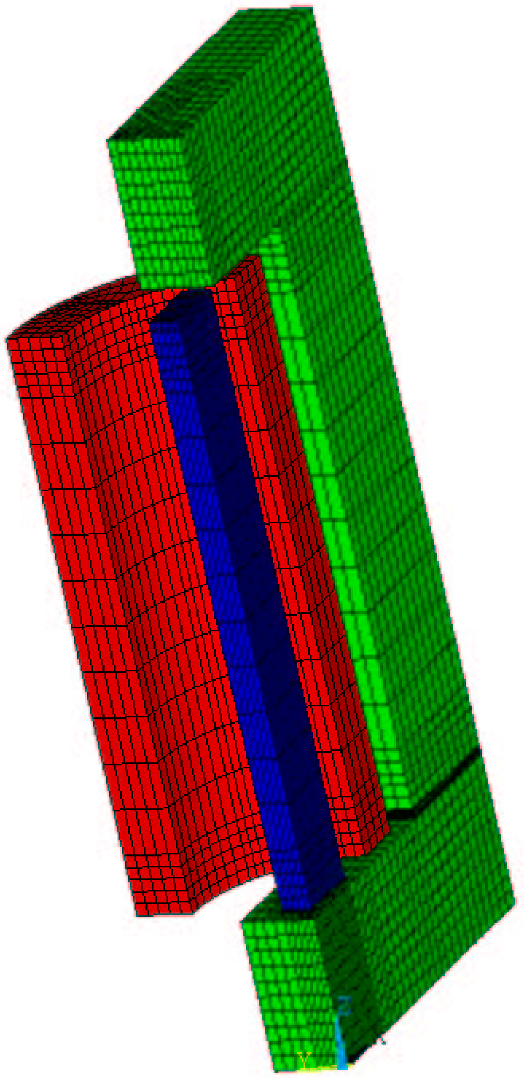
Computation with Stefan Reitzinger's AMG code Pebbles, CPU = PIII 1 GHz

TEAM 20 Benchmark problem

Coil and Iron core, small air gap.

Unknowns:	240E3
Iterations:	26
Solution time:	90 sec

Computations by Manfred Kaltenbacher,
University Erlangen, Germany
Using the code Pebbles



Schwarz-Preconditioning for High order H(curl) elements

The global stiffness matrix is split into the according unknowns:

$$A = \begin{pmatrix} A_{\mathcal{N}_0\mathcal{N}_0} & A_{\mathcal{N}_0 E} & A_{\mathcal{N}_0 F} & A_{\mathcal{N}_0 C} \\ A_{E\mathcal{N}_0} & A_{EE} & A_{EF} & A_{EC} \\ A_{F\mathcal{N}_0} & A_{FE} & A_{FF} & A_{FC} \\ A_{C\mathcal{N}_0} & A_{CE} & A_{CF} & A_{CC} \end{pmatrix}.$$

A cheap preconditioner is the $\mathcal{N}_0$-E-F-I block Jacobi-preconditioner

$$C = \begin{pmatrix} A_{\mathcal{N}_0\mathcal{N}_0} & 0 & 0 & 0 \\ 0 & \widetilde{A}_{EE} & 0 & 0 \\ 0 & 0 & \widetilde{A}_{FF} & 0 \\ 0 & 0 & 0 & A_{CC} \end{pmatrix}.$$

The Nedelec-0 block plays a special role: It is solved exactly, or, an h-version preconditioner is applied.

Space splitting and local complete sequence property

The potential FE-space is split into Vertex-Edge-Face-Cell blocks

$$W_{p+1} = W_V + \sum_E W_E + \sum_F W_F + \sum_C W_C \quad \subset H^1$$

and the $H(\mathrm{curl})$ by lowest order Nedelec-(high order)Edge-Face-Cell based hierarchic spaces

$$V_p = V_{\mathcal{N}_0} + \sum_E V_E + \sum_F V_F + \sum_C V_C \quad \subset H(\mathrm{curl})$$

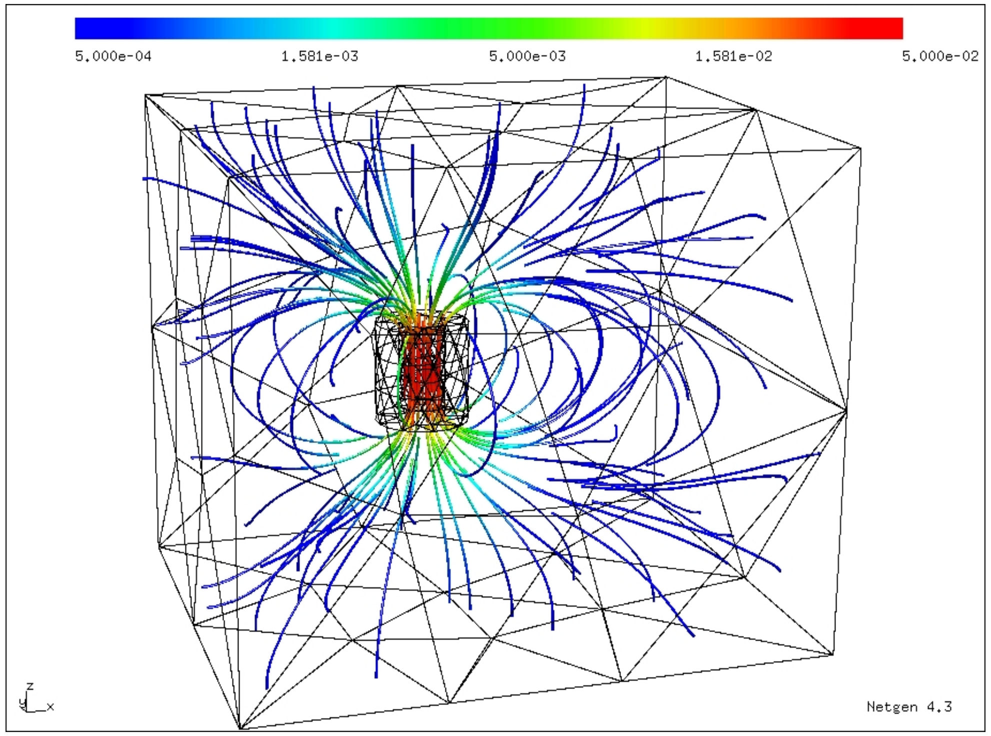
The $H(\mathrm{curl})$-splitting is compatible with the kernel $V_0 = \nabla W_{p+1}$, if the sequences associated with the edge, face, and cell nodes are complete (local complete sequence property):

$$\begin{array}{lllllll}
W^V_{h,\,p+1=1} & \xrightarrow{\nabla} V^{\mathcal{N}_0}_h & \xrightarrow{\mathrm{curl}} & Q^{\mathcal{RT}_0}_h & \xrightarrow{\mathrm{div}} & S_{h,\,0} \\
W^E_{p_E+1} & \xrightarrow{\nabla} V^E_{p_E} & & & & \\
W^F_{p_F+1} & \xrightarrow{\nabla} V^F_{p_F} & \xrightarrow{\mathrm{curl}} & Q^F_{p_F-1} & & \\
W^C_{p_I+1} & \xrightarrow{\nabla} V^C_{p_I} & \xrightarrow{\mathrm{curl}} & Q^C_{p_I-1} & \xrightarrow{\mathrm{div}} & S^C_{p_I-2}
\end{array}$$

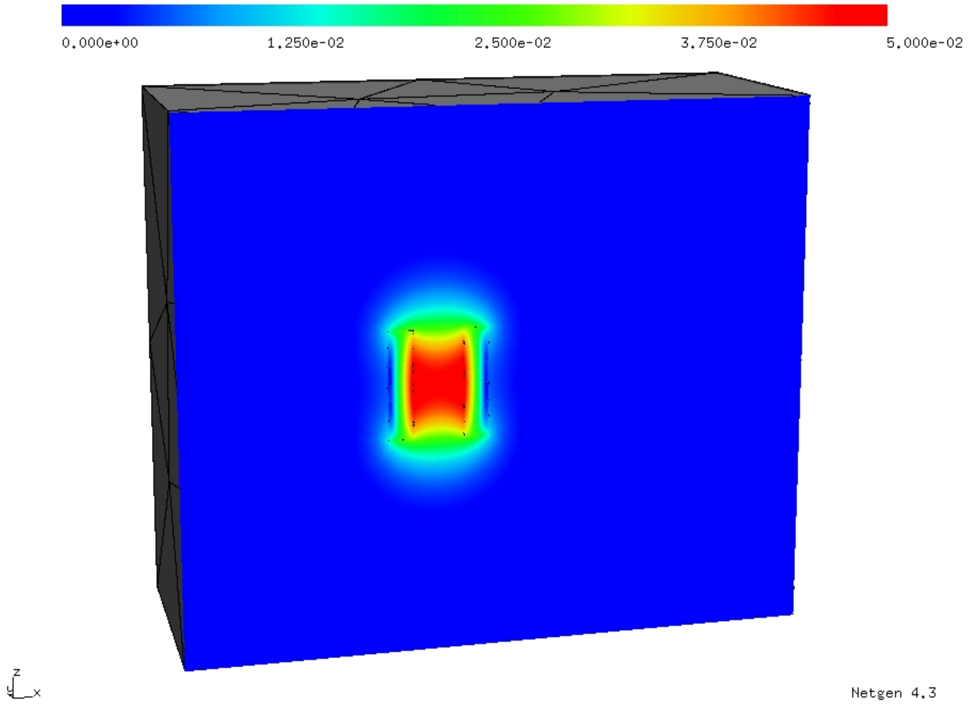
Magnetostatic boundary value problem - Numerical Results

Simulation of the magnetic field induced by a coil with prescribed currents (regularized formulation)

$$\text{Find} A \in H(\text{curl}) : \quad (\mu^{-1} \operatorname{curl} A, \operatorname{curl} v) + \epsilon\,(u, v) = (j, v) \qquad \forall v \in H(\text{curl})$$



Field-lines induced by a coil, p=6.



Absolute value $|B| = |\operatorname{curl} A|$.

Tetrahedral mesh with 2035 curved elements.

Reduced Basis Gauging

- regularization term for lowest-order subspace
- skipping higher-order gradients basis functions

Reduced-base vs. full-space regularization in simulation of coil-problem:

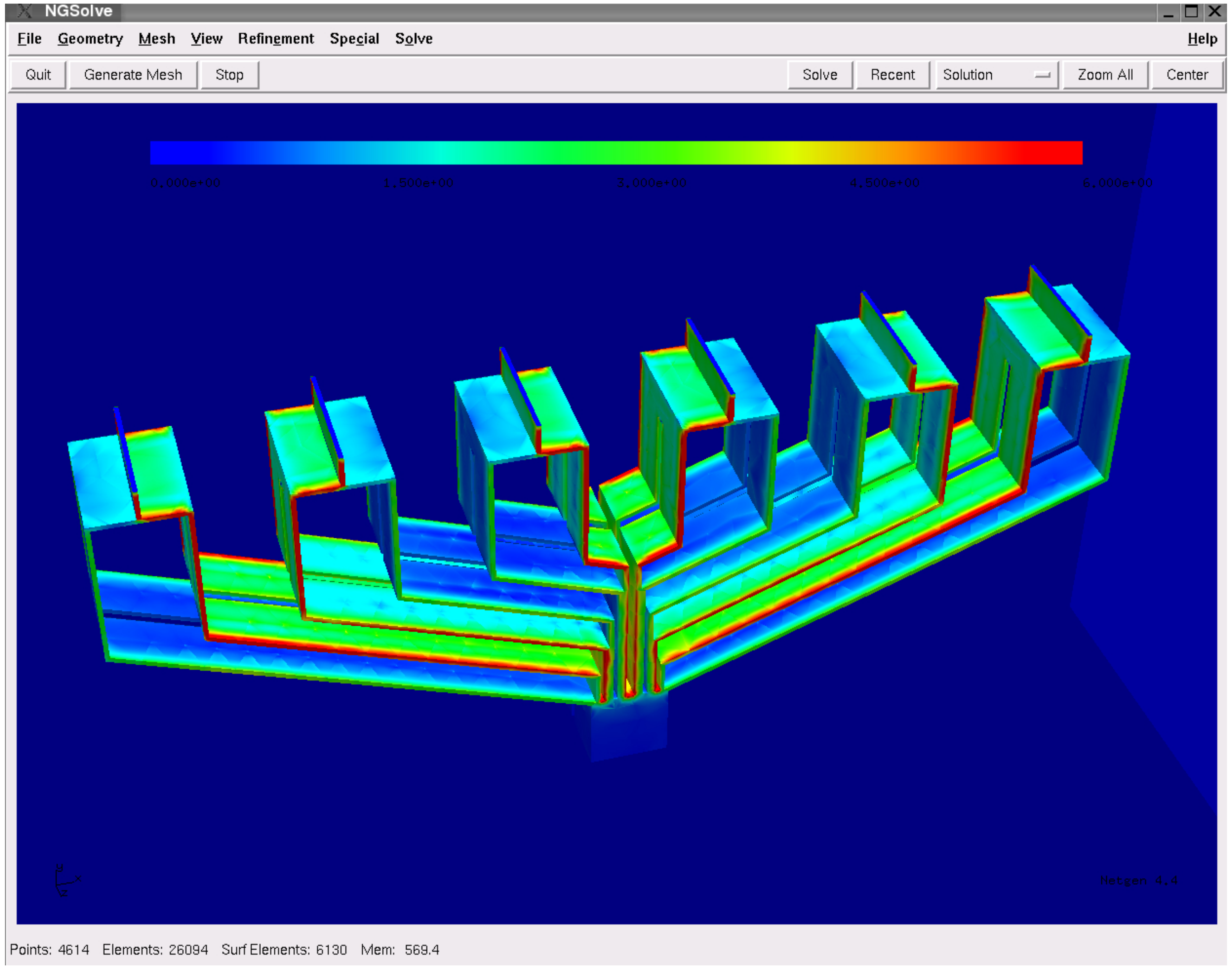
In reduced system about a third less shape functions $\rightarrow \approx 55\%$ faster integration

p	dofs	reduced/full	$\kappa(C^{-1}A)$	iterations	solver time
2	19719	full	7.9	20	1.9 s
2	10686	reduced	7.9	21	0.7 s
3	50884	full	24.2	32	9.8 s
3	29130	reduced	18.2	31	2.9 s
4	104520	full	71.4	48	40.5 s
4	61862	reduced	32.3	40	10.7 s
5	186731	full	179.9	69	137.9 s
5	112952	reduced	55.5	49	31.9 s
6	303625	full	421.0	97	427.8 s
6	186470	reduced	84.0	59	87.4 s
7	286486	reduced	120.0	68	209.6 s

Note: the computed $B = \mathrm{curl}\ A$ are the same for both versions.

Connection of a transformer switch (bus-bar)

Gradients can be skipped in non-conducting domains in Eddy-current problems.

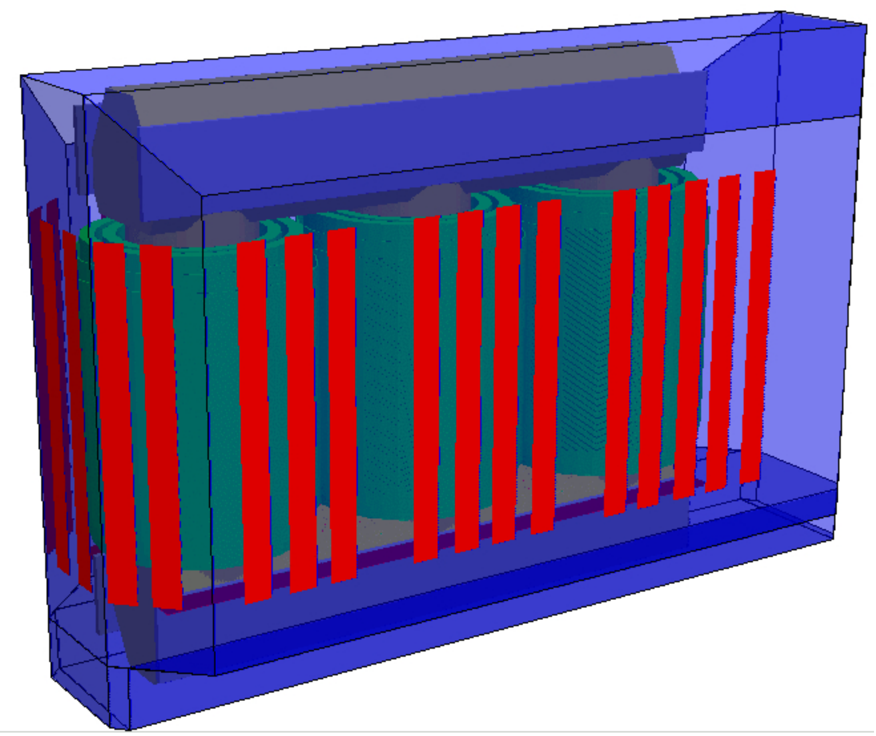


Full base for $p = 3$ in conductor, reduced base for $p = 3$ in air
$n \approx 450k$, 20 min on P4 Centrino, 1600MHz

Magnetic field simulation in a transformer

Project with Siemens / EBG Transformatorenbau, Linz:

Three phase transformer

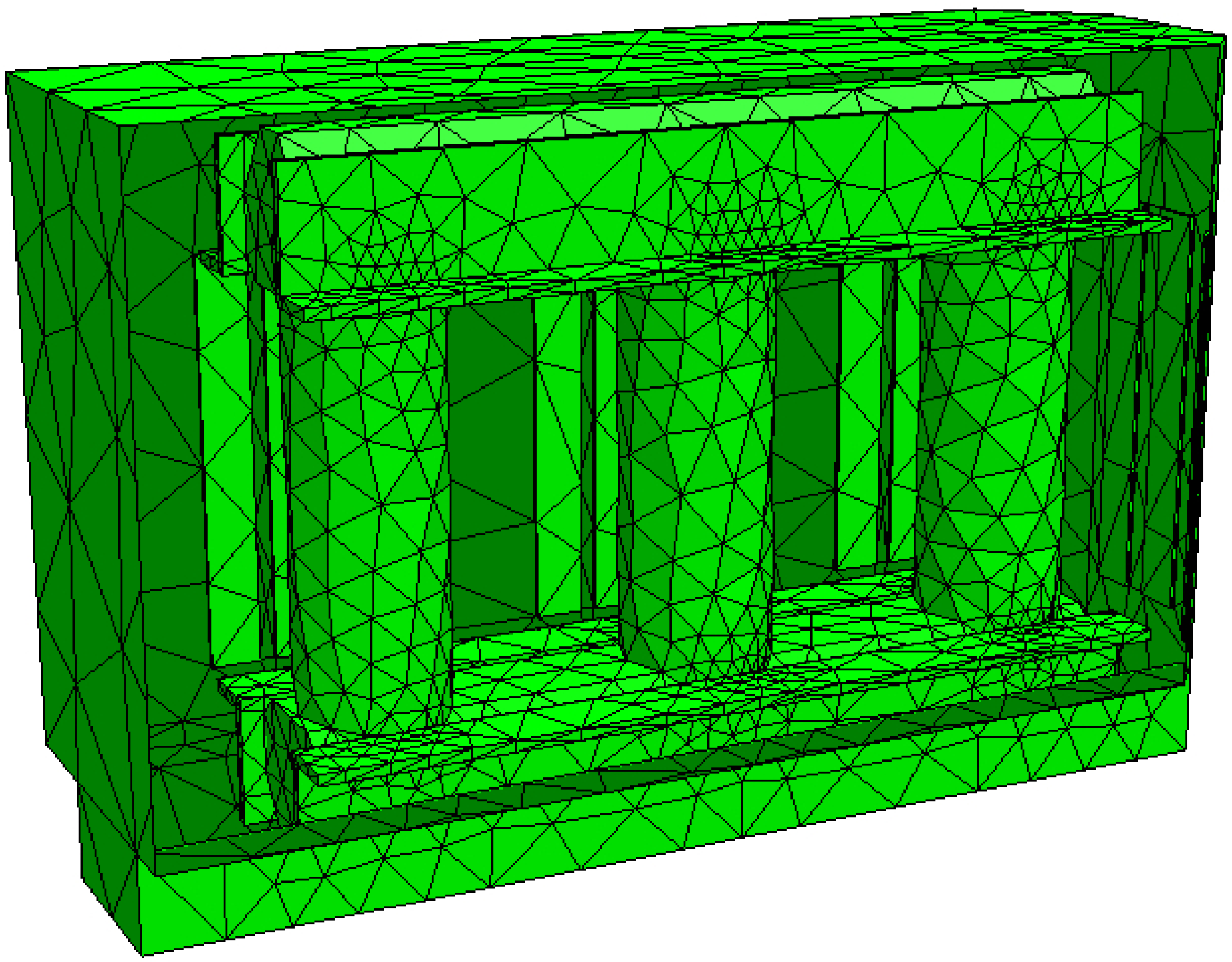


- prescribed current sources in coils
- main flux though layered core
- flux penetrating the casing causes eddy currents
- thin shields collecting stray fluxes

Model

- Time harmonic, low frequency
- Nonlinear terms due to saturation in casing

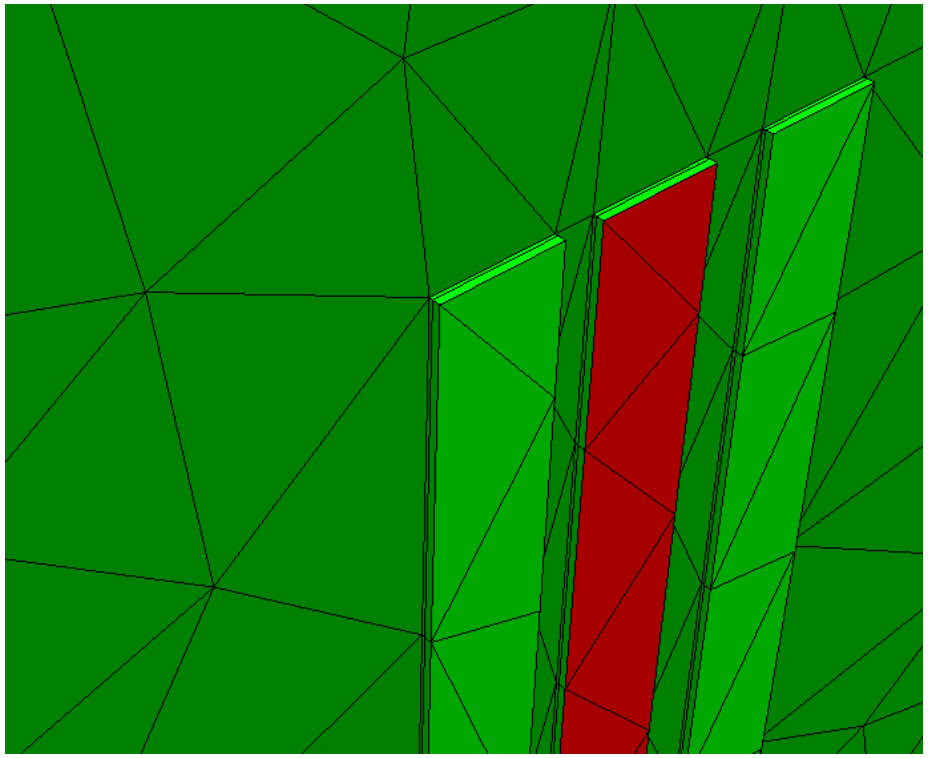
Coarse Mesh to resolve the geometry



22k elements, 26k complex dofs

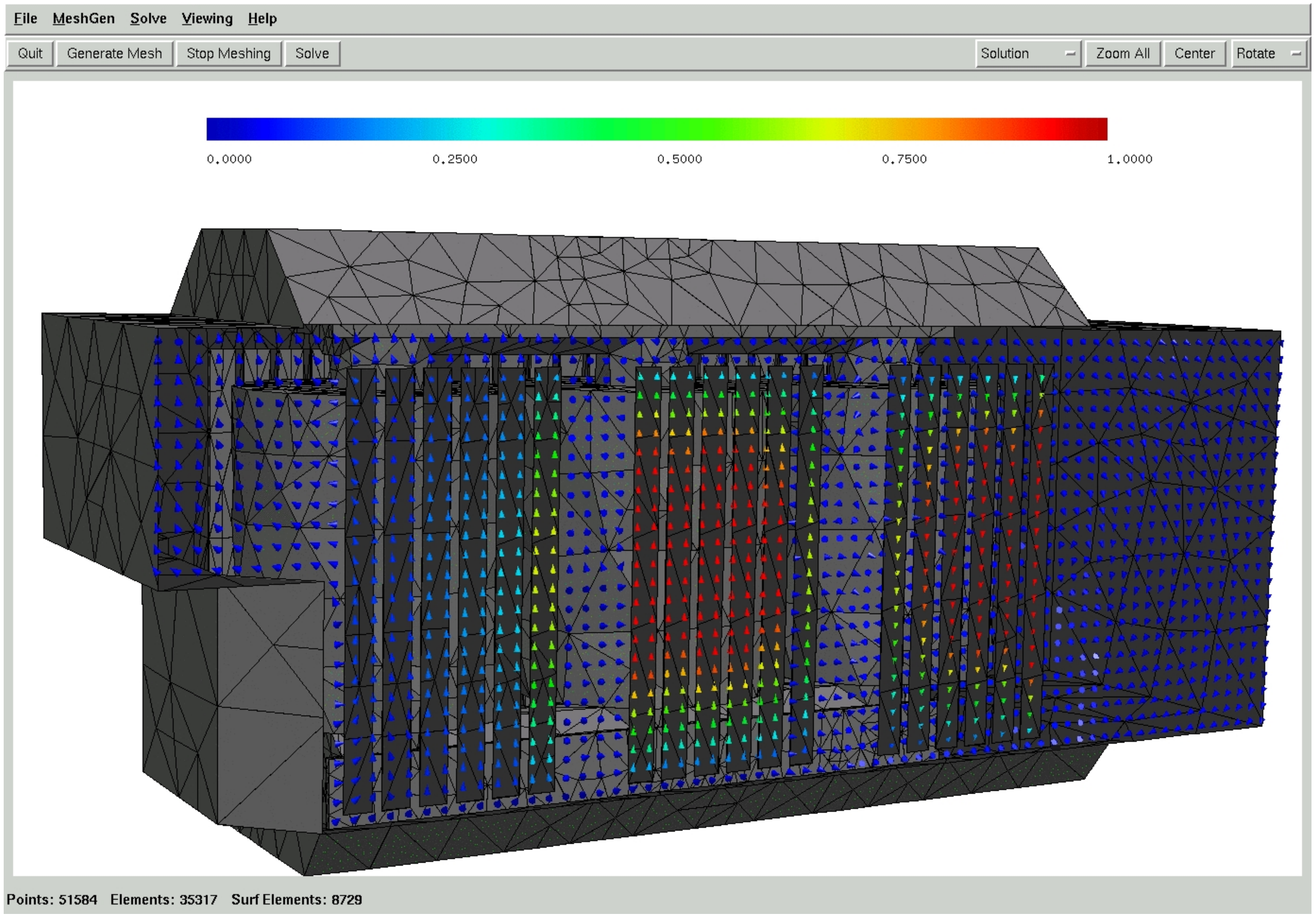
Mesh generated by Netgen

Anisotropic elements for thin shields

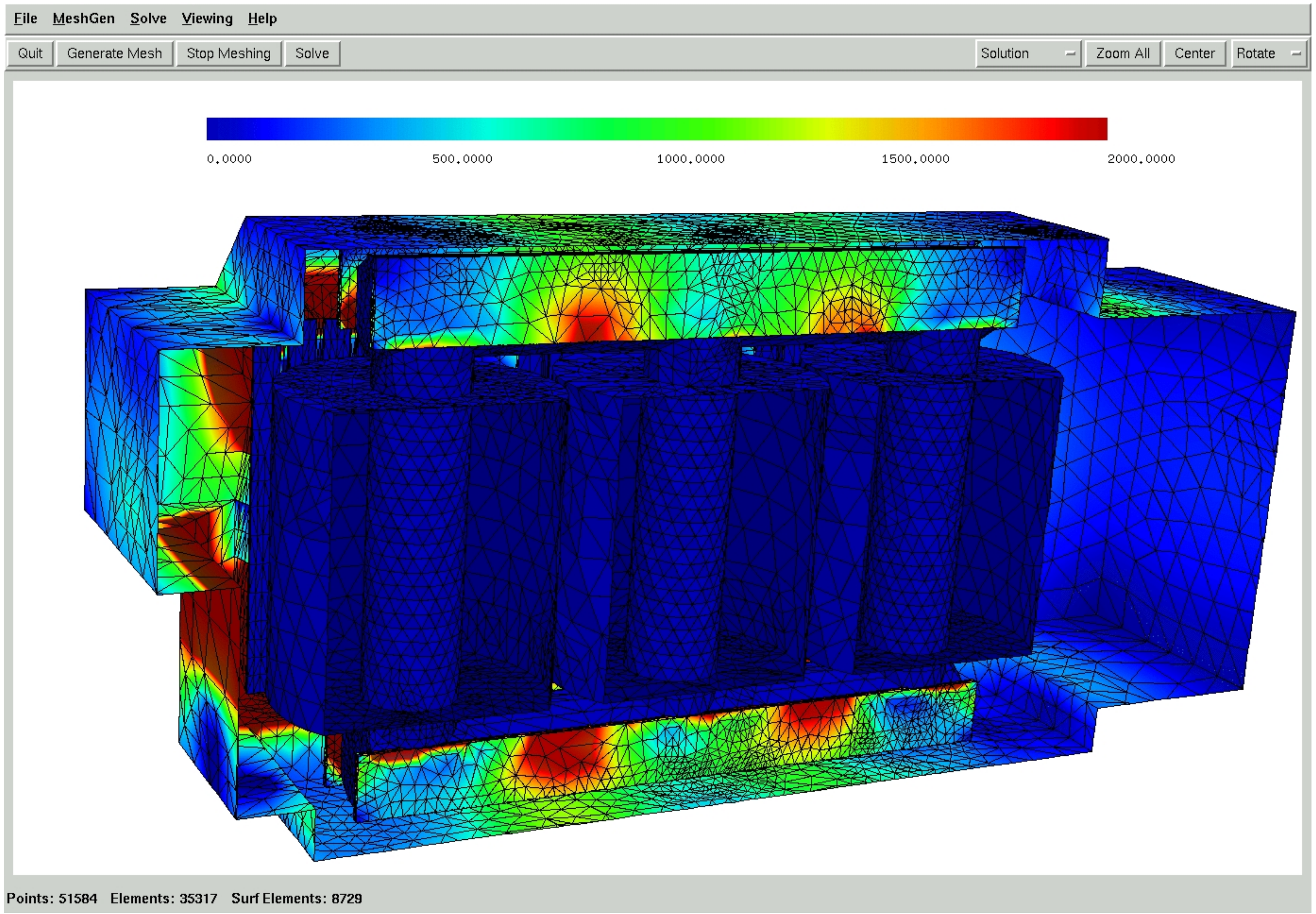


- Total object size: 6m, Thickness of shields: 2cm
- Prism elements in and behind shields, pyramid transition elements

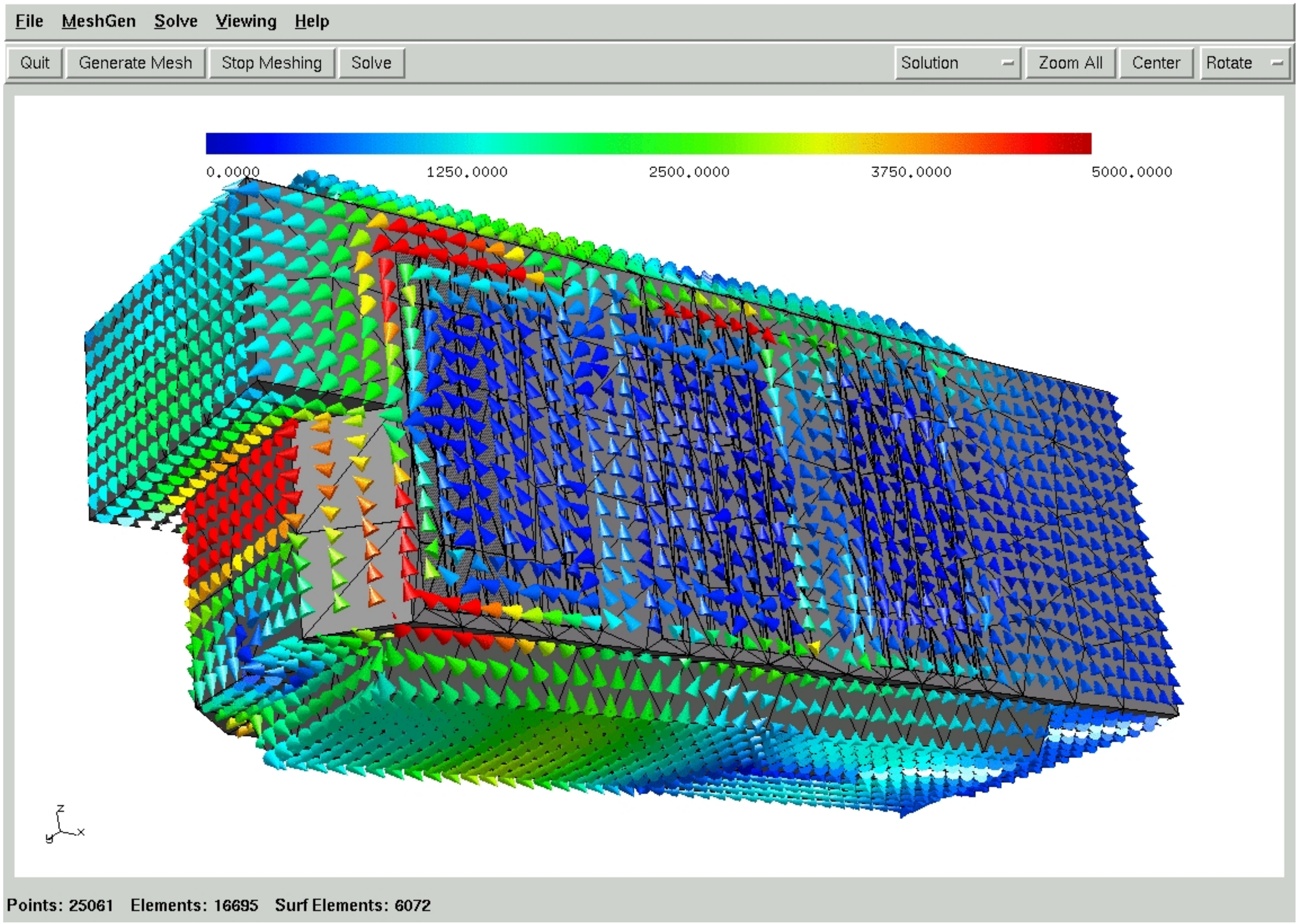
Magnetic flux density



Loss density in pressing plates



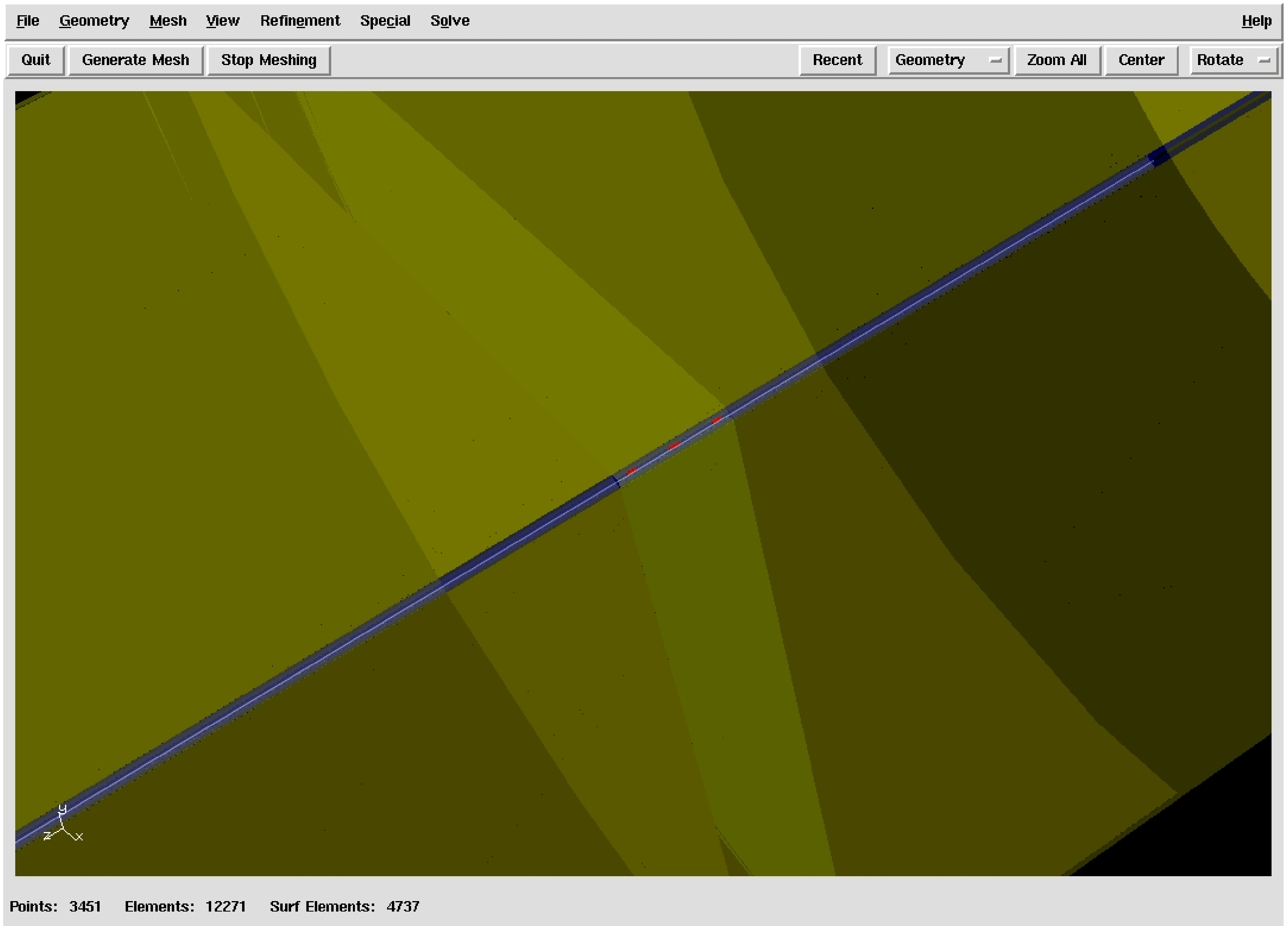
Eddy current density



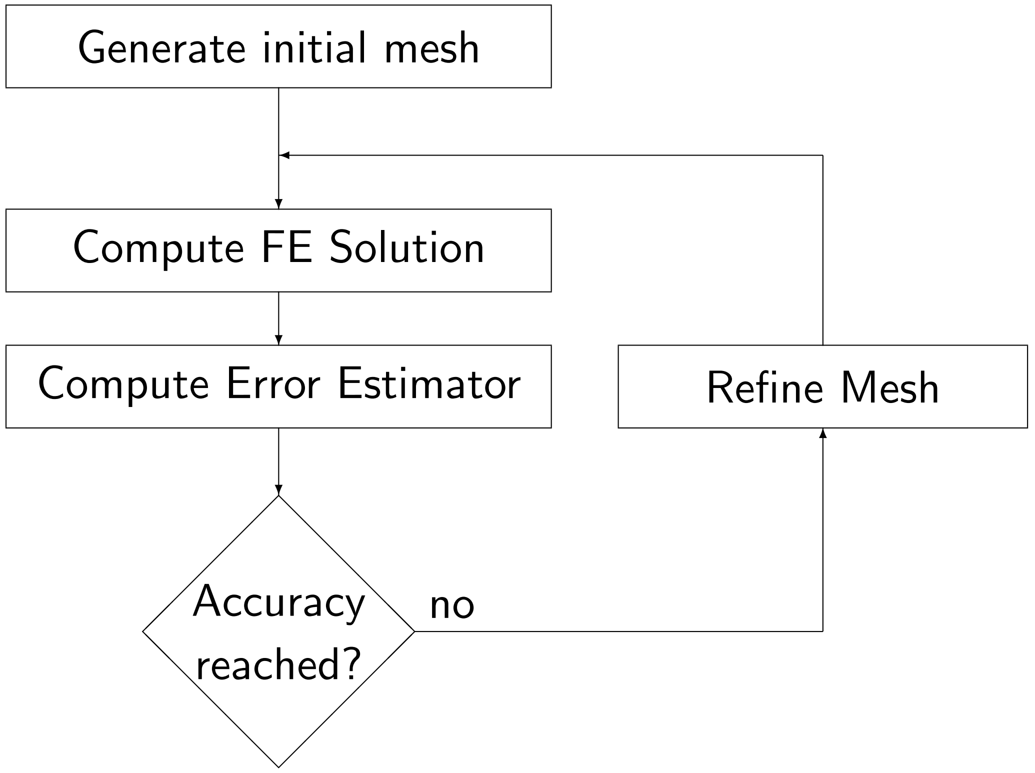
Simulation data

- Simulation with our code Netgen / NgSolve
- Second order type 2 - Nédélec elements
- 4 Levels of adaptive refinement, 500k complex unknowns
- 2 Newton iterations per level
- about 20 QMR-Multigrid iterations per Newton iteration
- Simulation time on PC, 2.4 GHz: 10 min

Bore-hole Electromagnetics



Local mesh refinement based on a posteriori error estimators



Energy norm error estimators

Infinite dimensional variational problem: Find $A \in V$ such that

$$B(A, v) = f(v) \qquad \forall\, v \in V,$$

Finite element problem: Find $A_h \in V_h$ such that

$$B(A_h, v_h) = f(v_h) \qquad \forall\, v_h \in V_h,$$

Element-wise energy norm error estimators:

$$\eta^2(A_h) = \sum_T \eta_T^2(A_h),$$

such that

$$\|A - A_h\|_B \simeq \eta(A_h).$$

Goal driven error estimates

One is interested in some quantity $y(A)$ depending on the solution, e.g., the closed loop voltage.
R. Rannacher's feedback ee and T. Oden's goal driven ee focus on the error

$$y(A) - y(A_h)$$

The key is to define the dual problem with the functional as r.h.s:

$$B(w, v) = y(v)$$

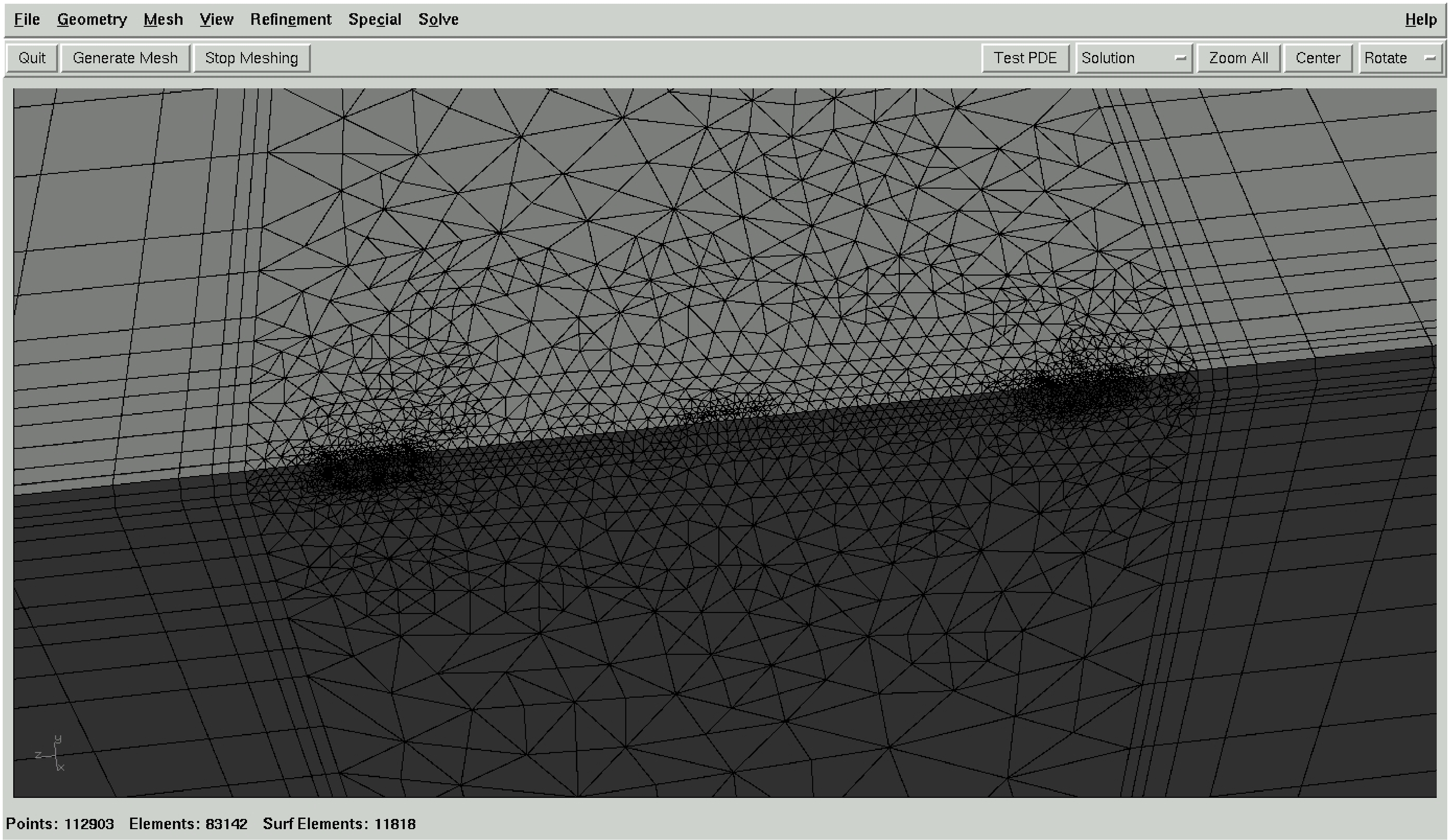
One observes

$$y(A) - y(A_h) = y(A - A_h) = B(w, A - A_h) = B(w - w_h, A - A_h)$$

One version of the goal driven error estimator is

$$y(A) - y(A_h) \simeq \sum_T \eta_T(w_h)\eta_T(A_h)$$

Goal: Compute closed loop voltage in coil 3

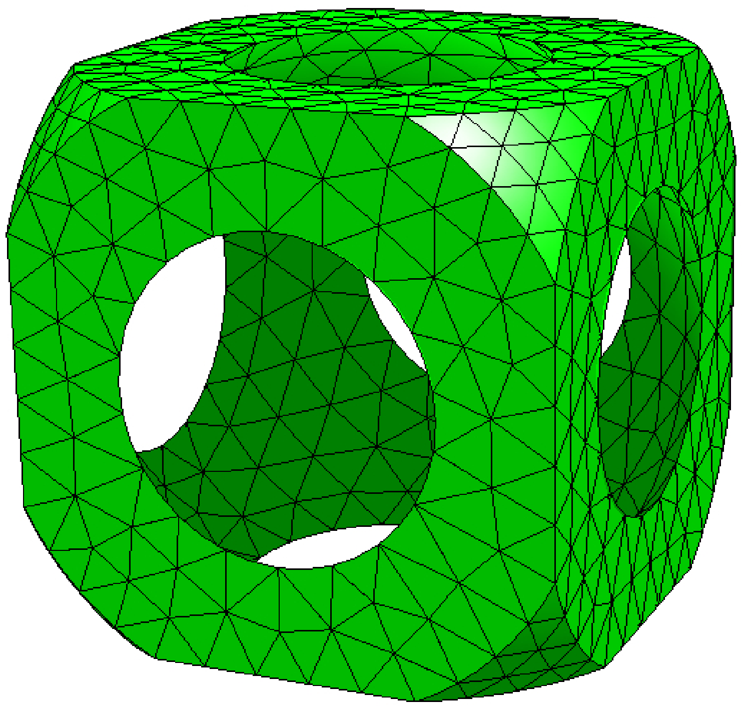


Netgen/NgSolve Software

- **NETGEN**: An automatic tetrahedral mesh generator
  - Internal CSG based modeller
  - Geometry import from IGES/Step or STL
  - Delaunay and advancing front mesh generation algorithms
  - Arbitrary order curved elements
  - Visualization of meshes and fields
  - Open Source (LGPL), 100-150 downloads / month
- **NgSolve**: A finite element package
  - Mechanical and magnetic field problems
  - Iterative solvers with multigrid preconditioning
  - Adaptive mesh refinement
  - High order finite elements
  - Intensively object oriented C++ (Compile time polymorphism by templates)
  - Open Source, available via CVS

Constructive Solid Geometry (CSG) in Netgen

Complicated objects are described by Eulerian operations applied to (simple) primitives.

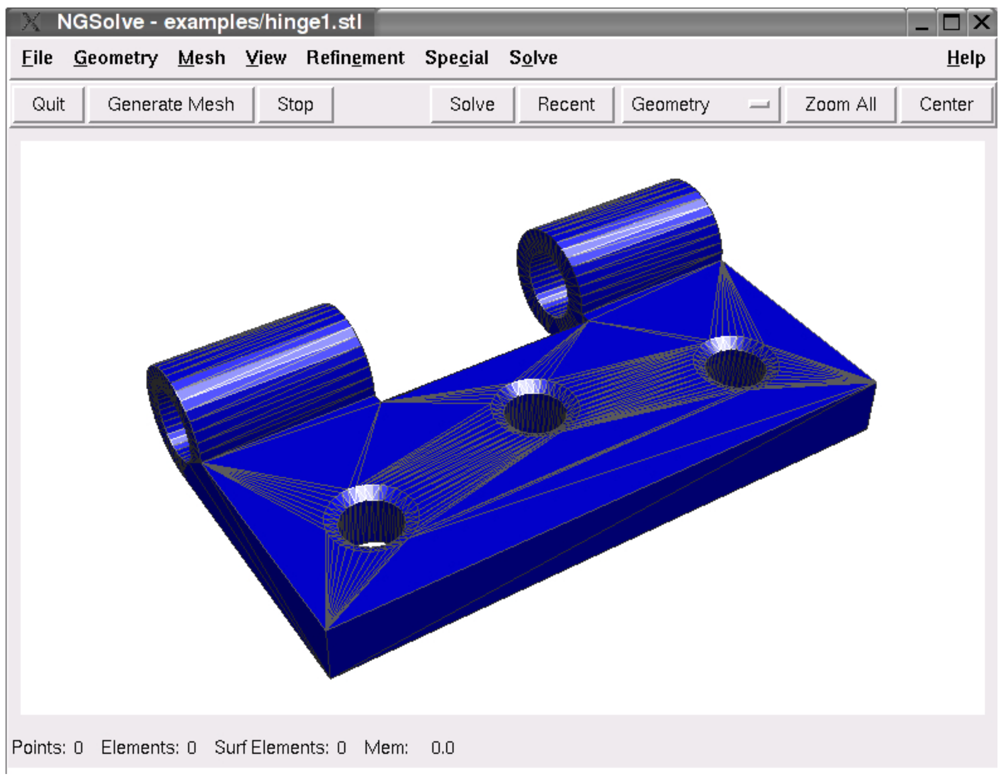


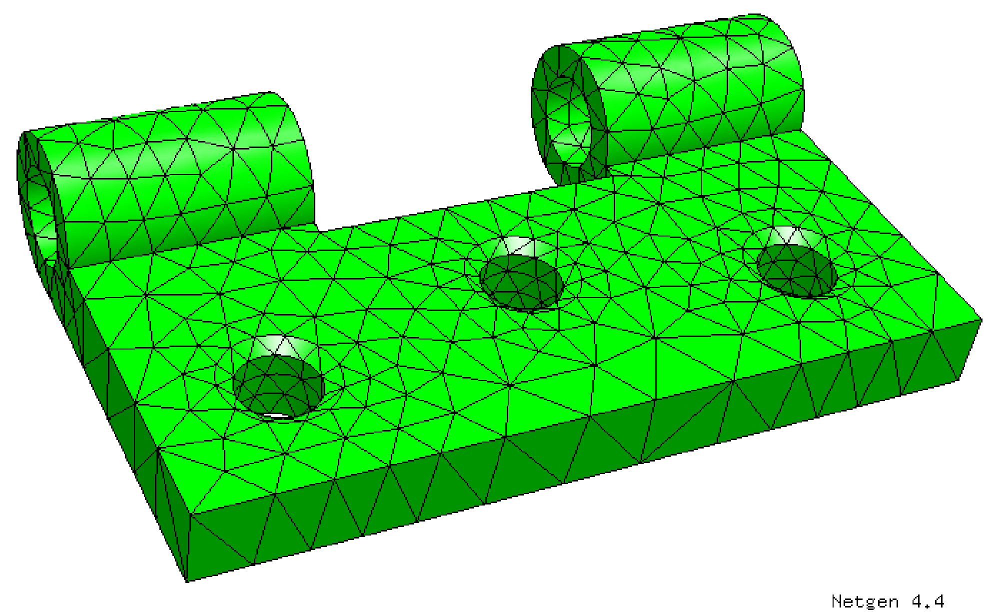
```
solid cube =
        plane (0, 0, 0; 0, 0, -1)
    and plane (0, 0, 0; 0, -1, 0)
    and plane (0, 0, 0; -1, 0, 0)
    and plane (100, 100, 100; 0, 0, 1)
    and plane (100, 100, 100; 0, 1, 0)
    and plane (100, 100, 100; 1, 0, 0);
solid main =
        cube
    and sphere (50, 50, 50; 75)
    and not sphere (50, 50, 50; 60);
```

Very useful for simple to moderate complexity

Surface Triangulated Geometry (STL)

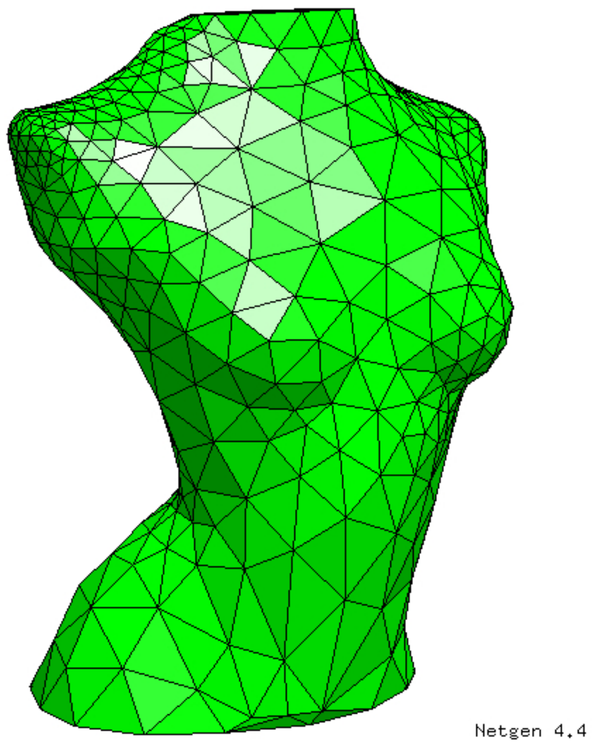
Geometry is defined by a surface triangulation:
(with J. Gerstmayr)

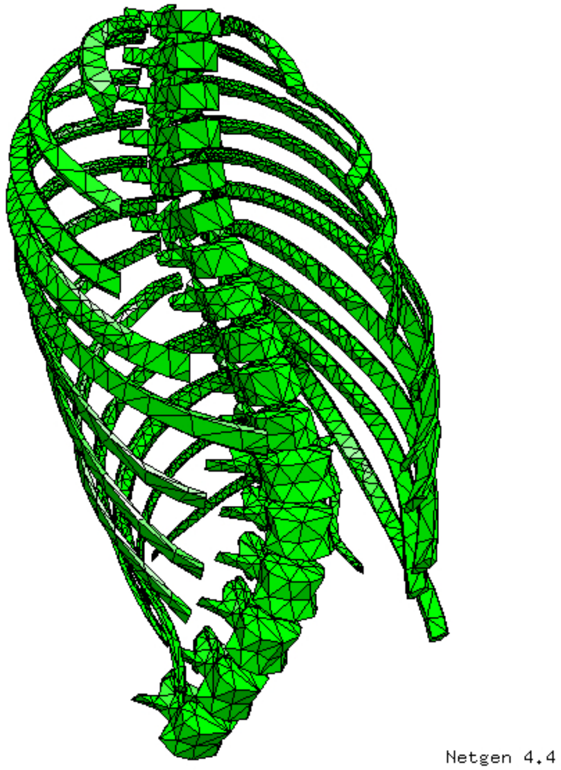


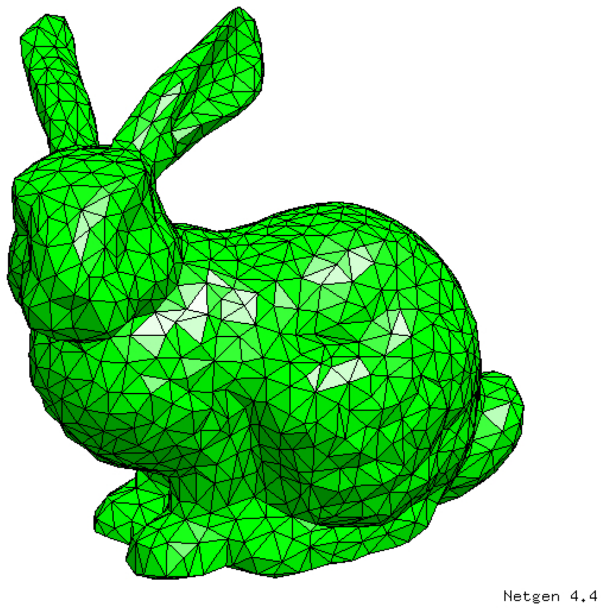


Stretched elements are suited for approximating the geometry

Biomechanical applications



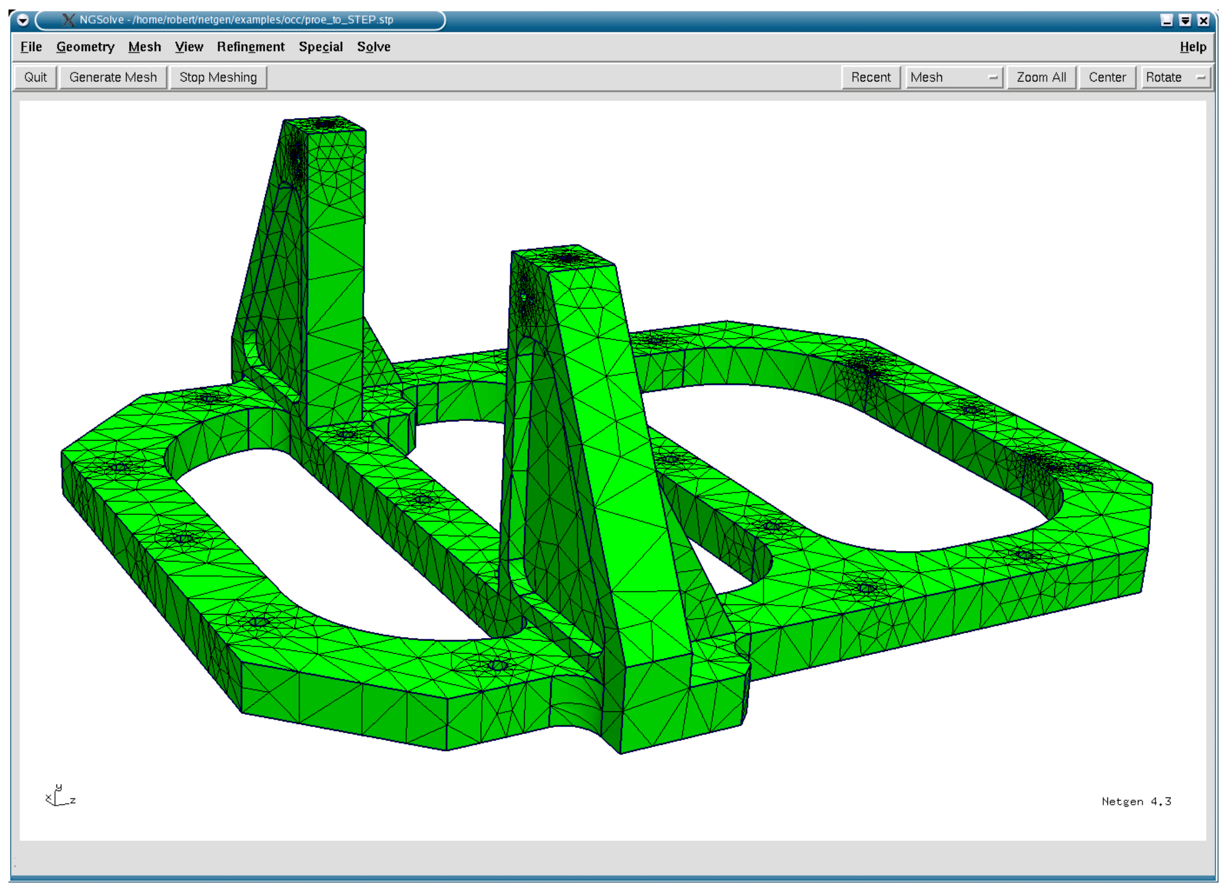




Qinghu Liao, Polytechnique Montreal: Computation of deformation of scoliosis
Bone geometry from X-ray, trunk from optical 3D camera, STL goemetry

Meshing from standard IGES/Step files

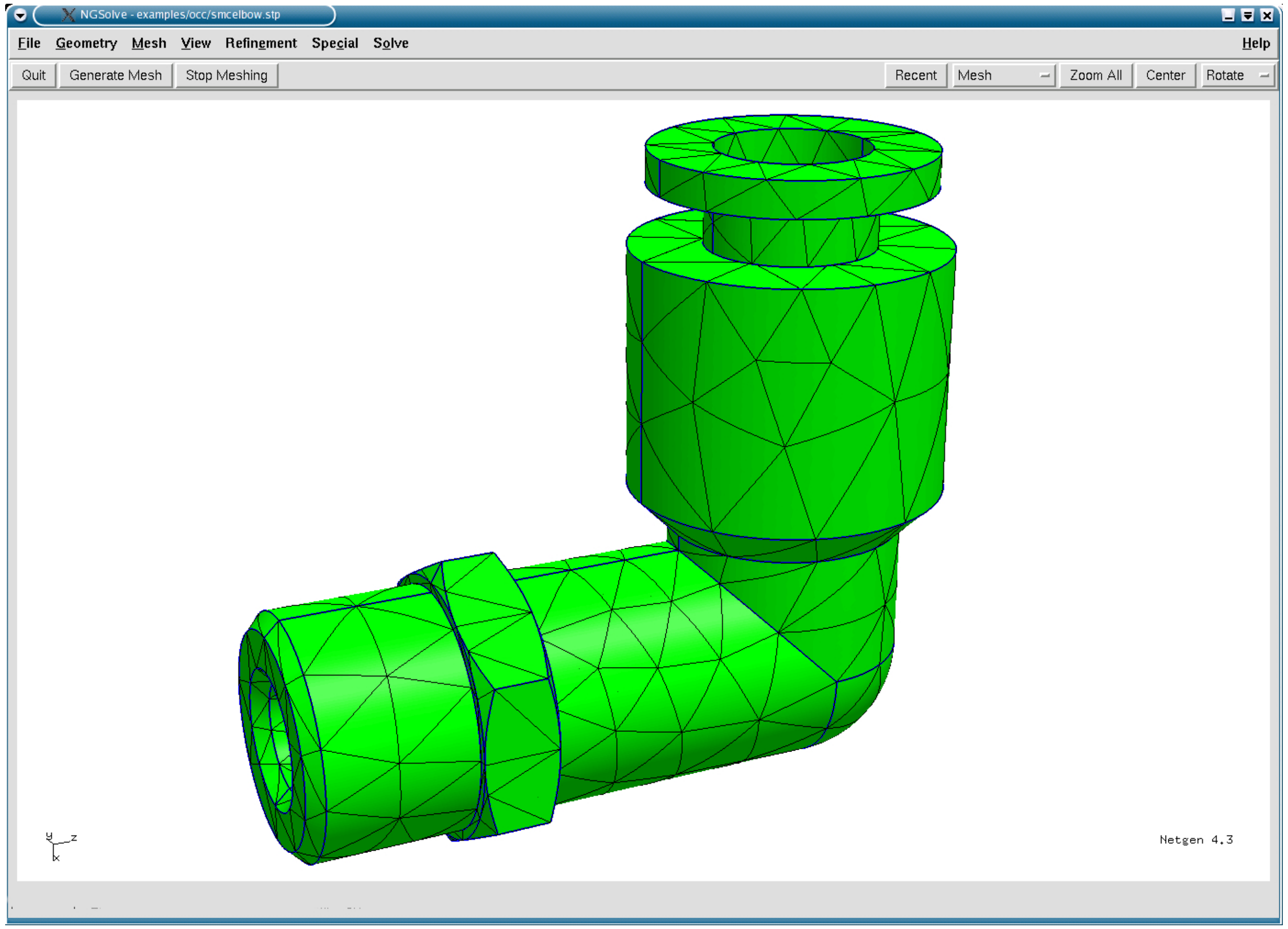
We (R. Gaisbauer) have integrated the opensource geometry kernel OpenCascade (by OpenCascade S.A.) into Netgen.



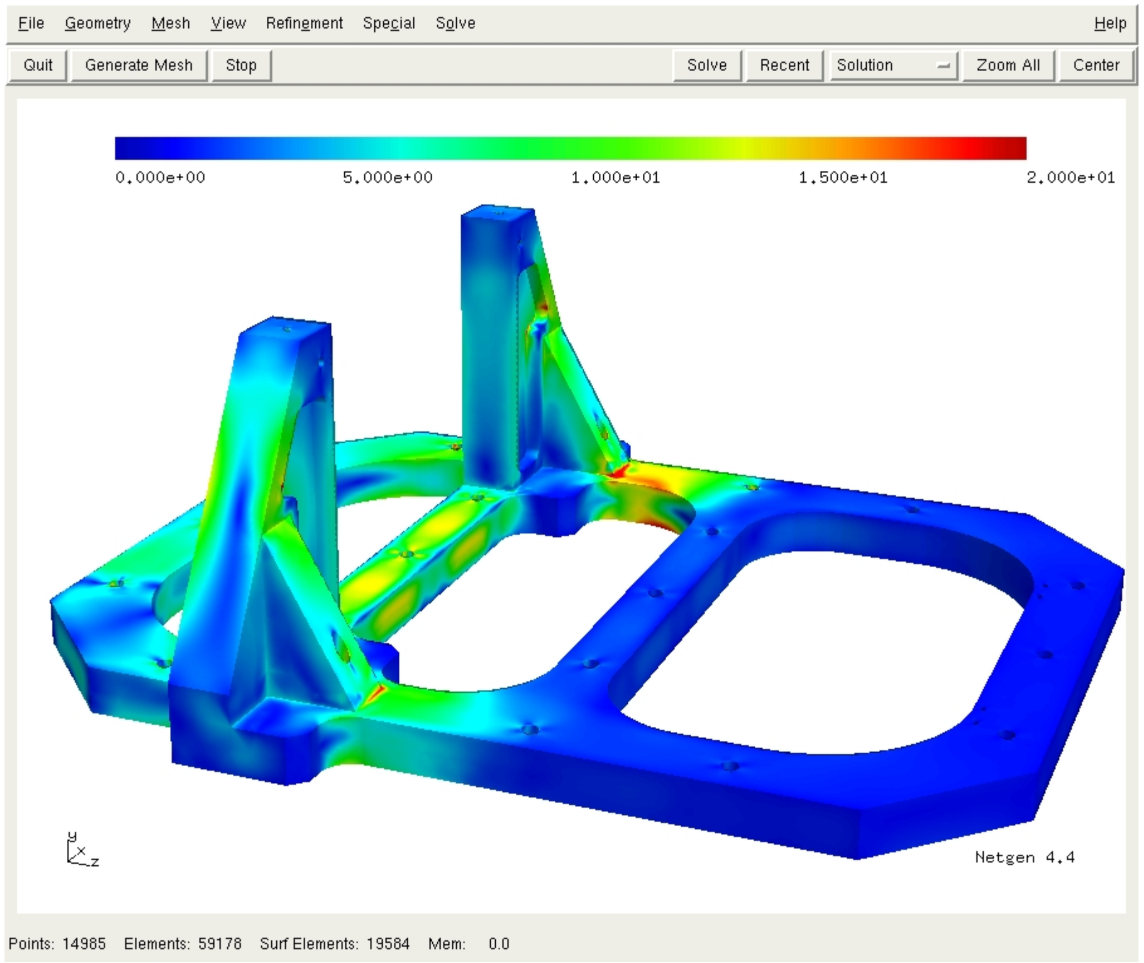
Work with true geometry (splines, nurbs, extrusion, rotational objects etc.)
Can access the geometry via a programming interface (topology, metric)

Curved elements

Curved elements of arbitrary order by **projection based interpolation** [Demkowicz]:
Element deformation is described by hierarchical finite element basis functions. Define edge-modes by H_0^1-projection on edges, and face-modes by H_0^1-projections on faces.



Von-Mises Stresses in a Machine Frame (linear elasticity)



Simulation with Netgen/NgSolve

45553 tets, p = 5, 3×1074201 unknowns, 40 min on 3 GHz 64-bit PC 6 GB RAM

NGSolve script file for Poisson example:

$$\text{Find } u \in H^1 \text{ s.t.} \qquad \int_\Omega \lambda \nabla u \cdot \nabla v \, dx + \int_{\partial\Omega} \alpha u v \, ds = \int_\Omega f v \, dx \qquad \forall\, v \in H^1$$

```
define coefficient lam      1,
define coefficient alpha    1e5, 1e5, 1e5, 0,
define coefficient cf       sin(x)*y,

define fespace v -h1 -order=4
define gridfunction u -fespace=v

define bilinearform a -fespace=v -symmetric
laplace lam
robin alpha

define linearform f -fespace=v
source cf

define preconditioner c -type=multigrid -bilinearform=a -smoothingsteps=1 -smoother=block

numproc bvp np1 -bilinearform=a -linearform=f -gridfunction=u -preconditioner=c -prec=1e-8
```

Central NGSolve classes

- FiniteElement:
  Provides shape functions and derivatives on reference element
- ElementTransformation:
  Represents mapping to physical elements, computes Jacobian
- Integrator:
  Computes element matrices and vectors
- FESpace:
  Provides global dofs, multigrid-transfers and smoothing blocks
- BilinearForm/LinearForm:
  Maintains definition of forms, provides matrix and vectors
- PDE:
  Container to store all components

Element matrix computation

Element matrices for many problems are of the form

$$A_{ij}^T = \int_T DB(\varphi_i) \cdot B(\varphi_j)\, dx$$

with differential operator $B(u) = \nabla u$, $\mathrm{curl}\ u$, $\varepsilon(u)$, u, ... and coefficient tensor D.

Computed by matrix-matrix products

$$A^T = \sum_{Int.Point} \omega_k J(x_k) BDB^T$$

with B-matrix

$$B_{ij} = B_j(\varphi_i) \qquad i = 1 \ldots \text{dim element space}, \quad j = 1 \ldots \text{dim D-matrix}$$

Element matrix computation: Implementation

```
template <class DIFFOP, class DMATOP>
void BDB_Integrator::ComputeElementMatrix (FiniteElement & fel,
                                           ElementTransformation & eltrans,
                                           Matrix<double> & elmat)
{
  MatrixFixHeight<DIM_DMAT, double> bmat (fel.GetNDof()), dbmat(fel.GetNDof());
  Mat<DIM_DMAT,DIM_DMAT> dmat;

  const IntegrationRule & ir = GetIntegrationRule (fel);
  elmat = 0;
  for (int i = 0; i < ir.GetNIP(); i++)
    {
      IntegrationPoint<DIM_ELEMENT,DIM_SPACE> ip(ir, i, eltrans);

      DIFFOP::GenerateMatrix (fel, ip, bmat);
      dmatop.GenerateMatrix (ip, dmat);
      double fac = ip.GetJacobiDet() * ip.Weight();

      dbmat = fac * (dmat * bmat);
      elmat += Trans (bmat) * dbmat;
    }
}
```

Conclusions and Ongoing work

We have now

- high order finite elements for scalar and vectorial problems
- preconditioning: algebraic and geometric multigrid, p-version
- a posteriori error estimates

we are working on

- High frequency Maxwell solver
- Scattering on periodic structures (Lithography, Photonic Crystalls)
- Parallelization of Netgen/NgSolve

Open-source software Netgen/NgSolve available from

www.hpfem.jku.at