IETI-DP Solvers in Simulation and Optimization of Electrical Machines

Rainer Schneckenleitner

Institue of Computational Mathematics (NUMA)
Johannes Kepler University Linz (JKU)
Austria

July 4, 2019

FWF Der Wissenschaftsfonds.
GEOMETRY NFN S117 SIMULATION

References

P. Gangl.
Sensitivity-Based Topology and Shape Optimization with Application to Electrical Machines.
Dissertation, Johannes Kepler University Linz, 2016

P. Gangl, U. Langer, A. Laurain, H. Meftahi, and K. Sturm.
Shape optimization of an electric motor subject to nonlinear magnetostatics.
SIAM Journal on Scientific Computing, 37(6):B1002–B1025, 2015

C. Hofer and U. Langer.
Dual-primal isogeometric tearing and interconnecting solvers for multipatch dG-IgA equations.
Comput. Methods Appl. Mech. Engrg., 316:2–21, 2017

R. Schneckenleitner.
Isogeometrical Analysis based Shape Optimization.
Master's thesis, Johannes Kepler University Linz, December 2017

A. Wächter and L. T. Biegler.
On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming.
Mathematical Programming, 106(1):25–57, 2006

References

P. Gangl, U. Langer, A. Mantzaflaris, and R. Schneckenleitner.
Isogeometric simulation and shape optimization with applications to electrical machines.
Technical report, ArXiv, 2018

C. Hofreither and S. Takacs.
Robust multigrid for isogeometric analysis based on stable splitting of spline spaces.
SIAM Journal on Scientific Computing, 55(4):2004 – 2024, 2017

S. Takacs.
Robust approximation error estimates and multigrid solvers for isogeometric multi-patch discretizations.
Mathematical Models and Methods in Applied Sciences, 28(10):1899 – 1928, 2018

S. Takacs.
Fast multigrid solvers for conforming and non-conforming multi-patch isogeometric analysis.
arXiv: 1902.01818, 2019

S. Takacs.
A quasi-robust discretization error estimate for discontinuous galerkin isogeometric analysis.
arXiv: 1901.03263, 2019

Outline

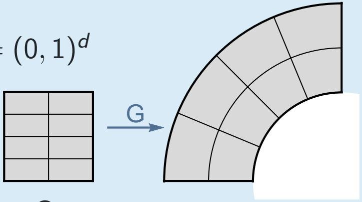
What is IgA?

- New paradigm for solving PDEs
- **Idea:** One and the same method that can be used for Computer Aided Design (CAD) and Numerical Simulation
- Exact representation of geometric objects from CAD systems without the need of meshing like in FEM

What is IgA?

- Based on splines
- Univariate splines $S_{p,k,h}(0,1)$
  - degree p
  - smoothness k
    $(S_{p,k,h}(0,1) = \{u \in C^k(0,1) : u|_{[ih,(i+1)h)} \in \mathbb{P}_p\})$
  - grid size h
- Tensor-product splines on $\hat{\Omega} := (0,1)^d$
- Global geometry function G:

  $\hat{\Omega} \to \Omega = G(\hat{\Omega})$



- IgA space in the physical domain Ω:
  $S_{p,p-1,h}(\Omega) = S_{p,p-1,h}(\hat{\Omega}) \circ G^{-1} = \{u : u \circ G \in S_{p,p-1,h}(\hat{\Omega})\}$

What is IgA?

- More complex geometries cannot be represented by one geometry map
- Use of multipatch domains:
  Per patch geometry functions G_k:
  $\overline{\Omega} = \bigcup_{k=1}^{K} \overline{G_k(\hat{\Omega})}$
- IgA space in the physical domain Ω:
  $S_{p,p-1,h}(\Omega) = \{u \in C^0(\Omega) : u \circ G_k \in S_{p,p-1,h}(\hat{\Omega}) \forall_{k=1,\dots,K}\}$

Why IgA for shape optimization?

- IgA has approximation power of a high-order method:

$$\inf_{u_h \in S_{p,p-1,h}} \|u - u_h\|_{L^2} \lesssim h^{p+1} |u|_{H^{p+1}}$$

- IgA has the problem size of a low-order method:

$$dim\ S_{p,p-1,h} \approx (n+p)^d$$

- Integration of geometry optimization in CAD environments
- No conversion of design-suitable and analysis-suitable models
  $\rightarrow$ Exact representation of the geometry
  $\rightarrow$ No systematic error

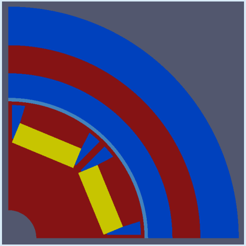
References to Shape Optimization with IgA

- D. Fußeder, B. Simeon, and A.-V. Vuong. Fundamental aspects of shape optimization in the context of isogeometric analysis.
  Computer Methods in Applied Mechanics and Engineering, 286, 04 2015
- J. Kiendl, R. Schmidt, R. Wüchner, and K.-U. Bletzinger. Isogeometric shape optimization of shells using semi-analytical sensitivity analysis and sensitivity weighting.
  Computer Methods in Applied Mechanics and Engineering, 274:148 – 167, 2014
- K. Kostas, M. Fyrillas, C. Politis, A. Ginnis, and P. Kaklis. Shape optimization of conductive-media interfaces using an IGA-BEM solver.
  Computer Methods in Applied Mechanics and Engineering, 340:600 – 614, 2018
- X. Qian. Full analytical sensitivities in NURBS based isogeometric shape optimization.
  Computer Methods in Applied Mechanics and Engineering, 199(29):2059 – 2071, 2010
- W. A. Wall, M. A. Frenzel, and C. Cyron. Isogeometric structural shape optimization.
  Computer Methods in Applied Mechanics and Engineering, 197(33):2976 – 2988, 2008
- D. Nguyen, J. Gravesen, and A. Evgrafov. *Isogeometric analysis and shape optimization in electromagnetism.*
  PhD thesis, 2012
- L. Blanchard, R. Duvigneau, A.-V. Vuong, and B. Simeon. Shape Gradient for Isogeometric Structural Design.
  Journal of Optimization Theory and Applications, 161(2):361–367, May 2014
- K. Solbakken. Isogeometric Analysis based Shape Optimization.
  Master's thesis, Norwegian University of Science and Technology, May 2015

Outline

Real IPM Motor and Computational Domain

(a) Interior permanent magnet (IPM) electric motor



(b) Quarter of a cross section of an IPM motor

Figure: Real world IPM motor[1] vs. computational domain

[1] We acknowledge the permission to use this photo taken by the Linz Center of Mechatronics (LCM). The motor was produced by Hanning Elektro-Werke GmbH & Co KG.

Optimization w.r.t. Linear Magnetostatics

- Optimization of the runout performance:

$$\min_{D} J(u) := \int_{\Gamma} |B(u) \cdot n_{\Gamma} - B_d|^2 \, ds = \int_{\Gamma} |\nabla u \cdot \tau_{\Gamma} - B_d|^2 \, ds$$

subject to: find $u \in H_0^1(\Omega)$ such that

$$\langle A_D u, v \rangle = \langle F, v \rangle \qquad \forall v \in H_0^1(\Omega) \tag{1}$$

with

$$\langle A_D u, v \rangle = \int_{\Omega} \nu_D(x) \nabla u \cdot \nabla v \, dx,$$
$$\langle F, v \rangle = \int_{\Omega} J_3 v + \nu_M M^{\perp} \cdot \nabla v \, dx.$$

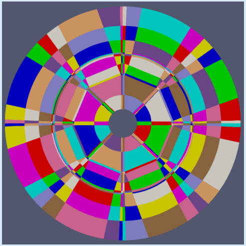
Outline

IETI-DP in a Nutshell

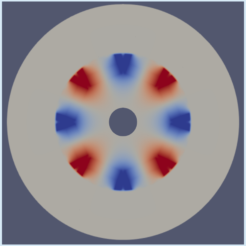
- Is a nonoverlapping domain decomposition method
- IETI-DP is an abbreviation for **D**ual-**P**rimal **I**sog**E**ometric **T**earing and **I**nterconnecting
- Similar to FETI-DP
- Basic idea:
  - Tearing of the computational domain into N subdomains
  - Posing the problem on a suitable interface space
  - DP approach requires for the solvability properly chosen primal variables, e.g., vertex evaluation, face averages, ...

Simulation with IETI-DP

- State solution in the conforming case with IETI-DP



(a) Full cross section of an IPM motor suitable for cG



(b) State on the full cross section of an IPM motor

- Multipatch domain with 372 patches and spline degree $p = 3$

Simulation with IETI-DP

- Now we can compare the solution techniques for the full cross section

# dofs	SuperLU	IETI-DP	speedup
23612	8.9 sec	5.7 sec	1.56
72 572	36.0 sec	17.0 sec	2.12
250 844	193.0 sec	69.8 sec	2.77
928 796	1943.0 sec	463.0 sec	4.20
3 570 332	–	1179.0 sec	–

Table: SuperLU vs. IETI-DP.

Simulation with IETI-DP

- Use of a parallel IETI-DP algorithm

# cores	1	2	4	8	16	32	64	128
time	1179	577	325	164	89	43	22	14
scaling	–	2.04	1.78	1.98	1.84	2.07	1.95	1.57

Table: Strong scaling with IETI-DP and 3 570 332 dofs, time in sec.

- We have an efficient PDE solver ✓

Optimization with Ipopt and IETI-DP

- Ipopt (**I**nterior **P**oint **Opt**imizer)
- Sign of the Jacobian determinant parametrization,

$$det(\mathrm{J}_G) = \sum_{k,\ell=1}^{M,N} c_{k,\ell} M_k^{2p-1}(u) N_\ell^{2q-1}(v)$$

  is used as constraint to prevent for self intersections
- Move the design variables
- Computation of the inner control coefficients from the boundary by a spring model of the mesh
  - An inner control coefficient $d_{i,j}$ satisfies

$$d_{i,j} = \frac{d_{i,j-1} + d_{i+1,j} + d_{i,j+1} + d_{i-1,j}}{4}$$

Optimization with Ipopt and IETI-DP

- Ipopt options:
  - Limited memory BFGS-method
  - NLP error tolerance: 10^{-6}
  - Relative error in the objective change: 10^{-6}
  - 3 acceptable iterations

Optimization with Ipopt and IETI-DP

- Optimization result with Ipopt:

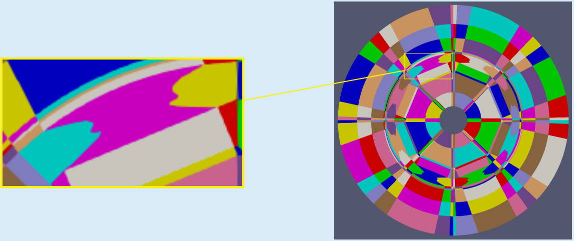


Figure: Optimized domain with Ipopt after 95 optimization iterations, the objective dropped from $4.266 \cdot 10^{-4}$ down to $2.587 \cdot 10^{-4}$

Optimization with Ipopt and IETI-DP

- Optimization result with Ipopt:

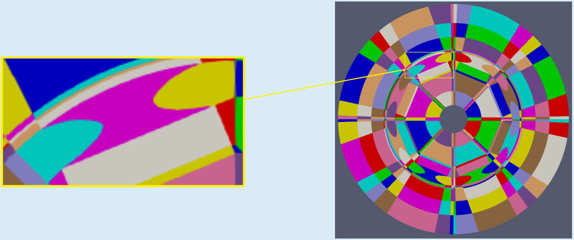


Figure: Optimized domain with Ipopt after 130 optimization iterations, the objective dropped from $4.266 \cdot 10^{-4}$ down to $2.436 \cdot 10^{-4}$

Outline

Preliminary Work to Capture the Rotation

- Hanging nodes have to be allowed
- We generalized the ideas in "Reparameterization and Adaptive Quadrature for the Isogeometric Discontinuous Galerkin method" by Seiler and Jüttler

The Reparameterization Technique

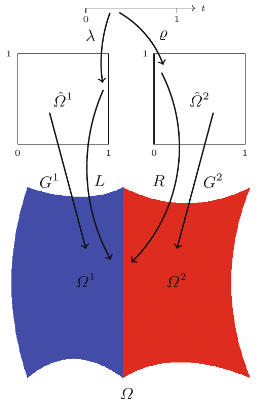


Figure: Multipatch domain Ω with geometry maps G^1, G^2

- Interface $e := G^1(1,[0,1]) = G^2(0,[0,1])$
- $L = G^1|_{(G^1)^{-1}(e)}$ and $R = G^2|_{(G^2)^{-1}(e)}$
- $\lambda : [0,1] \to \{1\} \times [0,1]$
- $\varrho : [0,1] \to \{0\} \times [0,1]$
- Relation $L \circ \lambda = R \circ \varrho$

The Reparameterization Technique

- Fix the reparameterization λ
- The reparameterization technique consists of 2 main steps:
  1. For a given number of samples, we compute

$$\varrho_i = R^{-1} \circ L \circ \lambda \left(\frac{i}{N} \right)$$

by solving

$$\varrho_i = \underset{\xi \in \{0\} \times [0,1]}{argmin} \left\| L \circ \lambda \left(\frac{i}{N} \right) - R(\xi) \right\|$$

for $i = 0, \ldots, N$

The Reparametrization Technique

- The reparametrization technique consists of 2 main steps:
  2. After choosing a suitable spline space we solve

$$\sum_{i=1}^{N}\left(\sum_{j=1}^{m} c_j N_j\left(\frac{i}{N}\right)-\varrho_i\right)^2 \rightarrow min!$$

The Reparametrization Technique

- Integration along the interface
  - Choose one master patch
  - Collect all breakpoints of the two patches
  - Compute preimages of breakpoints for the master patch
  - Put quadrature nodes between each of the breakpoints
  - Map the quadrature nodes via the reparameterization to the parameter domain of the second patch

Numerical Tests for the Reparameterization

- We consider the dG-IgA problem:

Find $u \in V_h : a(u,v) = F(v) \qquad \forall v \in V_h = \{v | v|_{\Omega^k} \in V_h^k\}$

with the bilinear form

$$a(u,v) = \sum_{k=1}^{M} a_1^k(u,v) - \frac{1}{2} \sum_{e \in \Gamma_c \cup \Gamma_D} (a_{2,1}^e(u,v) + a_{2,2}^e(u,v)) + \sum_{e \in \Gamma_c \cup \Gamma_D} a_3^e(u,v)$$

and the right-hand side

$$F(v) = \int_\Omega fv \; dx$$

Numerical Tests for the Reparameterization

- The quantities in the bilinear form are given by

$$a_1^k(u,v) = \int_{\Omega^k} \nabla u \cdot \nabla v \; dx$$

$$a_{2,1}^e(u,v) = \int_e \{\nabla u \cdot n\}^e [v]^e ds$$

$$a_{2,2}^e(u,v) = \int_e \{\nabla v \cdot n\}^e [u]^e ds$$

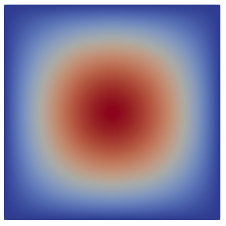
$$a_3^e(u,v) = \int_e \alpha [u]^e [v]^e ds$$

- $\{\cdot\}^e, [\cdot]^e$ denote the average and the jump across e, respectively and $\alpha \sim \frac{p^2}{h}$ is some suitably chosen parameter

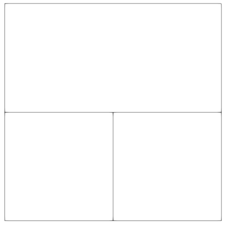
Problem Setting

- $f = 2\pi^2 sin(\pi x) sin(\pi y)$
- The exact solution of the problem is $sin(\pi x) sin(\pi y)$
- Homogeneous boundary conditions

Numerical Example 1



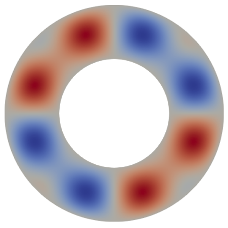
(a) Solution of the problem



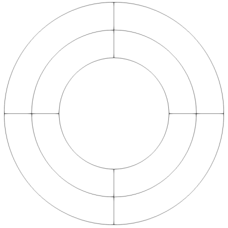
(b) Splitting of the domain

Figure: Solution with B-splines of degree 3

Numerical Example 2



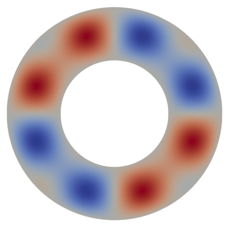
(a) Solution of the problem



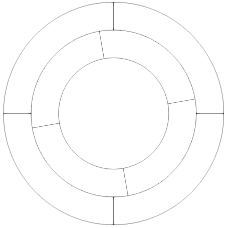
(b) Splitting of the domain

Figure: Solution with NURBS of degree 3

Numerical Example 2



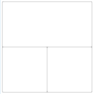
(a) Solution of the problem



(b) Splitting of the domain

Figure: Solution with NURBS of degree 3

Convergence Test of Example 1



# dofs	L^2 error	conv. rate	H^1 error	conv. rate
75	0.00108993	0	0.0544043	0
147	0.000160162	2.767	0.0113958	2.25522
363	8.21729e-06	4.285	0.000902003	3.65922
1083	4.82416e-07	4.090	8.37942e-05	3.42821
3675	2.9596e-08	4.029	8.53316e-06	3.2957
13467	1.83882e-09	4.009	9.30043e-07	3.19771
51483	1.14684e-10	4.003	1.06861e-07	3.12156
201243	7.17393e-12	3.999	1.27334e-08	3.06904

Table: L^2 and H^1 convergence rates for example 1 with B-splines of degree **3**

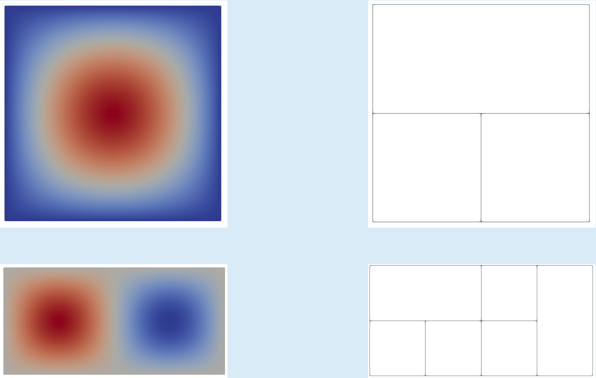
How to Handle Rotations of Elec. Machines?

- dG-IETI method
  - Would be completely new
  - No cG conforming meshes for the subdomains necessary
  - No theory so far (theory relies on matching vertices)
- Quasi p-robust multipatch multigrid
  - Developed mainly by Stefan Takacs at RICAM
  - Theory not complete
  - First experiments are promising
- ...
- ...

dG-IETI-DP

- Theory so far needs vertex values as primal variables
- Hanging nodes occur in the consideration of rotating machines
  $\rightarrow$ Vertex values are not feasible any more
  $\rightarrow$ Only edge averages are allowed

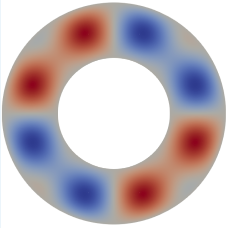
dG-IETI-DP



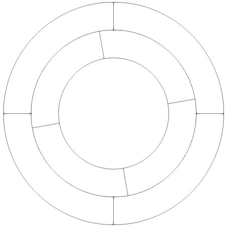
(a) Solution of the problem (b) Splitting of the domain

Figure: Solution with B-splines of degree 3

dG-IETI-DP



(a) Solution of the problem

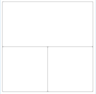


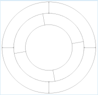
(b) Splitting of the domain

Figure: Solution with NURBS of degree 3

Tests with dG-IETI-DP

- Unit square with degree 3, NURBS circular rings with degree 3





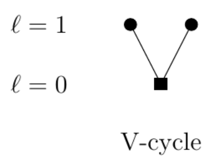
Unit square			NURBS circular rings		
# dofs	Solving time	CG-Iterations	# dofs	Solving time	CG-Iterations
363	0.21	4	392	0.97	23
1 083	0.37	5	968	1.43	32
3 675	0.89	6	2 888	2.50	33
13 467	2.95	7	9 800	5.53	31
51 483	13.84	8	35 912	16.93	34
201 243	82.19	9	137 288	71.48	33

Table: Solving time in sec and iterations for dG-IETI-DP, Error tolerance 1e-5

Institute of NuMa Computational Mathematics JKU

Multigrid in a Nutshell

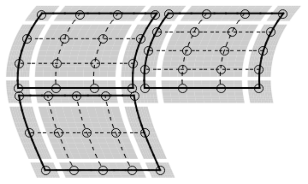
- Iterative method for solving large linear systems $Ax = f$
- Use a hierarchy of discretizations
- Basic algorithm consists of 3 steps:
  - Smoothing: $x^k \leftarrow x^k + P^{-1}(f - Ax^k)$
  - Restriction of the residual r
  - Prolongation of the correction $w = A^{-1}r$ and setting $x^{k+1} = x^k + w$



- Can be used as solver as well as preconditioner for e.g. CG

Quasi p-robust Multigrid

- Key idea is a stable splitting of the spline spaces



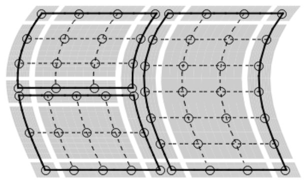


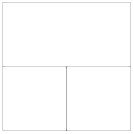
Figure: Decomposition of dofs

Quasi p-robust Multigrid

(a) Solution of the problem (b) Splitting of the domain

Tests with Multigrid

- Test on the unit square



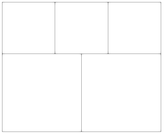
Unit square

r\p	1	2	3	4	5
2	7/44	6/70	7/102	8/140	10/184
3	8/184	7/234	8/290	9/352	12/420
4	8/752	8/850	9/954	11/1064	15/1180
5	8/3040	9/3234	10/3434	11/3640	18/3852
6	8/12224	9/12610	10/13002	12/13400	20/13804
7	9/49024	9/49794	11/50570	12/51352	20/52140
8	9/196352	10/137890	11/199434	12/200984	20/202540

Table: Iterations/#dofs, Error tolerance 1e-8

Tests with Multigrid

- Test on the rectangular domain



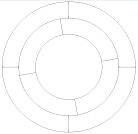
Rectangular domain

r\p	1	2	3	4	5
2	8/84	6/130	5/186	6/252	6/328
3	8/328	6/414	7/510	7/616	7/732
4	8/1296	6/1462	7/1638	8/1824	9/2020
5	8/5152	7/5478	8/5814	9/6160	10/6516
6	9/20544	7/21190	8/21846	9/22512	11/23188
7	9/82048	7/83334	8/84630	9/85936	11/87252
8	9/327936	7/330502	8/333078	10/335664	11/338260

Table: Iterations/#dofs, Error tolerance 1e-8

Tests with Multigrid

- Test on the circular rings



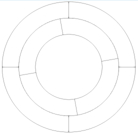
Circular rings

r\p	1	2	3	4	5
2	13/160	14/240	15/336	16/448	17/576
3	19/576	19/720	22/880	22/1056	30/1248
4	19/2176	21/2448	23/2736	26/3040	31/3360
5	20/8448	21/8976	26/9520	29/10080	36/10656
6	23/33280	23/34320	26/35376	30/36448	37/37536
7	23/132096	23/134160	27/136240	30/138336	37/140448
8	23/526336	23/530448	27/534576	30/538720	38/542880

Table: Iterations/#dofs, Error tolerance 1e-8

Tests with Multigrid

- Test on the circular rings with improved scaling



Circular rings

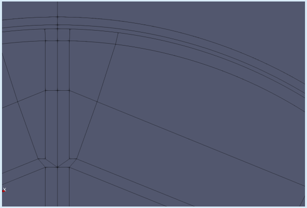
r\p	1	2	3	4	5
2	10/160	11/240	14/336	14/448	12/576
3	15/576	15/720	17/880	18/1056	27/1248
4	17/2176	16/2448	19/2736	20/3040	27/3360
5	20/8448	17/8976	20/9520	22/10080	29/10656
6	21/33280	18/34320	21/35376	22/36448	32/37536
7	22/132096	19/134160	22/136240	23/138336	32/140448
8	23/526336	19/530448	22/534576	23/538720	36/542880

Table: Iterations/#dofs, Error tolerance 1e-8

Outline

dG-IETI on a NURBS based geometry

- Solution on a NURBS based model of the IPM
- T-junctions but no rotation
- Same solution as in the conforming case



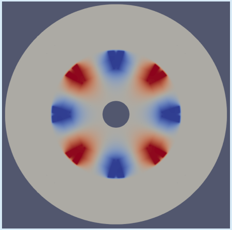


Figure: Discretization and solution of linear magnetostatics

dG-IETI on a NURBS based geometry

- We are now able to consider an arbitrary position of the motor:

Decompositon Solution

Outline

Summary

- So far only the linear static case is considered
- Need additional techniques to capture rotation
- Introduction of dG-IETI-DP and multipatch multigrid
- First results are promising
- dG-IETI-DP and multigrid still require some investigation
- Also reparametrization seems to influence the results

Outlook

- Provide a theory for dG-IETI-DP
- Provide a theory for multipatch multigrid
- Introduction of the nonlinearity in ν, i.e., $\nu = \nu(x, |\nabla u|)$
- Apply multipatch multigrid to the motorsimultation
- Proper treatment of highly distorted geometries