ECMI Modelling Week, July 17–24, 2016, Sofia, Bulgaria

Group 3

Design Optimization of an Electric Motor

Robert Bauer

Technische Universität Dresden, Germany
bauerrobert45@yahoo.com

Pedro Rodriguez-Barbeito

University of Santiago de Compostela, Spain
prodbarbeito.93@gmail.com

Anne Ryelund Nielsen

Technical University of Denmark, Denmark
AnneRyelund@hotmail.com

Szymon Sobieszek

Wroclaw University of Science and Technology, Poland
208614@student.pwr.edu.pl

Lea Miko Versbach

Lund University, Sweden
mat14lve@student.lu.se

Instructor: **Peter Gangl**

Johannes Kepler University Linz, Austria
gangl@numa.uni-linz.ac.at

Abstract. The optimization of electrical machines is of utter importance in industry. In this project, we consider the design optimization of an electric motor consisting of different components. The efficiency of the motor heavily depends on its geometry. For a fixed geometry, the arising magnetic field can be computed as the solution of a partial differential equation (PDE). The goal of the project is to determine the shapes and sizes of specific parts of the electric motor such that the corresponding magnetic field yields an optimal behavior with respect to some prescribed criteria (e.g., transmitted power, reduction of noise and vibration).

3.1 Introduction

In industry, the optimization of electrical machines is of great importance. In this project, we consider the design optimization of an electric motor which consists of the following components:

- Stator, consisting of ferromagnetic material (outer part),
- Rotor, consisting of ferromagnetic material (inner part),
- Air regions,
- Coils (in the stator),
- Permanent magnets possible.

Figure 3.1 shows an example of a real world motor with nine coils and six magnets.

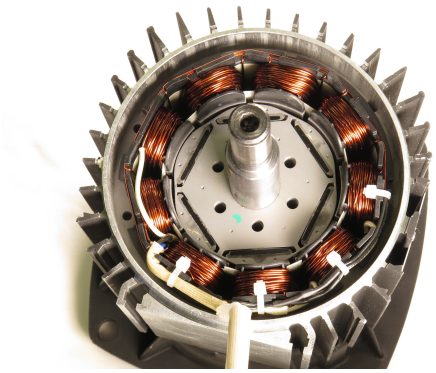
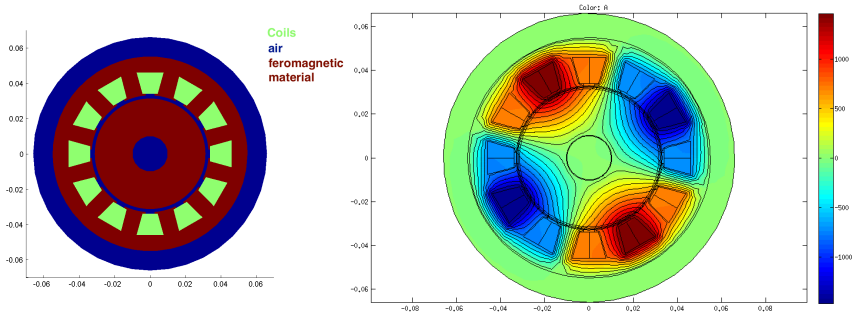


Figure 3.1: Real world motor

We consider a so-called synchronous reluctance motor with nine coils and no magnets. If electric current is induced in the coils, a magnetic field is produced in the entire motor, visualized in Figure 3.2.



(a) Model of an electric motor (b) Magnetostatic potential

Figure 3.2: Model of an electric motor

This magnetic field depends among other things on the geometry of the motor. Thus, the efficiency of the motor depends on its geometry. For a fixed geometry, the arising magnetic field can be computed as the solution of a partial differential equation (PDE).
The goal of the project is to determine a design of the rotor (see Figure 3.3) such that the corresponding magnetostatic potential maximizes the rotational force of the motor.

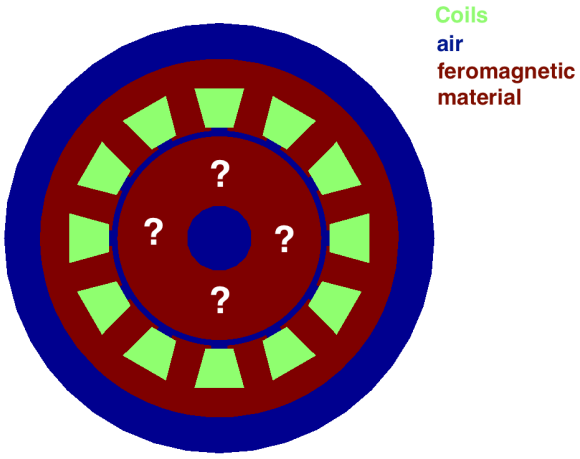


Figure 3.3: The goal is to optimize the design of the rotor.

3.2 Mathematical Model

Derivation of the Magnetostatic PDE

In this section we will derive the mathematical model which we will use to optimize the shape of our electric motor, the so-called *2D magnetostatic* model. The derivation is based on [5] and [4]. The model is based on Maxwell's equations, established by James Clark Maxwell in 1862. In differential form, they are given

by:

$$\begin{aligned}\nabla \times H &= \bar{J} + \frac{\partial D}{\partial t},\\ \nabla \times E &= -\frac{\partial B}{\partial t},\\ \nabla \cdot B &= 0,\\ \nabla \cdot D &= \rho.\end{aligned}$$

Here, for differentiable vector fields $F : \Omega \subset \mathbb{R}^3 \to \mathbb{R}^3$ and a spatial variable $x = (x_1, x_2, x_3) \in \Omega$, we used the following operators in cartesian coordinates

$$\nabla \times F := \begin{pmatrix} \partial F_3/\partial x_2 - \partial F_2/\partial x_3 \\ \partial F_1/\partial x_3 - \partial F_3/\partial x_1 \\ \partial F_2/\partial x_1 - \partial F_1/\partial x_2 \end{pmatrix},$$

which is called the *curl* or *rotation* of F and is also a vector field and

$$\nabla \cdot F := \mathrm{div} F := \frac{\partial F_1}{\partial x_1} + \frac{\partial F_2}{\partial x_2} + \frac{\partial F_3}{\partial x_3},$$

the *divergence* of F, which is a scalar field. The physical quantities involved are

- H ... *magnetic field* or *magnetic excitation* $[A/m]$,
- B ...*magnetic induction* or *magnetic flux density* $[Vs/m^2]$,
- E ... *electric field* $[V/m]$,
- D ... *electric flux density* or *electric induction* $[As/m^2]$,
- $\bar{J}$ ... *(surface) current density* $[As/m^2]$ and
- ρ ... *(volume) charge density* $[As/m^3]$.

All these functions are time dependent to the time $t \in \mathbb{R}$ and of course, H, B, E, D and J are vector fields and ρ is a scalar field. For uniqueness of our system of equations we assume the following material laws:

$$\begin{aligned}B &= \mu H,\\ D &= \varepsilon E,\\ \bar{J} &= \sigma E + J_i,\end{aligned}$$

involving

- μ ... *magnetic permeability* $[Vs/Am]$
- ε ... *electric permittivity* $[As/Vm]$
- σ ... *electric conductivity* $[A/Vm]$

- J_i ... *impressed current density* $[A/m^2]$.

In general, μ and ϵ are non-linear rank 2 tensors, depending on space, time and the fields E and H. But in our problem, these quantities are scalar and we consider isotropic, materials without hysteresis and assume

$$B = \mu(x, |H|)H.$$

Now we can consider the *magnetic reluctivity* ν defined by,

$$H = \nu(x, |B|)B.$$

In our case, we consider only non-conducting regions, i.e. $\sigma = 0$ and we further assume that all fields are time independent, i.e. $\frac{\partial}{\partial t} = 0$. The equations for the magnetic and electric fields are therefore decoupled into two independent systems and we are just looking at the simplified magnetostatic equations:

$$\begin{aligned} \nabla \times H &= J_i \\ \nabla \cdot B &= 0 \\ H &= \nu(x, |B|)B. \end{aligned}$$

Because of the divergence free magnetic flux density B, there exists a vector potential A, depending on a space variable x, such that $B = \nabla \times A$. Now we got

$$\begin{aligned} J_i &= \nabla \times H \\ &= \nabla \times [\nu(x, |B|)B] \\ &= \nabla \times [\nu(x, |\nabla \times A|)\nabla \times A]. \end{aligned}$$

In our project we deal with the magnetostatic equation only in an open and bounded 2D domain $\Omega \subset \mathbb{R}^2$, which describe the geometry of the electric motor. In addition the ferromagnetic material regions Ω^{ferro} and air regions Ω^{air} (also including the coils) are disjoint, i.e $\bar{\Omega} = \bar{\Omega}^{\text{ferro}} \cup \bar{\Omega}^{\text{air}}$, $\Omega^{\text{ferro}} \cap \Omega^{\text{air}} = \emptyset$ and we say that ν only depends on the kind of material:

$$\nu(x, |\nabla \times A|) = \begin{cases} \nu_0 & \text{if } x \in \Omega^{\text{air}} \\ \bar{\nu}(|\nabla \times A|) & \text{if } x \in \Omega^{\text{ferro}}. \end{cases}$$

Furthermore we say that the current density J_i is perpendicular to the magnetic field H, and so for $x = (x_1, x_2)$ we have

$$J_i = \begin{pmatrix} 0 \\ 0 \\ J_{i_3}(x_1, x_2) \end{pmatrix}, \; H = \begin{pmatrix} H_1(x_1, x_2) \\ H_2(x_1, x_2) \\ 0 \end{pmatrix}.$$

From the B-H-relation we also get

$$B = \begin{pmatrix} B_1(x_1, x_2) \\ B_2(x_1, x_2) \\ 0 \end{pmatrix}.$$

With the identity $\nabla \times A = B$ we get

$$0 = B_3 = (\nabla \times A)_3 = \partial_2 A_1 - \partial_1 A_2,$$

and to fulfill this we use the approach

$$A = \begin{pmatrix} 0 \\ 0 \\ A_3(x_1, x_2) \end{pmatrix}.$$

Now we use the notation $u := A_3$, $J := J_{i_3}$ and find

$$|\nabla \times A| = | \begin{pmatrix} \partial_2 A_3 \\ -\partial_1 A_3 \\ 0 \end{pmatrix} | = |\nabla u|.$$

Now we can simplify our magnetostatic equation as follows:

$$\begin{aligned} J_i = \begin{pmatrix} 0 \\ 0 \\ J_{i_3}(x_1, x_2) \end{pmatrix} &= \nabla \times [\nu(x, |\nabla \times A|) \nabla \times A] \\ &= \nabla \times \begin{pmatrix} \nu(x, |\nabla u|)\partial_2 u \\ -\nu(x, |\nabla u|)\partial_1 u \\ 0 \end{pmatrix} \\ &= \begin{pmatrix} 0 \\ 0 \\ \partial_1[-\nu(x, |\nabla u|)\partial_1 u] - \partial_2[\nu(x, |\nabla u|)\partial_2 u] \end{pmatrix} \\ &= \begin{pmatrix} 0 \\ 0 \\ -\nabla \cdot [\nu(x, |\nabla u|)\nabla u] \end{pmatrix}, \end{aligned}$$

and get that

$$-\mathrm{div}[\nu(x, |\nabla u|)\nabla u] = J. \tag{3.1}$$

In our model, the domain Ω is a circle where the electric motor Ω_{motor} is contained, i.e $\Omega_{\mathrm{motor}} \subset \Omega$ and

$$\mathrm{dist}(\Omega_{\mathrm{motor}}, \partial\Omega) > 0.$$

It can be shown that the condition

$$u = 0 \text{ on } \partial\Omega \tag{3.2}$$

implies this other boundary condition

$$B \cdot n = 0 \text{ on } \partial\Omega, \tag{3.3}$$

which is the actual boundary condition that we want to impose. Combining this statement with (3.1) and the continuity of u and the fluxes $\nu \frac{\partial u}{\partial n}$ across the

material interfaces $\overline{\Omega}^{ferro}$, we can formulate the mathematical problem in the classical sense:

Find $u \in C^2(\Omega^{\text{air}} \cup \Omega^{\text{ferro}}) \cap C(\bar{\Omega})$ such that

$$
\begin{aligned}
-\text{div } [\nu(x, |\nabla u|)\, \nabla u(x_1, x_2)] = J \quad &\text{in} \quad \Omega, && \text{(3.4a)}\\
u = 0 \quad &\text{on} \quad \partial\Omega, && \text{(3.4b)}\\
[\![u]\!] = 0 \quad &\text{on} \quad \overline{\Omega}^{\text{air}} \cap \overline{\Omega}^{\text{ferro}} && \text{(3.4c)}\\
[\![\nu(x, \nabla u)\frac{\partial u}{\partial n}]\!] = 0 \quad &\text{on} \quad \overline{\Omega}^{\text{air}} \cap \overline{\Omega}^{\text{ferro}} && \text{(3.4d)}
\end{aligned}
$$

where we denote by $[\![v]\!]$ the jump of a function v across the interface $\overline{\Omega}^{\text{air}} \cap \overline{\Omega}^{\text{ferro}}$.

Weak Formulation of the Magnetostatic PDE

To give a weak formulation of the problem in (3.4) we start multiplying equation (3.4a) time a test function, $v \in H_0^1(\Omega)$, and integrating it over the domain, which yields:

$$\int_\Omega -\text{div } [\nu(x, |\nabla u|)\, \nabla u(x_1, x_2)]\, v \, d\Omega = \int_\Omega Jv \, d\Omega \tag{3.5}$$

We can use use Green's identity, which stands that for any two given fields, $\boldsymbol{\psi}, \phi \in C^2(\Omega)$ the following equality, where $\mathbf{n}$ is the outer normal vector for Ω, holds

$$\int \phi \, \text{div}\, \boldsymbol{\psi} \, d\Omega + \int \boldsymbol{\psi} \cdot \nabla\phi \, d\Omega = \int_{\partial\Omega} (\mathbf{n} \cdot \boldsymbol{\psi})\, \phi \, d\Gamma. \tag{3.6}$$

Using it in (3.5) for $\boldsymbol{\psi} = \nu(x, |\nabla u|)\, \nabla u(x_1, x_2)$ and $\phi = v$, we get the following:

$$\int_\Omega \nu\, \nabla u(x_1, x_2) \nabla v(x_1, x_2)\, d\Omega = \int_\Omega Jv(x_1, x_2)\, d\Omega + \int_{\partial\Omega} \mathbf{n} \cdot \nu\, \nabla u(x_1, x_2) v(x_1, x_2)\, d\Gamma. \tag{3.7}$$

Then we can write a weak formulation for problem (3.4), which reads:

Find $u \in H^1(\Omega)$ with u = 0 over $\partial\Omega$ such that

$$\int_\Omega \nu(x, |\nabla u|)\, \nabla u(x_1, x_2) \nabla v(x_1, x_2)\, d\Omega = \int_\Omega Jv(x_1, x_2)\, d\Omega, \tag{3.8}$$

$\forall\, v \in H_0^1(\Omega)$.

The weak formulation of the equation is important since it is the starting point to apply the *Finite Element Method*, which is the method used by the Matlab's solver to solve these kinds of PDEs.

Calculation of the Torque

We have reached a way to calculate the magnetic field generated by our electric motor, nevertheless, the aim of the project is to optimize the force generated by

this magnetic field so we should be able to calculate this force.

This force we are talking about is called Torque and in our case, It's defined as follows:

$$\mathbf{T} = \frac{1}{\mu_0} \int_{\Gamma_0} \nabla u^T Q(x) \nabla u \, d\Gamma_0, \tag{3.9}$$

being

$$Q(x) = \frac{1}{\sqrt{x_1^2 + x_2^2}} \begin{pmatrix} x_1 x_2 & \frac{x_2^2 - x_1^2}{2} \\ \frac{x_2^2 - x_1^2}{2} & -x_1 x_2 \end{pmatrix} \tag{3.10}$$

and Γ_0 a circle standing in the middle of the air gap between the rotor and the stator.

This formula can be deduced from the general torque formula which reads

$$\mathbf{T} = \mathbf{F} \times \vec{OP}$$

when appling a force $\mathbf{F}$ on a given point P from a fixed one O. And using the Maxwell stress tensor. More information about this deduction can be consulted in [3].

3.3 Forward Problem

In this section we take a look at the forward problem, where we want to compute the torque for a given geometry of the rotor. To create this geometry we have limited ourselves to setting up just a quadrilateral in a first quarter of the rotor. We reflect the created shapes to the other quarters to obtain complete geometry. In order to setup the geometry we have created a Matlab function with the input vector $\vec{x}$ consisting of a three shape components x, y, z. The variables x, y, z represents the normalized distances seen in Figure 3.4. With this setup we can only adjust the shape of our quadrilaterals by changing these three variables and we have reduced the forward problem to a system with three degrees of freedom. (The three triangular domains seen in bottom left corner of Figure 3.4 are some extra domains we added to make it easier to set up a different geometry later.)

After creating the geometry we set up different parameters for each part of a motor (for example air and ferromagnetic material). Last thing in this step is to impose some boundary conditions (the potential u equal to zero on the outer boundary). An example of a final geometry is seen in Figure 3.5.

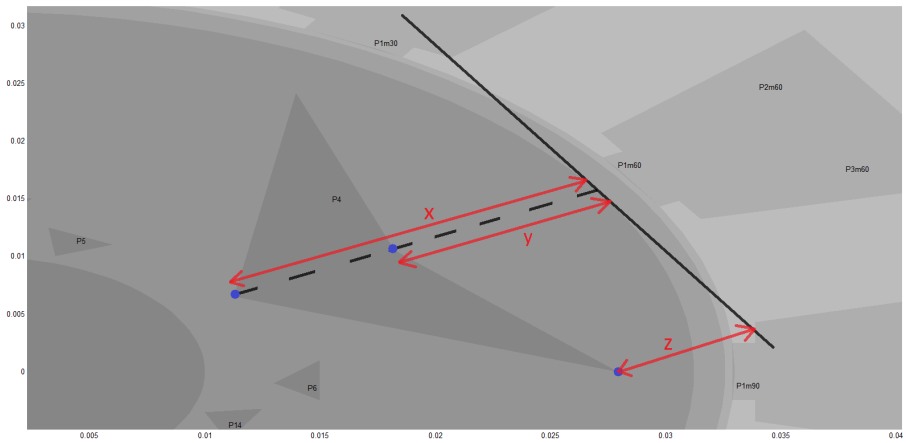


Figure 3.4: The figure shows how the variables x, y and z are defined as distances.

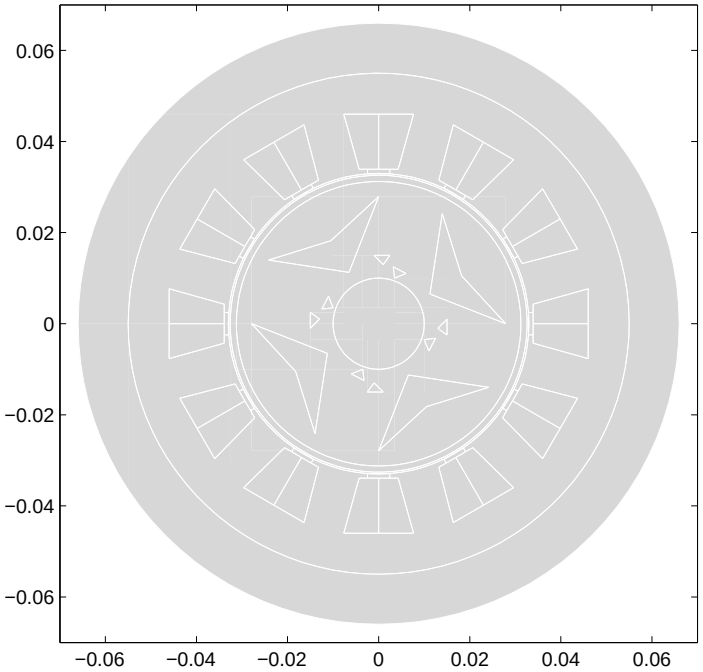


Figure 3.5: An example of a geometry.

The second step in the forward problem is to create a mesh which consists of the small triangles needed for the Finite Element method. The mesh itself is created with the help of the Matlab PDE toolbox. After that we can solve numerically our equation using nonlinear Matlab solver (which is based on the Newton's method). The mesh and the solution are seen in Figure 3.6 and Figure 3.7.

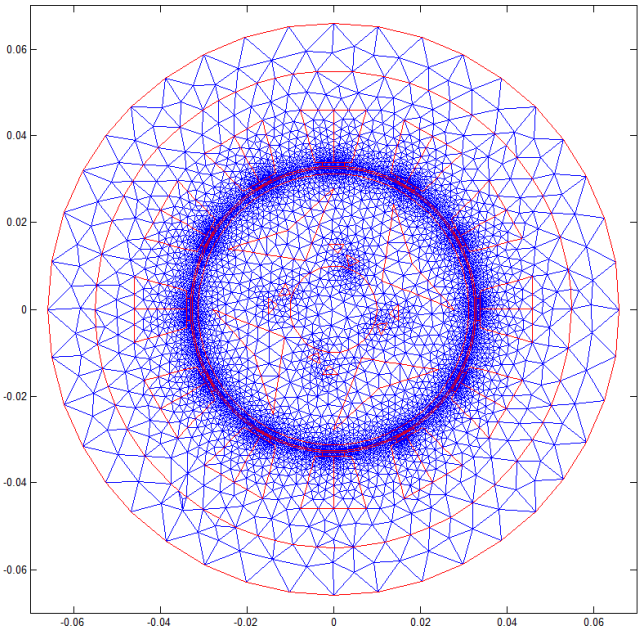


Figure 3.6: Mesh required for the Finite Element method.

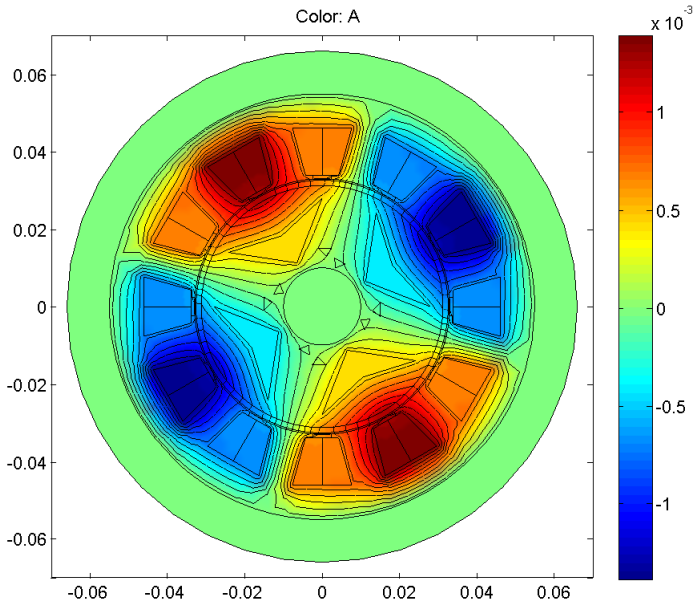


Figure 3.7: Numerical solution to the PDE on the geometry given in the example.

Finally, having all the inputs we can calculate the gradient of the potential which we need to compute the torque defined by (3.9). After having described how to

solve the forward problem we are ready to look at the following inverse problem: what is the geometry that gives the highest torque?

3.4 Optimization Problem

In this section we describe how the inverse problem can be seen as an optimization problem. In the previous section we described the forward problem: given an input vector $\vec{x}$ we obtain a certain Torque depending on the shape of Ω_{air}. Let us write this dependence with the following notation $\Omega_{air}^{\vec{x}}$. Thus, the torque generated by this configuration would be $T(u(\Omega_{air}^{\vec{x}}))$, where $u(\Omega_{air}^{\vec{x}})$ is the solution for the 2D magnetostatic model PDE we exposed in section 2.

Since all this is set up our aim now is to calculate the vector $\vec{x}$ that yields the maximum torque. This can be written mathematically as an optimization problem:

Find $\vec{p} \in \mathbb{R}^N$ such that

$$T(u(\Omega_{air}^{\vec{p}})) = \max_{\Omega_{air}^{\vec{x}}} \{T(u(\Omega_{air}^{\vec{x}}))\}, \tag{3.11a}$$

$$\text{subject to} - \operatorname{div}\left[\nu(\boldsymbol{x}, |\nabla\, u(\Omega_{air}^{\vec{x}})|)\, \nabla u(\Omega_{air}^{\vec{x}})\right] = J_z. \tag{3.11b}$$

For timing reasons we restricted to the case where $N = 3$, but the formulation of the problem still the same.

In order to solve this problem we encounter several issues. The first one is uniqueness of the solution. We cannot expect to find a unique solution since we don't know whether T is globally convex or not. For any given initial values we cannot be sure whether the solution is a local maximum or the global one. The second issue is that each evaluation of the torque, for a given $\Omega_{air}^{\vec{x}}$ domain, would take a great amount of time to compute (up to several minutes). this is why methods needing a great amount of evaluations to run are not suitable for this problem.

This first issue we were talking about was inevitable. For the second one, we considered to use the adjoint state method, described in [2]. Nevertheless, we had some problems with its implementation due to the nonlinearity of the problem. Finally, we decided to apply a gradient method to the problem. In the implementation we took into account that we have to limit the number of evaluations on each iteration step.

3.5 Optimization Algorithm

The optimization aims to maximize the torque T of the motor. The torque is influenced by the shape of domains with air in the ferromagnetic material. Thus,

the optimization is done by modifying the shape of an initial polygon with air in order to find the maximal torque. In Figure 3.8, this scheme is visualized.

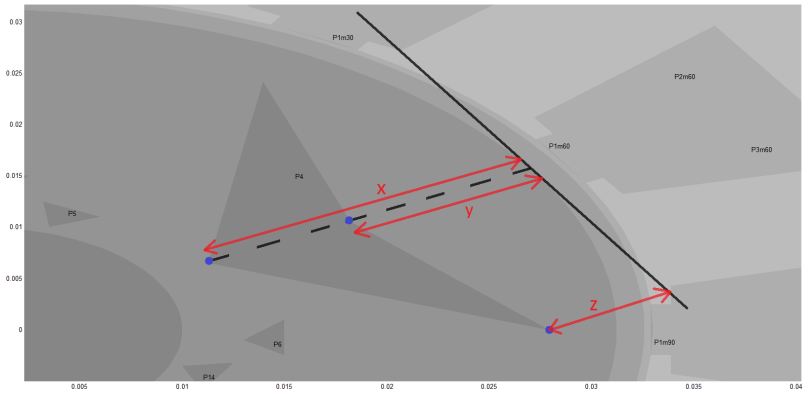


Figure 3.8: Idea of the optimization algorithm: move the corners along the red arrows

The Steepest Descent Method

The optimization algorithm is based on the steepest descent method, described in [1]. This is a first order optimization algorithm to find the local minimum of a continuously differentiable function f. Since the gradient vector $\nabla f(x)$ in a given point x gives the direction in x where f has the largest growth, $-\nabla f(x)$ indicates the direction with the maximal descent and is called the direction of steepest descent. Thus, this direction is a good choice as a search direction in a minimization problem. Based on this idea, the steepest descent method is an iterative method of the following scheme:
Choose an initial guess x_0. Compute iteratively x_n, $n \geq 1$, by

$$x_{n+1} = x_n - \alpha \nabla f(x_n),$$

for a constant $\alpha > 0$ small enough.
This gives $f(x_0) \geq f(x_1) \geq f(x_2) \geq f(x_3) \geq \ldots$, and the sequence $\{x_n\}_{n \geq 0}$ should converge to the local minimum. The convergence can be guaranteed by having more assumptions on the function f and particular choices of α.

Description of the Algorithm

In our optimization problem we want to maximize the torque T, this can be done by using the described steepest descent method to minimize $-T$. In each iteration we need to compute the gradient of T, but since the torque is not described by an analytical expression it is only possible to approximate the gradient. In order to limit the number of torque computations the gradient is approximated by its first order finite difference approximation. In each iteration a new point is calculated and it is controlled if this new point is inside the valid domain, i.e. inside the ferromagnetic material. If the point is invalid, a

new iteration point is calculated with decreased step size α. If a valid iteration point is found, the function values at both points are compared and only a decreasing function value for each iteration step is accepted. Finally, the gradient of the torque in the new iteration point is approximated. The algorithm stops, if the distance between two successive iteration points is smaller than a given tolerance (0.01) and we say that the algorithm converged. The algorithm does not converge, if the maximal number of iterations (20) is exceeded. A schematic description of the algorithm is listed below.

- given starting configuration x_0, tolerance tol and step size α
- calculate $T(x_0)$, approximate $\nabla T(x_0)$
- while $i <$ maxiter
  - $i = i + 1$
  - $x_i = x_{i-1} + \alpha \nabla T(x_{i-1})$
  - while x_i is invalid
    - $\alpha = \frac{\alpha}{2}$
    - compute $x_i = x_{i-1} + \alpha \nabla T(x_{i-1})$
  - calculate $T(x_i)$
  - while $T(x_i) < T(x_{i-1})$
    - $\alpha = \frac{\alpha}{2}$
    - compute $x_i = x_{i-1} + \alpha \nabla T(x_{i-1})$
  - approximate $\nabla T(x_i)$
  - break if $\|x_i - x_{i-1}\| <$ tol

3.6 Numerical Results, Interpretation and Conclusion

In this section we will present the numerical results obtained by applying the described optimization routine. We conducted several numerical experiments, common to all of them was the fact that the z-value was fixed. This was done to further reduce the degrees of freedom in our problem and thereby reducing the computational time. Even with this simplification of the problem, each numerical experiment lasted up to 30 minutes. From the numerical experiments with different initial configurations, we saw some tendencies: often the algorithm converges after a couple of iterations to a local optimum, which is very close to the initial conditions. But for some initial configurations the optimization algorithm manages to find a local optimum further away from the initial conditions. Some of these results are listed in Table 3.1.
For all the experiments listed in Table 3.1 we see that the torque is increased approximately 10% from the initial to the optimized configuration. In order to compare the optimized configurations we have plotted the numerical solutions to the PDE on the initial- and optimized geometry for each of the three

Table 3.1: Selected numerical results

	Initial configuration	**Optimized configuration**	**No. iterations**
Experiment 1	$x_0 = 0.2$ $y_0 = 0.1$ $T_0 = 0.002784$	$x_{opt} = 0.7212$ $y_{opt} = 0.6212$ $T_{opt} = 0.003011$	6 iterations
Experiment 2	$x_0 = 0.6$ $y_0 = 0.2$ $T_0 = 0.002749$	$x_{opt} = 0.6093$ $y_{opt} = 0.4670$ $T_{opt} = 0.003013$	4 iterations
Experiment 3	$x_0 = 0.8$ $y_0 = 0.2$ $T_0 = 0.002761$	$x_{opt} = 0.5930$ $y_{opt} = 0.5590$ $T_{opt} = 0.003021$	4 iterations

experiments. These plots are seen in Figure 3.9, Figure 3.10 and Figure 3.11, respectively.
Based on these selected results we observe that the quadrilaterals change to a similar shape in the optimized configuration. This shape is a thin "curve" facing outwards. Without having the necessary physical background, we cautiously conclude that the rotational force of the motor is increased when the air gaps in the rotor follows the equipotential lines of the magnetostatic potential.
This is the result of our small numerical study. In order to draw larger conclusions, it is necessary to conduct more extensive studies. In the following section you will find a discussion about how to structure potential future work.

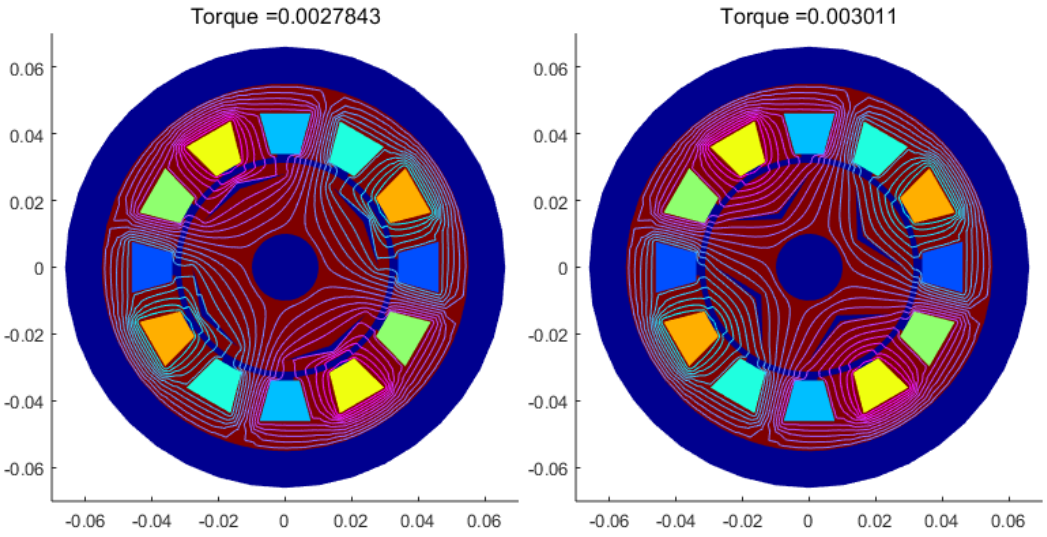


Figure 3.9: Numerical experiment 1. Left: Starting configuration. Right: Optimized configuration.

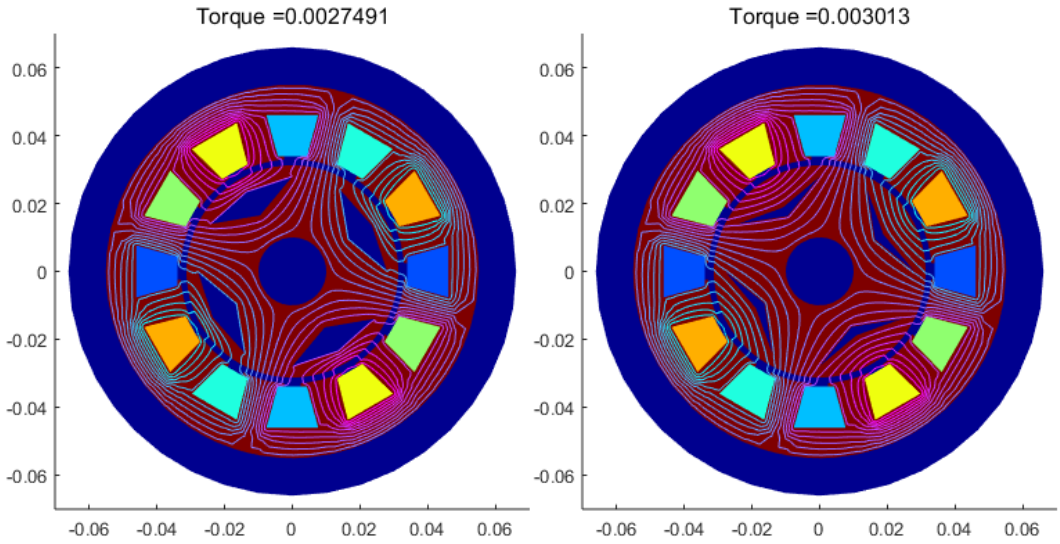


Figure 3.10: Numerical experiment 2. Left: Starting configuration. Right: Optimized configuration.

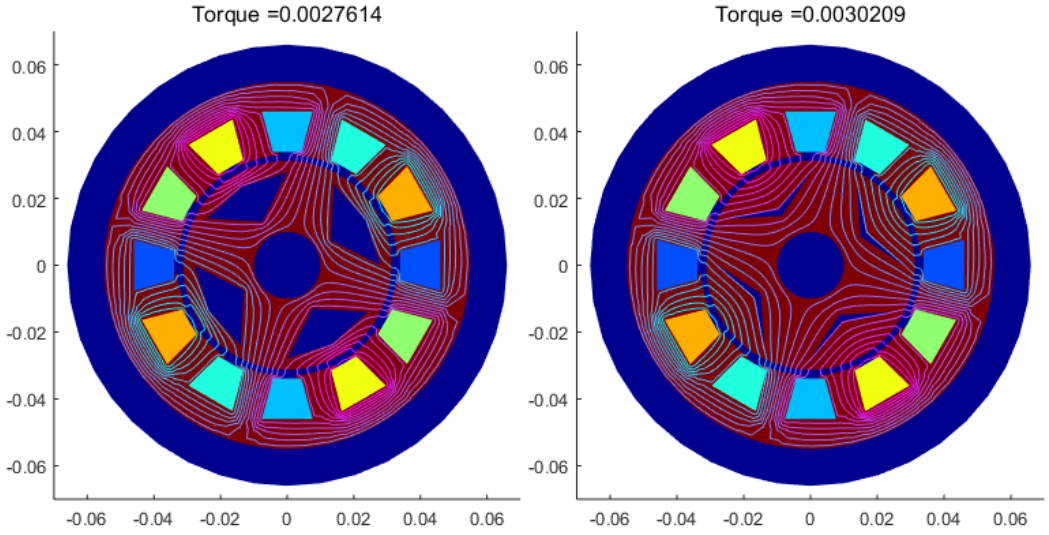


Figure 3.11: Numerical experiment 3. Left: Starting configuration. Right: Optimized configuration.

3.7 Future Work

Due to the limited amount of time for this project, some questions regarding the topic remain open for future work.

- This project was solved by using the PDE toolbox in MATLAB. Doing so, we encountered a number of drawbacks, which lead to some limitations in the solution. For example, the automatic numeration of the domains led to a fixed number of possible polygons, which is not desireable in the optimization. Thus, it would be of andvantage to use other packages or other programming languages to solve the PDE and to compute the torque.
- The optimization algorithm we used is based on a steepest descent method. There might exist more efficient and faster optimization algorithms which are applicable to the given problem.
- I this project, the given polygons have a maximum of six degrees of freedom to change their shape. An improvement would be to extend this number and compare the efficiency of the optimization algorithm.

Bibliography

[1] Lars-Christer Böiers. *Mathematical Methods of Optimization*. Studentlitteratur AB, Lund, 2010.

[2] Peter W. Christensen & A. Klarbring. *An introduction to structural optimization*. Solid Mechanics and its Applications V.153 Springer, 2009.

[3] Elisabeth Frank & Walter Zulehner. *Free-form Optimization of Electric Machines based on Shape Derivatives*(Master Thesis). Johannes Kepler University, Linz, 2010.

[4] Peter Gangl & Ulrich Langer. *Topology Optimization in Electrical Engineering*(Master Thesis). Johannes Kepler University, Linz, 2012.

[5] Clemens Pechstein & Ulrich Langer *Multigrid-Newton-Methods for Nonlinear Magnetostatic Problems*(Master Thesis). Johannes Kepler University, Linz, 2004.