

DELFT UNIVERSITY OF TECHNOLOGY

REPORT 12-06

ADJOINT SENSITIVITY IN ELECTRICAL IMPEDANCE TOMOGRAPHY
USING COMSOL MULTIPHYSICS

W. MULCKHUYSE, D. LAHAYE, A. BELITSKAYA

ISSN 1389-6520

Reports of the Department of Applied Mathematical
Analysis

Delft 2012

Copyright © 2012 by Department of Applied Mathematical Analysis, Delft, The Netherlands.

No part of the Journal may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission from Department of Applied Mathematical Analysis, Delft University of Technology, The Netherlands.

Adjoint Sensitivity in Electrical Impedance Tomography using COMSOL Multiphysics

Wouter Mulckhuyse

TNO Science and Industry, Delft, The Netherlands

E-mail: wouter.mulckhuyse@tno.nl

Domenico Lahaye

Department of Electrical Engineering, Mathematics and Computer Science,

TU Delft, Delft, The Netherlands

E-mail: d.j.p.lahaye@tudelft.nl

Alena Belitskaya

SRON, Utrecht, The Netherlands

E-mail: A.Belitskaya@sron.nl

Abstract

In 3D reconstruction problems the number of design variables is large and the forward model is typically computationally expensive. Reconstruction using gradient-based optimization algorithms requires many gradient evaluations. For such problems the computational cost of a finite difference approach to gradient calculation is prohibitive since the number of forward model evaluations per gradient calculation is equal to the number of design variables. We introduce the adjoint method which drastically reduces the amount of computations required for gradient calculation. Also, we introduce COMSOL Multiphysics as a valuable modeling and prototyping tool to be used for forward model evaluation and gradient calculation through the adjoint method. An example in Electrical Impedance Tomography (EIT) is provided to illustrate the use of the adjoint method for practical problems.

1. Introduction

In Electrical Impedance Tomography the conductivity profile throughout a conductive object is sought for using measurements of the electric field caused by injected excitation currents. Our approach is to measure the magnetic flux density instead of the electric field. The advantage is that there is no need for galvanic contact between sensor and the object. This allows for measuring through a thin layer of non-conducting material (e.g. a coating or isolation material). Magnetic flux density measurements are taken just above the outer surface of the object. Excitation currents are injected at the outer surface via galvanic connections.

A reconstruction method is required to estimate a conductivity profile from the magnetic flux density measurements. The reconstruction problem can be formulated as a partial differential equation (PDE) constrained minimization problem. These problems are often solved by gradient-based optimization algorithms. Such algorithms require the first and second order derivatives of the cost functional with respect to the design

variables. Calculating the derivatives using finite differences requires a minimum of one forward model evaluation per design variable. The forward model in our example is expensive to compute and the number of design variables is large. Consequently, the finite difference approach is not an option for the derivative computation. As an alternative the adjoint method provides a way of computing the gradients with a computation cost that is independent of the number of design variables. The adjoint method requires only one forward model and one adjoint model evaluation per gradient computation.

In this paper we introduce the adjoint method for least squares problems and illustrate its use by an example related to Electrical Impedance Tomography. We will also show that COMSOL Multiphysics is an excellent prototyping tool for reconstruction problems of this kind.

2. The EIT Example

2.1 Overview

The case we consider in this paper is measuring the thickness profile of the central section of a $50 \times 50 \times 0.1$ [cm³] aluminium plate. The thickness profile is derived from the conductivity profile of the plate, which we measure using EIT techniques. Excitation currents are injected through electrodes near the edge of the plate at opposite sides. Each side has 13 equally spaced electrodes. Measurements are performed by activating

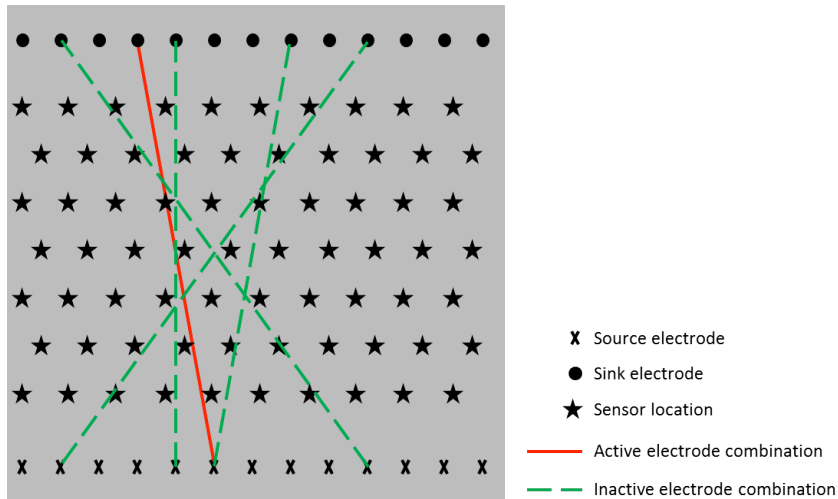


Figure 1. Measurement scenarios.

a single electrode pair and measuring the magnetic flux density at the sensor positions. At any time only two electrodes, a source and a sink, are active. The source and the sink electrode are always at opposite sides of the plate. By taking measurements with different electrode pairs and at different excitation frequencies we generate the full set of measurements that will be used to compute the conductivity profile of the plate. A schematic overview of the measurement scenarios is depicted in Figure 1. The solid line depicts the active electrode combination and the dashed lines depict inactive electrode combinations. Note that more valid electrode combinations exist than the ones drawn in the figure. Excitation currents are sinusoidal and the frequency ranges between 1 and 10 Hz. Sensors at more than 10 [cm] distance from the center of the active current path are

ignored because of the limited accuracy of the sensors. For reconstruction purposes we subdivide the plate into a set of conductivity cells of $1 \times 1 \times 0.1$ [cm³] resulting in 25000 design variables. The conductivity of each cell is assumed to be constant. A single unique design variable is associated with the conductivity of each cell. The objective of the reconstruction algorithm is to estimate the conductivity within each cell. The conductivity profile is then mapped to a thickness profile.

The EIT example as described above is based on one of our current projects. A laboratory setup is available for experimentation. The setup consists of two aluminium plates of $50 \times 50 \times 0.1$ [cm³]. One plate has a constant thickness (i.e. no defects), the other



Figure 2. Experimental setup.

plate has a spherical defect in the middle. Measurement of the magnetic flux density above the plate is automated. The sensor is located on a 3-axis translation stage which can position the sensor at any desired location. Figure 2 shows the translation stage above the plate with the spherical defect.

2.2 Forward Model

The forward model maps a conductivity profile and a current path into magnetic flux density at the sensor positions. Since the excitation current is sinusoidal and the frequencies are low, we will use as our forward model the following quasi-static time harmonic Maxwell equations.

$$\nabla \times \left(\frac{1}{\mu} \nabla \times A \right) + j\omega\sigma A + \sigma \nabla V = J_e \quad (1)$$

$$-\nabla \cdot (j\omega\sigma A + \sigma \nabla V - J_e) = 0 \quad (2)$$

where:

A Magnetic vector potential

V	Electric scalar potential
σ	Conductivity
ω	Excitation frequency [rad/s]
μ	Magnetic permeability

From the solution to A and V the magnetic flux density (B) and the electric field (E) can be calculated as follows.

$$B = \nabla \times A \quad (3)$$

$$E = -\nabla V \quad (4)$$

We have implemented the forward model in COMSOL Multiphysics which saves us the complex and tedious task of developing a Maxwell solver.

2.3 Inverse Model

The problem of computing the conductivity profile of an object from magnetic flux density measurements is formulated as a constrained least squares minimization problem.

$$\min_{\underline{\sigma} \in \mathcal{S}} \frac{1}{2} \| F(\underline{\sigma}) - \underline{d} \|^2_{\mathbf{y}} + \alpha L(\underline{\sigma}) \quad \text{subject to } \underline{\sigma}_{min} \leq \underline{\sigma} \leq \underline{\sigma}_{max} \quad (5)$$

where:

$F(\sigma)$	The state to observation functional which maps, for a given conductivity profile, the solution of the forward model (the state space) to a solution at the sensor positions (the observation space).
σ	The conductivity profile.
σ_{min}	The minimum value allowed for electrical conductivity.
σ_{max}	The maximum value allowed for electrical conductivity. Typically this is the electrical conductivity of the object's material.
d	The measured data.
$L(\sigma)$	A regularization functional.

The cost functional returns a scalar that is a measure for the discrepancy between the observed magnetic flux density and the computed magnetic flux density given a specific conductivity profile. This discrepancy decreases as the cost functional decreases. In practical situations the cost functional typically consists of a data discrepancy term and a regularization term. The purpose of the regularization term is to favor specific solution properties by increasing the penalty for solutions with undesired properties and/or to improve numerical stability and convergence behavior of the reconstruction process. A box constraint on conductivity is included to enforce that we obtain solutions that are physically meaningful.

3. The Adjoint Method

The adjoint method offers a way to compute the gradient of the cost functional with respect to the design variables requiring one forward model computation and one adjoint model computation only. The computational cost is therefore independent of the number of design variables. A good description of the adjoint method for gradient computation for least squares problems has been given by Vogel⁽¹⁾. Below we largely

reproduce his treatment of the subject with some minor adaptations towards our example application.

3.1 Definitions

We start with the definition of the vectors that we will use in our the description of the adjoint method.

$$\underline{\sigma} \in \mathbb{R}^K, \underline{\mathbf{d}} \in \mathbb{R}^N, \underline{\mathbf{u}} \in \mathbb{C}^P \quad (6)$$

The vector $\underline{\sigma}$ is the design variable vector. In our example each element of this vector corresponds to the conductivity of a unique conductivity grid cell. The number of conductivity grid cells is equal to K . The vector $\underline{\mathbf{d}}$ is the observation vector. Each element of $\underline{\mathbf{d}}$ corresponds to a unique magnetic flux density measurement. We measure the norm of the magnetic flux density, i.e. $\underline{\mathbf{d}}$ is real valued. The number of measurements is equal to N . The vector $\underline{\mathbf{u}}$ is the state vector for which each element corresponds to the complex valued computed magnetic flux density in a unique grid cell of the forward model's simulation domain. The number of magnetic flux density grid cells in the forward model is equal to P .

We introduce the function spaces U , S and Y which denote the state space, design variable space and the observation space, respectively.

The forward model operator $M(\underline{\sigma})$ corresponds to the finite element discretization of (1) and (2) on our problem's geometry and includes appropriate boundary conditions. The forward model maps a conductivity profile to a magnetic flux density throughout the simulation domain.

$$\mathbf{M}(\underline{\sigma}) \underline{\mathbf{u}} = \underline{\mathbf{s}} \quad (7)$$

The vector $\underline{\mathbf{s}}$ is the source term of the Maxwell's equations and corresponds to J_e in (1). If the forward model is well-posed its solution is given by

$$\underline{\mathbf{u}} = \mathbf{M}^{-1}(\underline{\sigma}) \underline{\mathbf{s}} \quad (8)$$

Note that in our case the forward matrix is large and sparse and therefore we are not able to compute an explicit inverse. Instead, $\underline{\mathbf{u}}$ will be computed using an iterative solver. From the forward model solution, the state variable $\underline{\mathbf{u}}$, we need to extract the computed magnetic flux density at the sensor locations. We introduce the functional $F : \mathbb{R}^K \rightarrow \mathbb{R}^N$ which maps the solution vector $\underline{\mathbf{u}}$ from the state space to a real-valued solution vector in the observation space that can be directly compared to the observation vector $\underline{\mathbf{d}}$.

$$F(\underline{\sigma}) = \mathbf{C} \underline{\mathbf{u}} \quad (9)$$

The $N \times P$ matrix \mathbf{C} that performs the actual mapping is commonly called the state to observation map. Note that the matrix \mathbf{C} performs the mapping of the complex valued magnetic flux density to the real valued norm of the magnetic flux density. Using (7), (9) and the observation vector $\underline{\mathbf{d}}$ we can formulate the reconstruction problem as a PDE constrained minimization problem with box constraints

$$\min_{\underline{\mathbf{u}} \in \mathcal{U}, \underline{\sigma} \in \mathcal{S}} \frac{1}{2} \| F(\underline{\sigma}) - \underline{\mathbf{d}} \|_{\mathcal{Y}}^2 + \alpha L(\underline{\sigma}) \text{ subject to } \begin{cases} \mathbf{M}(\underline{\sigma}) \underline{\mathbf{u}} = \underline{\mathbf{s}} \\ \underline{\sigma}_{min} \leq \underline{\sigma} \leq \underline{\sigma}_{max} \end{cases} \quad (10)$$

In (10) α denotes the regularization parameter and L denotes the regularization functional. Applying (8) to (9) yields

$$F(\underline{\sigma}) = \mathbf{C} \mathbf{M}^{-1}(\underline{\sigma}) \underline{\mathbf{s}} \quad (11)$$

This allows us to remove the PDE constraint and reformulate (10) into a minimization problem with box constraints on the design variable

$$\min_{\underline{\sigma} \in \mathcal{S}} \frac{1}{2} \| F(\underline{\sigma}) - \underline{\mathbf{d}} \|_{\mathcal{Y}}^2 + \alpha L(\underline{\sigma}) \text{ subject to } \underline{\sigma}_{min} \leq \underline{\sigma} \leq \underline{\sigma}_{max} \quad (12)$$

The gradient of the overall cost functional is obtained by taking the sum of the gradient of the data discrepancy term and the gradient of the regularization term. Since the gradient of the regularization term is often inexpensive to compute we will ignore the regularization term for the remainder of our discussion. The least squares data discrepancy term J_{LS} is the part of the cost functional that involves expensive forward model computations. From here on we focus on J_{LS} .

$$J_{LS}(\underline{\sigma}) = \frac{1}{2} \| F(\underline{\sigma}) - \underline{\mathbf{d}} \|_{\mathcal{Y}}^2 \quad (13)$$

In the manipulations that follow we will make use of the following identities from matrix calculus

$$\mathbf{M}(\underline{\sigma}) \mathbf{M}^{-1}(\underline{\sigma}) = \mathbf{I} \quad (14)$$

$$\frac{d}{d\underline{\sigma}} \mathbf{M}^{-1}(\underline{\sigma}) = -\mathbf{M}^{-1}(\underline{\sigma}) \frac{d\mathbf{M}}{d\underline{\sigma}} \mathbf{M}^{-1}(\underline{\sigma}) \quad (15)$$

$$\left. \frac{d}{d\tau} \mathbf{M}(\underline{\sigma} + \tau \underline{\mathbf{e}}_i) \right|_{\tau=0} = \frac{d\mathbf{M}}{d\underline{\sigma}} \underline{\mathbf{e}}_i = \frac{d\mathbf{M}}{d\sigma_i} \quad (16)$$

$$\langle \mathbf{M}(\underline{\sigma}) \underline{\mathbf{s}}, \underline{\mathbf{r}} \rangle_{\mathcal{Y}} = \langle \underline{\mathbf{s}}, \mathbf{M}^*(\underline{\sigma}) \underline{\mathbf{r}} \rangle_{\mathcal{U}} \quad (17)$$

Note that identity (15) follows directly from differentiating (14) with respect to $\underline{\sigma}$. Also note that (17) relates the inner product in observation space to the inner product in state space.

3.2 Gradient Computation

We now proceed with the computation of the gradient of J_{LS} with respect to the conductivity vector. In order to compute an element of the gradient we perturb J_{LS} in the corresponding element of the conductivity vector

$$\left[\underline{\mathbf{g}}_{LS} \right]_i = \left. \frac{d}{d\tau} J_{LS}(\underline{\sigma} + \tau \underline{\mathbf{e}}_i) \right|_{\tau=0} \quad (18)$$

Defining the residual $\underline{\mathbf{r}}(\underline{\sigma})$ as the difference between computed and observed data allows us to reformulate (18)

$$\underline{\mathbf{r}}(\underline{\sigma}) = F(\underline{\sigma}) - \underline{\mathbf{d}} \quad (19)$$

$$\left[\underline{\mathbf{g}}_{LS} \right]_i = \frac{1}{2} \frac{d}{d\tau} \left[\underline{\mathbf{r}}(\underline{\sigma} + \tau \underline{\mathbf{e}}_i)^* \underline{\mathbf{r}}(\underline{\sigma} + \tau \underline{\mathbf{e}}_i) \right] \Big|_{\tau=0} \quad (20)$$

$$= \frac{d}{d\tau} \underline{\mathbf{r}}(\underline{\sigma} + \tau \underline{\mathbf{e}}_i)^* \Big|_{\tau=0} \underline{\mathbf{r}}(\underline{\sigma}) \quad (21)$$

$$= \left\langle \frac{d}{d\tau} F(\underline{\sigma} + \tau \underline{\mathbf{e}}_i) \Big|_{\tau=0}, \underline{\mathbf{r}}(\underline{\sigma}) \right\rangle_{\mathcal{Y}} \quad (22)$$

$$= \left\langle \mathbf{C} \frac{d}{d\tau} \mathbf{M}^{-1}(\underline{\sigma} + \tau \underline{\mathbf{e}}_i) \Big|_{\tau=0} \underline{\mathbf{s}}, \underline{\mathbf{r}}(\underline{\sigma}) \right\rangle_{\mathcal{Y}} \quad (23)$$

Result (23) requires us to compute the derivative of the inverse operator $\mathbf{M}^{-1}(\underline{\sigma})$ with respect to $\underline{\sigma}$. This is an expensive computation that we want to avoid. Applying identity (15) to (23) allows us to reformulate (23) such that we only need the derivative of $\mathbf{M}(\underline{\sigma})$ with respect to $\underline{\sigma}$. This derivative can be computed efficiently since we can differentiate the governing equations of $\mathbf{M}(\underline{\sigma})$ analytically. Using the equations for the derivative of the governing equations we can then build the derivative of $\mathbf{M}(\underline{\sigma})$ with respect to $\underline{\sigma}$.

$$\left[\underline{\mathbf{g}}_{LS} \right]_i = - \left\langle \mathbf{C} \mathbf{M}^{-1}(\underline{\sigma}) \left(\frac{d\mathbf{M}}{d\underline{\sigma}} \underline{\mathbf{e}}_i \right) \mathbf{M}^{-1}(\underline{\sigma}) \underline{\mathbf{s}}, \underline{\mathbf{r}}(\underline{\sigma}) \right\rangle_{\mathcal{Y}} \quad (24)$$

If we use (24) to compute the elements of the gradient we still need to solve K times. In our case the computational cost is prohibitive. We now switch to the adjoint formulation of the gradient computation. Applying identity (17) to (24) gives

$$\left[\underline{\mathbf{g}}_{LS} \right]_i = \left\langle \left(\frac{d\mathbf{M}}{d\underline{\sigma}} \underline{\mathbf{e}}_i \right) \mathbf{M}^{-1}(\underline{\sigma}) \underline{\mathbf{s}}, [\mathbf{M}^*(\underline{\sigma})]^{-1} \mathbf{C}^* \underline{\mathbf{r}}(\underline{\sigma}) \right\rangle_{\mathcal{U}} \quad (25)$$

By introducing the adjoint variable $\underline{\mathbf{z}}$ we can rewrite this result in terms of the state variable $\underline{\mathbf{u}}$ and the adjoint variable. We now arrive at the final result

$$\underline{\mathbf{z}} = \mathbf{M}^{-1}(\underline{\sigma}) \mathbf{C}^* \underline{\mathbf{r}}(\underline{\sigma}) \quad (26)$$

$$\left[\underline{\mathbf{g}}_{LS} \right]_i = \left\langle \left(\frac{d\mathbf{M}}{d\underline{\sigma}} \underline{\mathbf{e}}_i \right) \underline{\mathbf{u}}, \underline{\mathbf{z}} \right\rangle_{\mathcal{U}}, \quad i = 1, \dots, K. \quad (27)$$

To compute the gradient we need the derivative of the forward model with respect to the design variables. This derivative can be obtained without significant computational cost as explained above. We also need the solution $\underline{\mathbf{u}}$ to the forward model. However, we needed that anyway to compute the data discrepancy term of the cost functional. Consequently, the computation of $\underline{\mathbf{u}}$ for gradient computation does not add extra computation. Finally, we need to compute the solution to the adjoint equation. The computational cost for solving the adjoint model is comparable to solving the forward model. So at the extra cost of one adjoint model solve we can compute all elements of the gradient.

3.3 Hessian Computation

The entries of the true Hessian are computed by perturbing the design variable in two directions.

$$[\mathbf{H}]_{ij} = \frac{d}{d\eta} \left[\frac{d}{d\tau} \left[\frac{1}{2} \mathbf{r}(\underline{\sigma} + \tau \mathbf{e}_i + \eta \mathbf{e}_j)^* \mathbf{r}(\underline{\sigma} + \tau \mathbf{e}_i + \eta \mathbf{e}_j) \right] \right]_{\tau=0} \bigg|_{\eta=0} \quad (28)$$

$$= \left[\frac{d^2}{d\eta d\tau} \left[\mathbf{r}(\underline{\sigma} + \tau \mathbf{e}_i + \eta \mathbf{e}_j)^* \right] \mathbf{r}(\underline{\sigma} + \tau \mathbf{e}_i + \eta \mathbf{e}_j) + \dots \right. \\ \left. \frac{d}{d\tau} \mathbf{r}(\underline{\sigma} + \tau \mathbf{e}_i + \eta \mathbf{e}_j)^* \frac{d}{d\eta} \mathbf{r}(\underline{\sigma} + \tau \mathbf{e}_i + \eta \mathbf{e}_j) \right]_{\tau=0, \eta=0} \quad (29)$$

Instead of using the true Hessian we will use the Gauss-Newton approximation to the Hessian. The Gauss-Newton approximation is obtained by ignoring the first term in (29), which gives

$$[\mathbf{H}^{\text{GN}}]_{ij} = \frac{d}{d\tau} \mathbf{r}(\underline{\sigma} + \tau \mathbf{e}_i)^* \bigg|_{\tau=0} \frac{d}{d\eta} \mathbf{r}(\underline{\sigma} + \eta \mathbf{e}_j) \bigg|_{\eta=0} \quad (30)$$

$$= \left\langle \frac{d}{d\tau} \mathbf{r}(\underline{\sigma} + \tau \mathbf{e}_i) \bigg|_{\tau=0}, \frac{d}{d\eta} \mathbf{r}(\underline{\sigma} + \eta \mathbf{e}_j) \bigg|_{\eta=0} \right\rangle_{\mathcal{Y}} \quad (31)$$

We obtain the adjoint formulation of the Gauss-Newton approximation of the Hessian in a similar manner as for the gradient, see equations (20) through (24).

$$[\mathbf{H}^{\text{GN}}]_{ij} = \left\langle \mathbf{CM}(\underline{\sigma})^{-1} \left(\frac{d\mathbf{M}}{d\underline{\sigma}} \mathbf{e}_i \right) \underline{\mathbf{u}}, \mathbf{CM}(\underline{\sigma})^{-1} \left(\frac{d\mathbf{M}}{d\underline{\sigma}} \mathbf{e}_j \right) \underline{\mathbf{u}} \right\rangle_{\mathcal{Y}} \quad (32)$$

$$= \left\langle \left(\frac{d\mathbf{M}}{d\underline{\sigma}} \mathbf{e}_i \right) \underline{\mathbf{u}}, \mathbf{M}^*(\underline{\sigma})^{-1} \mathbf{C}^* \mathbf{CM}(\underline{\sigma})^{-1} \left(\frac{d\mathbf{M}}{d\underline{\sigma}} \mathbf{e}_j \right) \underline{\mathbf{u}} \right\rangle_{\mathcal{U}} \quad (33)$$

The result shows that $2K$ solves are needed to obtain the elements of \mathbf{H}^{GN} . In our case the computational cost of this approach is still prohibitive. A Quasi-Newton method like BFGS is much cheaper in terms of computation and is therefore a better alternative for the approximate Hessian computation. For large scale problems a limited memory BFGS version exists. An in-depth discussion of the reconstruction algorithm is outside the scope of this paper. For detailed information on gradient-based optimization algorithms see Nocedal⁽²⁾.

3.5 Application to COMSOL Multiphysics

In order to validate our ideas with respect to the reconstruction algorithm we required a prototyping environment with the following characteristics

- Easy implementation of the forward model for arbitrary geometries.
- The freedom to experiment with different reconstruction algorithms.

For forward model computation COMSOL Multiphysics is a convenient choice. COMSOL Multiphysics is an excellent Finite Element Modeling tool that has some attractive features with respect to our application

- A quasi-static electromagnetic application mode which contains a solver for the governing equations of our forward model. For a detailed description of FEM with respect to electromagnetics see Jin⁽³⁾
- Sensitivity computation through the adjoint method as part of the forward model evaluation. Sensitivity computations are with respect to a user defined objective function.
- MATLAB code generation capabilities. This feature allows models implemented in COMSOL Multiphysics to be evaluated from MATLAB scripts.

It is worth noting that COMSOL Multiphysics also has an optimization toolbox which, in many cases, provides adequate functionality for the solution of inverse problems. However, it was rendered too limited for our purposes.

The combination of MATLAB and COMSOL was selected for our prototyping environment. MATLAB is used for implementation of the reconstruction algorithm. COMSOL Multiphysics is used for forward model and the gradient computation (using the adjoint method). The MATLAB code generation feature from COMSOL Multiphysics is essential because it allows us to call the forward model evaluation and gradient computations from our MATLAB code.

4. COMSOL Case Study

A simple study case is defined in COMSOL to investigate how the presence of a defect (e.g. a corrosion spot) on a conductive body (e.g. Al plate) influences on the magnetic flux density. Two models have been implemented in COMSOL, where one contains the Al plate with a defect and another is Al plate without a defect. A defect has a hemispherical shape with a diameter 8 [cm] and deep 1 [mm] as depicted in Figure 3.

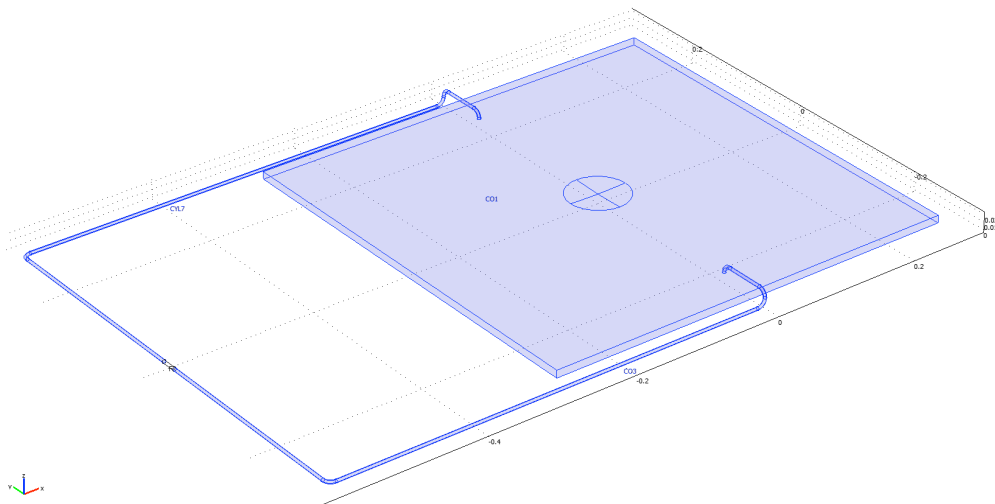


Figure 3: Plate geometry.

In this section we describe the steps required to build a model in COMSOL. First, an application mode needs to be selected, which determines the physical phenomena to be modeled. Then, the geometry of the object under investigation needs to be drawn. A geometry consists of subdomains (volumes), boundaries (surfaces), edges (lines) and points.

The geometry for an Al plate with a hemispherical defect drawn in COMSOL is shown in Figure 3. It is important to note that only one current excitation path is shown here and the wires connecting the current source are also part of the model since they influence the magnetic flux density at the sensor positions significantly. When the geometry is ready, the material properties of the subdomains, the boundary conditions and the excitation of the model must be defined.

When the geometry and physical properties of the material are properly defined, we are ready to mesh our model. The trade-off for 3D problems is the mesh density versus computation time and memory requirements. For an accurate simulation, a fine mesh is required. However, in many cases a fine mesh is not needed everywhere in the simulation domain. COMSOL has selective meshing and interactive meshing features that allow the meshes to be tailored according to specific needs.

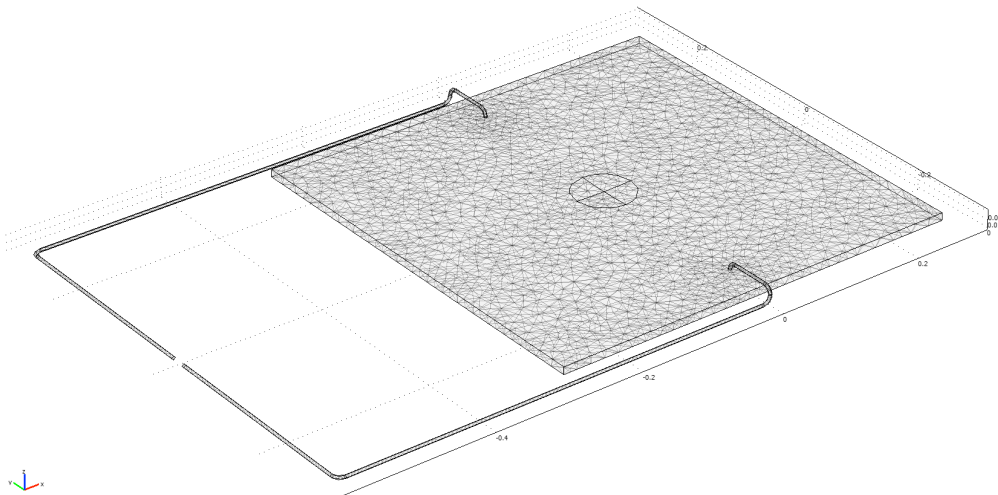


Figure 4: The meshed model.

The meshed geometry of the forward model is shown in Figure 4. When meshing is completed we must select and configure the solver that will be used to solve the model. COMSOL supports a whole range of solvers each with its specific strengths and weaknesses. Solver selection and configuration is therefore not a simple matter but requires some study. For our model we select a geometric multigrid solver. Now we are ready to run the model and obtain a solution.

In Figure 5 and Figure 6 the obtained simulation solutions for the X component of the magnetic flux density, denoted as B_x , and the Z component of the current density, denoted as J_z , for each coordinate on plate are shown. We expect that both, B_x and J_z ,

will increase near the defect because the current flow is redirected near the defect. The simulation results confirm our expectation.

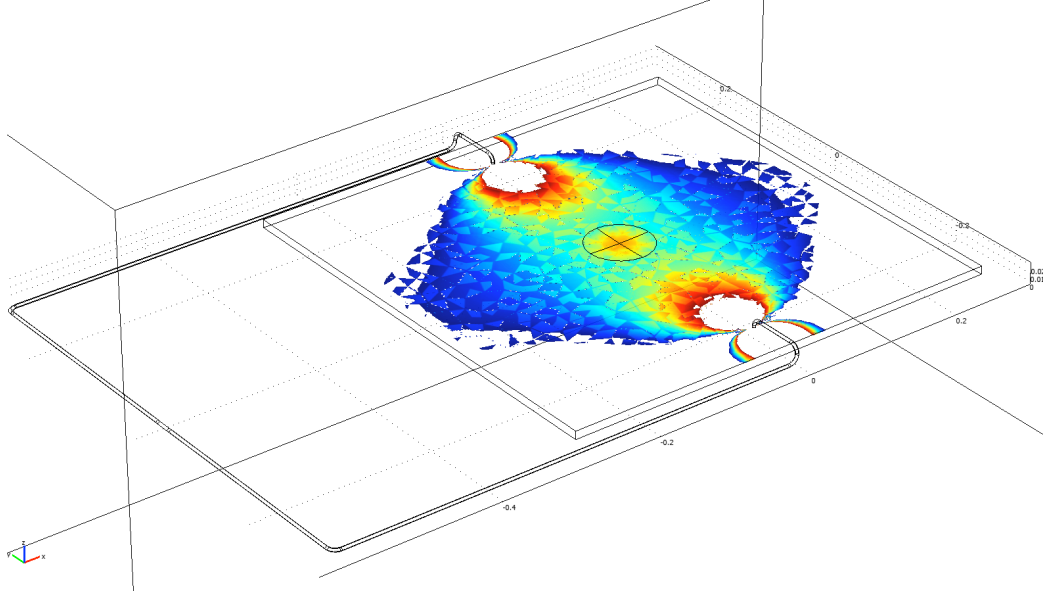


Figure 5: X component of magnetic flux density, denoted as B_x .

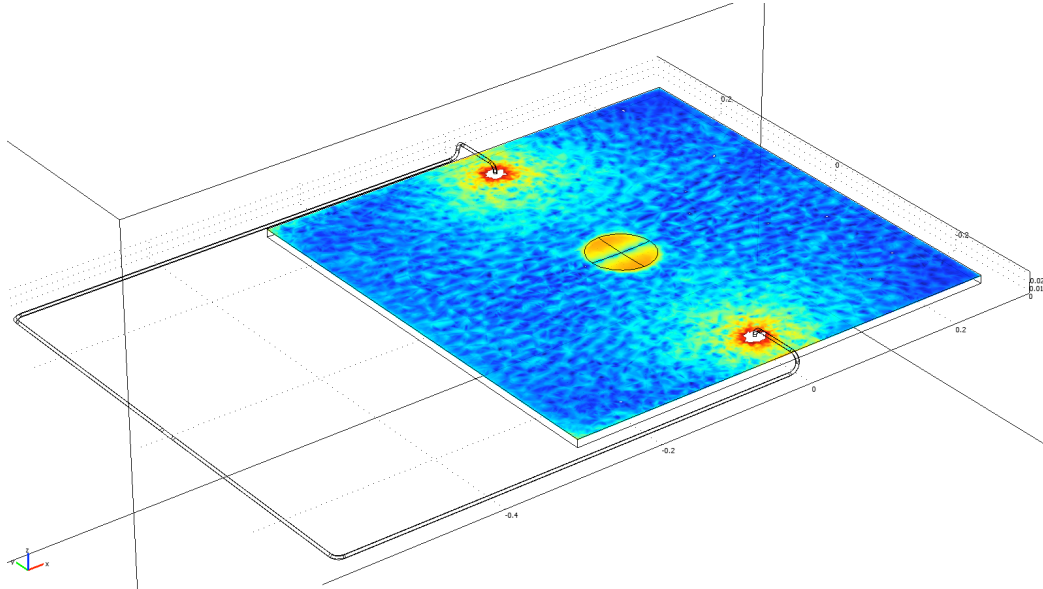


Figure 6: Z component of current density, denoted as J_z .

5. Conclusions

We have demonstrated the computational benefits of the adjoint method for computation of the gradient of a least squares cost functional with respect to the design variables. The adjoint method allows us to compute the gradient from one state and one adjoint model solve regardless of the number of design variables. The adjoint method

applied to Hessian computation is still computationally expensive. The Gauss-Newton approximation of the Hessian requires $2K$ forward model evaluations, where K is the number of design variables. Methods based on the gradient history, like BFGS, are computationally more efficient. The combination of COMSOL Multiphysics and MATLAB provides an excellent prototyping environment for solving PDE constrained optimization problems.

References

- (1) C. R. Vogel, *Computational Methods for Inverse Problems*, SIAM, Philadelphia, 2002.
- (2) J. Nocedal and S. J. Wright, *Numerical Optimization 2nd Edition*, Springer, New York, 2006.
- (3) J. Jin, *The Finite Element Method in Electromagnetics*, John Wiley & Sons, New York, 2002.