

**Numerical study of an evolutionary algorithm  
for electrical impedance tomography**

**Dissertation**

zur Erlangung des Doktorgrades  
der Mathematisch-Naturwissenschaftlichen Fakultäten  
der Georg-August Universität Göttingen

**Vorgelegt von :**

Harry Eckel

aus Kassel

Göttingen 2007

D7

Referent: Prof. Dr. Rainer Kreß

Koreferent: Prof. Dr. Roland Potthast

Tag der mündlichen Prüfung: 08. Januar 2008

# Abstract (English)

We consider the two-dimensional inverse electrical impedance problem without a-priori information. There, we want to reconstruct the conductivity inside a conducting object from currents and voltages applied at its exterior boundary. In our case we model the conductivity as *piecewise constant*, i.e., we define closed nonintersecting interface curves inside the object under consideration and require the conductivity to be constant between these interfaces. For the data at the exterior boundary we consider both the *continuum model* and the *complete electrode model*. In the first model we assume the full Cauchy data to be given, whereas in the second model we are given discrete values of currents and voltages at the electrodes.

We solve this problem by an *boundary integral equation method*. It is based on a system of nonlinear integral equations arising from Green's representation formula, from which the unknown conductivities and the unknown shapes of the interfaces are obtained iteratively via linearization. The method is an extension of a method that has been suggested by Kress and Rundell [39] for the case of one perfectly conducting inclusion.

For the dynamical adaptation of the regularization parameters occurring in the method we propose an *evolutionary algorithm*. This algorithm is furthermore used to provide an initial guess for the iterative solution by coupling it together with the *factorization method* [9] for the continuum model and some Newton-type *finite-element method* [51] for the complete electrode model. We describe the boundary integral equation method and the evolutionary algorithm in detail and illustrate its feasibility by various numerical examples.

At the end we also apply the evolutionary algorithm on *real data*. For this, some additional modifications to the algorithm turned out to be necessary. We describe these modifications, and finally two results for real data are presented.



# Abstract (Deutsch)

Wir untersuchen das inverse elektrische Impedanzproblem ohne gegebene Vorab-Informationen. Dabei geht es darum, aus Strom- und Spannungsmessungen am Rande eines leitenden Objektes die Leitfähigkeit innerhalb des Objektes zu rekonstruieren. In unserem Fall modellieren wir die Leitfähigkeiten als *stückweise konstant*, d.h. wir definieren geschlossene Kurven innerhalb des zu untersuchenden Objektes, die sich nicht überschneiden dürfen, und nehmen an, dass die Leitfähigkeit zwischen diesen Kurven jeweils konstant ist. Für die Eingangsdaten betrachten wir sowohl das *kontinuierliche Modell* als auch das *Complete Electrode Model*. Bei ersterem nehmen wir an, dass die vollständigen Cauchy-Randdaten gegeben sind, und bei letzterem gehen wir von diskreten Werten für Strom und Spannung aus.

Wir lösen dieses Problem über eine *Randintegralgleichungsmethode*. Diese basiert auf einem System nichtlinearer Integralgleichungen, die mit Hilfe der Greenschen Formel hergeleitet werden. Durch Linearisierung und iterative Lösung dieses Systems erhält man die Werte der unbekannt inneren Kurven und Leitfähigkeiten. Die Methode stellt eine Erweiterung einer Idee von Kress und Rundell [39] für den Fall einer perfekt leitenden Inklusion dar.

Die dynamische Anpassung der Regularisierungsparameter, die bei dieser Methode vorkommen, geschieht durch einen *Evolutionären Algorithmus*. Dieser wird weiterhin dazu verwendet, eine Startlösung für die Randintegralgleichungsmethode zu bestimmen. Dazu koppelt er die Methode im kontinuierlichen Fall mit der *Faktorisierungsmethode* [9] und für das Complete Electrode Model mit einer Newton-artigen *Finite Elemente Methode* [51]. Die Randintegralgleichungsmethode und der Evolutionäre Algorithmus werden ausführlich beschrieben und anhand zahlreicher Beispiele getestet.

Am Ende wenden wir den Algorithmus auch noch auf *reale Daten* an. Dafür mussten gewisse zusätzliche Modifikationen des Algorithmus vorgenommen werden. Diese Modifikationen werden erläutert, und am Schluss werden zwei Beispiele für reale Daten präsentiert.



# Contents

<b>Introduction and outline</b>	<b>1</b>
<b>1 The general EIT problem</b>	<b>9</b>
1.1 Mathematical modeling of the physical properties . . . . .	9
1.1.1 Interior conditions: Simplifying the full Maxwell equations . . . . .	10
1.1.2 Boundary conditions: From the continuum model to the complete electrode model . . . . .	11
1.2 Formulation of the direct and the inverse problem . . . . .	14
1.2.1 The direct problem . . . . .	14
1.2.2 The inverse problem . . . . .	17
1.3 A survey of existing methods . . . . .	17
<b>2 A novel boundary-element method</b>	<b>20</b>
2.1 Modeling the EIT problem as a transmission problem . . . . .	20
2.1.1 Uniqueness of the inverse transmission problem for one boundary data pair . . . . .	24
2.2 Solution of the direct problem . . . . .	25
2.2.1 Solution of the direct problem for the continuum model . . . . .	26
2.2.2 Solution for the complete electrode model . . . . .	30
2.3 Solution of the inverse problem . . . . .	34
2.3.1 Deriving a system of integral equations . . . . .	35

2.3.2	Linearization and iterative solution . . . . .	37
2.3.3	Extensions for the complete electrode model . . . . .	39
2.3.4	Regularization of the linearized system . . . . .	40
<b>3</b>	<b>Two methods for obtaining an initial guess</b>	<b>42</b>
3.1	The factorization method . . . . .	42
3.1.1	Basic idea . . . . .	42
3.1.2	The factorization method for inner inclusion levels . . . . .	43
3.1.3	Numerical implementation . . . . .	44
3.2	A simple Finite Element method . . . . .	45
3.2.1	Repartition of the domain and basis functions . . . . .	45
3.2.2	Solution of the direct problem via an FEM . . . . .	46
3.2.3	Solution of the inverse problem by a Newton-type iteration . . . . .	47
<b>4</b>	<b>Basic concepts of evolutionary algorithms</b>	<b>50</b>
4.1	Principles and notation . . . . .	50
4.2	Main parts of one generation step . . . . .	52
4.2.1	Recombination . . . . .	52
4.2.2	Mutation . . . . .	53
4.2.3	Selection . . . . .	53
4.3	Further development of evolutionary algorithms . . . . .	54
<b>5</b>	<b>The concept of our evolutionary algorithm</b>	<b>55</b>
5.1	Definition of an individual and of the fitness functional . . . . .	55
5.1.1	Definition of an individual . . . . .	55
5.1.2	Definition of the fitness functional . . . . .	56
5.2	Creation of the initial population . . . . .	57
5.2.1	Selecting the points . . . . .	57
5.2.2	Converting the point sets into subdomains . . . . .	58



5.2.3	Global subdomain indices . . . . .	60
5.2.4	An example . . . . .	62
5.3	One generation step . . . . .	65
5.3.1	Recombination . . . . .	65
5.3.2	Mutation . . . . .	67
5.3.3	Selection . . . . .	71
5.4	Further extensions . . . . .	72
5.4.1	An intermediate step . . . . .	72
5.4.2	Creating a new population for an inner inclusion level . . . . .	73
5.4.3	Partial data sets . . . . .	75
5.5	A flow diagram of the algorithm . . . . .	76
<b>6</b>	<b>Numerical examples</b>	<b>78</b>
6.1	Parameter and example choice . . . . .	78
6.1.1	Parameters of the evolutionary algorithm . . . . .	78
6.1.2	Fitness functional and substitution function . . . . .	80
6.1.3	Adjacent and trigonometric current patterns . . . . .	81
6.1.4	Presentation of the examples for the synthetic data . . . . .	82
6.2	Results for the continuum model . . . . .	85
6.2.1	One inclusion level . . . . .	85
6.2.2	Results without using the factorization method . . . . .	88
6.2.3	Multiple inclusion levels . . . . .	88
6.2.4	Noisy data . . . . .	91
6.2.5	Summary and interpretation of the results . . . . .	97
6.3	Results for the complete electrode model using synthetic data . . . . .	97
6.3.1	Results for exact data . . . . .	98
6.3.2	Results for noisy data . . . . .	103
6.3.3	Summary and interpretation of the results . . . . .	108
6.4	Testing with real data . . . . .	109

6.4.1	Preparing for real data: Incomplete adjacent current patterns . . . . .	110
6.4.2	Preparing for real data: Shrinking the electrode size . . . . .	112
6.4.3	Preparing for real data: Systematic measurement errors . . . . .	114
6.4.4	Preparing for real data: Approximate conversion from 3D data to 2D data . . . . .	117
6.4.5	Results for real data . . . . .	119

<b>Conclusion and outlook</b>	<b>121</b>
-------------------------------	------------

# Introduction and outline

Electrical impedance tomography (EIT) is a novel noninvasive evaluation method that creates images of the electrical conductivity  $\sigma$  of an inhomogeneous medium  $\Omega$  by applying currents  $I_\ell$  at a number of electrodes  $\mathcal{E}_\ell$  at the boundary and measuring the resulting voltages  $U_\ell$  at the electrodes. This medium can be for example a drill core where the conductivity provides information on the materials it contains, or the method can be applied on the human brain or thorax to detect anomalies in the tissue.

In the first case we usually have a *diagnostic situation*. We do not assume to have any a-priori information about the material contained in the drill core. On the other hand the conductivity does not change in time. In the second case we have a *monitoring situation*, where the conductivity changes in time (due to breathing respectively brain activity). On the other hand one already has some a-priori information about the physiology, and very often only the *change of conductivity* between two moments in time is of interest. These two examples illustrate that before starting with the mathematical modeling we have to know

- if the conductivity is time-invariant (*static conductivity*) or it changes in time (*dynamic conductivity*).
- if we are interested in the absolute values of the conductivity (*absolute conductivity reconstruction*) or just in the change of the conductivity between two different states or with respect to a reference conductivity (*relative conductivity reconstruction*).
- if we have any *a-priori information* at hand.

These criteria are important for the decision which method is appropriate for the problem under consideration and which requirements it has to fulfill. If we have a dynamic conductivity, we do not have the time to make lengthy computations. So usually no iterative reconstruction methods should be used.

If only the relative change of conductivity is researched, the problem can be approximately linearized by a relatively simple approach like in the widely used method described in [5]. Furthermore the *systematic measurement errors*<sup>1</sup> are much smaller, as the difference of two measurements with the same device is taken.

---

<sup>1</sup>which are errors that come from the measuring device (for further description see subsection 6.4.3)

Finally if some a-priori information is known, one should try to incorporate it into the reconstruction algorithm. This can be done in the initialization of the method, or this information also can be incorporated into the method itself like in [50].

In our case we will limit ourselves to *static absolute conductivity reconstructions with no a-priori information*, which corresponds to the drill core example mentioned beforehand.

### Modeling the real-world problem

After the considerations made above one has to define how the real-world problem is mathematically modeled. First one has to decide if the model is made in two or in three dimensions. Obviously the 3D model is more realistic, but it is also more difficult to be computed. Especially if the electrodes are only placed at one cross-section we cannot expect to obtain good reconstructions above and below it. Therefore it seems to constitute an acceptable loss of information if we only consider a two-dimensional cross-section of the object under consideration. Furthermore we will see in chapter 6 that it is basically possible to approximate the 3D measurements by a 2D model.

Also the boundary measurements can be modeled in different ways. In subsection 1.1.2 we will introduce two different models for the boundary conditions: The *continuum model* and the *complete electrode model* (CEM). In the first model currents and voltages are considered as continuous functions  $f, g$  over the boundary. This means that the *potential function*  $u$  fulfills the boundary conditions

$$u = f \text{ and } \sigma \frac{\partial u}{\partial \nu} = g \text{ on } \partial\Omega. \quad (1)$$

In the second model currents and voltages are considered as discrete values at the electrodes in the sense that

$$u + z_\ell \sigma \frac{\partial u}{\partial \nu} = U_\ell \text{ on } \mathcal{E}_\ell \quad (2)$$

$$\int_{\mathcal{E}_\ell} \sigma \frac{\partial u}{\partial \nu} dS = I_\ell \quad (3)$$

$$\sigma \frac{\partial u}{\partial \nu} = 0 \text{ on } \mathcal{N}, \quad (4)$$

where  $z_\ell$  denotes the *contact impedance* of the electrode  $\mathcal{E}_\ell$  and  $\mathcal{N}$  the space between the electrodes.

Although the complete electrode model is more realistic than the continuum model, we will consider both of them. The reason is that the continuum model is theoretically and numerically much easier to work with, and so certain aspects of our algorithm can be tested more easily.

Finally we have to decide how we model the conductivity inside the object. In our case we consider it as being *piecewise constant*. This means that we define closed non-intersecting curves  $\Gamma_i$  inside the object, and the conductivity  $\sigma_i$  between two curves is assumed to be constant. So the curves are considered as *interfaces* between different conductivities. The task in this case consists in the reconstruction of the interface curves and the conductivities. Figure 1 shows an example configuration for this modeling. Note that as indicated in the example we allow a finite number of layers with different conductivities included in each other. The maximum number of those layers we call the *inclusion level*.

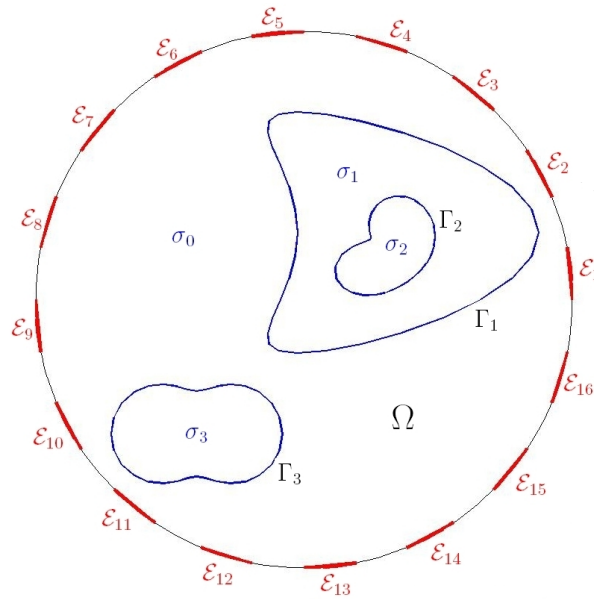


Figure 1: Example of a configuration with 16 electrodes and three inhomogeneities.

### Direct and inverse problem

The problem described above is the *inverse EIT problem*. By contrast the *direct EIT problem* consists in the determination of the voltages with given currents and a given conductivity distribution (direct Neumann problem) or the determination of the currents with given voltages and a given conductivity (direct Dirichlet problem). Physically the direct problem consists in the computation of the effect of a given cause, whereas the inverse problem describes the determination of the cause from the effect.

Mathematically inverse problems are usually *ill-posed*. Especially the solution is not *stable*, such that small changes or errors in the input data can lead to large changes in the result. For that reason usually some kind of *regularization* (see for example [16]) is required for the numerical solution. Also the *uniqueness* of the solution is not always guaranteed. In our case there only exists a theoretical uniqueness result for the

continuum model with infinitely many input data pairs (see [2]). So for the numerical solution for our examples we have to rely on some experimental heuristics to choose an appropriate number of boundary data pairs.

### A boundary-element method

The main method used is a novel boundary-element method based on an idea recently published by Kress and Rundell ([39]). It is described in detail in **chapter 2**.

We first need to introduce some notations. By  $\Gamma_0 = \partial\Omega$  we denote the exterior boundary curve and by  $\Gamma := \bigcup_{i=1}^N \Gamma_i$  the union of the interior interface curves. Denoting by  $h$  the trace of  $u$  on the interior interfaces we then define the double-layer potentials over  $\Gamma$  as

$$(K_0h)(x) := \int_{\Gamma} \frac{\partial\Phi(x,y)}{\partial\nu(y)} h(y) ds(y), \quad x \in \Gamma_0, \quad (5)$$

$$(Kh)(x) := \int_{\Gamma} \frac{\partial\Phi(x,y)}{\partial\nu(y)} h(y) ds(y), \quad x \in \Gamma. \quad (6)$$

Over the exterior curve  $\Gamma_0$  we define the combined potential

$$w(f,g)(x) := \int_{\Gamma_0} \left( \sigma_0 \frac{\partial\Phi(x,y)}{\partial\nu(y)} f(y) - \Phi(x,y)g(y) \right) ds(y), \quad x \in \Gamma. \quad (7)$$

with  $f, g$  as in (1). By using Green's integral theorem (see [37]) and taking care of the jump relations of the double-layer potentials we then obtain

$$\begin{aligned} -\frac{1}{2} \sigma_0 f + K_0h &= w_0(f,g) \quad \text{on } \Gamma_0, \\ -\frac{1}{2} \mu h + Kh &= w(f,g) \quad \text{on } \Gamma, \end{aligned} \quad (8)$$

where  $\mu$  depends on the (piecewise constant) conductivities  $\sigma_i$  and the potential  $w_0(f,g)(x)$  for  $x \in \Gamma_0$  is defined in an obvious way.

The equations (8) together with (1) in the continuum model and (2)–(4) in the complete electrode model now constitute a system of integral equations, which can be used for the solution of the direct problem as well as for the solution of the inverse problem.

Following the idea of Kress and Rundell we use it for the solution of the inverse problem. Then in the continuum case we are given the functions  $f, g$ , and in the CEM case we are given the voltages and currents  $U, I$ . Therefore the unknowns are the interface curves  $\Gamma$  (given through the operators  $K$ ), the conductivities  $\sigma$  (given through

$\mu$ ) and the potential functions  $h$ . In the complete electrode model also  $f$  and  $g$  are unknown.

The system is then linearized with respect to each unknown and approximately solved in an iterative way. The linearization with respect to  $f, g, h$  and  $\mu$  is straightforward, whereas with respect to  $\Gamma$  we have to determine the *Fréchet derivatives* (see [44]) of (5)–(7). For this we further need a *parameterization* of each curve  $\Gamma_i$ , which in our case is given by

$$z_i(t) = \xi_i + r_i(t)(\cos t, \sin t), \quad t \in [0; 2\pi[,$$

with a center point  $\xi_i \in \mathbb{R}^2$  and a  $2\pi$  periodic  $C^2$  function

$$r_i(t) = \sum_{k=0}^{m_i} a_{ik} \cos kt + \sum_{k=1}^{m_i} b_{ik} \sin kt, \quad (9)$$

where  $m_i$  is a previously defined polynomial degree and  $a_{ik}, b_{ik}$  are the Fourier coefficients that determine the size and shape of  $\Gamma_i$ . Note that unlike in a usual Newton-type iteration we do not need to solve the forward problem in each iteration step, as this method linearizes integral equations rather than the boundary-to-data map.

It can be shown (see subsection 2.1.1) that only one boundary data pair is not enough to uniquely determine a solution to the inverse EIT problem. Therefore we always assume that we are given  $M > 1$  data pairs. Then for each of them we are given one set of equations of the form (8) together with the respective boundary conditions. The values of the potential function  $u$ , and therefore the unknowns  $f, g, h$  are different for each data pair, but the values for  $\sigma$  and  $\Gamma$  have to be equal for all of them. Therefore we patch the linearized equations for each boundary data pair together and solve them for  $\Gamma$  and  $\mu$  in a least-squares sense.

#### *Regularization and a-priori information*

As the inverse EIT problem is ill-posed the solution of the linearized system requires regularization. In our case we use the well-established *Tikhonov regularization*, which instead of solving an equation  $A\varphi = b$  solves the regularized equation

$$(\alpha \cdot Id + A^*A)\varphi = A^*b \quad (10)$$

where the notation  $\alpha \cdot Id$  means that we take the identity matrix and replace its diagonal with the values of the *penalty vector*  $\alpha$ . We allow different Tikhonov parameters for each unknown, and we also keep the polynomial degree  $m_i$  in (9) variable. Furthermore these parameters should not be kept fixed during the whole iteration, but they rather should be adapted within each iteration step.

Secondly we note that in the presented form the method imperatively needs some a-priori information. We have to supply it with the number of interface curves as well as with their approximate position to create an initial guess for the iterative solution of the linearized system. This implies that if we assume not to be given any a-priori information we have to use a different method to obtain it.

### The evolutionary algorithm

To provide the necessary a-priori information to the boundary-element method and to adapt dynamically its regularization parameters we use an *evolutionary algorithm*. The basic ideas of this evolutionary algorithm also have been used in [15] for the solution of an inverse scattering problem.

Evolutionary algorithms originate from the area of optimization as they try to minimize a given functional. The two main differences to conventional optimization methods are

- They contain random elements, such that when running them several times for the same input data, the result is not necessarily always the same.
- They do not consider only one approximation to the given problem at each iteration step, but a whole set of them (called a *population*).

A survey of the basic principles and notations of evolutionary algorithms can be found in **chapter 4**.

Following the terms of evolutionary algorithms a population contains several *individuals*. In our case of piecewise constant conductivities an individual is a set of interface curves together with their conductivities and regularization parameters. This constellation will also be called a *subdomain* within this paragraph. The number and position of the subdomains can vary within one population.

At the beginning of our evolutionary algorithm, we have to create a starting population, for which we need an initial guess for our boundary-element method. As we assume not to be given any a-priori information, we introduce in **chapter 3** two methods which do not need any a-priori information. We use the *factorization method* (see [9]) for the continuum model and some *Newton-type finite element method* from [51] for the complete electrode model.

These methods provide us with some information about the conductivity on a pre-defined grid. We then have to convert this representation into a domain representation with piecewise constant conductivities. For this we define a strictly monotone sequence of thresholds  $\theta$ , and for each threshold we choose those points for which the result of the method lies beyond this threshold. Each set of connected points found this way is then converted into a subdomain. This way for each threshold we create a new individual for our starting population. Figure 2 shows an example for a conversion from a finite-element representation (left) into a domain representation (right).

Once we are given a population we can create new individuals for the next *generation* from the properties of the old ones by *recombination*. If we have multiple domain constellations, not necessarily all subdomains are taken also for the new individual. So especially if one individual contains artifacts which are not included in another individual, with a certain chance the recombination of those two does not contain the artifact anymore.



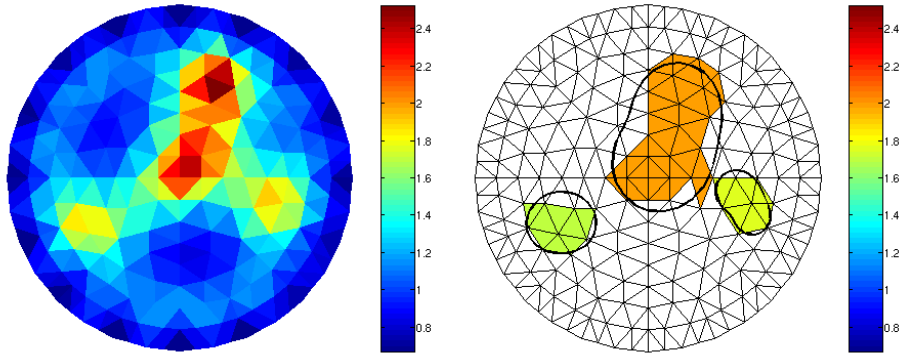


Figure 2: Example for a conversion between an FEM and a domain representation.

Then the newly created individuals are submitted to *mutation*. There, first its regularization parameters are randomly changed. With these modified parameters we then execute a given number of iteration steps of our boundary-element method. This way we hope to find not only a better approximation to the solution of the inverse problem, but also good regularization parameters.

We always create more individuals than we had before, so finally we have to *select* the best of them, which constitute our next generation. To evaluate them we have to define a *fitness functional*, which assigns to each individual a positive real number. In our case we solve for each individual the direct Neumann problem for the domain constellation it represents. Then we take the norm of their difference to the given (measured) input data  $\|f_{\text{calc}} - f_{\text{meas}}\|$  respectively  $\|U_{\text{calc}} - U_{\text{meas}}\|$  as our fitness functional. Although the problem is ill-posed, this measure usually seemed to be sufficient to evaluate the quality of an approximation.

## Numerical results

We tested our evolutionary algorithm for the continuum and the complete electrode model with synthetic data created for different domain configurations. We chose examples with one and with two inclusion levels. A detailed description can be found in chapter 6 together with the plots of the results. In all cases there was no considerable difference between the performance of the algorithm for the continuum model and for the complete electrode model.

For those examples with one inclusion level and with exact input data, the algorithm was always able to find the correct domain geometry (number and approximate position of the subdomains). In most cases it was also able to find the exact shape and almost the exact conductivity of the subdomains.

For noisy input data and one inclusion level the algorithm found the correct domain geometry in almost all cases. Obviously it could not find the exact shape and conductivity of the subdomains, but it still found the correct contrast and the approximate size

in all cases with the correct geometry.

For two inclusion levels, however, the results were less satisfactory. For exact data the correct shape and conductivity was only found in those examples containing one subdomain at the inner inclusion level with the same contrast as the outer subdomain. This was not the case for those examples with more than one inner subdomain or where the contrast of the inner subdomain was different. Of course those examples also could not be reconstructed for noisy data. The configuration with one subdomain at the inner inclusion level with the same contrast could be approximately reconstructed for 1% and 2% noise, but not for 5%.

Finally we also applied our algorithm on real data provided by the Institute for Anesthesiology at the medical faculty of the University of Göttingen. In this case certain assumptions we made for the synthetic data were not valid anymore, such that we had to modify the algorithm. These modifications are described in section 6.4. Then, for the examples we chose, we were able to find the correct geometry and the approximate size and conductivity of the unknown objects.

### Organization of the chapters

In **chapter 1** we introduce the basic notations of the general EIT problem, starting from the mathematical modeling of the physical properties. Then the direct and inverse EIT problem is defined, and the chapter is concluded with a survey of those methods which have provided some ideas to the algorithm we developed. In **chapter 2** we describe a novel boundary element method for the inverse EIT problem with piecewise constant conductivities. Then in **chapter 3** we shortly describe two already existing methods which compute the inverse EIT problem on a given grid. These methods are used in our evolutionary algorithm to provide an initial guess for the boundary-element method. In **chapter 4** we give a short introduction into the general ideas and notations of evolutionary algorithms. The concept and implementation of our particular evolutionary algorithm is then described in **chapter 5**. Finally in **chapter 6** we present different numerical results for various examples of synthetic data and for different configurations to illustrate the feasibility of our evolutionary algorithm for the EIT problem. The chapter will be concluded with some ideas to deal with the difficulties for real data, and at the end we will also present an approximation for certain real data. The main results of these numerical test are summarized again in the **conclusion**, where we also give an outlook to some ideas for further investigation.

# Chapter 1

## The general EIT problem

In this first chapter we will first present the mathematical model for the real-world EIT problem from which we will then deduce the formulation of the direct and the inverse problem. This chapter will conclude with a survey of existing methods for the solution of the inverse EIT problem.

### 1.1 Mathematical modeling of the physical properties

We will limit ourselves to the *two-dimensional EIT model*, which means that we consider a cross-section of an object with infinite height. So we first define by  $\Omega \subset \mathbb{R}^2$  a bounded and simply connected domain representing a cross-section of the object under consideration. For simplicity we assume the boundary curve  $\partial\Omega$  to be  $\mathcal{C}^2$  smooth. At the surface of the object  $L$  electrodes are attached. They are considered as  $L$  closed subsets  $\mathcal{E}_\ell \subset \partial\Omega$ ,  $\ell = 1, \dots, L$  such that  $\mathcal{E}_i \cap \mathcal{E}_j = \emptyset$  for  $i \neq j$ . We denote by  $\mathcal{N} := \partial\Omega \setminus \cup \mathcal{E}_\ell$  the space between the electrodes.

The *currents* which are applied at the electrodes are given as a vector  $I \in \mathbb{R}^L$  where by  $I_\ell$  we denote the current applied at the  $\ell$ -th electrode. Analogously the *voltages* are modeled as a vector  $U \in \mathbb{R}^L$  where  $U_\ell$  is the voltage measured at the  $\ell$ -th electrode.

As the total current has to be conserved, we have

$$\sum_{\ell=1}^L I_\ell = 0, \quad (1.1)$$

and to guarantee uniqueness for  $U$  we usually also require

$$\sum_{\ell=1}^L U_\ell = 0. \quad (1.2)$$

The quantity we want to recover inside  $\Omega$  is an *isotropic electrical conductivity*. It is defined as a map  $\bar{\Omega} \rightarrow \mathbb{R}^+$  that assigns for a given point  $x = (x_1, x_2) \in \Omega$  a positive

real number. For further simplicity we assume for most of the time that  $\sigma$  is piecewise continuous inside  $\bar{\Omega}$ . We will now derive the mathematical conditions relating the conductivity to the boundary measurements.

### 1.1.1 Interior conditions: Simplifying the full Maxwell equations

In general, electromagnetic effects are described by *Maxwell's equations* [42] from which we need the following relations:

$$\operatorname{rot} E = -\frac{\partial B}{\partial t} \quad (1.3)$$

$$\operatorname{rot} H = J + \frac{\partial D}{\partial t} \quad (1.4)$$

$$E = -\operatorname{grad} u - \frac{\partial A}{\partial t} \quad (1.5)$$

where the *electric field*  $E$ , the *electric displacement*  $D$ , the *electric current*  $J$ , the *magnetic field*  $H$ , the *magnetic potential*  $A$  and the *magnetic induction*  $B$  are vector fields and the electric potential  $u$  is a scalar field. We assume that  $u$  is piecewise twice continuously differentiable inside  $\Omega$ .

Following [51] we will now simplify these equations by making some assumptions. First we assume that the conducting medium is isotropic and also *linear* which leads to the relations

$$D = \epsilon E, \quad B = \mu H, \quad J = \sigma E$$

where  $\epsilon$  is the electric permittivity and  $\mu$  the magnetic permeability. Next we assume that the injected currents are time-harmonic such that (1.3) and (1.4) can be simplified to

$$\operatorname{rot} E = -i\omega\mu H \quad (1.6)$$

$$\operatorname{rot} H = J + i\omega\epsilon E \quad (1.7)$$

where  $\omega$  is the frequency of the current. Finally at low frequencies the magnetic and capacitive effects can be neglected such that we can omit  $-i\omega\mu H$  and  $i\omega\epsilon E$  and obtain from (1.5), (1.6) and (1.7) that

$$E = -\operatorname{grad} u \quad (1.8)$$

$$\operatorname{rot} H = \sigma E \quad (1.9)$$

Taking the divergence in equation (1.8) and substituting into (1.9) we obtain the *basic EIT equation for linear isotropic medium under quasistatic conditions*

$$\operatorname{div} \sigma \operatorname{grad} u = 0 \quad \text{in } \Omega \quad (1.10)$$

which from now on we will consider as our basic equation.

For certain theoretical considerations as well as for the finite-element method in section 3.2 we need to allow *weak solutions* to (1.10).

**Definition 1.1** A function  $u \in H^1(\Omega)$  is a weak solution to the basic EIT equation (1.10) if for all test functions  $v \in C_0^\infty(\Omega)$  it satisfies the relation

$$\int_{\Omega} \sigma \operatorname{grad} u \cdot \operatorname{grad} v \, dx = 0. \quad (1.11)$$

Here the notation  $v \in C_0^\infty(\Omega)$  means that in addition to being infinitely differentiable  $v$  is equal to zero in a neighborhood of  $\partial\Omega$ .

### 1.1.2 Boundary conditions: From the continuum model to the complete electrode model

Next we have to model the relation between the applied currents and measured voltages on the boundary  $\partial\Omega$  and the conductivity  $\sigma$  inside  $\Omega$ . In all cases this relation passes via the potential function  $u$ , which is related to  $\sigma$  by equation (1.10). Following [48] we will start with the simple *continuum model*, which then will be transformed via some intermediate steps into the *complete electrode model*.

In the *continuum model* we assume that voltage and currents are continuous functions over  $\partial\Omega$ . The voltages are given as the values of the potential function  $u$  on  $\partial\Omega$ , and the currents are given via its normal derivative, because by the Gauss divergence theorem we have

$$\int_{\partial\mathcal{D}} \sigma \frac{\partial u}{\partial \nu} \, ds = \int_{\mathcal{D}} \operatorname{div} \sigma \operatorname{grad} u \, dx = 0$$

for all test domains  $\mathcal{D} \subset \Omega$ . Therefore  $\sigma \frac{\partial u}{\partial \nu}$  on  $\partial\Omega$  corresponds to the local current sources

$$\sigma \frac{\partial u}{\partial \nu} = I \text{ on } \partial\Omega.$$

This also means that the total current that is induced at the surface of  $\Omega$  has to be equal to zero such that we have the *zero flux condition*

$$\int_{\partial\Omega} \sigma \frac{\partial u}{\partial \nu} = 0 \quad (1.12)$$

which we will assume to be always fulfilled by  $\sigma$  and  $u$ . In this context we further introduce the subspaces

$$H_0^{\pm 1/2}(\partial\Omega) := \left\{ \varphi \in H^{\pm 1/2}(\partial\Omega) : \int_{\partial\Omega} \varphi \, dS = 0 \right\} \quad (1.13)$$

of the Sobolev spaces  $H^{\pm 1/2}(\partial\Omega)$ .

Now we can define

**Definition 1.2** *In the continuum model we are given two functions  $f, g$ , and the electric potential  $u$  has to fulfill the conditions (1.10), (1.12) and*

$$u = f \text{ on } \partial\Omega \quad (1.14)$$

and

$$\sigma \frac{\partial u}{\partial \nu} = g \text{ on } \partial\Omega. \quad (1.15)$$

For potential functions  $u \in H^1(\Omega)$  these boundary conditions have to be interpreted in the sense of the trace operators, such that  $f \in H^{1/2}(\partial\Omega)$  and  $g \in H_0^{-1/2}(\partial\Omega)$ .

This model is theoretically and numerically easy to work with, and for that reason we will use it for the first version of our algorithm. However it does not yet incorporate the currents  $I$  and voltages  $U$ . To do this the first step is to replace the function  $g$  by a piecewise constant function in the sense that we require

$$\sigma \frac{\partial u}{\partial \nu} = \frac{I_\ell}{|\mathcal{E}_\ell|} \text{ on } \mathcal{E}_\ell, \ell = 1, \dots, L \quad (1.16)$$

and

$$\sigma \frac{\partial u}{\partial \nu} = 0 \text{ on } \mathcal{N}, \quad (1.17)$$

where  $|\mathcal{E}_\ell|$  denotes the length of the electrode. This is then called the *gap model*. Physically condition (1.17) means that between the electrodes no current can enter or leave the object. By condition (1.16) we assume that the *total current* flowing through  $\mathcal{E}_\ell$  is equal to  $I_\ell$ , and that this current is the same at each point of  $\mathcal{E}_\ell$ .

The gap model now models the currents  $I$ , but still not the voltages  $U$ . This is done in the next step where we take into account that in practice the electric potential must be constant over a whole electrode. This leads to the condition

$$u = U_\ell \text{ on } \mathcal{E}_\ell, \ell = 1, \dots, L. \quad (1.18)$$

Furthermore the assumption that the current is the same at each point of an electrode is *not* fulfilled in practice, but we only know that the *total current* over  $\mathcal{E}_\ell$  must be equal to  $I_\ell$ . So (1.16) has to be weakened into

$$\int_{\mathcal{E}_\ell} \sigma \frac{\partial u}{\partial \nu} = I_\ell \text{ on } \mathcal{E}_\ell, \ell = 1, \dots, L. \quad (1.19)$$

The resulting model is called the *shunt model*. In [48] it has been shown that when the data which are numerically computed for that model are compared to experimental data, this model constitutes a considerable improvement compared to the previous models.

For the final *complete electrode model* (CEM) an additional effect is also taken into consideration: Electrochemical effects lead to a thin high-resistive layer between the object under investigation and the electrode surface. This creates an additional resistance, which is proportional to the current that flows at that point. Therefore equation (1.18) finally has to be replaced by

$$u + z_\ell \sigma \frac{\partial u}{\partial \nu} = U_\ell \text{ on } \mathcal{E}_\ell, \ell = 1, \dots, L,$$

where  $z_\ell$  denotes the *contact impedance* of the electrode  $\mathcal{E}_\ell$ . In our case the contact impedances are always supposed to be known, as they only depend on the measuring device and not on the interior conductivity. The shunt model can be seen as the limiting case for  $z_\ell \rightarrow 0$ . So summarizing for the CEM we have the following definition:

**Definition 1.3** *In the complete electrode model (CEM) for  $L$  electrodes we are given a voltage vector  $U \in \mathbb{R}^L$  satisfying (1.2), a current vector  $I \in \mathbb{R}^L$  satisfying (1.1) and a set of contact impedances  $z_\ell, \ell = 1, \dots, L$ .*

*The electric potential  $u \in H^1(\Omega)$  has to fulfill the conditions (1.10), (1.12) and*

$$u + z_\ell \sigma \frac{\partial u}{\partial \nu} = U_\ell \text{ on } \mathcal{E}_\ell \tag{1.20}$$

$$\int_{\mathcal{E}_\ell} \sigma \frac{\partial u}{\partial \nu} dS = I_\ell \tag{1.21}$$

$$\sigma \frac{\partial u}{\partial \nu} = 0 \text{ on } \mathcal{N} \tag{1.22}$$

Following [48] the modeling error of the CEM when comparing the numerical data to experimental data is only 0.1% such that this model seems to be a very good approximation to reality. On the other hand there are more difficulties in treating this model theoretically and numerically, as it is no longer a standard Dirichlet or Neumann problem. Especially it can be shown that

**Theorem 1.1** *In the complete electrode model the normal derivative  $\frac{\partial u}{\partial \nu}$  has singularities at the edges of the electrodes. Denoting by  $d$  the distance to an endpoint of an electrode  $\mathcal{E}_\ell$ , the behavior of these singularities in the vicinity  $\mathcal{E}_\ell$  is  $o(\frac{1}{\sqrt{d}})$  for  $z_\ell > 0$ . If  $z_\ell = 0$ , that is for the shunt model, the behavior is  $\sim \frac{1}{\sqrt{d}}$ .*

*Proof.* see [12]. □

We will see in chapter 2 that this property leads to some difficulties for the numerical implementation. In [12] even a sharper characterization of the singularities for  $z_\ell > 0$  has been proven, which however will not be exploited in our case.

## 1.2 Formulation of the direct and the inverse problem

We will now continue with the formulation of the direct and inverse EIT problem, for the continuum as well as for the complete electrode model. Although we are mainly interested in the solution of the inverse problem we also need to define the direct problem, which has to be solved for the creation of synthetic data and will also be needed during our evolutionary algorithm.

Looking at the definitions of the continuum and complete electrode model we see that in both cases we have two different boundary data which are related to each other via  $\sigma$  and  $u$ . So we can define

**Definition 1.4** *The tuple  $(f, g)$  is called a continuous boundary data pair or also a Cauchy data pair, and the tuple  $(U, I)$  is called discrete boundary data pair. One part of the tuple is called compatible with the other one if there exist some  $\sigma$  and  $u$  such that the conditions (1.14)–(1.15) respectively (1.20)–(1.22) are fulfilled.*

### 1.2.1 The direct problem

In the direct problem we are given the conductivity  $\sigma$  and one part of the boundary data pair. The task is to compute the other one such that we have

**Definition 1.5** *In the continuum model the direct Dirichlet problem is:*

*Given  $f$  and  $\sigma$ , find  $g$*

*The direct Neumann problem is:*

*Given  $g$  and  $\sigma$ , find  $f$*

*In the complete electrode model the direct Dirichlet problem<sup>1</sup> is:*

*Given  $U$  and  $\sigma$ , find  $I$*

*The direct Neumann problem is:*

*Given  $I$  and  $\sigma$ , find  $U$*

Concerning the existence of a solution to the direct problem a proof for the continuum model in the weak formulation can be found for example in [26], and for the CEM we refer to [48]. These proofs are based on the Lax-Milgram theorem. For the uniqueness we will show

**Theorem 1.2** *The direct Dirichlet problem in the continuum model has at most one solution. The solutions of the direct Neumann problem only differ by an additive constant.*

*Proof.* Let  $u_1, u_2$  be two potential functions fulfilling  $u_1 = u_2 = f$  on  $\partial\Omega$  in the Dirichlet case or  $\sigma \frac{\partial u_1}{\partial \nu} = \sigma \frac{\partial u_2}{\partial \nu} = g$  in the Neumann case.

---

<sup>1</sup>Although one usually only speaks of ‘Dirichlet’ and ‘Neumann’ problem in the continuum case we will also do this in the discrete case to have a uniform notation for both models.



Then the function  $u := u_1 - u_2$  fulfills  $u = 0$  respectively  $\sigma \frac{\partial u}{\partial \nu} = 0$  on  $\partial\Omega$ . We have

$$\begin{aligned} \int_{\Omega} \sigma |\text{grad } u|^2 dx &= \int_{\Omega} \left( \sigma |\text{grad } u|^2 + \underbrace{u \text{div } \sigma \text{grad } u}_{=0} \right) dx \\ &= \int_{\Omega} \text{div} (u \sigma \text{grad } u) dx = \int_{\partial\Omega} u \sigma \frac{\partial u}{\partial \nu} ds \end{aligned}$$

where the last equality results from Gauss divergence theorem. But in fact if either  $u = 0$  or  $\sigma \frac{\partial u}{\partial \nu} = 0$  on  $\partial\Omega$  the last expression is also equal to zero, and so we have

$$\int_{\Omega} \sigma |\text{grad } u|^2 dx = 0.$$

As  $\sigma > 0$  and  $|\text{grad } u|^2 \geq 0$  inside  $\Omega$  we conclude that  $\text{grad } u = 0$  inside  $\Omega$  from which follows that  $u \equiv \text{const}$ . In the Dirichlet case we additionally know that  $u = 0$  on  $\partial\Omega$  such that also  $u \equiv 0$  inside  $\Omega$ .  $\square$

With the same idea we can also prove uniqueness for the complete electrode model, which reads

**Theorem 1.3** *The direct Dirichlet and the direct Neumann problem in the complete electrode model have both at most one solution.*

*Proof.* Like in the proof of theorem 1.2 we denote by  $u_1, u_2$  two solutions to the direct problem, and their difference is denoted by  $u$ . We further denote by  $\tilde{U}$  the difference of the voltages obtained by inserting  $u_1$  and  $u_2$  into (1.20), and by  $\tilde{I}$  we denote the difference of the currents by inserting them into (1.21).

As in the proof of theorem 1.2 we have

$$\int_{\Omega} \sigma |\text{grad } u|^2 dx = \int_{\partial\Omega} u \sigma \frac{\partial u}{\partial \nu} ds. \quad (1.23)$$

In the Dirichlet case by construction we have that  $\tilde{U} = 0$ , so from (1.20) we obtain

$$u + \sigma z_{\ell} \frac{\partial u}{\partial \nu} = 0 \text{ on } \mathcal{E}_{\ell}. \quad (1.24)$$

Together with (1.22) this implies that we can write (1.23) as

$$\underbrace{\int_{\Omega} \sigma |\text{grad } u|^2 dx}_{\geq 0} = - \underbrace{\sum_{\ell=1}^L \frac{1}{z_{\ell}} \int_{\mathcal{E}_{\ell}} u^2 ds}_{\leq 0}. \quad (1.25)$$

So we can conclude that

$$\int_{\Omega} \sigma |\text{grad } u|^2 dx = 0.$$

As in the proof of theorem 1.2 it then follows that  $u \equiv \text{const}$  in  $\Omega$ . From (1.24) we finally see that  $u = \frac{\partial u}{\partial \nu} = 0$  on  $\partial\Omega$ , and inserting this into (1.21) yields  $\tilde{I} = 0$ . Therefore we have uniqueness.

In the Neumann case we have  $\tilde{I} = 0$  by construction, such that (1.21) leads to

$$\int_{\mathcal{E}_\ell} \sigma \frac{\partial u}{\partial \nu} ds = 0. \quad (1.26)$$

We further obtain from (1.20) that

$$u = \tilde{U}_\ell - \sigma z_\ell \frac{\partial u}{\partial \nu} \text{ on } \mathcal{E}_\ell. \quad (1.27)$$

Inserting this into (1.23) and using again (1.22) yields

$$\begin{aligned} \int_{\Omega} \sigma |\text{grad } u|^2 dx &= \sum_{\ell=1}^L \left( \int_{\mathcal{E}_\ell} \left( \tilde{U}_\ell - \sigma z_\ell \frac{\partial u}{\partial \nu} \right) \sigma \frac{\partial u}{\partial \nu} ds \right) \\ &= \sum_{\ell=1}^L \tilde{U}_\ell \int_{\mathcal{E}_\ell} \sigma \frac{\partial u}{\partial \nu} - \sum_{\ell=1}^L z_\ell \int_{\mathcal{E}_\ell} \sigma^2 \left( \frac{\partial u}{\partial \nu} \right)^2 ds. \end{aligned} \quad (1.28)$$

From (1.26) it follows that the first term in (1.28) is equal to zero. As the second term must be smaller or equal than zero, we can conclude with the same argument as for the Dirichlet case that  $u = \frac{\partial u}{\partial \nu} = 0$  on  $\partial\Omega$ . Then from (1.27) we see that  $\tilde{U} = 0$ . Therefore we have uniqueness. □

The mappings which for a given  $\sigma$  map one part of a boundary data pair onto the other part are defined as follows:

**Definition 1.6** *The operator  $\Lambda_\sigma : H_0^{-1/2}(\partial\Omega) \rightarrow H_0^{1/2}(\partial\Omega)$  mapping the Neumann data  $g$  onto the Dirichlet data  $f$  is called the Neumann-to-Dirichlet operator (NtoD-operator).*

*The matrix  $R_\sigma : \mathbb{R}^L \rightarrow \mathbb{R}^L$  is called the resistivity matrix, mapping a current pattern  $I$  onto a voltage pattern  $U$ .*

So in the continuum model the direct Neumann problem can also be formulated as applying  $\Lambda_\sigma$  to  $g$ , and the direct Dirichlet problem as applying  $\Lambda_\sigma^{-1}$  to  $f$ . In the CEM it is applying  $R_\sigma$  to  $I$  respectively  $R_\sigma^{-1}$  to  $U$ . From the boundary conditions (1.14)–(1.15) and (1.20)–(1.22) we can easily see that  $\Lambda_\sigma$  and  $R_\sigma$  are linear operators.

## 1.2.2 The inverse problem

For the inverse problem both parts of a boundary data pair are given, and we are looking for the conductivity inside  $\Omega$ . We will see in 2.1.1 that even for an already simplified case only one boundary data pair is **not** enough to uniquely determine  $\sigma$ . So we assume that we are given a multiple number of boundary data pairs such that we can formulate

**Definition 1.7** *In the continuum model the inverse EIT problem is*

*Given  $M$  boundary data pairs  $(f_k, g_k)$ ,  $k = 1, \dots, M$ , find  $\sigma$*

*In the complete electrode model the inverse EIT problem is*

*Given  $M$  boundary data pairs  $(U_k, I_k)$ ,  $k = 1, \dots, M$ , find  $\sigma$*

Concerning the uniqueness, until now it only has been proven that

**Theorem 1.4** *If the complete Neumann-to-Dirichlet map  $\Lambda_\sigma$  is known and  $\sigma \in L^\infty$  then  $\sigma$  is uniquely determined.*

*Proof.* see [2]

□

But unfortunately this question is still open for the inverse problem given only a finite number of boundary data pairs - which is always the case in practical examples. For the complete electrode model there is no uniqueness result available at all until now.

## 1.3 A survey of existing methods

To conclude this chapter we will give a short survey of already existing methods to solve the inverse EIT problem. As quite a lot of them have already been developed, we will focus on those techniques, which have provided some ideas that will be used later in our evolutionary algorithm.

An important classification is made between *iterative* and *noniterative* methods. Iterative methods start from a given approximation for  $\sigma$  and compute an update  $\delta\sigma$  for it. This step is repeated until some stopping criterion is satisfied. Noniterative methods directly compute an approximation for  $\sigma$ . Following [7] a standard way for iterative methods is trying to minimize the norm of the output data, that is, the operator norm

$$\|\Lambda_\sigma - \Lambda_{\text{meas}}\| \quad (1.29)$$

between the true NtoD-map  $\Lambda_\sigma$  and the measured map  $\Lambda_{\text{meas}}$  has to be minimized. As the full NtoD-map is not known in practice, the functional  $\|f - f_{\text{meas}}\|$  respectively  $\|U - U_{\text{meas}}\|$  has to be minimized instead. A standard way for doing this is via a Newton-type method like the one which will be described in section 3.2. Other methods can be found for example in [54], [11] or [55].

Noniterative methods use some properties of the NtoD-operator, which is then transformed in an adequate way. Usually they do not need any a-priori information about the conductivity. The factorization method presented in 3.1 is one example for this type of method. A related approach is suggested by Ikehata and Siltanen [29], and the recently introduced d-bar method [34] is using some Fourier transform inside  $\Omega$ . All these methods compute  $\sigma$  inside the whole object  $\Omega$  in one single step. The layer-stripping method proposed in [49] first computes  $\sigma$  near the surface and then assumes  $\sigma$  to be known at this outermost layer and proceeds the same way for the next inner layer until arriving at the center of the object.<sup>2</sup> We will see in 3.1.2 that the factorization method can also be modified with a similar idea to deal with multiple inclusion levels.

In the implementation of all of the methods cited above the conductivity  $\sigma$  is represented as values on a given grid. Depending on the method, this grid can be a cartesian grid or a finite-element mesh. Usually if no a-priori information about  $\sigma$  is present, the grid will be equidistant. This leads to the fact that areas of interest are discretized in the same way as areas of non-interest. Furthermore the denseness of the grid is also a mean for the regularization of the solution. So on a sparse grid the problem is usually less ill-posed than on a dense one (see [32]). Starting from these two facts the idea of *adaptive multigrid methods* as in [6] has been developed. They first perform computations on a sparse grid before passing to a dense grid to obtain more precise approximations. This dense grid is adapted on the basis of the solution found before.

An alternative to computing  $\sigma$  on a grid are *boundary element methods* (BEM). They define some closed curves inside  $\Omega$  and assume  $\sigma$  to be constant between two curves. The computation is then done on these curves. This approach first has been proposed for EIT by Hofmann [25, 26] who also used a Newton-type iterative method for the inverse solution with continuous boundary data. Recently also some BEM approaches for the complete electrode model have been proposed as in [3] or [4], where in [3] only the direct EIT problem is solved via a BEM, and in [4] the three-dimensional inverse problem is solved. For the inverse two-dimensional EIT problem with CEM boundary conditions boundary-element methods have not been tried out yet.

To distinguish between boundary element-methods and the methods where  $\sigma$  is computed on a grid, we will speak of *grid-based methods* when we refer to the latter ones. Compared to grid-based methods, boundary-element methods have the advantage that they concentrate on the regions of interest. The size of the inhomogeneities is variable, and the spatial dimension of the problem to be solved is reduced by one. On the other hand these methods in their initial form imperatively need some a-priori information about the number and position of the inhomogeneities with piecewise constant conductivities.

Finally level-set methods as in [13] are grid-based methods that also assume the conductivity to be piecewise constant. In these methods for each subdomain a level-set

---

<sup>2</sup>Even if this method uses several steps where the output of the previous step is used the method is considered as noniterative as for a given point inside  $\Omega$  it computes  $\sigma$  only once.

function is defined which is positive if the point lies inside the subdomain and negative otherwise. Compared to boundary-element methods, they have the advantage that they do not need to know beforehand how many inhomogeneities there are, as they are able to split and merge areas together.

# Chapter 2

## A novel boundary-element method

We now start with the description of the novel boundary-element method, which will be used as the main method in our algorithm. As already mentioned it is an extension of a method previously proposed for the Dirichlet boundary condition by Kress and Rundell [39]. The main difference of this method to usual Newton-type methods as in [25] is that it linearizes the integral equations on the different curves rather than the boundary-to-data map. This avoids having to solve the full forward problem in each iteration step.

In boundary-element methods for EIT the conductivity inside  $\Omega$  is considered as *piecewise constant* and the EIT problem is reduced to the *transmission problem*. We start this chapter with the definition of the transmission problem and present some considerations about uniqueness for the inverse problem for one boundary data pair. Then we describe the solution of the direct transmission problem before we proceed with the solution of the inverse problem using our boundary-element method.

### 2.1 Modeling the EIT problem as a transmission problem

For the transmission problem we denote the boundary of  $\Omega$  by  $\Gamma_0 := \partial\Omega$  and assume that  $\Gamma_1, \dots, \Gamma_N$  are  $N$  closed curves contained in  $\Omega$  satisfying  $\Gamma_i \cap \Gamma_j = \emptyset$  for  $i \neq j$ . By  $B_i$  we denote the interior of  $\Gamma_i$  for  $i = 0, \dots, N$  and set

$$D_i := B_i \setminus \bigcup_{B_j \subset B_i} B_j, \quad i = 0, \dots, N.$$

Then we define a *predecessor* function  $p : \{1, \dots, N\} \rightarrow \{0, \dots, N\}$  such that  $p(i) \neq i$  and  $\Gamma_i \subset \partial D_{p(i)}$  for  $i = 1, \dots, N$ , that is,  $\Gamma_i$  is an interior boundary curve of  $D_{p(i)}$  and an exterior boundary curve of  $D_i$ .

Figure 2.1 gives an example of a configuration with  $N = 4$  and  $p(1) = p(2) = 0$ ,  $p(3) = p(4) = 1$ . Note that as indicated in the example we allow some of the interior

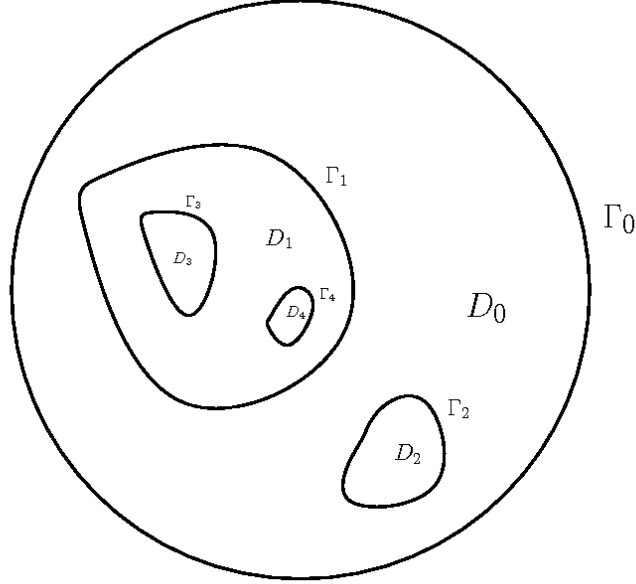


Figure 2.1: Example of a configuration.

subdomains  $D_1, \dots, D_N$  to be multiply connected, i.e., we allow a finite number of layers with different conductivities included in each other. The maximum number of layers we call the *inclusion level*. We denote by  $\nu$  the outward unit normal to each of the curves  $\Gamma_i$  for  $i = 0, \dots, N$ . Furthermore, we set

$$\Gamma := \bigcup_{i=1}^N \Gamma_i. \quad (2.1)$$

With each subdomain  $D_i$  we associate a positive constant conductivity  $\sigma_i$  and define the piecewise constant conductivity  $\sigma$  in  $D$  by setting  $\sigma|_{D_i} = \sigma_i$  for  $i = 0, \dots, N$ . So each curve  $\Gamma_i$  can be seen as an *internal interface*. We further define for each internal interface the *transmission quotient*  $\alpha_i := (\sigma_i)/(\sigma_{p(i)})$ . For the potential function we define the restrictions  $u_i := u|_{D_i}$  for  $i = 0, \dots, N$ . Analogously to (1.14) and (1.15) on  $\Gamma_0 = \partial\Omega$  we set

$$f := u|_{\Gamma_0} \text{ and } g := \sigma \frac{\partial u}{\partial \nu}|_{\Gamma_0}.$$

We note that in  $\Omega \setminus \Gamma$  we can eliminate  $\sigma$  from equation (1.10) as  $\sigma$  is constant, and we obtain the *Laplace equation*

$$\Delta u = 0 \text{ in } \Omega \setminus \Gamma. \quad (2.2)$$

Since  $u$  is continuous inside  $\Omega$ , we especially have

$$u_{p(i)} = u_i \quad \text{on } \Gamma_i, \quad i = 1, \dots, N. \quad (2.3)$$

For the normal derivatives we require

$$\sigma_{p(i)} \frac{\partial u_{p(i)}}{\partial \nu} = \sigma_i \frac{\partial u_i}{\partial \nu} \quad \text{on } \Gamma_i \quad i = 1, \dots, N, \quad (2.4)$$

because we can prove

**Theorem 2.1** *If  $u \in H^1(\Omega)$  satisfies (2.2), (2.3) and (2.4), then  $u$  is a weak solution of*

$$\operatorname{div} \sigma \operatorname{grad} u = 0 \quad \text{in } D. \quad (2.5)$$

*Proof.*  $u$  is a weak solution of (2.5) if it satisfies for all test functions  $v \in C_0^\infty(\Omega)$

$$\int_{\Omega} \sigma \operatorname{grad} u \cdot \operatorname{grad} v = 0.$$

Using Green's theorem indeed we have

$$\begin{aligned} \int_{\Omega} \sigma \operatorname{grad} u \cdot \operatorname{grad} v \, dx &= \sum_{j=0}^N \sigma_j \int_{D_j} \operatorname{grad} u \cdot \operatorname{grad} v \, dx \\ &= \sum_{j=0}^N \int_{\Gamma_j} \underbrace{\left( \sigma_{p(j)} \frac{\partial u_{p(j)}}{\partial \nu} - \sigma_j \frac{\partial u_j}{\partial \nu} \right)}_{=0} v \, ds = 0. \end{aligned}$$

□

So summarizing

**Definition 2.1** *For the direct transmission problem in all cases we are given a set  $\{\Gamma_1, \dots, \Gamma_N\}$  of interface curves with their respective conductivities  $\sigma_i$ , and we have to determine the potential function  $u$ . Also in all cases we require  $u$  to fulfill the conditions (2.2), (2.3) and (2.4) as well as the zero flux condition (1.12). Additionally*

- *For the direct Dirichlet problem in the continuum case we are given the values of the function  $f \in H_0^{1/2}(\Gamma_0)$ , we require  $u$  to fulfill  $u = f$  on  $\Gamma_0$ , and we have to determine  $g = \sigma_0 \frac{\partial u}{\partial \nu}$  on  $\Gamma_0$ .*
- *For the direct Neumann problem in the continuum case we are given the values of  $g \in H_0^{-1/2}(\Gamma_0)$ , we require  $u$  to fulfill  $\sigma_0 \frac{\partial u}{\partial \nu} = g$  on  $\Gamma_0$ , and we have to determine  $f = u$  on  $\Gamma_0$ .*
- *For the direct Dirichlet problem in the discrete (CEM) case we are given a voltage vector  $U \in \mathbb{R}^L$ , we require  $u$  to fulfill*

$$\sigma \frac{\partial u}{\partial \nu} = 0 \quad \text{on } \mathcal{N} \quad (2.6)$$



and

$$u + z_\ell \sigma \frac{\partial u}{\partial \nu} = U_\ell \text{ on } \mathcal{E}_\ell, \quad (2.7)$$

and we have to determine

$$I_\ell = \int_{\mathcal{E}_\ell} \sigma \frac{\partial u}{\partial \nu} dS. \quad (2.8)$$

- For the direct Neumann problem in the discrete case we are given a current vector  $I \in \mathbb{R}^L$ , we require  $u$  to fulfill the conditions (2.6),

$$\int_{\mathcal{E}_\ell} \sigma \frac{\partial u}{\partial \nu} dS = I_\ell \quad (2.9)$$

and

$$u + z_\ell \sigma \frac{\partial u}{\partial \nu} - U_\ell = 0 \text{ on } \mathcal{E}_\ell, \quad (2.10)$$

and we have to determine  $U$ , which is given through (2.10).

Note that in the first three cases the computation of the potential function  $u$  and the computation of the unknown part of the boundary data pair can be done separately, in the sense that we first compute  $u$ , which is then inserted into the equation that determines the unknown part of the boundary data. This is **not** possible in the last case, where condition (2.10) is needed for the computation of  $u$  as well as for the computation of  $U$ . So  $u$  and  $U$  have to be computed simultaneously. This will be relevant not only in our numerical computations, but also in the proof of existence and uniqueness of our ansatz in theorem 2.3.

The inverse problem is given as

**Definition 2.2** For the inverse transmission problem in the continuum model we are given a finite number of boundary pairs  $(f_k, g_k)$  of Dirichlet and Neumann data on  $\Gamma_0$  with  $k = 1, \dots, M$ . where the Neumann data  $g_k$  require the zero flux condition (1.12).

For the inverse transmission problem in the complete electrode model we are given a finite number of boundary pairs  $(U^k, I^k)$  of voltages and currents with  $k = 1, \dots, M$ . where the currents  $I^k$  require the zero flux condition (1.1).

In both cases the task consists in reconstructing the shape and location of the interfaces  $\Gamma_i$  and the conductivities  $\sigma_i$  for  $i = 1, \dots, N$ , such that the  $M$  potential functions  $u_k$  are required to satisfy (2.2), (2.3) and (2.4).

Note that in this formulation we assume the background conductivity  $\sigma_0$  to be known.

### 2.1.1 Uniqueness of the inverse transmission problem for one boundary data pair

To get an idea of how many data pairs we might need at least we will now look at the case  $N = M = 1$  in the continuum model, i.e. we have only one single Cauchy data pair  $(f, g)$  and one unknown subdomain with unknown conductivity. Further in this section we normalize by setting  $\sigma_0 = 1$ . So we denote by  $a := a_1 = \sigma_1$  the transmission constant of the interior interface curve. First we show that one Cauchy pair is **not** enough to uniquely determine one inclusion and its conductivity.

For this consider the simplified problem of  $\Gamma_0 = \partial D$  being the unit circle and  $\Gamma_1$  a circle with center 0 and radius  $r < 1$  and transmission constant  $a > 0$ .

As Neumann boundary data we choose  $g(\theta) = \cos(m\theta)$ ,  $\theta \in [0; 2\pi[$ ,  $m \in \mathbb{N}$ . Then the solution of the direct problem can be analytically computed from the Neumann boundary condition and the two transmission conditions (2.3) and (2.4). We obtain that the restrictions  $u_0 = u|_{D_0}$  for the electric potential inside  $D_0$  and  $u_1 = u|_{D_1}$  for the potential inside  $D_1$  can be written in the form

$$\begin{aligned} u_0 &= A\rho^m \cos(m\theta) + B\rho^{-m} \cos(m\theta) \\ u_1 &= C\rho^m \cos(m\theta) \end{aligned}$$

with  $(\rho, \theta)$  being the polar coordinates of a point in  $\bar{\Omega}$  and coefficients  $A, B, C \in \mathbb{R}$  which can be explicitly computed knowing  $r$  and  $a$ .

This implies that for the inverse problem we are given the Dirichlet data in the form  $f = (A + B) \cos m\theta$ . The coefficient  $\alpha := A + B$  is the only variable part, but one coefficient cannot be enough to determine both  $r$  and  $a$ .

With the same argument one can also prove that when two unknown concentric circles are given and only one boundary data pair is available, the problem is not uniquely solvable, even if either the radius or the conductivities are known.

Now if we assume that either the shape or the conductivity of the unknown inclusion is known, we can at least prove that the problem is unique for the unknown conductivity.

**Theorem 2.2** *In the transmission problem with one interior subdomain  $D_1$  the transmission constant  $a$  is uniquely determined by one Cauchy data pair  $(f, g)$  with  $f \neq \text{const}$  if the interior subdomain  $D_1$  is known.*

*Proof.* Let  $a_1$  and  $a_2$  be two transmission constants with the same boundary data  $f$  and  $g$ . We want to show that  $a_1 = a_2$ .

First we define functions  $u_1, u_2$  in  $D_0$  and functions  $v_1, v_2$  in  $D_1$  such that

$$u_1 = u_2 = f \text{ on } \Gamma_0 \tag{2.11}$$

$$u_1 = v_1 \text{ and } \frac{\partial u_1}{\partial \nu} = a_1 \frac{\partial v_1}{\partial \nu} \text{ on } \Gamma_1 \tag{2.12}$$

$$u_2 = v_2 \text{ and } \frac{\partial u_2}{\partial \nu} = a_2 \frac{\partial v_2}{\partial \nu} \text{ on } \Gamma_1 \quad (2.13)$$

as solutions to the direct Dirichlet problem. By assumption we have

$$\frac{\partial u_1}{\partial \nu} = \frac{\partial u_2}{\partial \nu} = g \text{ on } \Gamma_0. \quad (2.14)$$

We denote by  $w := u_1 - u_2$  and  $\tilde{w} := v_1 - v_2$  the difference of both functions. From (2.11) and (2.14) we see that  $w = (\partial w)/(\partial \nu) = 0$  on  $\Gamma_0$ . So we can apply Holmgren's theorem (see e.g. [45]) for  $w$  in  $D_0$  and conclude that  $w \equiv 0$  in  $D_0$ , such that we especially have that  $w = 0$  on  $\Gamma_1$ .

From (2.12) and (2.13) it now follows that also  $\tilde{w} = 0$  on  $\Gamma_1$ . Then as  $\tilde{w}$  is harmonic inside  $D_1$  we can conclude from the maximum principle (see for example [37]) that  $\tilde{w} \equiv 0$  in  $D_1$ , and so also  $(\partial \tilde{w})/(\partial \nu) = 0$ . This implies that

$$\frac{\partial v_1}{\partial \nu} = \frac{\partial v_2}{\partial \nu} \text{ on } \Gamma_1,$$

and so on  $\Gamma_1$  we can write

$$(a_1 - a_2) \frac{\partial v_1}{\partial \nu} = a_1 \frac{\partial v_1}{\partial \nu} - a_2 \frac{\partial v_2}{\partial \nu} = \frac{\partial u_1}{\partial \nu} - \frac{\partial u_2}{\partial \nu} = \frac{\partial w}{\partial \nu} = 0$$

which means that  $a_1 = a_2$  if  $(\partial v_1)/(\partial \nu)$  is not identically 0 on  $\Gamma_1$ .

But if  $(\partial v_1)/(\partial \nu) = 0$  on  $\Gamma_1$ , then  $v_1$  must be equal to a constant in  $D_1$  (see [37]). We denote this constant by  $c$ . We particularly have that  $v_1 = c$  on  $\Gamma_1$ . From the transmission conditions (2.12) it then follows that on  $\Gamma_1$  we have  $u_1 = c$  and  $(\partial u_1)/(\partial \nu) = 0$ .

Next we define a function  $\hat{u} := u_1 - c$ . This function is 0 on  $\Gamma_1$  as well as its normal derivative. So we can apply Holmgren's theorem in  $D_0$  and conclude that  $\hat{u} \equiv 0$  in  $D_0$  such that especially on  $\Gamma_0$  we have

$$f - c = u_1 - c = \hat{u} = 0.$$

But this implies that  $f$  is constant, which is a contradiction to the assumption of the theorem. This means that  $(\partial v_1)/(\partial \nu)$  cannot be identically 0, and so we must have  $a_1 = a_2$ .  $\square$

For an unknown shape with known conductivities there is no general result available until now. Partial results requiring some restrictions on the shape can be found in [19], [30] and [1]. Also note that the proof for the transmission constant cannot be generalized for multiple subdomains in an obvious way.

## 2.2 Solution of the direct problem

We now start with the description of the method for the solution of the direct problem where we begin with the solution of the direct Neumann problem in the continuum model. Afterwards we describe the modifications for the other direct problems.

### 2.2.1 Solution of the direct problem for the continuum model

To find a function that for given interface curves  $\Gamma_1, \dots, \Gamma_N$ , given conductivities  $\sigma_1, \dots, \sigma_N$  and given Neumann boundary data  $g$  satisfies the conditions (2.2), (2.3) and (2.4) we choose the single-layer approach

$$u(x) := \sum_{j=0}^N \int_{\Gamma_j} \Phi(x, y) \varphi(y) ds(y) \quad (2.15)$$

with an unknown density function  $\varphi$ . Here

$$\Phi(x, y) := -\frac{1}{2\pi} \ln |x - y|, \quad x \neq y \quad (2.16)$$

is the fundamental solution of the Laplace equation (2.2) in  $\mathbb{R}^2$ .

From the conditions (1.15) and (2.4) which  $u$  has to fulfill, we then derive a system of linear integral equations. On the exterior boundary curve  $\Gamma_0$  equation (1.15) is satisfied by the approach (2.15) if

$$\sum_{j=0}^N \int_{\Gamma_j} \frac{\partial \Phi(x, y)}{\partial \nu(x)} \varphi(y) ds(y) + \frac{1}{2} \varphi(x) = \frac{1}{\sigma_0} g(x), \quad x \in \Gamma_0, \quad (2.17)$$

where  $\frac{1}{2} \varphi(x)$  arises from the jump of the normal derivative of the single-layer potential on  $\Gamma_0$  when the limit coming from inside  $D_0$  is taken. The jumps occurring on each interior interface curve  $\Gamma_i$  lead to

$$\frac{\partial u_+}{\partial \nu}(x) = \sum_{j=0}^N \int_{\Gamma_j} \frac{\partial \Phi(x, y)}{\partial \nu(x)} \varphi(y) ds(y) - \frac{1}{2} \varphi(x), \quad x \in \Gamma_i \quad (2.18)$$

and

$$\frac{\partial u_-}{\partial \nu}(x) = \sum_{j=0}^N \int_{\Gamma_j} \frac{\partial \Phi(x, y)}{\partial \nu(x)} \varphi(y) ds(y) + \frac{1}{2} \varphi(x), \quad x \in \Gamma_i. \quad (2.19)$$

Setting  $a_i := \sigma_i / \sigma_{p(i)}$  and using (2.18) and (2.19) shows that condition (2.4) is satisfied provided

$$(1 - a_i) \sum_{j=0}^N \int_{\Gamma_j} \frac{\partial \Phi(x, y)}{\partial \nu(x)} \varphi(y) ds(y) - \frac{1 + a_i}{2} \varphi(x) = 0, \quad x \in \Gamma_i. \quad (2.20)$$

The equations (2.17) and (2.20) now constitute a system of integral equations of the second kind which can be solved for the unknown  $\varphi$ . Because of the continuity of the

single-layer potential over all interface curves  $\Gamma_i$  the condition (2.3) is fulfilled for any  $\varphi$ .

To write the system in operator form we define the *derivative of the single-layer potential* as

$$(K'_{ij})\varphi(x) := \int_{\Gamma_j} \frac{\partial \Phi(x, y)}{\partial \nu(x)} \varphi(y) ds(y), \quad x \in \Gamma_i. \quad (2.21)$$

Then we can write the system as

$$\frac{1}{2}Id + \begin{pmatrix} K'_{00} & K'_{01} & \dots & K'_{0N} \\ -\frac{1-a_1}{1+a_1} K'_{10} & -\frac{1-a_1}{1+a_1} K'_{11} & \dots & -\frac{1-a_1}{1+a_1} K'_{1N} \\ \vdots & \vdots & \ddots & \vdots \\ -\frac{1-a_N}{1+a_N} K'_{N0} & -\frac{1-a_N}{1+a_N} K'_{N1} & \dots & -\frac{1-a_N}{1+a_N} K'_{NN} \end{pmatrix} \begin{pmatrix} \varphi_0 \\ \vdots \\ \varphi_N \end{pmatrix} = \begin{pmatrix} g/\sigma_0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad (2.22)$$

where we denote by  $\varphi_i := \varphi|_{\Gamma_i}$  the density function restricted to the curve  $\Gamma_i$  and by  $Id$  the identity matrix<sup>1</sup>. In [37] it is shown that the kernel of  $K' := K'_{00}$  is weakly singular, such that the *Riesz theory* can be applied. Further it is proven that the nullspace of the operator  $1/2Id + K'$  has dimension one. Then following [26] one can show that the same is true for the nullspace of the system (2.22) and that the system is uniquely solvable up to an additive constant.

To obtain a unique solution we additionally extend the system (2.22) with the condition

$$\int_{\Gamma_0} \varphi(y) ds(y) = 0, \quad (2.23)$$

which can be deduced from the zero flux condition (1.12) (see [37]).

The density function  $\varphi$  can then be inserted into the ansatz (2.15) to obtain the potential function  $u$ . To write this in operator form we define the *single-layer operator* as

$$(S_{ij})\varphi(x) := \int_{\Gamma_j} \Phi(x, y) \varphi(y) ds(y), \quad x \in \Gamma_i \quad (2.24)$$

such that  $f$  can be written as

$$f = \sum_{i=0}^N S_{i0} \varphi_i.$$

For the solution of the direct Dirichlet problem we replace the single-layer approach (2.15) on  $\Gamma_0$  by a double-layer approach. This leads to

$$u(x) := \int_{\Gamma_0} \frac{\partial \Phi(x, y)}{\partial \nu(y)} \varphi(y) ds(y) + \sum_{j=1}^N \int_{\Gamma_j} \Phi(x, y) \varphi(y) ds(y). \quad (2.25)$$

---

<sup>1</sup>to avoid notational confusion with the current  $I$

The operators which are needed are the *double-layer potential*

$$(K_{ij}\varphi)(x) := \int_{\Gamma_j} \frac{\partial\Phi(x,y)}{\partial\nu(y)} \varphi(y) ds(y), \quad x \in \Gamma_i \quad (2.26)$$

and its derivative

$$(T_{ij}\varphi)(x) := \int_{\Gamma_j} \frac{\partial^2\Phi(x,y)}{\partial\nu(x)\partial\nu(y)} \varphi(y) ds(y), \quad x \in \Gamma_i. \quad (2.27)$$

The system to be solved is deduced in the same way as for the Neumann problem and can be written as

$$\frac{1}{2}Id - \begin{pmatrix} K_{00} & S_{01} & \cdots & S_{0N} \\ \frac{1-a_1}{1+a_1}T_{10} & \frac{1-a_1}{1+a_1}K'_{11} & \cdots & \frac{1-a_1}{1+a_1}K'_{1N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1-a_N}{1+a_N}T_{N0} & \frac{1-a_N}{1+a_N}K'_{N1} & \cdots & \frac{1-a_N}{1+a_N}K'_{NN} \end{pmatrix} \begin{pmatrix} \varphi_0 \\ \vdots \\ \varphi_N \end{pmatrix} = \begin{pmatrix} f \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad (2.28)$$

Following [37] the operator  $1/2Id - K$  has zero-dimensional nullspace, and with the same argument from [26] as for the Neumann case one can show that this is also the case for the nullspace of the system (2.28). So from the Riesz theory it follows that (2.28) is uniquely solvable.

The obtained density function  $\varphi$  is finally inserted into the normal derivative of (2.25) which leads to

$$g = \sigma_0 \left( T_{00}\varphi_0 + \sum_{i=1}^N K'_{i0}\varphi_i \right).$$

### 2.2.1.1 Numerical discretization

For the numerical solution the system (2.22) has to be discretized, and therefore a parameterization is required. For  $j = 0, \dots, N$  we parameterize each of the closed curves in the form

$$\Gamma_j := \{z_j(t) : t \in [0; 2\pi[ \} \quad (2.29)$$

with  $2\pi$  periodic  $C^2$ -smooth functions  $z_j : \mathbb{R} \rightarrow \mathbb{R}^2$  such that  $z_j$  is injective on  $[0; 2\pi[$ . In particular this implies that  $z'_j(t) \neq 0$  for all  $t \in [0; 2\pi[$ . We further set  $\psi(t) := \varphi(z(t))$ . Then the parameterized form of the single-layer operator (2.24) is given by

$$\tilde{S}(\psi, z)(t) = -\frac{1}{2\pi} \int_0^{2\pi} \ln |z(t) - z(\tau)| |z'(\tau)| \psi(\tau) d\tau, \quad t \in [0; 2\pi[, \quad (2.30)$$

and the parameterized form of its normal derivative (2.21) is given by

$$\tilde{K}'(\psi, z)(t) = \frac{1}{2\pi} \int_0^{2\pi} \frac{[z'(t)]^\perp \cdot [z(t) - z(\tau)] |z'(\tau)|}{|z(t) - z(\tau)|^2 |z'(t)|} \psi(\tau) d\tau, \quad t \in [0; 2\pi[. \quad (2.31)$$

Here, we used the notation  $a^\perp = (a_2, -a_1)$  for any vector  $a = (a_1, a_2)$ , that is,  $a^\perp$  is obtained by rotating  $a$  clockwise by 90 degrees. The dependence of the operators on the boundary  $\Gamma$  through the parameterization  $z$  is indicated through the notation  $\tilde{S}(\cdot, z)$  and  $\tilde{K}'(\cdot, z)$ .

It can be verified that the kernel function in (2.21) is smooth for  $t = \tau$  with diagonal term

$$\frac{[z'(t)]^\perp \cdot z''(t)}{4\pi |z'(t)|^2} \quad (2.32)$$

so that the system (2.22) does not contain any singularity to be treated. The same is true for the system which has to be inverted for the solution of the Dirichlet problem. There the double-layer potential (2.26) is parameterized as

$$\tilde{K}(\psi, z)(t) = \frac{1}{2\pi} \int_0^{2\pi} \frac{[z'(\tau)]^\perp \cdot [z(t) - z(\tau)]}{|z(t) - z(\tau)|^2} \psi(\tau) d\tau, \quad t \in [0; 2\pi[. \quad (2.33)$$

Its kernel is also smooth for  $t = \tau$  with the same diagonal term (2.32).

So the operators in (2.22) can be discretized by the usual Nyström method (see e.g. [37]) using the trapezoidal rule and equidistant discretization points.

However the kernels of the operators  $S_{00}$  and  $T_{00}$  used for the computation of  $g$  respectively  $f$  for a given  $\varphi$  are not smooth.  $S_{00}$  is a singular operator, and  $T_{00}$ , parameterized as

$$\tilde{T}(\psi, z)(t) = \frac{1}{2\pi} \int_0^{2\pi} \left\{ \frac{[z'(t)]^\perp \cdot [z'(\tau)]}{|z'(t)| |z(t) - z(\tau)|^2} - 2 \frac{[z'(\tau)]^\perp \cdot [z(t) - z(\tau)] [z'(t)]^\perp \cdot [z(t) - z(\tau)]}{|z'(t)| |z(t) - z(\tau)|^4} \right\} \psi(\tau) d\tau, \quad t \in [0; 2\pi[.$$

is even hypersingular. These singularities - respectively hypersingularities - of  $S_{00}$  and  $T_{00}$  have to be treated by splitting them off as described in [38].

### 2.2.2 Solution for the complete electrode model

For the complete electrode model we slightly have to modify the ansatz (2.15) for  $u$ . For notational convenience we define

$$(M\varphi)(x) := \begin{cases} \varphi(x) - \frac{1}{|\Gamma_0|} \int_{\Gamma_0} \varphi(y) ds(y), & x \in \Gamma_0 \\ \varphi(x), & x \in \Gamma_i, i = 1, \dots, N \end{cases} \quad (2.34)$$

and we use the ansatz

$$u(x) := \sum_{j=0}^N \int_{\Gamma_j} \Phi(x, y) (M\varphi)(y) ds(y) + \frac{1}{|\Gamma_0|} \int_{\Gamma_0} \varphi(y) ds(y) \quad (2.35)$$

instead of (2.15). Then the boundary conditions (1.20), (1.21) and (1.22) are satisfied provided

$$\begin{aligned} & \sum_{j=0}^N \int_{\Gamma_j} \left( \Phi(x, y) + z_\ell \sigma_0 \frac{\partial \Phi(x, y)}{\partial \nu(x)} \right) (M\varphi)(y) ds(y) \\ & + z_\ell \sigma_0 \frac{1}{2} (M\varphi)(x) - \frac{1}{|\Gamma_0|} \int_{\Gamma_0} \varphi(y) ds(y) = U_\ell, \quad x \in \mathcal{E}_\ell, \end{aligned} \quad (2.36)$$

$$\sigma_0 \int_{\mathcal{E}_\ell} \left( \sum_{j=0}^N \int_{\Gamma_j} \frac{\partial \Phi(\hat{x}, y)}{\partial \nu(\hat{x})} (M\varphi)(y) ds(y) + \frac{1}{2} (M\varphi)(\hat{x}) \right) ds(\hat{x}) = I_\ell, \quad \ell = 1, \dots, L \quad (2.37)$$

and

$$\sum_{j=0}^N \int_{\Gamma_j} \frac{\partial \Phi(x, y)}{\partial \nu(x)} (M\varphi)(y) ds(y) + \frac{1}{2} (M\varphi)(x) = 0, \quad x \in \mathcal{N}. \quad (2.38)$$

On the interior interfaces we have to replace  $\varphi$  by  $M\varphi$  in (2.20), and we obtain for  $i = 1, \dots, N$

$$(1 - a_i) \sum_{j=0}^N \int_{\Gamma_j} \frac{\partial \Phi(x, y)}{\partial \nu(x)} (M\varphi)(y) ds(y) - \frac{1 + a_i}{2} (M\varphi)(x) = 0, \quad x \in \Gamma_i. \quad (2.39)$$

Following definition 2.1 and the considerations made afterwards we solve the *discrete Dirichlet problem* by inverting the system consisting of (2.36), (2.38) and (2.39) to obtain the density function  $\varphi$ . We then insert it into (2.37) to obtain the currents  $I$ . For the solution of the *discrete Neumann problem* indeed we invert the system consisting of (2.36)–(2.39) to obtain simultaneously the values for  $\varphi$  and  $U$ . In both cases the system has exactly one solution, as we will prove now



**Theorem 2.3** *For the ansatz (2.35) the direct discrete Dirichlet and Neumann problem in the complete electrode model has exactly one solution.*

*Proof.* We start with the discrete Dirichlet case. To write the system described through (2.36), (2.38) and (2.39) in operator form we introduce the piecewise constant functions  $\chi, V$  on  $\Gamma_0$  defined as

$$\chi := \begin{cases} \frac{1}{\sigma_0 z_\ell} & \text{on } \mathcal{E}_\ell, \ell = 1, \dots, L \\ 0 & \text{on } \mathcal{N} \end{cases}$$

and

$$V := \begin{cases} \frac{U_\ell}{\sigma_0 z_\ell} & \text{on } \mathcal{E}_\ell, \ell = 1, \dots, L \\ 0 & \text{on } \mathcal{N}. \end{cases}$$

Next we define operators  $R_i : L^2(\Gamma_i) \rightarrow L^2(\Gamma_0)$  for  $i = 0, \dots, N$  by

$$R_i := K'_{0i}M + \chi S_{0i}M.$$

These operators are compact, as the operators  $K'_{0i}$  and  $S_{0i}$  are compact and the operators  $\chi$  and  $M$  are bounded. If we have no inclusions inside  $\Omega$  we can then write (2.36) and (2.38) as

$$\frac{1}{2}M\varphi + R_0\varphi + \chi \frac{1}{|\Gamma_0|} \int_{\Gamma_0} \varphi ds = V. \quad (2.40)$$

By defining the operator  $\tilde{R}_0 : L^2(\Gamma_0) \rightarrow L^2(\Gamma_0)$  as

$$\tilde{R}_0\varphi := R_0\varphi + \left(\chi - \frac{1}{2}\right) \frac{1}{|\Gamma_0|} \int_{\Gamma_0} \varphi ds$$

equation (2.40) can be transformed into

$$\frac{1}{2}Id\varphi + \tilde{R}_0\varphi = V$$

where  $\tilde{R}_0$  is still compact, as  $R_0$  is compact and any integral operator with constant kernel is compact and  $\chi$  is bounded.

For the inner inclusions we extend the system in a similar way as for the continuum case. We introduce the operator  $\mathcal{R} : L^2(\cup\Gamma_i) \rightarrow L^2(\cup\Gamma_i)$  as

$$\mathcal{R} := \begin{pmatrix} \tilde{R}_0 & R_1 & \dots & R_N \\ -\frac{1-a_1}{1+a_1}K'_{10} & -\frac{1-a_1}{1+a_1}K'_{11} & \dots & -\frac{1-a_1}{1+a_1}K'_{1N} \\ \vdots & \vdots & \ddots & \vdots \\ -\frac{1-a_N}{1+a_N}K'_{N0} & -\frac{1-a_N}{1+a_N}K'_{N1} & \dots & -\frac{1-a_N}{1+a_N}K'_{NN} \end{pmatrix}. \quad (2.41)$$

Then we can write the system consisting of (2.36), (2.38) and (2.39) as

$$\frac{1}{2}Id\varphi + \mathcal{R}\varphi = \tilde{V} \quad (2.42)$$

where  $\tilde{V}$  is the extension of  $V$  onto the interior curves filled up with zeros. The operator  $\mathcal{R}$  is compact, as all its parts are compact. Therefore the Riesz theory can be applied to (2.42), such that injectivity of the operator  $\frac{1}{2}Id + \mathcal{R}$  implies existence and uniqueness of a solution.

We note that the operator  $\mathcal{R}$  maps  $L^2(\cup\Gamma_i)$  onto the set of piecewise continuous functions on  $\cup\Gamma_i$ . This follows from the fact that the operators  $K'_{ij}$  have continuous kernels, such that they map  $L^2(\Gamma_j)$  onto  $\mathcal{C}(\Gamma_i)$  (see [37]). For the operators  $\tilde{R}_0, R_i$  ( $i = 1, \dots, N$ ) we refer to [23], where it is proven that the single-layer potential  $S$  - which has a weakly singular kernel - also maps  $L^2(\Gamma)$  onto  $\mathcal{C}(\Gamma_i)$ . So the operators  $\chi S_{0i}$  and therefore also the operators  $\tilde{R}_0, R_i$  map  $L^2(\Gamma_i)$  onto the set of piecewise continuous functions on  $\Gamma_0$ . This particularly implies that the jump relations of the potential operators remain valid on all  $\Gamma_i, i = 0, \dots, N$  in our case.

So if we now consider the homogeneous equation to (2.42) we especially see that by construction  $u$  defined by (2.35) solves the homogeneous CEM-Dirichlet problem. Using theorem 1.3 we then conclude that  $u = 0$  inside  $\Omega$ . Therefore also its normal derivative on the interior curves are zero. From the jump relations of the operator  $K'$  we obtain

$$\frac{\partial u_+}{\partial \nu} - \frac{\partial u_-}{\partial \nu} = -\varphi \text{ on } \Gamma_i, i = 1, \dots, N, \quad (2.43)$$

and therefore we must have  $\varphi = 0$  on all interior curves.

As furthermore the single-layer potential does not have any jump relations we conclude that for all  $x \in \mathbb{R}^2$  we can write  $u$  as

$$u(x) = \int_{\Gamma_0} \Phi(x, y)(M\varphi)(y)ds(y) + \frac{1}{|\Gamma_0|} \int_{\Gamma_0} \varphi ds.$$

From [37] we know that  $u$  tends to zero at infinity, so that we have uniqueness for the exterior Dirichlet problem. This implies that  $u = 0$  in  $\mathbb{R}^2$ , and like in (2.43) we conclude from the jump relations of the normal derivatives that  $\varphi = 0$  on  $\Gamma_0$ . Therefore from the Riesz theory it follows existence and uniqueness of a solution for the discrete Dirichlet case.

For the discrete Neumann case we have to consider the system consisting of (2.36)–(2.39), which has to be solved simultaneously for  $\varphi$  and  $U$ . Therefore we extend the operator  $\mathcal{R} : L^2(\cup\Gamma_i) \rightarrow L^2(\cup\Gamma_i)$  defined through (2.41) into an operator  $\mathcal{R}^\diamond : L^2(\cup\Gamma_i) \times \mathbb{R}^L \rightarrow L^2(\cup\Gamma_i) \times \mathbb{R}^L$  such that the equations (2.36)–(2.39) are equivalent to

$$\frac{1}{2}Id \begin{pmatrix} \varphi \\ U \end{pmatrix} + \mathcal{R}^\diamond \begin{pmatrix} \varphi \\ U \end{pmatrix} = \begin{pmatrix} 0 \\ I \end{pmatrix} \quad (2.44)$$

with a compact operator  $\mathcal{R}^\diamond$ . So the Riesz theory can be applied to (2.44). Now we can proceed like in the Dirichlet case. When we consider the homogeneous equation of (2.44) we particularly see that in that case by construction the function  $u$  defined by (2.35) solves the homogeneous CEM-Neumann problem. From theorem 1.3 we again conclude that  $u = 0$  inside  $\Omega$ , and with the same argument as for the Dirichlet case we then obtain that  $\varphi = 0$ . Finally equation (2.36) yields that also  $U = 0$ . This implies existence and uniqueness of a solution for the discrete Neumann case.  $\square$

Note that as in the continuum case the density function  $\varphi$  can be inserted into (2.15) or its derivative to obtain the Dirichlet and Neumann data  $f$  and  $g$ , which still have the same meaning as in the continuum model. This will be useful when we proceed with the system for the solution of the inverse problem.

### 2.2.2.1 Numerical discretization

For the numerical solution of the direct CEM problem we additionally require  $\varphi$  to fulfill the condition

$$\int_{\Gamma_0} \varphi ds = 0,$$

so that  $M\varphi = \varphi$  and the representations (2.35) and (2.15) coincide. So we can write  $u$  again in the form (2.15).

From [12] we know that  $g$  has singularities at the endpoints of each electrode, but that nevertheless the integral (2.37) is finite. By expanding the representation  $(I + 2K')\varphi = 2g$  into a Neumann series (see [37]) we see that the density  $\varphi$  must have singularities at the same points as  $g$ . So following [36], to discretize the integrals in (2.36)–(2.38) we have to choose a non-equidistant discretization of  $\Gamma_0$  which cancels out the singularities in the integral.

For this we are looking for a strictly increasing bijective  $\mathcal{C}^\infty$  substitution function  $\omega : [0; 2\pi] \rightarrow [0; 2\pi]$  fulfilling for all  $\tilde{t} \in \{t \mid g(z_0(t)) \text{ has a singularity}\}$  and some  $p \in \mathbb{N}$

$$\omega^{(j)}(\tilde{t}) = 0, \quad j = 1, \dots, p-1 \quad (2.45)$$

$$\omega^{(p)}(\tilde{t}) \neq 0. \quad (2.46)$$

If we now have an integral

$$\int_{\Gamma_0} K(x, y)\varphi(y)ds(y)$$

with some smooth kernel  $K(x, y)$  and a singular density  $\varphi$ , the parameterized form

$$\int_0^{2\pi} K(z_0(t), z_0(\tau))|z_0'(\tau)|\psi(\tau)d\tau$$

is replaced via the substitution rule by

$$\int_0^{2\pi} K(z_0(\omega(t)), z_0(\omega(\tau))) |z'_0(\omega(\tau))\omega'(\tau)| \psi(\omega(\tau)) d\tau, \quad (2.47)$$

and we expect that because of property (2.45) the singularities in the substituted integral (2.47) cancel out by multiplying  $\psi(\omega(\tau))$  with  $\omega'(\tau)$ . From the result cited in theorem 1.1 we see that a function satisfying for all  $\tilde{t}$

$$\lim_{\substack{z_0(t) \in \mathcal{E}_\ell \\ t \rightarrow \tilde{t}}} \frac{\omega'(t)}{\sqrt{|t - \tilde{t}|}} = 0 \quad (2.48)$$

cancels out the singularities. Note that from a theoretical point of view property (2.46) is not necessary, but following [37] using substitution functions where all derivatives are zero easily leads to numerical underflow.

The substituted integral can then be numerically computed with the usual trapezoidal rule. Note that also in the kernels for the derivatives of  $z_0$  we have to use the chain rule. Therefore for  $z = z_0$  (2.31) now becomes

$$\tilde{K}'(\psi, z_0)(\omega(t)) = \frac{1}{2\pi} \int_0^{2\pi} \frac{[z'_0(\omega(t))]^\perp \cdot [z_0(\omega(t)) - z_0(\omega(\tau))] |z'_0(\omega(\tau))\omega'(\tau)|}{|z_0(\omega(t)) - z_0(\omega(\tau))|^2 |z'_0(\omega(t))\omega'(t)|} \psi(\omega(\tau)) d\tau,$$

with diagonal term

$$\frac{[z'_0(\omega(t))]^\perp \cdot [z''_0(\omega(t))(\omega'(t))^2 + z'_0(\omega(t))\omega''(t)]}{4\pi |z'_0(\omega(t))|^2 |\omega'(t)|}.$$

## 2.3 Solution of the inverse problem

We now proceed with the solution of the inverse problem. The basic idea is to use Green's formula (see e.g. [37]) to deduce relations between integrals over the exterior boundary curve  $\Gamma_0$  and integrals over the interior interface curves  $\Gamma_i$ . We start with the solution for the continuum model as also described in [14]. Then we describe the extensions for the complete electrode model, which are similar to the extensions for the solution of the forward problem.

Further we will see that the system which is used for the solution of the inverse problem can also be applied for the solution of the direct problem.<sup>2</sup> However, to avoid an inverse crime when creating the synthetic data we deliberately solved the direct problem by the approach described in section 2.2.

<sup>2</sup>In fact, a quite similar approach has been used in [3] to do this.

### 2.3.1 Deriving a system of integral equations

In terms of the Cauchy data  $(f, g)$  on  $\Gamma_0$  and with  $\Phi(x, y)$  being the fundamental solution (2.16) of the Laplace equation we define the combined single- and double-layer potential

$$w(f, g)(x) := \int_{\Gamma_0} \left\{ \sigma_0 f(y) \frac{\partial \Phi(x, y)}{\partial \nu(y)} - g(y) \Phi(x, y) \right\} ds(y), \quad x \in \mathbb{R}^2 \setminus \Gamma_0. \quad (2.49)$$

For notational convenience we introduce the quotient

$$\mu_i := \frac{\sigma_{p(i)} + \sigma_i}{\sigma_{p(i)} - \sigma_i}, \quad i = 1, \dots, N, \quad (2.50)$$

and define the function  $\mu : \Gamma \rightarrow \mathbb{R}$  by setting  $\mu|_{\Gamma_i} := \mu_i$  for  $i = 1, \dots, N$ . Further we define the *weighted traces*  $h_i$  of  $u$  on  $\Gamma_i$  by

$$h_i := (\sigma_{p(i)} - \sigma_i)u|_{\Gamma_i}. \quad (2.51)$$

Recalling that  $\Gamma$  is defined by (2.1) as the union of all interior interfaces, we define the double-layer operators

$$(K_0 h)(x) := \int_{\Gamma} \frac{\partial \Phi(x, y)}{\partial \nu(y)} h(y) ds(y), \quad x \in \Gamma_0, \quad (2.52)$$

and

$$(K h)(x) := \int_{\Gamma} \frac{\partial \Phi(x, y)}{\partial \nu(y)} h(y) ds(y), \quad x \in \Gamma. \quad (2.53)$$

With these notations we will now prove

**Theorem 2.4** *If the piecewise harmonic function  $u$  in  $\Omega$  satisfies the transmission conditions (2.3) and (2.4) and has boundary values  $u = f$  and  $\sigma_0 \frac{\partial u}{\partial \nu} = g$  on  $\Gamma_0$  then the function  $h$  on  $\Gamma$  given by the traces  $h|_{\Gamma_i} := h_i$ ,  $i = 1, \dots, N$ , satisfies the integral equations*

$$\begin{aligned} K_0 h &= w_+(f, g) \quad \text{on } \Gamma_0, \\ -\frac{1}{2} \mu h + K h &= w(f, g) \quad \text{on } \Gamma, \end{aligned} \quad (2.54)$$

where the subscript indicates that  $w_+(f, g)$  is the limit obtained when approaching  $\Gamma_0$  from outside  $\Omega$ . Conversely, if  $h$  satisfies the integral equations (2.54) then

$$u(x) := \frac{1}{\sigma(x)} \left\{ \int_{\Gamma} \frac{\partial \Phi(x, y)}{\partial \nu(y)} h(y) ds(y) - w(f, g)(x) \right\}, \quad x \in \Omega \setminus \Gamma, \quad (2.55)$$

solves the transmission problem and has boundary values  $u = f$  and  $\sigma_0 \frac{\partial u}{\partial \nu} = g$  on  $\Gamma_0$ .

*Proof.* The function  $v := \sigma u$  clearly is piecewise harmonic in  $\Omega$ . Its normal derivatives are continuous across  $\Gamma_i$  and its jumps are given by

$$v_{p(i)} - v_i = h_i \quad \text{on } \Gamma_i$$

for  $i = 1, \dots, N$ . Therefore applying Green's integral theorem to  $v$  and the fundamental solution  $\Phi(x, \cdot)$  over each of the subdomains  $D_j$  and summing up, we find

$$v(x) = \int_{\Gamma} h(y) \frac{\partial \Phi(x, y)}{\partial \nu(y)} ds(y) - w(f, g)(x), \quad x \in \Omega \setminus \Gamma. \quad (2.56)$$

Letting  $x$  tend to  $\Gamma_0$  and observing the jump relations for the double-layer potentials we obtain the first equation of (2.54). Analogously letting  $x$  tend to  $\Gamma_i$  from inside  $D_i$  we obtain the second equation of (2.54).

Conversely, we assume that the functions  $h_j$  satisfy the integral equations (2.54) and define the piecewise harmonic function  $u$  via (2.2) in all of  $\mathbb{R}^2 \setminus (\Gamma \cup \Gamma_0)$ . Then, for  $v := \sigma u$  the first integral equation of (2.54) implies that for the limit  $v_+$  when approaching  $\Gamma_0$  from outside  $\Omega$  we have  $v_+ = 0$  on  $\Gamma_0$ . From the flux condition (1.12) it can be deduced that  $v$  is bounded at infinity. Therefore, from the uniqueness for the exterior Dirichlet problem it follows that  $v = 0$  in  $\mathbb{R}^2 \setminus \Omega$ . The jump relations for single- and double-layer potentials then imply that for the limits when approaching from inside  $\Omega$  we have

$$v_- = \sigma_0 f \quad \text{and} \quad \frac{\partial v_-}{\partial \nu} = g \quad \text{on } \Gamma_0. \quad (2.57)$$

Furthermore, from the jump relations for double-layer potentials, we deduce

$$\sigma_i v_{p(i)} - \sigma_{p(i)} v_i = -(\sigma_{p(i)} - \sigma_i) \left( [Kh - w] + \frac{1}{2} \mu_i h_i \right) \quad \text{on } \Gamma_i,$$

and therefore, in view of the second integral equation of (2.54), it follows that

$$\sigma_i v_{p(i)} - \sigma_{p(i)} v_i = 0 \quad \text{on } \Gamma_i. \quad (2.58)$$

Finally, since the normal derivative of the double-layer potential is continuous across  $\Gamma_i$  we have

$$\frac{\partial v_{p(i)}}{\partial \nu} - \frac{\partial v_i}{\partial \nu} = 0 \quad \text{on } \Gamma_i. \quad (2.59)$$

Summarizing, (2.57)–(2.59) imply that that  $u = v/\sigma$  fulfills the transmission conditions (2.3) and (2.4) and has boundary values  $u = f$  and  $\sigma_0 \frac{\partial u}{\partial \nu} = g$  on  $\Gamma_0$ .  $\square$

This system can be used for the solution of the direct problem with given interface curves  $\Gamma$  and conductivities  $\sigma$ . Then the system is linear with respect to the unknown functions  $h$  and also to the boundary data  $f$  and  $g$ . On the other hand - and this is our purpose - it can be used for the solution of the inverse problems, as it can be easily seen that for multiple boundary data pairs we have

**Corollary 2.5** *Assume that  $(f_k, g_k)$ ,  $k = 1, \dots, M$ , are compatible Cauchy pairs for the transmission problem. Then solving the inverse transmission problem in the continuum model with these data is equivalent to solving the system of integral equations*

$$\begin{aligned} K_0 h_k &= w_+(f_k, g_k) \quad \text{on } \Gamma_0, \\ -\frac{1}{2} \mu h_k + K h_k &= w(f_k, g_k) \quad \text{on } \Gamma. \end{aligned} \tag{2.60}$$

*These equations have to be satisfied for  $k = 1, \dots, L$  and the unknowns are the interface curves  $\Gamma$ , the conductivity  $\sigma$  (given through  $\mu$ ) and the densities  $h_k$ ,  $k = 1, \dots, M$ .*

Now, for the solution of the inverse problem, the system is not linear anymore with respect to all unknowns. Especially the interfaces  $\Gamma$  enter the system (2.60) through the double-layer boundary integral operators  $K_0$  and  $K$  in a nonlinear fashion such that the equations need to be linearized with respect to  $\Gamma$ .

Finally note that due to the lack of a uniqueness result for a finite number of Cauchy pairs we also do not have unique solvability for the system of integral equations in Corollary 2.5.

### 2.3.2 Linearization and iterative solution

We proceed by linearizing the system (2.60) for the unknowns  $\Gamma$ ,  $\mu$  and  $h$  where the linearized system is then solved in an iterative way. The parameterization of  $\Gamma$  is obtained by patching the parameterizations (2.29) together which gives

$$\Gamma := \{z(t) : t \in [0; 2N\pi]\} \tag{2.61}$$

by setting  $z(t) = z_j(t)$  for  $t \in [2(j-1)\pi; 2j\pi[$  for  $j = 1, \dots, N$ . For  $\psi := h(z(t))$ ,<sup>3</sup> the parameterized versions  $\tilde{w}(f, g, z)$ ,  $\tilde{K}_0(\psi, z)$  and  $\tilde{K}(\psi, z)$  of (2.49), (2.52) and (2.53) are obtained by using the parameterizations (2.30) for the single-layer and (2.33) for the double-layer operator.

Now given a current approximation for  $\Gamma$ ,  $\sigma$  and  $h_k$  the parameterized system is given by

$$\begin{aligned} \tilde{K}_0(\psi_k, z) &= \tilde{w}_+(f_k, g_k), \\ -\frac{1}{2} \mu \psi_k + \tilde{K}(\psi_k, z) &= \tilde{w}(f_k, g_k), \end{aligned} \tag{2.62}$$

The linearization with respect to  $\mu$  and  $\psi$  is straightforward. For  $\Gamma$  we have to compute the *Fréchet derivatives* of the integral operators  $\tilde{w}$ ,  $\tilde{K}_0$  and  $\tilde{K}$  with respect to  $z$  in

---

<sup>3</sup>where we reuse the notation of  $\psi$  from section 2.2, although it has a different meaning in this section,

direction  $\zeta$ . In [44] it has been shown that the Fréchet derivatives can be obtained by differentiating their kernels with respect to  $z$ . Therefore we have

$$\begin{aligned} d\tilde{w}(f, g, z; \zeta)(t) &= \frac{1}{2\pi} \int_0^{2\pi} \left\{ \frac{[z'_0(\tau)]^\perp \cdot \zeta(t)}{|z(t) - z_0(\tau)|^2} \right. \\ &\quad \left. - \frac{2[z'_0(\tau)]^\perp \cdot [z(t) - z_0(\tau)][z(t) - z_0(\tau)] \cdot \zeta(t)}{|z(t) - z_0(\tau)|^4} \right\} f(z_0(\tau)) d\tau \\ &\quad + \frac{1}{2\pi} \int_0^{2N\pi} \frac{[z(t) - z_0(\tau)] \cdot \zeta(t)}{|z(t) - z_0(\tau)|^2} g(z_0(\tau)) |z'_0(\tau)| d\tau, \quad t \in [0; 2N\pi[, \end{aligned}$$

$$\begin{aligned} d\tilde{K}_0(\psi, z; \zeta)(t) &= \frac{1}{2\pi} \int_0^{2N\pi} \left\{ \frac{2[z'(\tau)]^\perp \cdot [z(\tau) - z_0(t)][z_0(t) - z(\tau)] \cdot \zeta(\tau)}{|z_0(t) - z(\tau)|^4} \right. \\ &\quad \left. - \frac{[z'(\tau)]^\perp \cdot \zeta(\tau) + [\zeta'(\tau)]^\perp \cdot [z(\tau) - z_0(t)]}{|z_0(t) - z(\tau)|^2} \right\} \psi(\tau) d\tau, \quad t \in [0; 2\pi[, \end{aligned}$$

and

$$\begin{aligned} d\tilde{K}(\psi, z; \zeta)(t) &= \frac{1}{2\pi} \int_0^{2N\pi} \left\{ \frac{2[z'(\tau)]^\perp \cdot [z(\tau) - z(t)][z(t) - z(\tau)] \cdot [\zeta(t) - \zeta(\tau)]}{|z(t) - z(\tau)|^4} \right. \\ &\quad \left. - \frac{[z'(\tau)]^\perp \cdot [\zeta(\tau) - \zeta(t)] + [\zeta'(\tau)]^\perp \cdot [z(\tau) - z(t)]}{|z(t) - z(\tau)|^2} \right\} \psi(\tau) d\tau, \quad t \in [0; 2N\pi[. \end{aligned}$$

We note that the kernel of  $d\tilde{K}$  is also smooth with the diagonal values given by

$$-\frac{[z'(t)]^\perp \cdot z''(t) z'(t) \cdot \zeta'(t)}{2\pi |z'(t)|^4} + \frac{[z'(t)]^\perp \cdot \zeta''(t) + [\zeta'(t)]^\perp \cdot z''(t)}{4\pi |z'(t)|^2}.$$

Now we can write the linearized system, which for a given approximation  $z$ ,  $\mu$  and  $\psi_k$ , needs to be solved for  $\zeta$ ,  $\eta$  and  $\chi_k$ , to obtain the updates  $z + \zeta$  for the parameterization,  $\mu + \eta$  for the conductivity quotient (from which  $\sigma$  can be computed using (2.50)) and  $\psi_k + \chi_k$  for the densities.



$$\begin{aligned}
& \tilde{K}_0(\psi_k, z) + d\tilde{K}_0(\psi_k, z; \zeta) + \tilde{K}_0(\chi_k, z) \\
& = \tilde{w}_+(f_k, g_k), \\
& -\frac{1}{2} \mu \psi_k - \frac{1}{2} \eta \psi_k - \frac{1}{2} \mu \chi_k \\
& + \tilde{K}(\psi_k, z) + d\tilde{K}(\psi_k, z; \zeta) + \tilde{K}(\chi_k, z) \\
& = \tilde{w}(f_k, g_k) + d\tilde{w}(f_k, g_k, z; \zeta).
\end{aligned} \tag{2.63}$$

After having obtained the new approximation, this procedure is iterated in an obvious way. Also note that on the left-hand side of the resulting system all kernels that occur are smooth, so that no kernel singularities have to be treated.

### 2.3.3 Extensions for the complete electrode model

The system (2.60) can also be used to solve the inverse problem for the complete electrode model. Then the Cauchy data  $f$  and  $g$  also have to be considered as unknowns, and the system (2.62) must also be linearized with respect to them. Hence (2.63) becomes

$$\begin{aligned}
& \tilde{K}_0(\psi_k, z) + d\tilde{K}_0(\psi_k, z; \zeta) + \tilde{K}_0(\chi_k, z) \\
& = \tilde{w}_+(f_k, g_k) + \tilde{w}_+(\tilde{f}_k, \tilde{g}_k), \\
& -\frac{1}{2} \mu \psi_k - \frac{1}{2} \eta \psi_k - \frac{1}{2} \mu \chi_k \\
& + \tilde{K}(\psi_k, z) + d\tilde{K}(\psi_k, z; \zeta) + \tilde{K}(\chi_k, z) \\
& = \tilde{w}(f_k, g_k) + \tilde{w}(\tilde{f}_k, \tilde{g}_k) + d\tilde{w}(f_k, g_k, z; \zeta).
\end{aligned} \tag{2.64}$$

such that now besides the updates  $\zeta$ ,  $\eta$  and  $\chi_k$  also the updates  $\tilde{f}_k$  and  $\tilde{g}_k$  for  $f_k$  and  $g_k$  are computed. Furthermore we have to include the CEM boundary conditions (1.20) and (1.21) into the system which gives as linearized form

$$f_k + z_\ell g_k + \tilde{f}_k + z_\ell \tilde{g}_k = U_\ell^k \text{ on } \mathcal{E}_\ell \tag{2.65}$$

and

$$\int_{\mathcal{E}_\ell} g_k dS + \int_{\mathcal{E}_\ell} \tilde{g}_k dS = I_\ell^k \tag{2.66}$$

with given discrete boundary data pairs  $(U^k, I^k)$ . The third CEM boundary condition (1.22) provides  $g = 0$  on  $\mathcal{N}$ , such that when solving the system (2.64)–(2.66) only the restriction  $g|_{\cup \mathcal{E}_\ell}$  has to be considered as unknown.

As for the solution of the direct problem we also have to choose a non-equidistant discretization to deal with the singularities of  $g$  at the edges of the electrodes. So we have to choose again a substitution function  $\omega : [0; 2\pi] \rightarrow [0; 2\pi]$  fulfilling (2.45), (2.46) and (2.48). Also note that now the singular kernel of the single-layer potential in  $w$  is included in the left-hand side of the system, such that the kernel of the complete system is not smooth anymore. However we will not use a different numerical method to solve the system.

### 2.3.4 Regularization of the linearized system

Since the inverse problem is ill-posed, the system of linearized equations (2.63) respectively (2.64) also inherits this property. So in order to restore stability the solution requires regularization. For this we choose the well-established classical Tikhonov method which instead of solving an equation  $A\varphi = b$  solves the regularized equation

$$(\alpha \cdot Id + A^*A)\varphi = A^*b \quad (2.67)$$

where  $\alpha$  is a vector of *penalty terms* which are usually between 0 and 1. In our case the notation  $\alpha \cdot Id$  means that we take the identity matrix and replace its diagonal with the values of the vector  $\alpha$ . We found out by numerical tests that it is necessary to add penalty terms on the shape parameterization  $z$  and the density functions  $\psi_k$ . The conductivity quotient  $\mu$  only appears in the jump  $\frac{1}{2}\mu\psi$ , so it does not require a penalty term. The numerical tests also showed that for the solution of (2.64) no penalty term is required for  $f$  and  $g$ .

An additional regularization parameter is created through the approximation of the boundary curves. For further simplicity we assume that the boundary curves  $\Gamma_i$  are starlike, that is, we express

$$z_i(t) = \xi_i + r_i(t)(\cos t, \sin t), \quad t \in [0; 2\pi[, \quad (2.68)$$

with a  $2\pi$  periodic  $C^2$  function  $r_i : ]0; \infty[$  and a center point  $\xi_i \in \mathbb{R}^2$ . For the numerical computations we approximate the radial functions  $r_i$  by trigonometric polynomials

$$r_i(t) \approx \sum_{k=0}^{m_i} a_{i,k} \cos kt + \sum_{k=1}^{m_i} b_{i,k} \sin kt \quad (2.69)$$

of degree less than or equal to  $m_i \in \mathbb{N}$ . The polynomial degrees  $m_i$  also need to be considered as regularization parameters.

In addition, we found it necessary to work with different regularization parameters on each of the boundary parts  $\Gamma_i$ , that is, we have  $2N$  Tikhonov parameters for shape and density. However, we kept the regularization parameter the same for all Cauchy pairs. So together with the  $N$  polynomial degrees this leads to a total number of  $3N$  regularization parameters.

As to be expected for that many regularization parameters, the trivial trial and error selection cannot be employed anymore with a realistic chance for reasonable reconstructions. In particular, the parameters should not be kept fixed throughout the complete iteration process and should be decreased when the approximate solution gets closer to the true solution.

This dynamical adaptation of the regularization parameters will be the second task of our evolutionary algorithm - besides the determination of the number and geometry of subdomains.

# Chapter 3

## Two methods for obtaining an initial guess

We will now shortly describe two grid-based methods which will be used to obtain an initial guess for the previously described boundary-element method. They do not require any a-priori information except for the background conductivity  $\sigma_0$ , which we assume to be known.

The *factorization method*, which will be used for the continuum model, is a non-iterative method and so it does not need any initial guess at all. For the complete electrode model we will use a simple *finite element method* (FEM) taken from Marko Vauhkonen [51] using some Newton-type iteration for which we can simply take the homogeneous conductivity  $\sigma = \sigma_0$  as initial guess.

### 3.1 The factorization method

Originally the factorization method was introduced for inverse obstacle scattering problems by Kirsch [33]. It then has been extended to inverse impedance tomography by Brühl and Hanke [8, 9, 24]. In this section we normalize the background conductivity to  $\sigma_0 = 1$ , which can be done without loss of generality.

#### 3.1.1 Basic idea

As typical for the so-called sampling or probe methods, the main idea of the factorization method is to provide an indicator function  $F : \Omega \rightarrow \mathbb{R}$  that decides on whether a point  $x$  belongs to the area of background conductivity  $D_0$  or not. To describe this indicator function, we denote by  $\Lambda_\sigma$  the Neumann-to-Dirichlet operator (as defined by definition 1.6) for the (unknown) true conductivity  $\sigma$  and by  $\Lambda_1$  the corresponding operator for constant conductivity equal to one. The operator  $\Lambda_\sigma - \Lambda_1 : L^2(\Gamma) \rightarrow L^2(\Gamma)$  can be shown to be compact and self-adjoint. Consequently it has a complete orthonormal system  $v_k, k = 1, \dots, \infty$ , of eigenfunctions with real eigenvalues  $\lambda_k$ .

Now we introduce the dipole potential

$$D_x(\xi) := \frac{(x - \xi) \cdot d}{|x - \xi|^2}, \quad \xi \neq x, \quad (3.1)$$

for a source point  $x \in \Omega$ , an arbitrary direction  $d$  and a test function

$$\eta_x = D_x|_{\Gamma_0} - \Lambda_1 \frac{\partial}{\partial \nu} D_x - c \quad (3.2)$$

with a constant  $c$  such that  $\int_{\Gamma_0} \eta_x ds = 0$ . Then the indicator function of the factorization method is given by the series

$$F(x) := \sum_{k=1}^{\infty} \frac{\langle \eta_x, v_k \rangle^2}{\lambda_k} \quad (3.3)$$

where  $\langle \cdot, \cdot \rangle$  denotes the usual  $L^2$  inner product on  $L^2(\Gamma)$ . Provided that all conductivities  $\sigma_i, i = 1, \dots, N$  satisfy either  $\sigma_i > 1, i = 1, \dots, N$ , or  $\sigma_i < 1$  for  $i = 1, \dots, N$ , then as main feature of the factorization method the series in (3.3) converges if and only if  $x \notin \overline{D_0}$  (see [8]).

### 3.1.2 The factorization method for inner inclusion levels

In its original form presented above, the factorization method only works for one inclusion level, as it can only indicate if the conductivity at a given point differs from the background conductivity or not. However starting from the theory presented in [20] one can show [21] that basically the same principle can be employed if an inner subdomain is known.

So we now consider the case of  $N \geq 2, p(1) = 0$  and  $p(i) > 0$  for  $i = 2, \dots, N$ , i.e. we only have one inclusion  $D_1$  at inclusion level 1 containing at least one other inclusion inside. This can be done without loss of generality, as the procedure can be iterated for multiple subdomains at the same inclusion level. After this it can also be iterated for several inclusion levels, following the basic idea of the layer-stripping algorithm mentioned in section 1.3. For a given conductivity  $\kappa$  inside  $D_1$  (not necessarily equal to the true conductivity  $\sigma_1$ ) we define the operator  $\Lambda_\kappa$  as the NtoD-operator for conductivity  $\kappa$  inside  $\Omega \setminus \overline{D_0}$  and  $\sigma_0 = 1$  inside  $D_0$ . If the shape of  $D_1$  is exactly known, we can apply the factorization method again. First using [20] it can be shown [21]

**Theorem 3.1** *If the transmission constants  $a_1, \dots, a_N$  all fulfill either  $a_i > 1$  or  $a_i < 1$  we have:*

*If  $\sigma_1 > \sigma_0$  the operator  $\Lambda_\sigma - \Lambda_\kappa$  is negative definite if and only if  $\kappa \leq \sigma_1$ . If  $\sigma_1 < \sigma_0$  the operator  $\Lambda_\sigma - \Lambda_\kappa$  is positive definite if and only if  $\kappa \geq \sigma_1$ .*

This implies that given the shape of  $D_1$  we can determine  $\sigma_1$  at any precision by checking the definiteness of  $\Lambda_\sigma - \Lambda_\kappa$  by a simple bisection algorithm. Once knowing  $D_1$  and its conductivity we can then apply the factorization method again by using  $\Lambda_\sigma - \Lambda_\kappa$  instead of  $\Lambda_\sigma - \Lambda_1$ . Following [21] the dipole function (3.1) has to be replaced by some function

$$D_x^\kappa(\xi) := D_x(\xi) + K_x^\kappa(\xi) \quad (3.4)$$

with a function  $K_x^\kappa$  which is harmonic in  $\Omega \setminus \Gamma_1$ , continuous over  $\Gamma_1$  and fulfills on  $\Gamma_1$  the transmission condition

$$\sigma_0 \frac{\partial K_{x,+}^\kappa}{\partial \nu} - \kappa \frac{\partial K_{x,-}^\kappa}{\partial \nu} = (\kappa - \sigma_0) \frac{\partial D_x}{\partial \nu} \quad (3.5)$$

such that the modified dipole function (3.4) fulfills the transmission condition (2.4) on  $\Gamma_1$ . The test function  $\eta_x^\kappa$  is then defined as in (3.2) with  $D_x$  replaced by  $D_x^\kappa$  and the series (3.3) by

$$F^\kappa(x) := \sum_{k=1}^{\infty} \frac{\langle \eta_x^\kappa, v_k^\kappa \rangle^2}{\lambda_k^\kappa} \quad (3.6)$$

where  $v_k^\kappa$  are the eigenvectors and  $\lambda_k^\kappa$  the eigenvalues of  $\Lambda_\sigma - \Lambda_\kappa$ . The series converges if and only if  $x \notin \overline{D_0} \cup \overline{D_1}$ .

### 3.1.3 Numerical implementation

As already mentioned before, in reality we do not know the full Neumann-to-Dirichlet map but only a set of  $M$  Cauchy pairs  $(f_k, g_k)$ . So we have to replace  $\Lambda_\sigma - \Lambda_1$  by its projection on the finite-dimensional space  $\text{span}\{g_1, \dots, g_M\}$ . We will only describe the standard factorization method, as the discretization for the inner inclusion levels is done exactly the same way. If the  $g_k$  are linearly independent, we then obtain an  $M \times M$  matrix with  $M$  eigenvectors  $v_k^{(M)}$  and eigenvalues  $\lambda_k^{(M)}$ . The series (3.3) is replaced by the finite sum

$$F_M(x) := \sum_{k=1}^M \frac{\langle \eta_x^{(M)}, v_k^{(M)} \rangle^2}{\lambda_k^{(M)}} \quad (3.7)$$

by which we now have to decide whether the corresponding infinite series converges or not. This will be done via a threshold criterion which will be described in chapter 5.

Finally, although there is no theoretical justification for the factorization method in the case when some of the interior conductivities are larger than  $\sigma_0$  and some of them are smaller than  $\sigma_0$ , we also applied it for this case using some slightly modified version. So we replaced the eigenvalues  $\lambda_k$  by their absolute values and instead of (3.3) we examined

$$F^{\text{abs}}(x) := \sum_{k=1}^{\infty} \frac{\langle \eta_x, v_k \rangle^2}{|\lambda_k|} \quad (3.8)$$

and computed its finite approximation  $F_M^{\text{abs}}(x)$ . To check convergence one can always look at  $F^{\text{abs}}(x)$  instead of  $F(x)$ , because in the case of equal contrast all eigenvalues have the same sign. This implies that  $F^{\text{abs}}(x) = |F(x)|$ , and the only additional information  $F(x)$  can provide is the sign, which indicates if we have a contrast smaller or larger than 1. However, in case of different contrasts at one inclusion level the criterion in theorem 3.1 cannot be applied anymore. So in this case all conductivities have to be exactly known for the application of the factorization method for inner inclusion levels.

## 3.2 A simple Finite Element method

Finite Element methods (FEM) are the most common form of methods used for the solution of the EIT problem. However, as we only want to use them for an initial guess for the boundary-element method, we will limit ourselves to a simple Newton-type method implemented by M. Vauhkonen [51]. But note that for the evolutionary algorithm presented in the next chapter *any* finite element method that does not need a-priori information could be used instead.

For the FEM we start with the weak formulation (1.11) of the EIT problem. For the complete electrode model we additionally have to define a *discrete test function*  $V \in \mathbb{R}^L$ . Some computations, which can be found for example in [53], finally lead to the *FEM equation for the complete electrode model*

$$\int_D \sigma \text{grad } u \cdot \text{grad } v dx + \sum_{\ell=1}^L \frac{1}{z_\ell} \int_{\mathcal{E}_\ell} (u - U_\ell)(v - V_\ell) dS = \sum_{\ell=1}^L I_\ell V_\ell \quad (3.9)$$

where  $u, v \in H^1(\Omega)$ .

### 3.2.1 Repartition of the domain and basis functions

Now the domain under consideration  $\Omega$  has to be split into convex polyhedrons  $\Delta_i$ ,  $i = 1, \dots, N_p$ . They are required to have the property that the intersection of two different polyhedrons  $\Delta_i \cap \Delta_j$ ,  $i \neq j$  is either

- the empty set *or*
- exactly one corner of each polyhedron *or*
- one complete edge of each polyhedron.

This especially means that every corner of a polyhedron has also to be a corner of any neighboring polyhedron. ‘Hanging’ vertices are not allowed. Figure 3.1 shows an example of a valid and a non-valid repartition of domains.

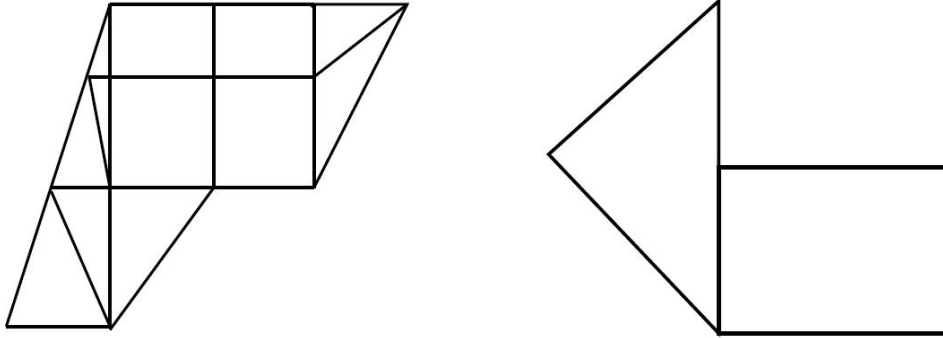


Figure 3.1: Example for a valid (left) and non-valid (right) FEM repartition. On the right we have a ‘hanging’ vertex. Taken from [41].

This definition will be especially important for the grid refinement used in the evolutionary algorithm described in chapter 5. Now we assume the conductivity  $\sigma$  to be constant on each polyhedron where in this case we take the *resistivity*  $\rho := \sigma^{-1}$  instead.

Denoting by  $N_n$  the number of nodes in the FEM grid a set of *basis functions*  $\phi_i$ ,  $i = 1, \dots, N_n$  is defined such that  $\phi_i = 1$  at node  $i$  and 0 at all other nodes. The potential function  $u$  can then be approximated as

$$u \approx \sum_{i=1}^{N_n} \alpha_i \phi_i \quad (3.10)$$

with variable coefficients  $\alpha_i$ .

Here a set of *triangles* is chosen as the simplest case in 2D for the polyhedrons, and for the basis functions the simplest case is to take *piecewise linear functions*.

### 3.2.2 Solution of the direct problem via an FEM

For the Newton iteration, which is used to solve the inverse problem, the solution of the forward problem is also required in each iteration step. So we start with the description of the forward solver introduced in [51]. The aim is to find the interior potential  $u$  and the voltages  $U$  on the electrodes given the currents  $I$  and the resistivity  $\rho = \sigma^{-1}$ . The potential  $u$  is approximated as in (3.10). To assure the condition (1.2) the voltage  $U$  is written as

$$U = \sum_{j=1}^{L-1} \beta_j \xi_j \quad (3.11)$$

where  $\xi_1 := [1, -1, 0, 0, \dots, 0]$ ,  $\xi_2 := [1, 0, -1, 0, \dots, 0]$ ,  $\dots$  are  $L - 1$  linearly independent vectors in  $\mathbb{R}^L$  and  $\beta_j$  the unknown coefficients. Equation (3.9) can then



be transformed into a linear system for the unknown coefficients  $\alpha$  and  $\beta$  which are written as  $b := (\alpha, \beta)^T$ . They are computed as the solution of the linear system

$$Ab = c \quad (3.12)$$

where the *stiffness matrix*  $A$  is a block matrix of the form

$$\begin{pmatrix} B & C \\ C^T & D \end{pmatrix}.$$

The elements of these matrices are defined as follows

$$B(i, j) := \int_D \sigma \operatorname{grad} \phi_i \cdot \operatorname{grad} \phi_j dx + \sum_{\ell=1}^L \frac{1}{z_\ell} \int_{\mathcal{E}_\ell} \phi_i \phi_j dS,$$

$$i, j = 1, \dots, N_n$$

$$C(i, j) := -\left( \frac{1}{z_1} \int_{\mathcal{E}_1} \phi_i dS - \frac{1}{z_{j+1}} \int_{\mathcal{E}_{j+1}} \phi_{j+1} ds \right),$$

$$i = 1, \dots, N_n, j = 1, \dots, L-1$$

$$D(i, j) := \sum_{\ell=1}^L \frac{1}{z_\ell} \int_{\mathcal{E}_\ell} (\xi_i)_\ell (\xi_j)_\ell dS$$

$$i, j = 1, \dots, L-1$$

and the data vector  $c \in \mathbb{R}^{N_n+L-1}$  is given as

$$\begin{pmatrix} 0 \\ \vdots \\ 0 \\ \sum_{\ell=1}^L I_\ell(\xi_1)_\ell \\ \vdots \\ \sum_{\ell=1}^L I_\ell(\xi_{L-1})_\ell \end{pmatrix}$$

Note that for most  $i, j$  we have  $\phi_i \phi_j = 0$  so that also  $B(i, j) = 0$ .

### 3.2.3 Solution of the inverse problem by a Newton-type iteration

The Newton method is a well-known classical method for the solution of operator equations (see for example [16]). The solution to the nonlinear operator equation

$$F(x) = 0$$

is found by iteratively solving the linearized equation

$$F'(x_n)r_n = -F(x_n) \quad (3.13)$$

to find the update  $r_n$  to improve on a current approximation  $x_n$ . Starting with some first guess  $x_0$  we define  $x_{n+1} = x_n + r_n$ .

In our case we define the function as  $F(\rho) := U_{\text{approx}}(\rho) - U_{\text{meas}}$ , i.e. we try to adapt the approximated resistivity  $\rho$  such that for fixed  $I$  the resulting voltages  $U_{\text{approx}}$  by solving the direct EIT problem correspond to the measured voltages  $U_{\text{meas}}$ . This means that the *Jacobian matrix*  $J := F'(\rho^{(n)})$  has to be computed. It is a  $(KL \times N_p)$  matrix that describes the derivative of the voltages  $U$  with respect to the conductivity inside each element

$$J = \begin{pmatrix} \frac{\partial U_1^1}{\partial \rho_1} & \cdots & \frac{\partial U_1^1}{\partial \rho_{N_p}} \\ \frac{\partial U_2^1}{\partial \rho_1} & \cdots & \frac{\partial U_2^1}{\partial \rho_{N_p}} \\ \vdots & \ddots & \vdots \\ \frac{\partial U_1^K}{\partial \rho_1} & \cdots & \frac{\partial U_1^K}{\partial \rho_{N_p}} \\ \vdots & \ddots & \vdots \\ \frac{\partial U_L^K}{\partial \rho_1} & \cdots & \frac{\partial U_L^K}{\partial \rho_{N_p}} \end{pmatrix}$$

In [56] it has been shown that the derivative of the stiffness matrix with respect to the conductivity  $\rho_k$  inside element  $\Delta_k$  is equal to

$$\frac{\partial A^{-1}c}{\partial \rho_k} = -A^{-1} \frac{\partial A}{\partial \rho_k} A^{-1}c = A^{-1} \frac{\partial A}{\partial \rho_k} b \quad (3.14)$$

where  $\frac{\partial A}{\partial \rho_k}$  can be computed as

$$\frac{\partial A(i, j)}{\partial \rho_k} = \frac{1}{\rho_k^2} \int_{\Delta_k} \text{grad } \phi_i \cdot \text{grad } \phi_j. \quad (3.15)$$

From this representation the derivatives  $\frac{\partial \beta}{\partial \rho}$  can be extracted, and from them the derivatives with respect to the voltages - and so the Jacobian - can be computed.

Also in this case the solution of the equation needs to be regularized. This time this is done by some kind of *generalized Tikhonov regularization* (see [52])

$$\rho^{(n+1)} := \rho^n + (J^T J + L_2^T L_2 \alpha)^{-1} (J^T F(\rho^n) - \alpha L_2^T L_2 \rho^{(n)}) \quad (3.16)$$

where  $L_2$  is a regularization matrix depending on the structure of the FEM mesh.

In our case we choose a quite strong regularization parameter  $\alpha$ , as we only need a crude initial guess about the conductivity. Note that the computation of  $\rho$  via (3.16) can lead to  $\rho^{(n)} < 0$  at some elements. As by definition of the conductivity this is not

possible, in this case we set  $\rho^{(n)} = \rho^{(n-1)}$  at the corresponding elements - unlike in the original method.

The computations are stopped and considered as successful if after less than a pre-defined number  $\tilde{\mu}$  of iterations the change  $\|\rho^{(n)} - \rho^{(n-1)}\|$  between two iteration results is smaller than a given  $\varepsilon$ . Otherwise, if after  $\tilde{\mu}$  iterations we still have  $\|\rho^{(n)} - \rho^{(n-1)}\| \geq \varepsilon$ , we conclude that the method does not converge for the chosen regularization parameter  $\alpha$  and restart the method with another  $\alpha$  greater than the previous one.

# Chapter 4

## Basic concepts of evolutionary algorithms

Evolutionary algorithms have been developed in the area of computational optimization. Their main idea is to employ concepts of biological evolution like recombination, mutation and selection to achieve the approximate solution of optimization problems. As the main difference to conventional optimization methods they use *stochastic components* for iteration, and they perform a *parallel search* on a whole population of solutions.

Different main *categories* of evolutionary algorithms (EA) have been developed independently from each other. The most common form of EA are *genetic algorithms* (GA) developed by J. H. Holland [27], followed by *evolutionary strategies* (ES) by I. Rechenberg [46] and H.-P. Schwefel [47] and *evolutionary programming* (EP) by Fogel, Owens and Walsh [18]. Also *genetic programming* (GP) by Koza [35] is considered as a separate concept, even if it was developed on the basis of GA. The concepts of these approaches differ by the representation of the solutions, notations and the focus of each technique. From a mathematical viewpoint, they all employ the same principles and ideas - the differences are mainly on the level of implementation and software technology. For more details about the differences between the different categories see for example [43].

### 4.1 Principles and notation

As evolutionary algorithms are inspired by natural principles, the notations which are used are taken from their biological equivalent. First an *individual*  $\mathcal{I}$  is a representation of a possible or approximate solution to the given problem. An individual can be seen as an ordered finite set of variables  $\varsigma_i$  which - depending on the category of the EA and the problem under consideration - might be restricted to certain values. These values can be for example natural numbers or even binary values, as it is the

case in genetic algorithms. A set of individuals is then called a *population* and will be denoted by  $\mathcal{P}$ . In contrast to usual iterative schemes, evolutionary algorithms do not only consider one approximation at each time but a whole population of several approximations/individuals.

To be able to compare individuals we have to define a *fitness functional*, which assigns to each individual  $\mathcal{I}$  a positive real number  $\mathcal{F}(\mathcal{I})$ , such that it turns a population into a well-ordered set. This functional represents the criterion with respect to which the problem under consideration is then minimized by the evolutionary algorithm. The value of the fitness functional for a given individual  $\mathcal{I}$  is also called *fitness value* of that individual.

Evolutionary algorithms are iterative methods. After the creation of some initial population the same steps are repeated until an individual satisfies a given criterion. In each iteration the individuals of an existing population are used to generate individuals of a new population. The existing population is then called *parents*, the resulting population *children*. We will denote by  $P$  the size of the parent population and by  $Q$  the size of the children population. The outcome of an iteration step is called *generation*, and the iteration step is also called *generation step*. A generation steps typically consists of *recombination*, *mutation* and *selection*. Figure 4.1 illustrates the main principles of an EA.

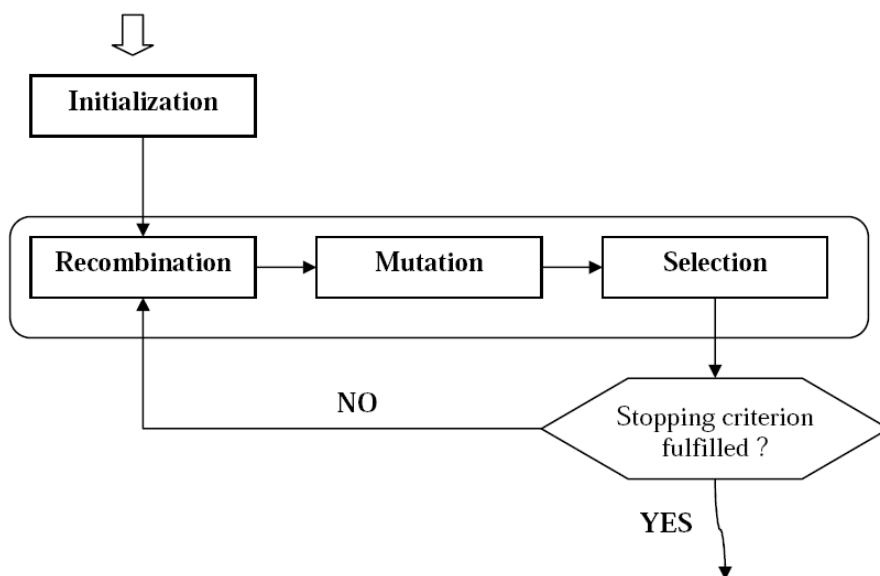


Figure 4.1: Main steps of an evolutionary algorithm.

## 4.2 Main parts of one generation step

The aim of each evolutionary algorithm is to find an individual with minimal fitness value. So in each generation step it is the aim to create new individuals and change them in a way that their fitness values are smaller than the fitness values of the last generation. In the following we will summarize the realization of the three main steps in each generation step: Recombination, mutation and selection.

Note that depending on the category of the EA and the problem under consideration the focus can rather lie on the recombination or on the mutation where even one of them can be completely omitted. This was especially the case in the first genetic algorithm, which omitted the mutation, and the first evolutionary strategy, which omitted the recombination.

### 4.2.1 Recombination

The creation of a new individual from two or more existing individuals by combination of their properties is called *recombination*. If an new individual is created as a copy of another individual this is called *replication* or cloning. In nature recombination is realized by the process where a child inherits properties of both of its parents.

To create a new individual each parameter  $\tilde{\zeta}_i$  of the new individual is set as a convex combination of the parameters  $\zeta_{ij}$ ,  $j = 1, \dots, P$ , of the parent population. This means that we randomly choose weights  $0 \leq \vartheta_{ij} \leq 1$  such that

$$\sum_{j=1}^P \vartheta_{ij} = 1 \quad (4.1)$$

and we set the parameters of the new individual as

$$\tilde{\zeta}_i := \sum_{j=1}^P \vartheta_{ij} \zeta_{ij}. \quad (4.2)$$

Usually one does not take all parent individuals into account, so there are usually some  $j$  for which  $\vartheta_{ij} = 0$ . Especially in *discrete recombination* there is only one index  $k$  for which  $\vartheta_{ik} = 1$ , and consequently we have  $\vartheta_{ij} = 0$  for all other  $j \neq k$ . Discrete recombination often occurs if we do not have real or complex values and is especially used for binary representations.

Another frequently used special case is the so-called *line recombination*, which means that the weights  $\vartheta_{ij}$  are the same for all parameters  $\zeta_i$ . Not covered by the above definition is the recombination via *crossover*, which is used in genetic algorithms but which will not be treated here (for more details see [27]).

### 4.2.2 Mutation

In its original form *mutation* always meant a stochastical change of the individual's properties - analogously to natural mutation - and is normally executed after the creation of a new individual by recombination/replication. If the variable  $\varsigma_i$  of the individual is real or complex, it is usually multiplied by a random real number  $r$  close to 1, or a quantity  $\varrho$  is added to it, whose absolute value is relatively small compared to  $\varsigma_i$ . In the discrete case some random natural number is added or subtracted, or in the binary case the value of  $\varsigma_i$  is inverted with a certain probability.

However, with this definition mutation is totally random and blind, as it does not take at all into account the properties of the problem under consideration. So the definition of mutation has been extended (see for example [40]) to include also some deterministic components, which are now adapted to the problem under consideration. So in a more general (and less biological) sense, mutation is understood as a change of the individual's properties involving a stochastical component at some point. This also means that unlike in the original definition of mutation the individual can be changed considerably, and the mutation can lead to completely new individuals. As it will be described in the next chapter, this type of mutation will also be used in our evolutionary algorithm.

### 4.2.3 Selection

The choice of the individuals for the next generation is called *selection*. In biology selection occurs if there are more individuals than resources which can aliment them. In this case only those individuals which are the best adapted to the environment survive. In EA this corresponds to those individuals with the lowest fitness value. Obviously this only makes sense if during one generation step more individuals than the number of those taken for the next generation have been created.

The most simple case of selection is the *deterministic rank-based selection*, which will also be used in our algorithm. It simply consists of ordering the individuals of the parent and children population with respect to their fitness values and then taking the best  $P$  of them. Depending on the question whether the parents can be re-selected, we call a  $(P, Q)$  selection a selection where only the children can be selected for the next generation, a  $(P, Q + 1)$  selection if also the best parent individual can be selected and a  $(P, Q + P)$  selection if all parents can be selected. The set of individuals from which the next parent population is selected is called *selection set*.

Which selection set is chosen depends on the type of problem to be solved. In a purely explorative search one should choose the  $(P, Q)$  selection to avoid getting stuck in a local optimum too soon. On the other hand the  $(P, Q + P)$  selection is used if we want to avoid the approximation to get worse or if we want to store intermediate results. The  $(P, Q + 1)$  selection is a compromise between them, which however already assures that the best approximation ever found is always kept. In our case

we will also use this kind of selection set, although with some small modifications (see chapter 5).

Beyond this simple selection model there exist more sophisticated models, which often include also stochastical components. Especially selection can also be postponed to the recombination process of the next generation. This means that first all new individuals (and - depending on the selection set - also some or all of the individuals of the last generation) are taken for the next generation, but then the weights  $v_{ij}$  for the recombination are not equally drawn but in favor of those individuals with a low fitness value. This corresponds more to the biological equivalent of fitness being defined as the capability to produce offspring. As these selection models will not be used here, we refer for example to [43] for more information.

### 4.3 Further development of evolutionary algorithms

The considerable increase of computational power in recent years made it possible to extend the classical concept of one population with a once given definition of recombination, mutation and selection. So several extensional concepts for EA have been developed, which will be shortly mentioned here.

One of these new concepts is the concept of *multiple populations* which use selection schemes and/or fitness functionals which are different from each other. In biology this corresponds to different environment conditions which require other qualities from an individual. This might be necessary if for example one has several optimization criteria which are equally important. In this case, for each criterion a separate population is created, where the fitness functional of each population is defined according to this criterion. Another reason for using this concept can be that one wishes to consider only individuals which fulfill certain restrictions and let them develop independently for a certain time.

Obviously multiple populations only make sense if also a certain exchange between them is allowed, at least under certain conditions. Corresponding to biological terms this exchange is then called *migration*.

Another important concept is the so-called *meta evolution*. This means that the parameters of the EA itself - like size of the population, definition of the selection or the parameter setup - are submitted to an optimization process, which can itself be again an evolutionary algorithm. The idea has already been pronounced by Rechenberg in 1973 (see [46]) as a *learning population* but could not be reasonably realized at that time due to limited computer power. Note that if the method for the Meta Evolution is an EA we also have to consider several populations which are then considered as individuals of the Meta-EA.



# Chapter 5

## The concept of our evolutionary algorithm

We now describe the concept of our particular evolutionary algorithm, which combines the different methods presented in the chapters 2 and 3. The boundary-element method from chapter 2 is used as our main method, and the grid-based methods from chapter 3 are used to provide a-priori information about the number, approximate positions and geometry of the unknown subdomains. Note that the basic ideas of our algorithm also have been used in [15] for the case of the inverse scattering problem.

We will start with the definition of an individual and the fitness functional. Then we proceed with the conversion of the results of the grid-based method into a boundary representation and with the description of recombination, mutation and selection.

To the 'classical' evolutionary algorithm two additional features have been added, which will be described in section 5.4. The chapter will conclude in section 5.5 with a flow diagram to illustrate the concept of our evolutionary algorithm.

### 5.1 Definition of an individual and of the fitness functional

#### 5.1.1 Definition of an individual

In our case an individual  $\mathcal{I}$  represents a conductive medium, characterized by its geometry and piecewise constant conductivity. It also contains information about the (discretized) potential functions and the regularization parameters. Within a population containing  $P$  individuals we denote by  $N_p$  the number of subdomains within an individual  $\mathcal{I}_p$ ,  $p = 1, \dots, P$ . This number can be different for each individual.

Then for  $i = 1, \dots, N_p$  each subdomain  $D_{ip}$  is characterized through the parameters

$$\mathbf{D}_{ip} \quad \boxed{(\alpha_r)_{ip} \quad (\alpha_\psi)_{ip} \quad m_{ip} \quad \xi_{ip} \quad r_{ip} \quad \sigma_{ip} \quad \psi_{ip,1}, \dots, \psi_{ip,M}}$$

Here the Tikhonov parameter  $(\alpha_r)_{ip}$  for the radial update, the Tikhonov parameter  $(\alpha_\psi)_{ip}$  for the density update and the polynomial degree  $m_{ip}$  constitute the regularization parameters associated with the subdomain  $D_{ip}$ . The other parameters describe the domain itself:  $\xi_{ip}$  is the center and  $r_{ip}$  is the radial function for the exterior interface curve  $\Gamma_{ip}$  as described through its  $2m_{ip} + 1$  Fourier coefficients according to (2.69),  $\sigma_{ip}$  is the constant conductivity inside the subdomain, and  $\psi_{ip,1}, \dots, \psi_{ip,M}$  are the (discretized) density functions on  $\Gamma_{ip}$ . In the case of the complete electrode model the individual also contains the continuous Dirichlet and Neumann boundary data  $f_{p,k}$  and  $g_{p,k}$  for  $k = 1, \dots, M$ .

### 5.1.2 Definition of the fitness functional

For the computation of the fitness functional we solve the direct transmission problem for the configuration that each individual represents, that is, for the corresponding interface curves and conductivities. Then we look at the norm of the difference between the computed solution and the given data. This norm will be denoted by  $\|\cdot\|$ , since due to the fact that we work with discretized data there is flexibility in the choice of a particular norm. Therefore for our particular choice we refer to chapter 6.

In the continuum case we consider the transmission problem with Neumann boundary condition  $g$ , and with our direct solver we obtain for an individual  $\mathcal{I}$  the corresponding Dirichlet data  $f_{\mathcal{I}}(g)$ . In the CEM case we analogously consider the direct transmission problem with input data  $I$ , and we obtain the corresponding voltages  $U_{\mathcal{I}}(I)$ . Here, in a slight abuse of notation, we use the same symbols  $f, U$  to denote the given input data and the result of the computation of the direct problem, which depends on the input data  $g, I$  and the configuration the individual  $\mathcal{I}$  represents.

Then we define the *partial fitness functional* as

$$\mathcal{F}_{\tilde{k}}(\mathcal{I}) := \|f_{\mathcal{I}}(g_{\tilde{k}}) - f_{\tilde{k}}\|$$

respectively

$$\mathcal{F}_{\tilde{k}}(\mathcal{I}) := \|U_{\mathcal{I}}(I^{\tilde{k}}) - U^{\tilde{k}}\| \quad (5.1)$$

with randomly selected  $\tilde{k}$ ,  $1 \leq \tilde{k} \leq M$  and the *complete fitness functional* as

$$\mathcal{F}(\mathcal{I}) := \sum_{k=1}^M \|f_{\mathcal{I}}(g_k) - f_k\|$$

respectively

$$\mathcal{F}(\mathcal{I}) := \sum_{k=1}^M \|U_{\mathcal{I}}(I^k) - U^k\|. \quad (5.2)$$

Using the partial fitness functional we require less time to compute it, and we introduce a certain stochastical component into the evaluation of the individual. The complete

fitness functional instead is a more accurate measure, but if we have many data sets, we require much more time to compute it. For exact data this measure usually seems good enough to decide on the quality of the approximations, but also for noisy data it is used, as no other information about the true solution is known.

We note that as an alternative one could also reverse the roles of the boundary conditions and consider the transmission problem with Dirichlet boundary condition with data  $f$  on  $\Gamma_0$  (respectively the voltages  $U$  on the electrodes), compute the Neumann data  $g_{\mathcal{I}}(f)$  (respectively the current  $I_{\mathcal{I}}(U)$ ) and use the norm of the difference  $g_{\mathcal{I}}(f) - g$  (respectively  $I_{\mathcal{I}}(U) - I$ ) as fitness functional.

## 5.2 Creation of the initial population

As we do not assume to be given any a-priori information, we first have to apply one of the grid-based methods (factorization or FEM) from chapter 3. Usually we obtain quite a heterogeneous set of values from it, and so we have several possibilities to convert them into a domain representation. Therefore we start our evolutionary algorithm with the creation of several subdomain sets. They are then transformed into those individuals which will constitute our starting population.

In this section we will speak of *configurations* when we explicitly mean a set of subdomains together with their conductivities, but without Tikhonov parameters and density functions. We call an *inhomogeneity* each subset of  $\Omega$  where the conductivity differs from the background conductivity  $\sigma_0$ . Especially each subdomain  $D_{ip}$ ,  $i = 1, \dots, N_p$ , within an individual  $\mathcal{I}_p$  is an inhomogeneity.

### 5.2.1 Selecting the points

For the factorization method we know that for those gridpoints  $x \in \Omega$ , for which the series  $F^{\text{abs}}(x)$  defined through (3.8) converges, we have a contrast between the conductivity at that point and the background conductivity. This is equivalent to  $x$  being part of an inhomogeneity. The sign of the series  $F(x)$  defined by (3.3) indicates if we have a *positive* ( $a_i > 1$ ) or a *negative* ( $a_i < 1$ ) contrast. Therefore when considering the finite approximation  $F_M^{\text{abs}}(x)$  we know that the probability that  $x$  is part of an inhomogeneity increases if the value of  $F_M^{\text{abs}}(x)$  decreases.

For the finite element method for each element  $\Delta_j$  we are directly given the difference between the initial conductivity  $\sigma^{(\text{old})}(j)$  and the conductivity  $\sigma^{(\text{new})}(j)$  obtained via the Newton-type iteration described in section 3.2. However, to assure that changes with positive and negative contrast are equally evaluated, we rather look at the *logarithmic change*

$$LC(j) := \ln \frac{\sigma^{(\text{new})}(j)}{\sigma^{(\text{old})}(j)} \quad (5.3)$$

instead. It is always well-defined, as we only have positive conductivities. Obviously the probability that  $\Delta_j$  is part of an inhomogeneity increases if also  $|LC(j)|$  increases. Furthermore the sign of  $LC(j)$  indicates if we have a positive or a negative contrast.

Starting from the above considerations we define a sequence of positive thresholds  $\{\theta_q\}$ , which we require to be strictly increasing in the case of the factorization method and to be strictly decreasing in the case of the FEM. Then for each threshold  $\theta_q$  we collect all those points  $x$  satisfying

$$F_M^{\text{abs}}(x) \leq \theta_q \quad (5.4)$$

respectively all those elements  $\Delta_j$  satisfying

$$|LC(j)| \geq \theta_q. \quad (5.5)$$

If for a predefined natural number  $S$  we now have a set of **at least  $S$  connected points (elements) with the same contrast**, we consider this set as a subdomain and store it. The number  $S$  should be chosen in such a way that we avoid getting small artifacts.

As thresholds we take the values of  $F_L^{\text{abs}}(x)$  respectively  $|LC(j)|$  themselves in the way that we order them by increasing (decreasing) values. This way we assure to obtain a different set of points for each threshold and to cover all possibilities.

## 5.2.2 Converting the point sets into subdomains

From the points we collected for the different thresholds we now have to create the individuals for our starting population. First we have to find a parameterization according to (2.69) for the interface curves of each subdomain. Then we have to assign to each subdomain a constant conductivity. Finally the missing parameters - density functions and Tikhonov parameters - have to be set to some initial values to complete the obtained configurations into individuals.

### 5.2.2.1 Creating an interface curve from a point set

In the case of the factorization method we determine for each set of connected grid-points the boundary of its convex hull, and in the FEM case we determine the boundary of the convex hull of the nodes of each set of connected elements. Denoting by  $\{\tilde{x}_1, \dots, \tilde{x}_n\} \subset \mathbb{R}^2$  the set of points we obtained this way, we now want to convert it into an interface curve  $\Gamma$  of the form (2.68). Therefore we have to determine its center point  $\xi$  and, for a given polynomial degree  $m$ , the  $2m + 1$  coefficients  $a_0, \dots, a_m, b_1, \dots, b_m$  of the trigonometric radial function  $r$  according to (2.69).

The center point  $\xi$  is set as the mean of all points, that is

$$\xi := \frac{1}{n} \sum_{i=1}^n \tilde{x}_i.$$

Now each point  $\tilde{x}_i$  can be represented in polar coordinates  $(\tilde{r}_i, \theta_i)$  with respect to  $\xi$  as center. From this we can derive a system of equations by requiring that for each point we have

$$r(\theta_i)(\cos \theta_i, \sin \theta_i) = \tilde{x}_i - \xi, \quad i = 1, \dots, n. \quad (5.6)$$

Since in general we have  $2n > 2m + 1$ , this system can then be solved in a least-squares sense for the unknown coefficients  $a_0, \dots, a_m, b_1, \dots, b_m$  of the radial function  $r$ .

### 5.2.2.2 Treating curve intersections

Now it can happen that some of the created curves intersect with others or also with themselves. As this constitutes a non-valid configuration we have to create rules how to treat this situation. These rules are also applied in other situations. Therefore we will speak of 'individuals' instead of 'configurations'.

First, to check for intersections we do the following: For two different curves  $\Gamma_x, \Gamma_y$  we denote their discretization points by  $x_1, \dots, x_n$  and  $y_1, \dots, y_n$ . We say that these curves intersect if there exist some indices  $i, j \in \{1, \dots, n\}$  such that the lines from  $x_i$  to  $x_{i+1}$  and from  $y_j$  to  $y_{j+1}$  intersect (where we have set  $x_{n+1} := x_1$  and  $y_{n+1} := y_1$ ). They intersect if and only if there exists a point  $z$  and two constants  $h_1, h_2$  fulfilling  $0 \leq h_1, h_2 \leq 1$  such that  $z$  can be written as

$$z = x_i + h_1(x_{i+1} - x_i) = y_j + h_2(y_{j+1} - y_j). \quad (5.7)$$

Therefore, we check for this property for all  $i, j = 1, \dots, n$ . To detect self-intersections of a curve  $\Gamma$  discretized as  $x_1, \dots, x_n$  we proceed in a similar way. Here we check for intersections between the lines from  $x_i$  to  $x_{i+1}$  and the lines from  $x_j$  to  $x_{j+1}$  for  $i = 1, \dots, n$  and  $j = i + 1, \dots, n$ .

In case of a self-intersection, the polynomial degree of the concerned curve is reduced, and a new curve is computed by a least-squares approximation of the system (5.6). This is repeated until there are no more self-intersections - which is certainly the case for  $m = 0$ .

For the intersection of different subdomains the idea is to *shrink* a subdomain by multiplying its radial function by a predefined number  $0 < \beta < 1$ . This is performed for all concerned subdomains one by one, starting with the innermost inclusion level. If afterwards there are still intersections, this procedure is repeated until there are no intersections anymore or a predefined maximum number of shrinks is reached. If then some subdomains still intersect, they are deleted from the individual.

If at the end of this process the individual does not contain any more subdomains, it is deleted from the population.

### 5.2.2.3 Setting the conductivity

Next we have to assign a conductivity to each created subdomain. In the case of the FEM we are already given some a-priori information about the conductivity. For each subdomain we compute the *area* of all constituting elements as

$$\mathcal{A}(\Gamma^*) := \sum_{i \in \Gamma^*} |\Delta_i|$$

and we set the initial conductivity as

$$\sigma^*(\Gamma^*) := \frac{1}{\mathcal{A}} \sum_{i \in \Gamma^*} \sigma(i) |\Delta_i|.$$

In the case of the factorization method we define an 'initial' conductivity  $\sigma^* > 1$ , and if the sign of  $F_L$  over the points constituting the subdomain is negative we assign the conductivity  $\sigma^*$  to the subdomain, otherwise we assign  $1/\sigma^*$ .

### 5.2.2.4 Selecting and completing the best configurations

Usually by the procedure described before we obtain more than  $P$  different configurations. Therefore to obtain an initial population consisting of  $P$  individuals we have to perform a selection. This is done by computing the *complete fitness functional* (5.2) for each configuration and then taking those  $P$  configurations with the lowest values.

Those configurations are then completed into individuals. For this we predefine initial Tikhonov parameters  $(\alpha_\psi^*, \alpha_r^*)$  and assign them to each subdomain. They should be rather strong, as they only constitute an initial guess, which will be weakened during mutation. For the densities  $\psi$  in the continuum case we solve the system (2.62) for each boundary data pair  $(f_k, g_k)$  for the unknown  $\psi$  and with fixed  $\Gamma, \sigma$ . In that case we have an overdetermined system, so that we have to solve it in a least-squares sense. In the CEM case we extend (2.62) with the CEM boundary conditions (1.20)–(1.22). This way we obtain for each boundary data pair  $(U^k, I^k)$  an overdetermined system for the unknowns  $\psi, f, g$ , which is also solved in a least-squares sense.

## 5.2.3 Global subdomain indices

We will see in subsection 5.3.1 that for the *recombination* we need to keep track on the evolvement of the subdomains during the evolutionary algorithm. To do this we introduce a set of *global subdomain indices*  $j = 1, \dots, N$ . For each configuration/individual  $\mathcal{I}_q$  we assign one of these indices to each of the subdomains it contains, where we require that each index is assigned at most once within one configuration/individual. If index  $j$  has been assigned to subdomain  $D_{iq}$  we say (by using a term of object-oriented programming) that subdomain  $D_{iq}$  is an *instance of the global subdomain*  $j$ . In this context we also define for each configuration/individual  $\mathcal{I}_q$  a list

of boolean variables  $\text{include}_q$ , where  $\text{include}_q(j)$  indicates if an instance of the global subdomain  $j$  is contained in  $\mathcal{I}_q$  or not.

We further define two lists of sets Split and Merge.  $\text{Split}(j)$  contains the indices of those subdomains which have contributed to the creation of subdomain  $j$ . This is equivalent to the possibility that an instance of subdomain  $j$  can be split into instances of those subdomains.  $\text{Merge}(j)$  on the other hand indicates into which larger instances of subdomains we can merge instances of the subdomain  $j$ .

If we have multiple inclusion levels we also define the lists predecessor and successor. Recalling the definition of the predecessor relation from chapter 2 we say that for a global subdomain index  $j$  the exterior curve of an instance of  $j$  is an interior curve of an instance from  $\text{predecessor}(j)$ . On the other hand,  $\text{successor}(j)$  contains those indices  $\tilde{j}$  satisfying  $j \in \text{predecessor}(\tilde{j})$ .

When creating the starting population we only have one inclusion level. Therefore predecessor and successor still remain empty at the beginning of our evolutionary algorithm. They are initialized when we look for subdomains at inner inclusion levels as described in subsection 5.4.2.

### 5.2.3.1 Initialization of Split and Merge

When creating the starting population we build up Split and Merge simultaneously with the initial configurations. If for the rest of this subsection we speak of 'points' we always mean 'gridpoints' in the continuum and 'elements' in the CEM case. For each threshold  $\theta_q$  ( $q \geq 2$ ) we check for each set  $\mathcal{S}_{i_q}$  of connected points that we created if it contains points that already have been chosen for the previous threshold  $\theta_{q-1}$ .

- If  $\mathcal{S}_{i_q}$  only contains points which have not been chosen for the previous configuration, the resulting subdomain is considered as an instance of a *new* global subdomain, which has no splitting or merging relation with the existing ones. Therefore we increase the number of global subdomain indices  $N$  by one, set  $\text{include}_q(N)$  to 1 for the actual configuration and  $\text{Split}(N)$  and  $\text{Merge}(N)$  to the empty set.
- If  $\mathcal{S}_{i_q}$  contains some points which already have been chosen for the previous configuration, but all those points have been part of *only one point set*  $\mathcal{S}_{i_{q-1}}$ , we consider the subdomain resulting from  $\mathcal{S}_{i_q}$  as an instance of the same global subdomain as the subdomain created from  $\mathcal{S}_{i_{q-1}}$ . Therefore no new global subdomain index is created. Denoting by  $\tilde{j}$  the global subdomain index assigned to the subdomain created from  $\mathcal{S}_{i_{q-1}}$ , we set  $\text{include}_q(\tilde{j})$  to 1.
- If  $\mathcal{S}_{i_q}$  contains points which already have been chosen before and there have been *several* point sets  $\mathcal{S}_{i_{1,q-1}}, \dots, \mathcal{S}_{i_{l,q-1}}$  containing some of those points, the subdomain resulting from  $\mathcal{S}_{i_q}$  is considered as a *merge* of those subdomains resulting from  $\mathcal{S}_{i_{1,q-1}}, \dots, \mathcal{S}_{i_{l,q-1}}$ . Then, like in the first case, we increase  $N$  by one,

set include $_q(N)$  to 1 for the actual configuration and Merge( $N$ ) to the empty set. Additionally, denoting by  $j_1, \dots, j_l$  the global subdomain indices associated with the subdomains created from  $\mathcal{S}_{\iota_1, q-1}, \dots, \mathcal{S}_{\iota_l, q-1}$  we then set

$$\text{Split}(N) = \{j_1, \dots, j_l\} \cup \left( \bigcup_{i=1}^l \text{Split}(j_i) \right)$$

and for  $i = 1, \dots, l$

$$\text{Merge}(j_i) = \text{Merge}(j_i) \cup \{N\}.$$

For the first threshold  $\theta_1$ , obviously only the first case is possible.

### 5.2.3.2 Maximum number of subdomains

We further define for each global subdomain index  $j$  the *maximum number of splits*

$$\tilde{M}(j) := \# \{l \mid l \in \text{Split}(j) \wedge \text{Split}(l) = \emptyset\} \quad (5.8)$$

Then for a predefined *maximum number of subdomains*  $\hat{N}$  we require that each configuration  $\mathcal{I}_q$  satisfies

$$\tilde{N}_q := \sum_{i=1}^{N_q} \max \left( \tilde{M}(\text{global}(i, q)), 1 \right) \leq \hat{N}, \quad (5.9)$$

where  $\text{global}(i, q)$  denotes the global subdomain index associated with the subdomain  $D_{i,q}$ . The reason why we require (5.9) is that we want to assure that no possible recombination (see subsection 5.3.1) can lead to an individual containing more than  $\hat{N}$  subdomains.

When creating the starting population it can be verified that when passing from threshold  $\theta_q$  to  $\theta_{q+1}$  with the rules defined above we always have  $\tilde{N}_{q+1} \geq \tilde{N}_q$ . If for  $\theta_{q+1}$  there is a set of connected points  $\mathcal{S}_{i, q+1}$  which does not contain any points chosen for  $\theta_q$  (case 1), then  $\tilde{N}_{q+1}$  increases with respect to  $\tilde{N}_q$ . If all sets of connected points for  $\theta_{q+1}$  satisfy the cases 2 or 3, we have  $\tilde{N}_{q+1} = \tilde{N}_q$ . Therefore, if for a given threshold  $\theta_q$  we have  $\tilde{N}_q = \hat{N}$ , from there on we only select those point sets which contain some points already chosen before. Point sets satisfying case 1 are ignored.

## 5.2.4 An example

We will now illustrate the previously described procedure by a simplified example. Here, we use the finite-element method for the initial guess. The true configuration consists of three subdomains and is shown on the left-hand side of figure 5.1. On the right-hand side of figure 5.1 we see the result of the finite-element method.



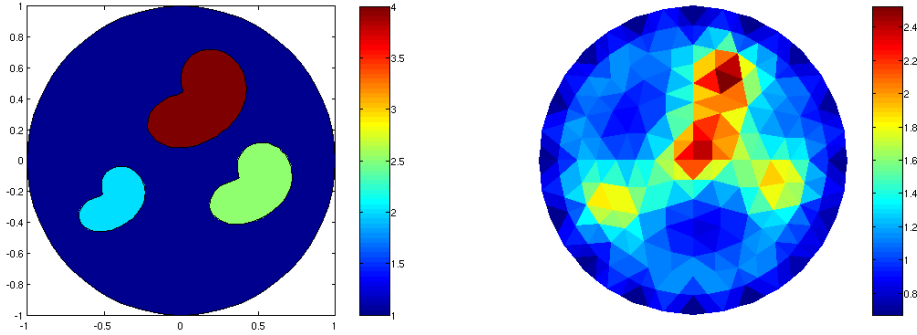


Figure 5.1: True configuration and result of the FEM.

Now we will show illustrations of all those configurations, for which the domain constellation has changed with respect to the previous one. We will also present the merging list as far as it has been built until then and the inclusion list for each of the presented configurations. The minimum number  $S$  of connected points has been set to 1 in this case to obtain better illustrations. The curve around the subdomains corresponds to the curve obtained by the least-squares approximation of (5.6) for an initial polynomial degree  $m^* = 2$ . The color of the selected elements indicates the constant conductivity assigned to that subdomain. The maximum number of subdomains  $\hat{N}$  has been set to 4 in this case.

The first two domain configurations look as follows:

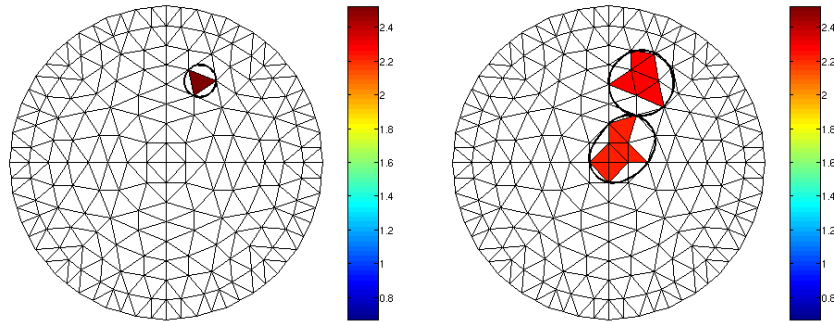


Figure 5.2: Domain configurations 1 and 2

We see that for configuration 2 the upper point set contains elements which have already been chosen for configuration 1. There, they are all contained in *one* point set. Therefore case 2 of the rules from subsection 5.2.3 is applied, and the upper subdomain of configuration 2 is considered as an instance of the same global subdomain 1 as the subdomain from configuration 1. So the inclusion lists of the two configurations are

Subdomain	1	Subdomain	1 2
include <sub>1</sub>	1	include <sub>2</sub>	1 1

Split and Merge are still empty at that moment. The next two domain configurations are the following:

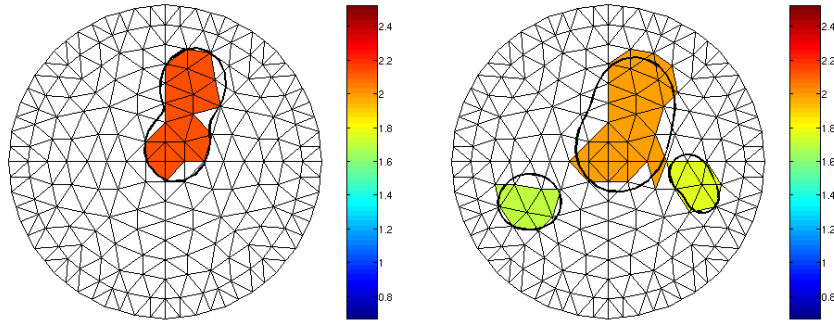


Figure 5.3: Domain configurations 3 and 4

Now the subdomain included in configuration 3 contains elements which are contained in two *different* point sets of configuration 2. Therefore case 3 of the rules from 5.2.3 is applied. We create a new global subdomain index 3, and instances of that index are considered as a merging of instances of 1 and 2. Note that this also implies  $\tilde{M}(3) = 2$ , and therefore we have  $\tilde{N}_3 = 2$ . So now Split and Merge look as follows:

Subdomain	1	2	3
Split	–	–	1, 2
Merge	3	3	–

The inclusion list for configuration 3 is given as:

Subdomain	1	2	3
include <sub>3</sub>	0	0	1

For configuration 4 we have found one point set satisfying case 2 for the global subdomain number 3 and two point sets satisfying case 1. So they are considered as instances of the new global subdomains 4 and 5. The inclusion list for configuration 4 is then

Subdomain	1	2	3	4	5
include <sub>4</sub>	0	0	1	1	1

Split and Merge did not receive any new entries, so that now they look like

Subdomain	1	2	3	4	5
Split	–	–	1, 2	–	–
Merge	3	3	–	–	–

However, due to the fact that we have found two point sets satisfying case 1, we now have  $\tilde{N}_4 = \hat{N} = 4$ . So, from now on we only consider those point sets satisfying the cases 2 or 3.

When the threshold is further decreased, we obtain the following configurations:

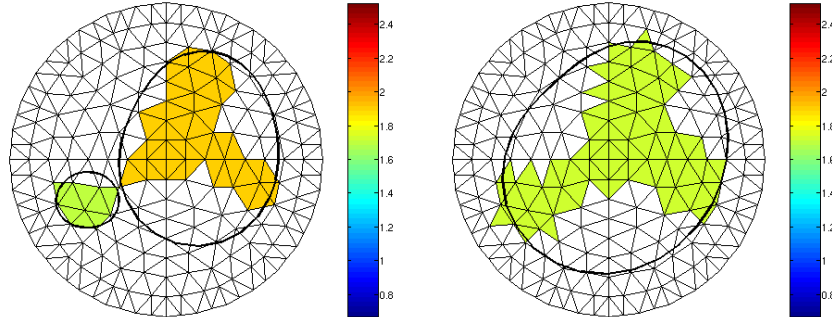


Figure 5.4: Domain configurations 5 and 6

After the creation of configuration 5 Split and Merge look as follows:

Subdomain	1	2	3	4	5	6
Split	—	—	1, 2	—	—	1, 2, 3, 4
Merge	3, 6	3, 6	6	6	—	—

After the creation of configuration 6 we have:

Subdomain	1	2	3	4	5	6	7
Split	—	—	1, 2	—	—	1, 2, 3, 4	1, 2, 3, 4, 5, 6
Merge	3, 6, 7	3, 6, 7	6, 7	6, 7	7	7	—

The inclusion lists for configurations 5 and 6 are

Subdomain	1	2	3	4	5	6
include <sub>5</sub>	0	0	0	0	1	1

Subdomain	1	2	3	4	5	6	7
include <sub>6</sub>	0	0	0	0	0	0	1

Finally, when the search for new configurations is finished, we fill up the inclusion lists of all configurations with zeros, such that all lists then contain the same number of entries.

## 5.3 One generation step

We will now describe the realization of one generation step, consisting of recombination, selection and mutation.

### 5.3.1 Recombination

Each generation step starts with the creation of  $Q$  new individuals by recombination. For each new individual we randomly draw with equal probability two parent individuals (which are also allowed to be twice the same individual). In this section they

will be denoted by  $\mathcal{I}_1, \mathcal{I}_2$ . Then the inclusion list of the new individual  $\mathcal{I}_3$  is created by *discrete recombination* (see subsection 4.2.1) of the two parent individuals' inclusion lists, where the weights are drawn separately for each entry. This means that if for a global index  $j$  an instance of  $j$  is contained in both parent individuals, also the new individual has to contain an instance of  $j$ . If only one of the parent individuals contains an instance of  $j$ ,  $\mathcal{I}_3$  contains an instance of  $j$  only with a 50% chance. After its creation the inclusion list might have to be modified, because due the geometry of the configurations we have to require that

- For each subdomain at an inner inclusion level contained in  $\mathcal{I}_3$  also an instance of one of its predecessors has to be contained in  $\mathcal{I}_3$ .
- An instance of  $j$  cannot be contained in an individual together with an instance of any of those indices contained in  $\text{Split}(j)$  or  $\text{Merge}(j)$ .

These rules will be illustrated by an example in 5.3.1.1.

For those subdomains whose global numbers are contained in the inclusion list of both parent individuals, the parameters of the new subdomains are computed by a *line recombination* (see subsection 4.2.1) of the parameters of the parents' subdomains. The other subdomains are copied from that individual that contains them. In the case that some of the subdomains we have finally chosen for  $\mathcal{I}_3$  intersect, the procedure described in 5.2.2.2 is applied again.

### 5.3.1.1 An example for the recombination of the inclusion lists

We will now illustrate the rules for the creation of the inclusion lists via an example. The two parent individuals are illustrated in figure 5.5 together with the global numbers of their subdomains.

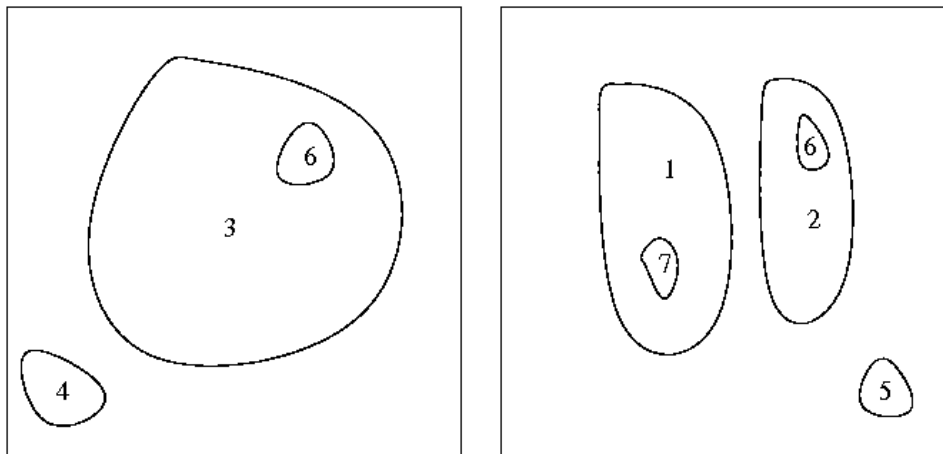


Figure 5.5: Example for two individuals

The global lists predecessor, successor, Split and Merge are the following:

Subdomain	1	2	3	4	5	6	7
predecessor	–	–	–	–	–	2, 3	1, 3
successor	7	6	6, 7	–	–	–	–
Split	–	–	1, 2	–	–	–	–
Merge	3	3	–	–	–	–	–

and the inclusion lists for the two parent individuals are

Subdomain	1	2	3	4	5	6	7
include <sub>1</sub>	0	0	1	1	0	1	0
include <sub>2</sub>	1	1	0	0	1	1	1

Then for the recombination of those inclusion lists we have the following conditions:

- An instance of subdomain 6 is included in both parent individuals, so also the new individual has to contain an instance of 6. This instance is set as a convex combination of the instances contained in  $\mathcal{I}_1$  and  $\mathcal{I}_2$ .
- As subdomain 6 is at an inner inclusion level, either the instance of subdomain 3 from  $\mathcal{I}_1$  or the instance of subdomain 2 from  $\mathcal{I}_2$  has to be contained in  $\mathcal{I}_3$ .
- If the instance of subdomain 3 has been chosen, the instances of the subdomains 1 and 2 from  $\mathcal{I}_2$  cannot be contained in  $\mathcal{I}_3$ , as  $\text{Split}(3) = \{1, 2\}$ . Otherwise, if the instance of subdomain 2 has been chosen,  $\mathcal{I}_3$  cannot contain the instance of subdomain 3.
- The instance of subdomain 7 from  $\mathcal{I}_2$  is chosen with a 50% chance. If it is included in  $\mathcal{I}_3$  also an instance of one of its predecessors has to be included. If previously the instance of subdomain 2 has been chosen as predecessor of 6, we now have to include the instance of subdomain 1 from  $\mathcal{I}_2$  as instances of 3 are excluded. Otherwise, if the instance of subdomain 3 from  $\mathcal{I}_2$  has been chosen, it is automatically set as the predecessor of the instance of subdomain 7 from  $\mathcal{I}_2$ .
- Finally the instance of 4 from  $\mathcal{I}_1$  and the instance of 5 from  $\mathcal{I}_2$  are both chosen with a probability of 50%, independently from anything else.

### 5.3.2 Mutation

After their creation the  $Q$  children individuals are submitted to mutation. There, each individual is mainly exposed to two actions:

- First the regularization parameters are submitted to a 'classical' mutation: With a predefined fixed probability  $p$  it is decided for each parameter if it is changed

or kept the same. The Tikhonov parameters that are selected for being changed are multiplied by a random number between previously defined positive values  $s_{\min}$  and  $s_{\max}$ . The trigonometric polynomial degrees to be changed are randomly either increased or decreased by 1. If they are increased, the new radial coefficients  $a_{m+1}$  and  $b_{m+1}$  are set to zero. If they are decreased, the new radial function  $r$  and the new center  $\xi$  are determined by a least-squares fit of the system (5.6).

- Then, secondly, we execute a fixed number  $It$  of iteration steps of the Tikhonov solution of (2.63) (respectively (2.64)–(2.66)) with the new parameters.

This way we hope to find not only a better approximation to the solution of the inverse problem, but also regularization parameters which are better adapted to the actual configuration.

After applying the boundary-element method we might obtain again some non-valid configurations. In these cases we proceed as follows

- Negative conductivities are set back to their value before the execution of the boundary-element method.
- Intersections between subdomains at *different* inclusion levels are treated the same way as described in 5.2.2.2.
- In case of intersections of different subdomains at the *same* inclusion level we *merge* the concerned subdomains together as it will be described in 5.3.2.1.
- In case of self-intersections we *split* the concerned subdomains as described in 5.3.2.2.

If a splitting or merging operation has been successfully performed, we execute again  $It$  iteration steps of the boundary-element method with the modified configuration. If we encounter intersections after this, we treat them as described in 5.2.2.2. However, subdomains which still intersect after the maximum number of shrinks are not deleted from the individual, but they are set back to their values before the mutation. The same is done for their regularization parameters, as they obviously have not been strong enough to lead to a valid configuration.

Finally, if the domain configuration has been changed due to a splitting or merging operation, the global merging list is adapted as described in 5.3.2.3.

### 5.3.2.1 Merging of subdomains

Denoting by  $\Gamma_1, \dots, \Gamma_s$  the boundary curves of those subdomains which intersect with each other, we first determine the exterior hull of their union  $\cup \Gamma_j$ . For this we compute

for each discretization point  $x \in \cup \Gamma_j$  the potential

$$d_j(x) := \int_{\Gamma_j} \frac{\partial \Phi(x, y)}{\partial \nu(y)} ds(y), \quad j = 1, \dots, s \quad (5.10)$$

Denoting by  $i(x)$  the index such that  $\Gamma_{i(x)}$  is the curve containing  $x$  and using the properties of the double-layer potential we then look at the sign of  $d_j(x)$ . If for all  $j \neq i(x)$  we have  $d_j(x) \geq 0$ , we add  $x$  to the exterior hull of the union.

Then, by taking points of the exterior hull as input set, we determine the center point  $\xi$  and the radial coefficients  $a_0, \dots, a_m, b_1, \dots, b_m$  for the curve of the merged subdomain by solving (5.6) in a least-squares sense. The polynomial degree  $m$  is set to the maximum encountered over all concerned subdomains. The remaining parameters of the new subdomain are set as a convex combination of the parameters of the composing subdomains, where the weights are chosen proportional to the area of each subdomain. This way we guarantee that large subdomains are taken more into account than small subdomains.

### 5.3.2.2 Splitting of a subdomain

For a subdomain  $D$  whose curve  $\Gamma$  contains self-intersections we denote by  $\{x_1, \dots, x_n\}$  the set of discretization points for  $\Gamma$ . We further denote by  $z_1, \dots, z_{\tilde{s}}$  the intersection points of  $\Gamma$  and by  $(\tilde{i}_k, \tilde{j}_k)$ ,  $k = 1, \dots, \tilde{s}$  those indices for which the lines from  $x_{\tilde{i}_k}$  to  $x_{\tilde{i}_k+1}$  and from  $x_{\tilde{j}_k}$  to  $x_{\tilde{j}_k+1}$  intersect at  $z_k$  according to (5.7). Then we order these indices as  $i_1, \dots, i_{2\tilde{s}}$  such that  $i_{j+1} > i_j$  for  $j = 1, \dots, 2\tilde{s} - 1$ .

For the splitting procedure the set of discretization points  $\{x_1, \dots, x_n\}$  is first divided into parts  $\Xi_1, \dots, \Xi_{2\tilde{s}}$ , where  $\Xi_j := \{x_{i_{j+1}}, \dots, x_{i_{j+1}}\}$  for  $j = 1, \dots, 2\tilde{s} - 1$  and  $\Xi_{2\tilde{s}} := \{x_{i_{2\tilde{s}+1}}, \dots, x_n, x_1, \dots, x_{i_1}\}$ . Then we patch some of these parts together in the way that for each intersection point  $z_k$  we connect the part ending at  $x_{\tilde{i}_k}$  with the part starting at  $x_{\tilde{j}_k+1}$  and the part ending at  $x_{\tilde{j}_k}$  with the part starting at  $x_{\tilde{i}_k+1}$ . Note that there are always cases where these parts coincide. This way we always obtain a set of  $s := \tilde{s} + 1$  closed curves  $\tilde{\Gamma}_1, \dots, \tilde{\Gamma}_s$ . For each of those curves containing at least 2 points we then determine a center point  $\xi$  and a radial function  $r$  via (5.6) to obtain a new curve which can be written in the form (2.68). The polynomial degree  $m$  for  $r$  is the same as for the original subdomain  $D$ . In the case that  $\tilde{\Gamma}_j$  contains less than  $2m + 1$  points, the polynomial degree is reduced for  $\tilde{\Gamma}_j$ .

Now, denoting by  $\Gamma_j$ ,  $j = 1, \dots, s$  the curves obtained via the newly computed radial function and by  $\{x_{j,1}, \dots, x_{j,n}\}$  the discretization points for  $\Gamma_j$ , we compute for each  $x_{j,k}$  the potential

$$d_i(x_{j,k}) := \int_{\Gamma_i} \frac{\partial \Phi(x, y)}{\partial \nu(y)} ds(y), \quad i = 1, \dots, s, \quad i \neq j. \quad (5.11)$$

If for all  $i \neq j$  and for all  $k = 1, \dots, n$  we have  $d_i(x_{j,k}) \geq 0$ , we know that  $\Gamma_j$  does not lie inside another newly created curve  $\Gamma_i$  or intersects with it. In that case  $\Gamma_j$  is completed into a subdomain  $D_j$  by assigning the remaining parameters of the original subdomain  $D$  to it. Otherwise the curve  $\Gamma_j$  is deleted.

Figure 5.6 shows an example of a subdomain  $D$ , whose curve  $\Gamma$  contains  $\tilde{s} = 2$  intersection points. Therefore  $\Gamma$  is first split into  $2\tilde{s} = 4$  parts  $\Xi_1, \dots, \Xi_4$ , which are then patched together into  $s = 3$  closed curves  $\tilde{\Gamma}_1, \dots, \tilde{\Gamma}_3$ . Next, these curves are transformed into starlike curves  $\Gamma_1, \dots, \Gamma_3$ , which can be written in the form (2.68). Finally one of those curves is deleted as it lies inside another curve.

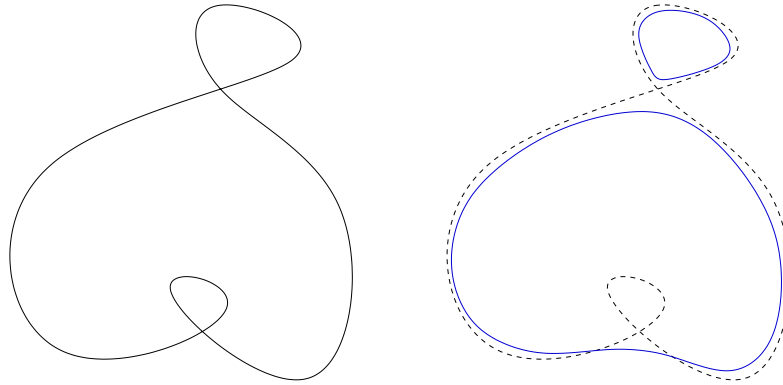


Figure 5.6: Example for splitting

### 5.3.2.3 Modifications of the splitting and merging list

If a splitting or merging operation has been successfully performed, we also have to adapt the global lists Split and Merge in an adequate way.

In case of a merging operation we denote by  $j_1, \dots, j_s$  the global subdomain indices of those subdomains which have contributed to the merging of the new subdomain. We look for an index  $\tilde{j}$  satisfying  $\{j_1, \dots, j_s\} \subset \text{Split}(\tilde{j})$ . If there exists such an index we consider the new subdomain as an instance of the global subdomain  $\tilde{j}$ , and we are finished.

Otherwise we have to create a new global subdomain index  $\varsigma$  for the merged subdomain, and we have to set  $\text{Split}(\varsigma) = \{j_1, \dots, j_s\}$  and  $\text{Merge}(j_i) = \text{Merge}(j_i) \cup \{\varsigma\}$  for  $i = 1, \dots, s$ . Then we also have to look for indices  $\hat{j}$  satisfying  $\text{Split}(\hat{j}) \subset \{j_1, \dots, j_s\}$ . Subdomains which are instances of those global indices also can be merged into an instance of  $\varsigma$ . Therefore for each of those  $\hat{j}$  we set  $\text{Split}(\varsigma) = \text{Split}(\varsigma) \cup \{\hat{j}\}$  and  $\text{Merge}(\hat{j}) = \text{Merge}(\hat{j}) \cup \{\varsigma\}$ . Finally, if we have inner inclusion levels, we set  $\text{successor}(\varsigma) = \bigcup_{k=1}^s \text{successor}(j_k)$  and then for all  $\tilde{j} \in \text{successor}(\varsigma)$  we set  $\text{predecessor}(\tilde{j}) = \text{predecessor}(\tilde{j}) \cup \{\varsigma\}$ .

In case of a splitting operation we now denote by  $s$  the number of subdomains  $D_1, \dots, D_s$  which finally have been created by the splitting procedure. We further



denote by  $j$  the global subdomain index of the subdomain which has been splitted and set  $t := \tilde{M}(j)$  with the maximum number of splits  $\tilde{M}(j)$  as defined by (5.8). Then we are given  $t$  indices  $j_k$ ,  $k = 1, \dots, t$  satisfying  $j_k \in \text{Split}(j) \wedge \text{Split}(j_k) = \emptyset$ . These are the global subdomain indices which can be assigned to the new subdomains  $D_1, \dots, D_s$

To decide which global subdomain index is assigned to which new subdomain we do the following: For each global index  $j_k$ ,  $k = 1, \dots, t$  we collect from the individuals of the *parent* population all subdomains which are instances of  $j_k$ . Then we determine  $\tilde{\xi}_{j_k}$  as the convex combination of the center points  $\xi$  of all those subdomains with equally chosen weights. We denote by  $\xi_i$ ,  $i = 1, \dots, s$  the center points of the subdomains  $D_1, \dots, D_s$  created by the splitting procedure.

- If  $s \leq t$  we start with the new subdomain  $D_1$ . We look for an index  $k_1$  satisfying  $|\xi_1 - \tilde{\xi}_{j_{k_1}}| \leq |\xi_1 - \tilde{\xi}_{j_k}|$  for all  $k = 1, \dots, t$ . We then set the global subdomain index of  $D_1$  to  $j_{k_1}$ . Then we continue with  $D_2$  by looking for  $k_2$  satisfying  $|\xi_2 - \tilde{\xi}_{j_{k_2}}| \leq |\xi_2 - \tilde{\xi}_{j_k}|$  for all  $k = 1, \dots, t$ ,  $k \neq k_1$ , and so on. This way we assign a different global subdomain index to all new subdomains  $D_1, \dots, D_s$ . The lists Split and Merge are not changed in this case.
- If  $s > t$  we start with index  $j_1$  and look for an index  $\hat{k}_1$  satisfying  $|\xi_{\hat{k}_1} - \tilde{\xi}_{j_1}| \leq |\xi_{\hat{k}} - \tilde{\xi}_{j_1}|$  for all  $\hat{k} = 1, \dots, s$ . We then assign the global subdomain index  $j_1$  to  $D_{\hat{k}_1}$ . Then we continue with  $j_2$  by looking for  $\hat{k}_2$  satisfying  $|\xi_{\hat{k}_2} - \tilde{\xi}_{j_2}| \leq |\xi_{\hat{k}} - \tilde{\xi}_{j_2}|$  for all  $\hat{k} = 1, \dots, s$ ,  $\hat{k} \neq \hat{k}_1$ , and so on. At the end of this procedure we are left with  $s - t$  subdomains  $D_{\hat{k}}$  to which no global subdomain index has been assigned yet. Therefore we have to create  $s - t$  new indices  $\varsigma_1, \dots, \varsigma_{s-t}$  and assign them to the remaining subdomains. Then we set  $\text{Split}(j) = \text{Split}(j) \cup \{\varsigma_1, \dots, \varsigma_{s-t}\}$  and for  $l = 1, \dots, s - t$  we set  $\text{Merge}(\varsigma_l) = \{j\}$ .

If we have multiple inclusion levels, we now denote by  $j_1, \dots, j_s$  those global indices assigned to  $D_1, \dots, D_s$ . We then look for those indices  $\hat{j}$  satisfying  $\hat{j} \in \text{successor}(j) \wedge \hat{j} \notin \bigcup_{k=1}^s \text{successor}(j_k)$ . To those indices we have to assign a predecessor from  $j_1, \dots, j_s$ . For this we collect again all instances of  $\hat{j}$  from the parent population and determine  $\tilde{\xi}_{\hat{j}}$  as the convex combination of the center points  $\xi$  of all those subdomains with equally chosen weights. Then, for  $k = 1, \dots, s$ , if the potential  $d_k(\tilde{\xi}_{\hat{j}})$  as defined by (5.11) is positive, we set  $\text{predecessor}(\hat{j}) = \text{predecessor}(\hat{j}) \cup \{j_k\}$  and  $\text{successor}(j_k) = \text{successor}(j_k) \cup \{\hat{j}\}$ .

### 5.3.3 Selection

Basically we use the deterministic rank-based selection, where the selection set is a  $(P, Q + 1)$  selection. First we compute the **partial** fitness functional (5.1) for each individual, where the index  $\tilde{k}$  is chosen randomly, such that it is the same for all individuals. Then the best  $P$  individuals with respect to that fitness functional are chosen.

However we always want to keep the best individual found so far in the population. So we then compute the **full** fitness functional for the best newly selected individual and compare it to the full fitness functional of the best parent individual. If the latter one is better, we keep the best old individual as best individual in the population and take only the best  $P - 1$  new individuals.

## 5.4 Further extensions

The particular properties of the problem under consideration made it necessary to modify the ‘classical’ scheme of an evolutionary algorithm as it has been described in chapter 4. This was done by adding two additional features to our program: An intermediate step between recombination and mutation in the case of different domain constellations and the interruption of the iterative EA scheme for the creation of a new population. Furthermore to reduce the computational effort we tried to use only a part of the boundary data pairs for the solution of the inverse problem. These features will be described now.

### 5.4.1 An intermediate step

For the mutation of each individual the whole boundary-element method has to be executed several times, which is quite a time-consuming process. In addition, if we have multiple domain constellations, the number of possible parameter combinations is quite large, because we have several domain constellations as well as different regularization parameters.

So to reduce the computational costs, we introduce some predefined  $\tilde{Q}$  satisfying  $P < \tilde{Q} < Q$ . For the  $Q$  children individuals created via recombination we only execute *one* step of the boundary-element method. Then we create an *intermediate population* by selecting the best  $\tilde{Q}$  children individuals with respect to the *partial* fitness functional. We call this *pre-selection*. We then perform a full mutation step only on those  $\tilde{Q}$  individuals of the intermediate population.

Another problem we faced during the numerical tests was that sometimes the algorithm found an artifact which was then part of all individuals in the population. As in the recombination all subdomains included in both parent individuals are always taken, these artifacts could not be deleted anymore. For that reason we *delete* with a given probability  $\tilde{p}$  from each of the  $\tilde{Q}$  selected individuals of the intermediate population a randomly selected subdomain which does not have any successor subdomains. This is consciously done **after** the pre-selection, as now the individuals modified this way are submitted to a full mutation step. So, if the deleted subdomain was really an artifact, there is some chance that the individual without the artifact is better than those including the artifact.

If all individuals contain the same domain constellation, i.e. the inclusion list is

the same for all of them, we create only  $\tilde{Q}$  individuals via recombination and omit the intermediate step. Then, also no subdomain is randomly deleted before the mutation.

### 5.4.2 Creating a new population for an inner inclusion level

If there are multiple inclusion levels the information about the subdomains' geometry cannot be extracted in one single step. If the grid-based method indicates a contrast at a given point/element, it is not clear at which inclusion level the corresponding subdomain is located. So for multiple inclusion levels it is not enough to convert the grid representation into a domain representation only once and then run the evolutionary algorithm.

Instead we have to proceed layerwise, starting with the outermost inclusion level, and then interrupt the EA at some point and newly apply the grid-based method to find the subdomains at the next inner inclusion level. Additionally even if the number and approximate location of all subdomains is a-priori known, our numerical experiments showed that it is quite advantageous to proceed layerwise and to determine first the shape and conductivity of the outermost layer's subdomains.

First, we have to define at which point the EA is interrupted. For this, we define a number  $C_{\max} \in \mathbb{N}$  and count the number  $C_n$  of generations, for which the best individual has not changed **and** all individuals of the parent population have contained the same domain constellation (i.e.  $\text{include}_p$  is the same for all  $p = 1, \dots, P$ ). If this number has reached  $C_{\max}$ , we interrupt the EA, as then obviously the algorithm is stuck in a local minimum for the actual domain configuration. So at that point we are looking for new subdomains at the next inclusion level. We recall that  $N_1$  denotes the number of subdomains in the best individual.

So we convert the configuration that the best individual represents into a grid-based representation. For this we define for each subdomain  $D_i$ ,  $i = 0, \dots, N_1$ , some kind of level set function  $\ell_i : \Omega \rightarrow \{0; 1\}$ , which is 1 for those points lying inside  $D_i$  and 0 otherwise. Obviously for all  $x \in \Omega$  we have  $\ell_i(x)\ell_j(x) = 0$  for  $i \neq j$ . In the case of the factorization method these level set functions are computed for all gridpoints  $x$ , and in the case of the FEM they are computed for the center of each element  $\Delta$ . In both cases this computation is done by using the potential  $d_j$  defined through (5.10). Then we assign the conductivity

$$\sum_{i=0}^{N_1} \sigma_i \ell_i(x) \tag{5.12}$$

to each point or element, respectively.

Next, we newly execute the respective grid-based method from chapter 3. The factorization method for inner inclusion levels is described in subsection 3.1.2. Before executing the finite-element method we perform a *grid refinement* following the idea

of adaptive multigrid methods mentioned in section 1.3. If for a predefined number  $\aleph$  and for  $i = 1, \dots, N_1$  there are at most  $\aleph$  elements for which we have  $\ell_i = 1$ , then we divide each of those elements into four as illustrated by the lower left triangle on the right-hand side of figure 5.7. To maintain the property of the FEM mesh that 'hanging' vertices are not allowed we also have to divide the neighboring elements into two as illustrated by the upper left triangle on the right-hand side of figure 5.7.

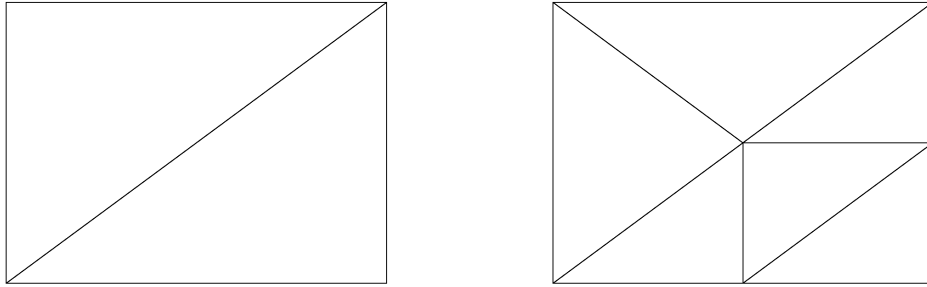


Figure 5.7: Refinement of the FEM partition The right lower triangle is part of the area to be refined. To assure the property that no hanging vertices are allowed the upper left triangle also has to be divided in two.

Then the Newton-type FEM method is executed in the same way as before with the initial conductivity  $\sigma^{(\text{old})}$  set to (5.12) instead of  $\sigma_0$ . Also the logarithmic change (5.3) is now computed with respect to (5.12), such that now inner inclusions can be detected.

The sequence of thresholds  $\theta$  is chosen in the same way as before. For each threshold  $\theta_q$  we collect all those points  $x$  satisfying

$$(F_M^\kappa)^{\text{abs}}(x) \leq \theta_q$$

respectively all those elements  $\Delta_j$  satisfying

$$|LC(j)| \geq \theta_q.$$

Then for each subdomain index  $i = 0, \dots, N_k$  each set of at least  $S$  connected points (elements) with the same contrast **and** with  $\ell_i(x) = 1$  is considered as a subdomain and stored. The predecessor of the new subdomains created from these points is then set to  $i$ , such that this way we can create subdomains at inner inclusion levels. As for the creation of our starting population we then select the best  $P$  new domain constellations with respect to the full fitness functional and complete them into individuals.

When we create new subdomains at inner inclusion levels, for Split and Merge the same rules as for the creation of the starting population are applied. For the lists

predecessor and successor we denote by  $j_{\text{new}}$  the global subdomain index of a newly created subdomain and by  $j$  the global subdomain index of its predecessor subdomain. Then we set  $\text{predecessor}(j_{\text{new}}) = \{j\}$  and  $\text{successor}(j) = \text{successor}(j) \cup \{j_{\text{new}}\}$ . Note that no other individual can contain an instance of some  $\tilde{j} \in \text{Split}(j) \cup \text{Merge}(j)$ , as by assumption when executing the above procedure all individuals of the parent individual contain the same domain constellation. Therefore we do not have to make other modifications to predecessor and successor.

Finally the newly created (parent) population (containing the new subdomains) has to be merged together with the old (parent) population (not containing the new subdomains). This is done by a modified version of the intermediate step described in 5.4.1: Instead of creating  $Q_2$  individuals via recombination we create the first  $2P$  individuals of the intermediate population as a replication of the individuals contained in the old and new parent population. The remaining  $Q_2 - 2P$  individuals are then created by recombination, where the parent individuals can be drawn from both parent populations. The rest of the generation step is then performed in the usual way, such that we end up with one population containing  $P$  individuals. So we can continue the EA in the same way as before. Note that by proceeding this way it is possible that none of the newly found subdomains are finally taken. This is done deliberately for the case that the true configuration does not contain any more subdomains than we already have found before the execution of the procedure described in this section.

### 5.4.3 Partial data sets

For the complete electrode model the use of the substitution function  $\omega$  leads to the need of a high discretization of the exterior boundary  $\Gamma_0$ . Therefore also the computational costs for the iterative solution of (2.64)–(2.66) increase considerably. To reduce them we try not to use all available boundary data pairs  $(U^k, I^k)$  for (2.64)–(2.66), but only a randomly selected part of them. To compare the performance of this modification with respect to using all data sets, we also try the same in the continuum case.

Following the considerations made in subsection 2.1.1 we admit that for each unknown subdomain  $D$  we need at least one boundary data pair for the reconstruction of its shape and another one to reconstruct its conductivity. Therefore if an individual contains  $N$  subdomains we set the minimum number of data sets to  $2N$ . The maximum number is set to  $4N$ .

When working with partial boundary data the previously described parts of the EA are extended in the following way:

- In the creation of our starting population 5.2 we assign to each individual a randomly created set of  $2N$  indices for the boundary data pairs.
- For the *recombination* 5.3.1 we create for each parent individual a binary list indicating for each boundary data index if it is included in the index list of that

individual. For each new individual we create this list by discrete recombination, analogously to the subdomain inclusion lists. As long as the number of indices in the new list is not between  $2N$  and  $4N$  this recombination is repeated with newly chosen weights.

- In the first part of the *mutation* 5.3.2 we add with probability  $p$  a randomly selected index to the list, if it contains less than  $4N$  indices. Equivalently a randomly chosen index is removed from the list with probability  $p$ , if it contains more than  $2N$  indices.

In the second part of the mutation we then execute our boundary-element method on the system consisting only of the linearized equations (2.64) respectively (2.64)–(2.66) for those boundary data pairs, whose indices are contained in the list. Analogously to the choice of the regularization parameters we hope that this way we find not only better approximations to the given problem, but also those boundary data pairs which contribute the most to the solution of the inverse problem.

- Finally, if the number of subdomains within an individual changes - due to a splitting or merging operation or by adding new subdomains by the procedure described in 5.4.2 - we randomly add indices to the list or remove indices from it, until it contains between  $2N$  and  $4N$  indices.

## 5.5 A flow diagram of the algorithm

To summarize this chapter we will finally illustrate our evolutionary algorithm by a flow diagram to illustrate how the different components described previously are related to each other. This flow diagram of the 'final' algorithm looks as follows:

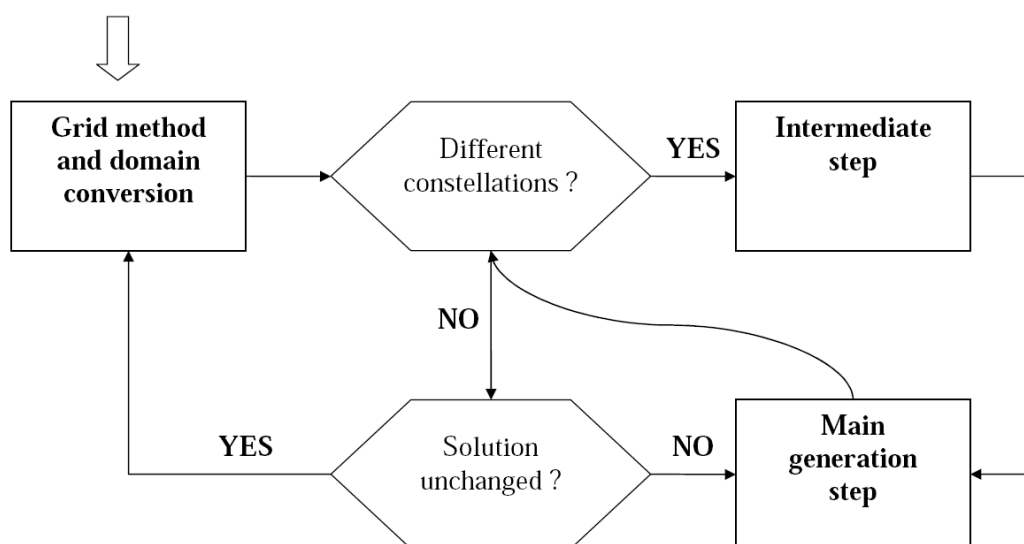


Figure 5.8: Flow diagram of the algorithm

- We start with the grid-based method (factorization method or FEM) and create an initial population for the EA as described in section 5.2.
- Then the main generation step consisting of recombination, mutation and selection described in section 5.3 is iteratively executed. If the population contains several domain constellations, we create  $Q$  children individuals via recombination and reduce the size of the children population to  $\tilde{Q}$  by the intermediate step described in 5.4.1. Otherwise we only create  $\tilde{Q}$  children individuals via recombination.
- If the best individual did not change during  $C_{\max}$  generations, we interrupt the EA and look for new subdomains as described in subsection 5.4.2.
- The newly created population is then merged together with the old population. The EA is then continued with the population we obtained via this merging procedure.

Concerning the stopping criterion, for exact data we execute a predefined number of generation steps of our algorithm. For noisy data we interrupt the EA as soon as the configuration which the best individual represents can only be distinguished from the true configuration at a precision which is below the given noise level. For details we refer to chapter 6.

# Chapter 6

## Numerical examples

In this last chapter we will now present various results of our algorithm for different configurations to illustrate its capabilities, but also its limits. We implemented the algorithm using MATLAB 7, and it was executed on Linux PCs with a 1.8 Ghz AMD Opteron CPU and 3 GB memory.

We will start with the description of the setup and with the presentation of the examples before we then present the results obtained for synthetic data in the continuum and the complete electrode model. In the last section we will describe the additional difficulties that occur when dealing with real data before finally presenting the results for a simple example with real data.

### 6.1 Parameter and example choice

As there are many parameters of the problem under consideration and also of the evolutionary algorithm we cannot make enough tests to vary all of them. Instead we have to choose which parameters of the EA should be kept fixed, and we have to choose particular examples.

#### 6.1.1 Parameters of the evolutionary algorithm

We start with a list of the parameters we used during our evolutionary algorithm. They were used during all runs if not otherwise stated. We always set the exterior curve  $\Gamma_0$  as the unit circle. For exact data we ran the algorithm in all cases over 200 generations and stored the result after each 10 generations.



The following parameters apply for both the continuum and the complete electrode model:

Parameter	Symbol	Value
background conductivity	$\sigma_0$	1
discretization points for inner curves		
direct problem		128
inverse problem		32
minimum number of connected points for domain conversion	$S$	5
maximum number of subdomains	$\hat{N}$	6
initial polynomial degree	$m^*$	2
maximum polynomial degree		5
initial Tikhonov parameter for $\psi$	$\alpha_\psi^*$	0.01
initial Tikhonov parameter for $r$	$\alpha_r^*$	1
number of parent individuals	$P$	4
number of children individuals	$Q$	40
size intermediate population	$\tilde{Q}$	12
probability of change in mutation	$p$	0.4
interval for multiplication of Tikhonov parameters for mutation	$[s_{\min}; s_{\max}]$	$[0.1; 0.9]$
number of iterations in mutation	$It$	20
shrink factor in the intersection case	$\beta$	0.9
maximum number of shrinks per subdomain		5
probability of random subdomain removal in the intermediate step	$\hat{p}$	0.4
number of generations before a new application of the grid-based method	$C_{\max}$	5

The following parameters only apply for the continuum model:

Parameter	Symbol	Value
discretization points on $\Gamma_0$		
direct problem		128
inverse problem		32
number of boundary data pairs	$M$	16
gridpoints for factorization method		$40 \times 40$

The following parameters only apply for the complete electrode model:

Parameter	Symbol	Value
number of electrodes	$L$	16
discretization points per electrode		
direct problem		40
inverse problem exact data		10
inverse problem noisy data		5
contact impedance	$z$	0.005
relative length of an electrode		50%
initial regularization parameter for FEM	$\alpha$	0.001
maximum iterations for FEM	$\hat{\mu}$	30
number of layers in the initial FEM mesh		4
maximum change between two FEM iterations to interrupt FEM	$\varepsilon$	$10^{-5}$
maximum number of elements for mesh refinement	$\aleph$	40

As we will see in subsection 6.1.3 for the CEM the number of (maximum) boundary data pairs is automatically determined by the number of electrodes.

### 6.1.2 Fitness functional and substitution function

The norms of the difference for the fitness functionals (5.1) and (5.2) are computed pointwise. We denote by  $n$  the number of discretization points on  $\Gamma_0$  and by  $x_j$  the  $j$ -th discretization point of  $\Gamma_0$ . We further denote by  $f_{\mathcal{I},k}(x_j)$  the value of  $f_{\mathcal{I}}(g_k)$  at the point  $x_j$  and by  $(U_{\mathcal{I}})_\ell^k$  the value of  $U_{\mathcal{I}}(I^k)$  at the  $\ell$ -th electrode. So we have for the partial fitness functional

$$\mathcal{F}_{\tilde{k}}(\mathcal{I}) := \|f_{\mathcal{I}}(g_{\tilde{k}}) - f_{\tilde{k}}\|_1 := \sum_{j=1}^n |f_{\mathcal{I},\tilde{k}}(x_j) - f_{\tilde{k}}(x_j)|$$

respectively

$$\mathcal{F}_{\tilde{k}}(\mathcal{I}) := \|U_{\mathcal{I}}(I^{\tilde{k}}) - U^{\tilde{k}}\|_1 := \sum_{\ell=1}^L |(U_{\mathcal{I}})_\ell^{\tilde{k}} - U_\ell^{\tilde{k}}| \quad (6.1)$$

with randomly selected  $\tilde{k}$ ,  $1 \leq \tilde{k} \leq M$ , and for the complete fitness functional

$$\mathcal{F}(\mathcal{I}) := \sum_{i=1}^M \|f_{\mathcal{I}}(g_i) - f_i\|_1 := \sum_{i=1}^M \sum_{j=1}^n |f_{\mathcal{I},i}(x_j) - f_i(x_j)|$$

respectively

$$\mathcal{F}(\mathcal{I}) := \sum_{i=1}^M \|U_{\mathcal{I}}(I^i) - U^i\|_1 := \sum_{i=1}^M \sum_{\ell=1}^L |(U_{\mathcal{I}})_\ell^i - U_\ell^i|. \quad (6.2)$$

We furthermore define the *relative distinguishability*  $\mathcal{G}(\mathcal{I})$  of an individual as

$$\mathcal{G}(\mathcal{I}) := \frac{\mathcal{F}(\mathcal{I})}{\sum_{i=1}^M \|f_i\|_1} \quad (6.3)$$

respectively

$$\mathcal{G}(\mathcal{I}) := \frac{\mathcal{F}(\mathcal{I})}{\sum_{i=1}^M \|U^i\|_1}. \quad (6.4)$$

It indicates at which precision the input data has to be given to distinguish between the approximation that the individual represents and the true configuration. For noisy data this means that an approximation whose relative distinguishability is lower than the noise level cannot be distinguished from the true configuration. This fact will be used for the stopping criterion of our evolutionary algorithm in the case of noisy data. For further description we refer to the subsections 6.2.4 and 6.3.2.

For the choice of the substitution function defined through (2.45)-(2.46) to treat the singularities occurring in the complete electrode model we define by  $t_\ell$  the parameter such that the  $\ell$ -th electrode starts at  $z_0(t_\ell)$ . We also define

$$\varpi_\ell(t) := 2\pi \frac{t - t_\ell}{|\mathcal{E}_\ell|}$$

for  $t \in \mathcal{E}_\ell$ . Then for the *direct problem* we choose the function

$$\omega(t) = \begin{cases} t_\ell + |\mathcal{E}_\ell| \frac{\varpi_\ell(t)^3}{\varpi_\ell(t)^3 + (2\pi - \varpi_\ell(t))^3} & \text{if } z_0(t) \in \mathcal{E}_\ell \\ t & \text{if } z_0(t) \in \mathcal{N} \end{cases} \quad (6.5)$$

and for the *inverse problem*

$$\omega(t) = \begin{cases} t - \frac{|\mathcal{E}_\ell|}{2\pi} \sin(\varpi_\ell(t)) & \text{if } z_0(t) \in \mathcal{E}_\ell \\ t & \text{if } z_0(t) \in \mathcal{N} \end{cases} \quad (6.6)$$

which both fulfill the condition (2.48).

### 6.1.3 Adjacent and trigonometric current patterns

As input current patterns we take *trigonometric* current patterns for the continuum case and *trigonometric* and *adjacent* current patterns for the CEM case.

In [10] it has been shown that trigonometric current patterns are optimal for the distinction of a circular inhomogeneity which lies at the center of  $D$ . Although it has not been shown that they are also the best current patterns for general conductivity distributions, we will use them as input current for the synthetic data.

In the continuum case they are given by

$$g_{2n-1}(\theta) = \cos(n\theta)$$

and

$$g_{2n}(\theta) = \sin(n\theta) \quad (6.7)$$

for  $n = 1, \dots, M/2$ , and in the CEM case they are given by

$$I_\ell^{2n-1} = \cos\left(n\frac{2\pi\ell}{L}\right)$$

and

$$I_\ell^{2n} = \sin\left(n\frac{2\pi\ell}{L}\right) \quad (6.8)$$

for  $n = 1, \dots, L/2$ . We note that  $I^L = -I^{L-1}$  so we only have  $L - 1$  linearly independent current patterns and omit the last one.

However in practice the question which input current is used depends rather on practical assumptions. For that reason very often adjacent current patterns are used as physically they can be produced quite easily. This is also the case for the example we considered for real data, and so we tried this kind of current patterns also for our synthetic examples in the CEM case. For the adjacent patterns current is applied to two neighboring electrodes such that for  $n = 1, \dots, L - 1$  we have

$$I_\ell^n = \begin{cases} 1 & \ell = n \\ -1 & \ell = n + 1 \\ 0 & \text{otherwise} \end{cases} \quad (6.9)$$

and  $I_L^L = 1$ ,  $I_1^L = -1$  for the  $L$ -th current pattern. So in this case we have  $M = L$  different current patterns.

Intuitively one can also argue in favor of the adjacent patterns that if the inhomogeneity lies near the boundary, one should rather take current patterns, which are large at the part of the boundary near the inhomogeneity and small on those parts which are further away.

### 6.1.4 Presentation of the examples for the synthetic data

Now we present the configurations we used as examples. As the focus does not lie on the reconstruction of different shapes but on different geometries and conductivities, we limit ourselves in all examples to *bean-shaped subdomains*. Their shape is given by the parameterization

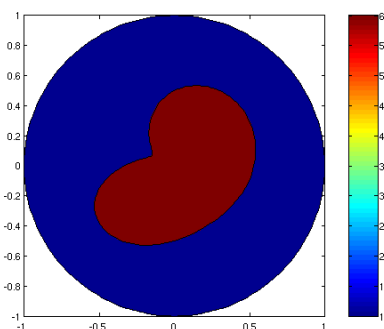
$$z(t) = C \frac{0.5 + 0.4 \cos t + 0.1 \sin 2t}{1 + 0.7 \cos t} (\cos t, \sin t), \quad t \in [0; 2\pi],$$

with a (varying) scaling factor  $C$ . They are non-convex but - as numerical tests showed - can already be quite well approximated by a trigonometric polynomial of degree 4, such that our choice of a maximum polynomial degree of 5 is justified. The capability of the algorithm to reconstruct different shapes was already illustrated in the initial

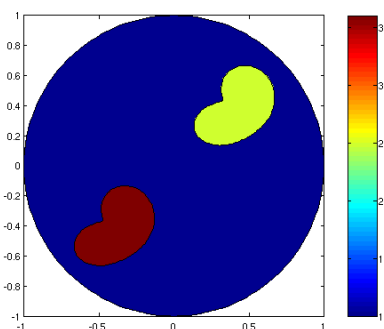
paper [39] for the Dirichlet case. Our numerical tests showed similar results for the transmission case, which however will not be presented here.

Concerning the geometries we consciously also chose some more complicated examples to look for the limits of the presented algorithm. The following examples have been chosen:

### Examples 1 and 2



$$\sigma_1 = 6$$

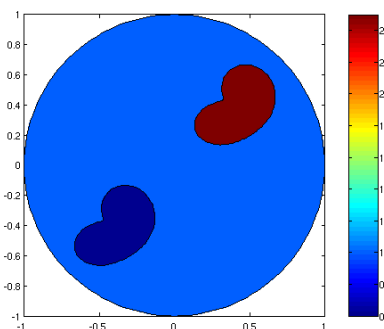


$$\sigma_1 = 2.5, \sigma_2 = 3.6$$

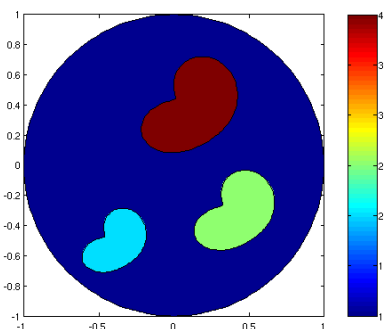
We start with the simple example of one subdomain in the middle of  $D$ . Here we do not expect any difficulties of the algorithm to find the correct configuration.

As a second example we choose two subdomains with equal contrast at the same inclusion level. Also here we do not expect serious difficulties.

### Examples 3 and 4



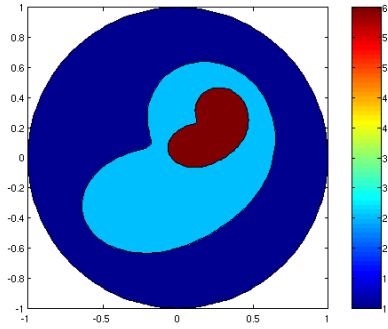
$$\sigma_1 = 2.5, \sigma_2 = 0.6$$



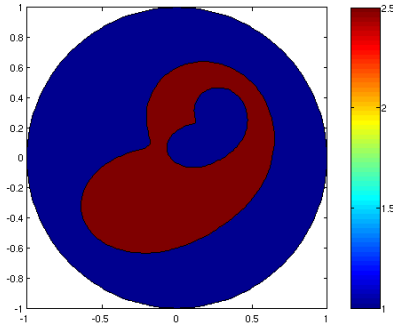
$$\sigma_1 = 4, \sigma_2 = 2, \sigma_3 = 2.5$$

Then the conductivity of the second subdomain is changed from 3.6 to 0.6 such that we now have two subdomains with a different contrast.

As our last example with one inclusion level we choose three subdomains.

**Examples 5 and 6**

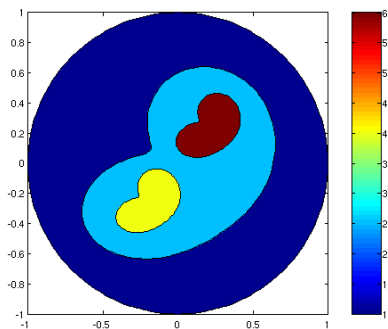
$$\sigma_1 = 2.5, \sigma_2 = 6$$



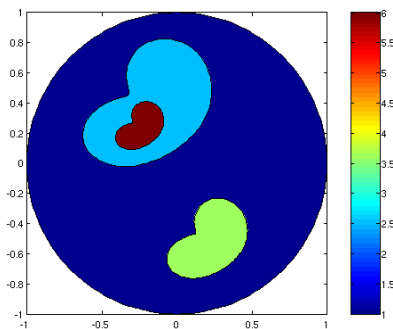
$$\sigma_1 = 2.5, \sigma_2 = 1$$

We then proceed with the examples with more than one inclusion level. We start with the case of one larger subdomain containing a smaller one where the contrast between the conductivities is positive on both interface curves.

Next we keep the interface curves and change the conductivity of the inner subdomain such that we now have a different contrast on the inner interface curve. We also tested this example with some  $\sigma_2 < \sigma_0 = 1$  but in these cases not even the outer subdomain could be reconstructed correctly. So  $\sigma_2 = 1$  is the lowest value we were able to choose.

**Examples 7 and 8**

$$\sigma_1 = 2.5, \sigma_2 = 6, \sigma_3 = 4$$



$$\sigma_1 = 2.5, \sigma_2 = 6, \sigma_3 = 3.6$$

Then we took example 5 again and added a second subdomain inside. Hence the task is to distinguish two inclusions at the second inclusion level.

Finally we choose two subdomains on the first inclusion level where a third subdomain is contained in one of them.

We also tested the algorithm for one example with three inclusion levels, but as the relative distinguishability of the innermost subdomain was too small we were not able to locate it, even with exact data.

## 6.2 Results for the continuum model

We start with the presentation of the results for the continuum model. Besides using the algorithm in the form described in chapter 5 we also made some runs with modified versions to test some features of the algorithm. So for exact data we also tested the performance of the splitting feature by starting with one subdomain in the middle as initial guess instead of using the factorization method. For noisy data we ran the algorithm 6 times for each example with and without the right domain configurations given beforehand. In all cases we ran the algorithm once for all configurations using the complete boundary data set and once we ran it using only a part of them as described in subsection 5.4.3. However we will not show the plots of all runs in this chapter.

We further measured the computing time for example 1 (containing one subdomain) and example 4 (containing 3 subdomains). This was done for one iteration step of the boundary-element method (including the creation of the system matrix) and for one whole generation step with  $\tilde{Q} = 12$  children individuals. There, as most time-consuming operations, we executed for each individual  $It = 20$  iteration steps of the BEM within the mutation, and we computed one partial fitness functional for the selection.

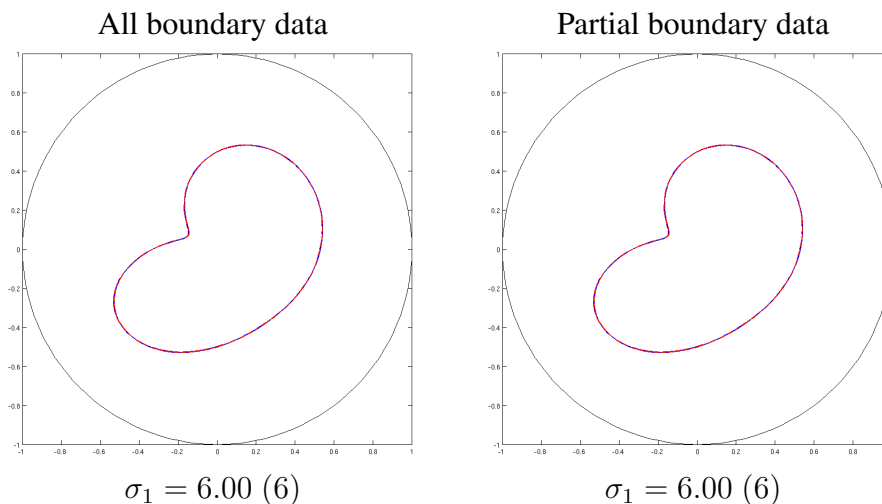
For example 1 the computing time for one iteration step was about 0.4 seconds when using all boundary data sets and about 0.2 seconds using only a part of them. For example 4 we needed about 4 seconds using all data sets and 0.3 seconds using partial data sets.

For one generation step the algorithm needed about 85 seconds for example 1 using all data sets, but only about 9 seconds using partial data sets. For example 3 we even had a more considerable speedup: The algorithm needed about 15 minutes using all data sets and only about 80 seconds using partial data sets.

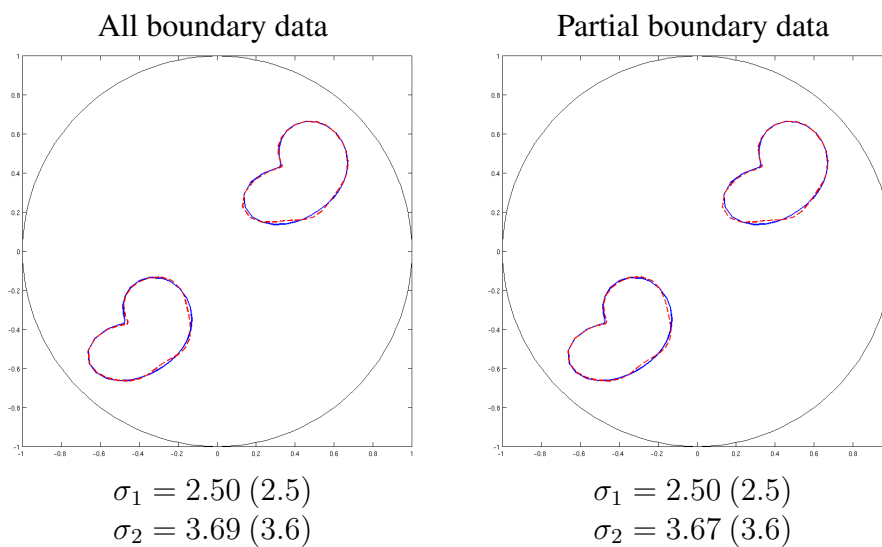
### 6.2.1 One inclusion level

For each of the four examples with one inclusion level we ran the algorithm as described in chapter 5 using the factorization method, once with the complete boundary data set and once with only a partial data set. The random number generator was set to the same value at the beginning of the evolutionary algorithm in both cases.

For each figure the result for the first case will be shown on the left and the result for the second case on the right. The solid blue lines indicate the interface curves of the true configuration, and the dotted red lines indicate the approximation found by the algorithm.

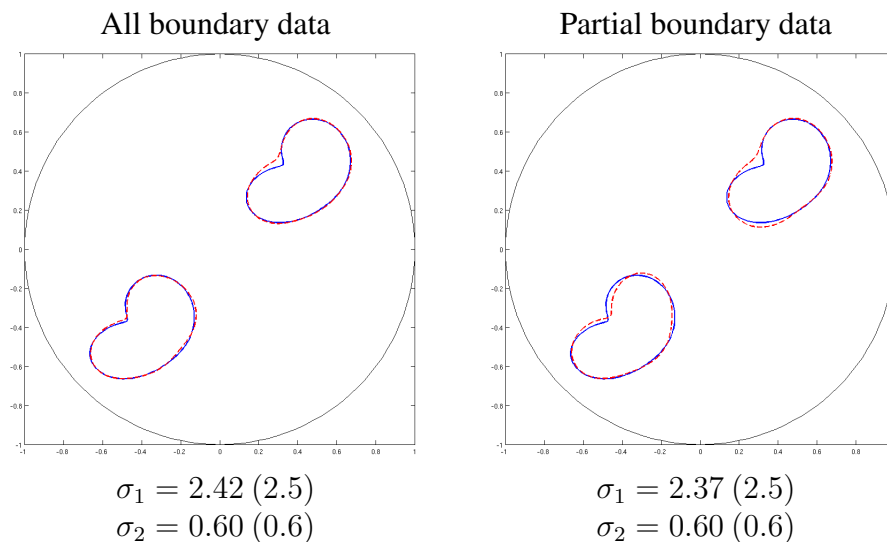
**Example 1:**

Here the approximated configuration was exactly the same for both runs and practically corresponds to the true configuration.

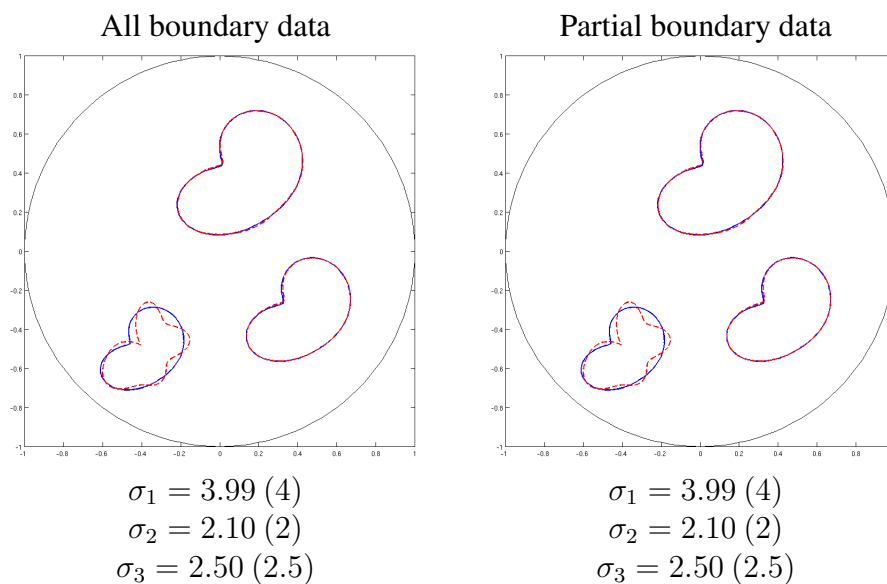
**Example 2:**

Also in this case the exact shape has been practically found but there have been small errors in the conductivity of the second subdomain.



**Example 3:**

Here the shape has not been exactly found. However the relative distinguishability is only  $2.25 \cdot 10^{-4}$  for the left solution and  $1.44 \cdot 10^{-4}$  for the right solution.

**Example 4:**

Here the algorithm has found two of the three subdomains almost perfectly, but obviously it had some problems with the shape of the smallest subdomain. The relative distinguishability between the approximated and the true configuration is only  $1.2 \cdot 10^{-4}$  for both cases.

## 6.2.2 Results without using the factorization method

We also tested the performance of the splitting feature to find out if the subdomains can be found without using the factorization method for the initial guess. For this we always started with a circle of radius 0.5 centered at the origin with initial conductivity  $\sigma^* = 2$ . As we did not use the factorization method, the algorithm was not interrupted after  $C_{\max}$  generations, and we did not try to find inner inclusions. For that reason we only ran the algorithm this way for the examples having only one inclusion level, i.e. the examples 1-4. Also here we ran the algorithm with the complete boundary data set as well as with partial data sets.

For the examples 1, 2 and 4 the algorithm was able to find the right number of subdomains, and the results looked similar to those presented previously. For that reason we do not show the plots here. The configuration from example 3, however, could not be found due to the different contrast of the two subdomains.

## 6.2.3 Multiple inclusion levels

When we tried to apply the algorithm as described in chapter 5 to the examples with more than one inclusion level, it turned out that unfortunately the factorization method did not work when the shape of the outer inclusion was not **exactly** known. So to be able to solve these examples we have to provide some a-priori information to the algorithm in another way. This was done in two different ways:

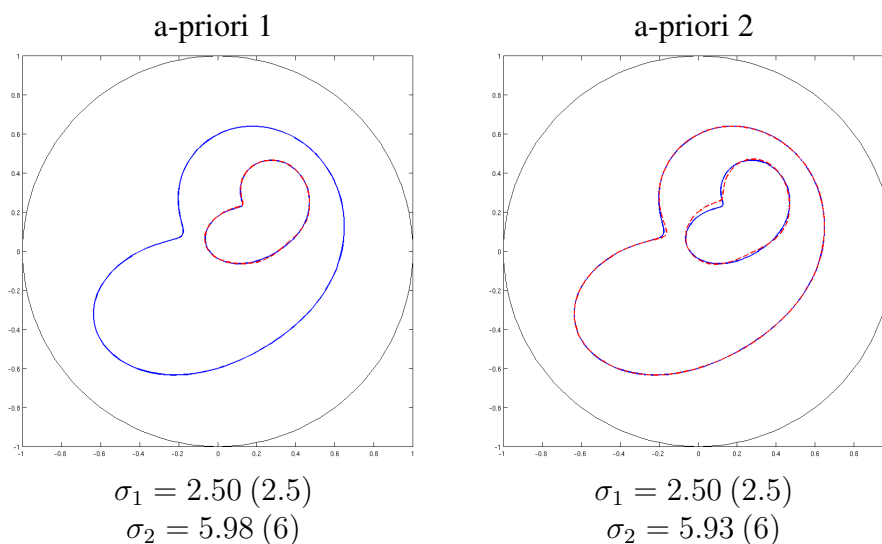
- We fixed the shape of the outer subdomain beforehand, approximated its conductivity by the bisection algorithm according to theorem 3.1 and then applied the factorization method as described in subsection 3.1.2. Then the evolutionary algorithm was applied to the population created this way without being interrupted again. The shape of the outer subdomain remained fixed, but its conductivity was allowed to vary.
- We predefined the location of **all** subdomains beforehand. For this at the beginning we created for each original subdomain a circle with radius 0.1 centered at the same place as the original subdomain. Its conductivity was set to  $\sigma^* = 2$  or  $1/\sigma^* = 0.5$ , depending on the contrast of the original subdomain.

However, when starting the algorithm we only created circles for the subdomains at the outermost inclusion level. Then the algorithm was run until we had  $C_{\max}$  generations without a change of the best individual. After that, instead of applying the factorization method, we inserted circles for the subdomains at the next inclusion level into each individual and continued the algorithm with the modified population. Here, unlike for the first method, the shape of *all* subdomains was allowed to vary over the whole run. However, in this case we deactivated the splitting and merging feature.

For the sake of completeness the second method was also applied to the examples 1-4 but did not produce any results which differed considerably from those presented in 6.2.1 and 6.2.2.

For the examples 5-8 we will plot on the left-hand side of each figure the result obtained for the a-priori given shape of the outer subdomain (which will be called 'a-priori 1') and on the right sides the results for the a-priori given subdomain locations ('a-priori 2'). They were obtained using the partial boundary data sets, but using the complete boundary data sets was leading to similar results.

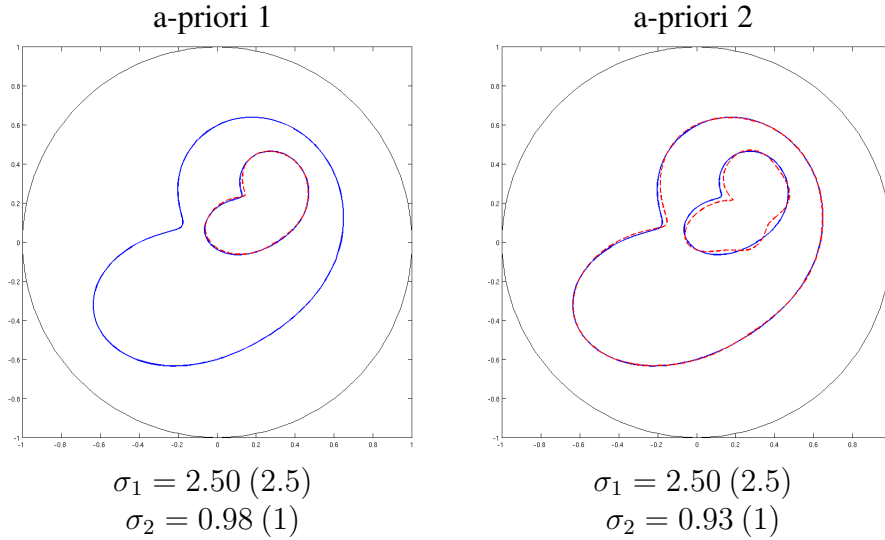
### Example 5:



Here the shape of the inner subdomain has been exactly found, when the curve of the outer subdomain was a-priori known and the factorization method was used to locate the inner inclusion. When the outer subdomain was not fixed beforehand, we had to increase the maximum polynomial degree for the outer subdomain to 8 to obtain the result shown on the right.

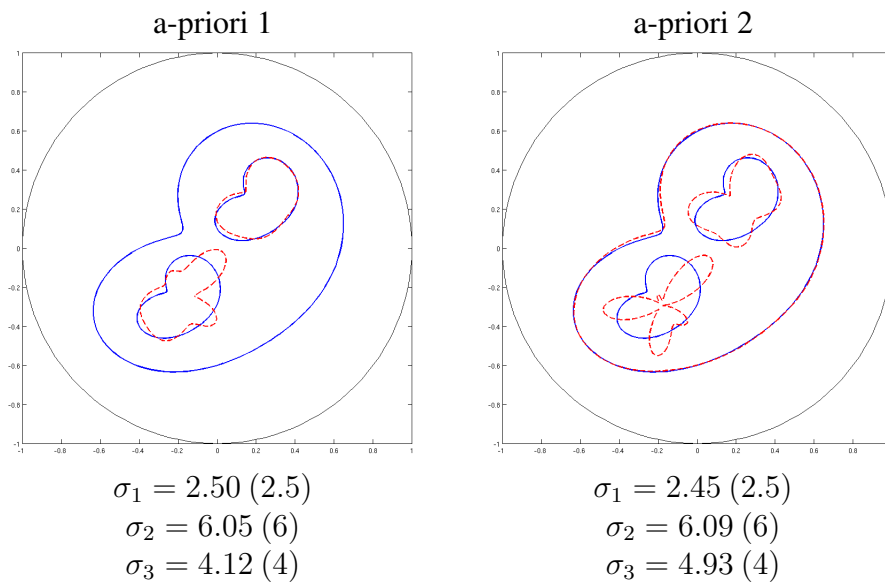
### Example 6:

For example 6 the conductivity was fixed beforehand, as theorem 3.1 does not apply in the case of a contrast change. However it was allowed to vary afterwards during the execution of the generation steps.



Also here the location and then the shape of the inner subdomain was found exactly when the outer subdomain was a-priori known, but some small errors occurred if this was not the case. This time the approximation also did not improve when we increased the maximum polynomial degree for the outer subdomain as in example 5. The relative distinguishability for the configuration on the right-hand side is  $2.25 \cdot 10^{-4}$ .

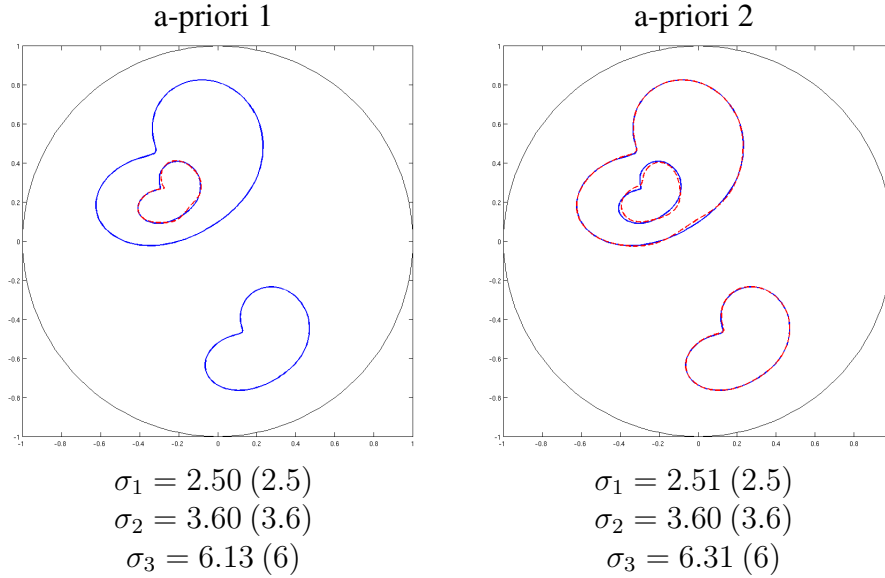
### Example 7:



For this example the second inner inclusion could not even be found when the outer subdomain was a-priori known, and the result looked even worse when the shape of the outer subdomain was also unknown. However the relative distinguishability to the

true solution is only  $1.6 \cdot 10^{-4}$  for **both** solutions. So this example illustrates very well the ill-posedness of the given problem.

**Example 8:**



Finally for the last example no great difficulties occurred for exact data. The difficulties for this example rather lie in the case of noisy data as we will see in the next section.

### 6.2.4 Noisy data

To simulate noisy data we added randomly created noise to the values of the Dirichlet data  $f$  to obtain the noisy data  $f^\delta := f + \delta$ . It was created such that it was equally distributed and that for a given noise level  $\hat{\delta}$  we have for the 2 norm

$$\frac{\|f^\delta - f\|}{\|f\|} \leq \hat{\delta} \quad (6.10)$$

As the factorization method turned out to be very sensitive to noise and did not work for most noisy cases, we tested the algorithm with a-priori given positions of the subdomains as in the second method of section 6.2.3. For the examples 2 and 4 we also tested the algorithm with one a-priori given subdomain as in section 6.2.2. As noise levels  $\hat{\delta}$  we chose 1%, 2% and 5%.

Furthermore the algorithm was interrupted as soon as the *noisy relative distinguishability*

$$\mathcal{G}^\delta(\mathcal{I}) := \frac{\sum_{i=1}^M \|f_{\mathcal{I}}(g_i) - f_i^\delta\|_1}{\sum_{i=1}^M \|f_i^\delta\|_1} = \frac{\sum_{i=1}^M \|f_{\mathcal{I}}(g_i) - f_i - \delta_i\|_1}{\sum_{i=1}^M \|f_i + \delta_i\|_1} \quad (6.11)$$

of the best individual was lower than the given noise level  $\hat{\delta}$  because obviously after that the algorithm would only optimize with respect to the noise. Especially note that as  $\|\delta\| \ll \|f\|$  we have

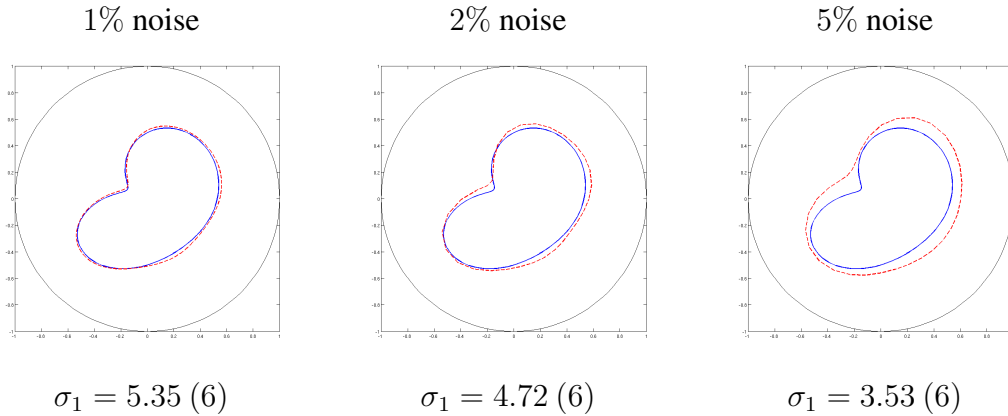
$$\sum_{i=1}^M \|f_i^\delta\|_1 = \sum_{i=1}^M \sum_{j=1}^n |f_i(x_j) + \delta_i(x_j)| = \sum_{i=1}^M \sum_{j=1}^n |f_i(x_j)| + \sum_{i=1}^M \sum_{j=1}^n \text{sgn}(f_i(x_j)) \delta_i(x_j)$$

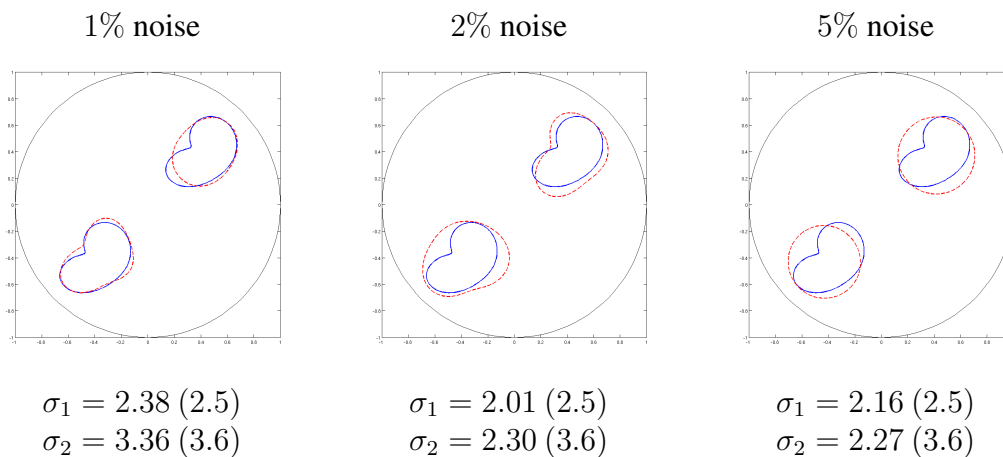
where the second sum is quite small because the statistical mean of  $\delta$  over all values is zero. So we have  $\|f^\delta\| \approx \|f\|$  which justifies the use of  $\|f^\delta\|$  in the denominator of (6.11).

We also ran the algorithm 6 times for each configuration to illustrate the stochastic character of the algorithm, where however over all runs the noise was kept the same for each example.

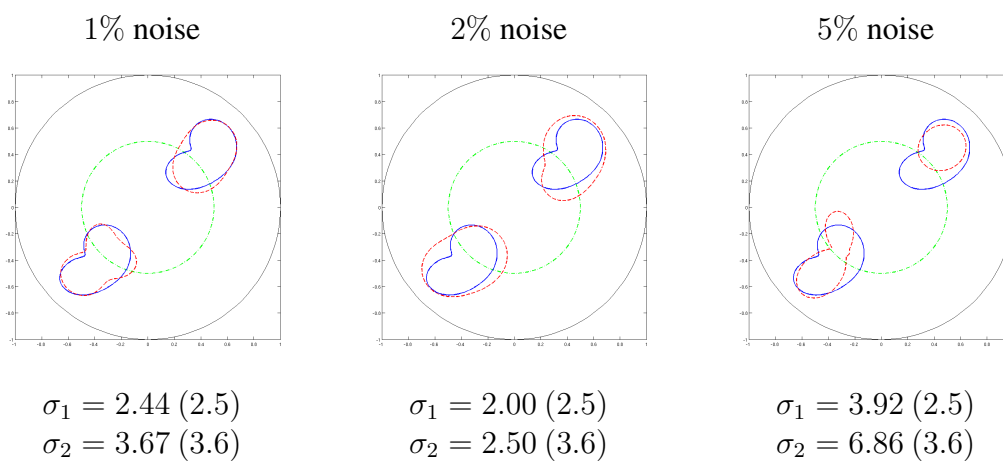
In the following we will always present the results for the first run with partial boundary data for each noise level with a-priori given subdomain locations, and for the examples 2 and 4 we will also present the results for the configuration with one initial subdomain. For the examples with two inclusion levels in some cases the algorithm already stopped before the inner inclusions could be inserted. For these cases obviously no plot will be shown.

### Example 1:



**Example 2:**

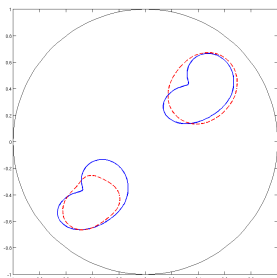
The results for one initial subdomain with application of the splitting feature look as follows :



The green dotted line indicates the initial guess.

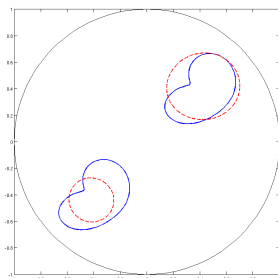
**Example 3:**

1% noise



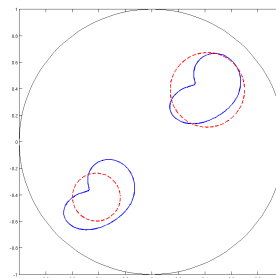
$$\begin{aligned}\sigma_1 &= 2.10 \text{ (2.5)} \\ \sigma_2 &= 0.52 \text{ (0.6)}\end{aligned}$$

2% noise



$$\begin{aligned}\sigma_1 &= 2.08 \text{ (2.5)} \\ \sigma_2 &= 0.36 \text{ (0.6)}\end{aligned}$$

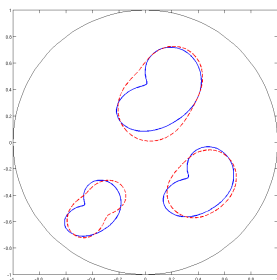
5% noise



$$\begin{aligned}\sigma_1 &= 2.27 \text{ (2.5)} \\ \sigma_2 &= 0.31 \text{ (0.6)}\end{aligned}$$

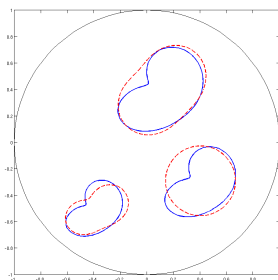
**Example 4:**

1% noise



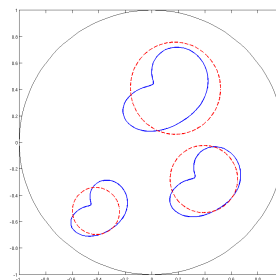
$$\begin{aligned}\sigma_1 &= 3.21 \text{ (4)} \\ \sigma_2 &= 2.05 \text{ (2)} \\ \sigma_3 &= 2.26 \text{ (2.5)}\end{aligned}$$

2% noise



$$\begin{aligned}\sigma_1 &= 3.06 \text{ (4)} \\ \sigma_2 &= 2.00 \text{ (2)} \\ \sigma_3 &= 2.36 \text{ (2.5)}\end{aligned}$$

5% noise

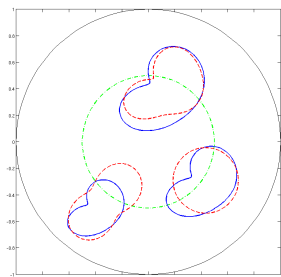


$$\begin{aligned}\sigma_1 &= 2.41 \text{ (4)} \\ \sigma_2 &= 2.06 \text{ (2)} \\ \sigma_2 &= 2.76 \text{ (2.5)}\end{aligned}$$



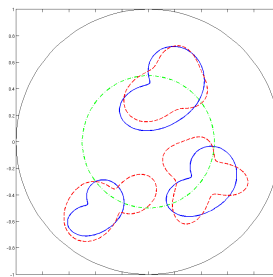
The results for one initial subdomain look as follows :

1% noise



$$\begin{aligned}\sigma_1 &= 5.14 (4) \\ \sigma_2 &= 1.70 (2) \\ \sigma_3 &= 2.40 (2.5)\end{aligned}$$

2% noise

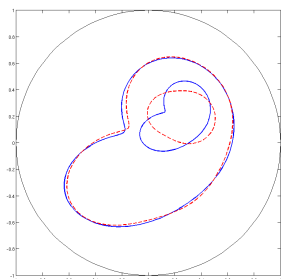


$$\begin{aligned}\sigma_1 &= 4.36 (4) \\ \sigma_2 &= 1.48 (2) \\ \sigma_3 &= 2.34 (2.5)\end{aligned}$$

For a noise level of 5% the algorithm was not able anymore to find the right number of subdomains.

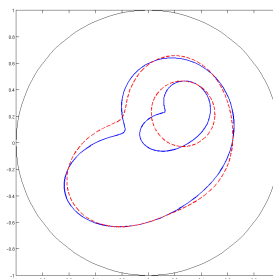
### Example 5:

1% noise



$$\begin{aligned}\sigma_1 &= 2.63 (2.5) \\ \sigma_2 &= 5.98 (6)\end{aligned}$$

2% noise

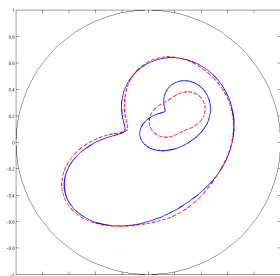


$$\begin{aligned}\sigma_1 &= 2.52 (2.5) \\ \sigma_2 &= 5.26 (6)\end{aligned}$$

Here for the noise level of 5% the relative distinguishability of the best individual already was below the noise level before the inner inclusion was inserted.

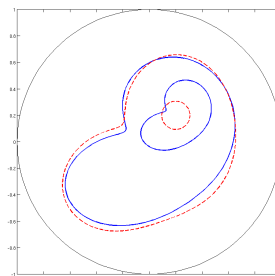
**Example 6:**

1% noise



$$\begin{aligned}\sigma_1 &= 2.30 \text{ (2.5)} \\ \sigma_2 &= 0.90 \text{ (1)}\end{aligned}$$

2% noise

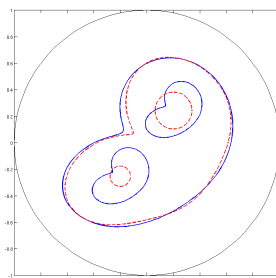


$$\begin{aligned}\sigma_1 &= 2.05 \text{ (2.5)} \\ \sigma_2 &= 0.75 \text{ (1)}\end{aligned}$$

Also here for the noise level of 5% the relative distinguishability was below the noise level before the inner inclusion was inserted.

**Example 7:**

1% noise



$$\begin{aligned}\sigma_1 &= 2.88 \text{ (2.5)} \\ \sigma_2 &= 6.36 \text{ (6)} \\ \sigma_2 &= 5.93 \text{ (4)}\end{aligned}$$

Here the relative distinguishability was already below the noise level of 2% before introducing the inner subdomains. Also for the noise level of 1% only one generation step could be executed after introducing the inner subdomains.

**Example 8:**

Finally for the last example the inner subdomain could not even be introduced for a noise level of 1% as the relative distinguishability of the inner subdomain is only 0.0083. So obviously in this case the inner subdomain cannot be recovered from noisy data.

### 6.2.5 Summary and interpretation of the results

For exact data the boundary element method in combination with the evolutionary algorithm was able to find an approximation which differed from the true configuration with a relative distinguishability of about  $10^{-4}$ . For the examples 1,2 and 8 this approximation corresponded practically to the true configuration whereas for the examples 3-7 some errors occurred. Nevertheless for the examples 3-6 the reconstruction was still quite good, which was not the case anymore for example 7. This example particularly illustrates the ill-posedness of the given problem.

For noisy data we especially had difficulties with the examples with multiple inclusion levels. For 5% noise the algorithm for any example was not able to locate the inner subdomain. This particularly illustrates the difficulty to find inner inclusions and the fact that the relative distinguishability is quite small in these cases. For the examples with one inclusion level indeed the results for noisy data seem to be quite satisfactory.

At the first inclusion level the method was able to find the right number of subdomains via using only the splitting feature, as long as the contrast of all subdomains was the same. This even still worked for most of the noisy cases. This illustrates that the incorporation of a feature of the previously mentioned level-set methods into this boundary-element method considerably increases its possibilities.

In all cases there was no significant difference between the results obtained using always the full set of boundary data and those using only a varying partial set of them whose size depends on the number of unknown subdomains. On the other hand we saw that when using only partial data sets the computing time considerably decreased. This fact will justify the use of partial data sets for the complete electrode model.

Concerning the factorization method it turned out to be extremely sensitive to noise. In most cases it was neither able to locate the correct number of subdomains for noisy Dirichlet data  $f$ , nor to find an inner inclusion when the shape of the outer subdomain was not exactly known - which also corresponds to some type of noise. For exact data, however, it seems to be a fast method for locating the number and position of a set of unknown subdomains, which can be easily implemented. Especially it also seems to work for contrast changes, although this case is still not justified theoretically.

## 6.3 Results for the complete electrode model using synthetic data

In this section we will present the results for the complete electrode model. For the continuum model we did not see any significant differences between the results using the complete boundary data set and those using partial data sets. On the other hand the computing time for one generation step was significantly lower. Therefore, for the CEM we ran the algorithm only with partial data sets. Nevertheless, for those examples with unsatisfactory results we then additionally tried to use the full data set,

but the results did not improve. Unlike for the continuum model we only ran the algorithm exactly as described in chapter 5 and did not look at special cases like a-priori information or relying on the splitting feature. For that reason we will present the results also in those cases where the correct configuration has not been found, without testing how much a-priori information would be needed to find it anyway.

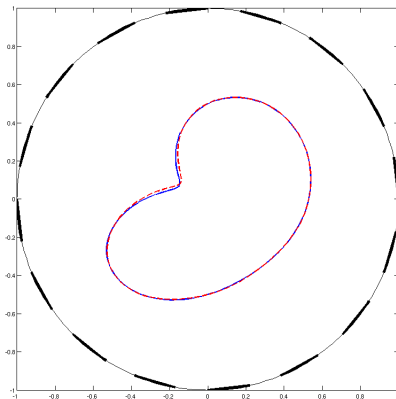
Like for the continuum model we also measured the computing time for one iteration step of the boundary-element method and for one generation step of the evolutionary algorithm for the examples 1 and 4. For example 1 the time needed for one iteration step was about 4 seconds; the time for one generation step was about 7 minutes. For example 4 the boundary-element method needed about 17 seconds for one iteration step, and for one generation step the evolutionary algorithm needed about 30 minutes.

### 6.3.1 Results for exact data

For exact data we ran the algorithm for each example once for adjacent current patterns and once for trigonometric current patterns as input data. We discretized each electrode with 10 discretization points. We also tried to use only 5 discretization points per electrode but the results were less satisfactory. In all figures the left plot will show the result for adjacent and the right plot the result for trigonometric current patterns. The bold parts of the exterior boundary indicate the electrodes. Unlike the factorization method the FEM was able to find inner inclusions also in those cases where the outer subdomain was not exactly known. For that reason we will present the results for multiple inclusion levels in the same section as those for one inclusion level.

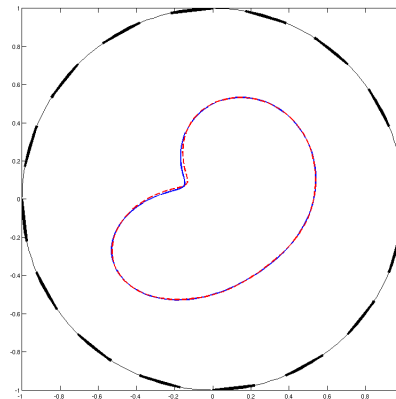
#### Example 1:

adjacent patterns



$$\sigma_1 = 5.98 (6.0)$$

trigonometric patterns

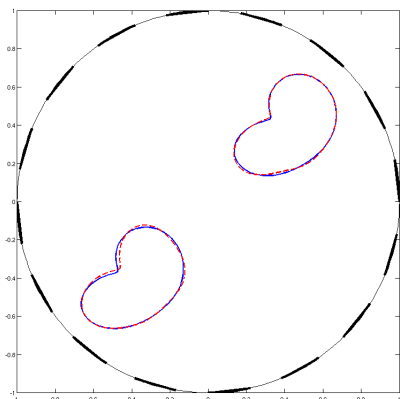


$$\sigma_1 = 6.02 (6.0)$$

As in the continuum model the algorithm did not have difficulties in the reconstruction of one subdomain. However the conductivity is not exactly found this time.

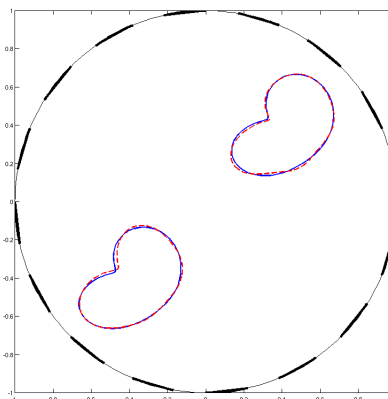
**Example 2:**

adjacent patterns



$$\begin{aligned}\sigma_1 &= 2.49 (2.5) \\ \sigma_2 &= 3.57 (3.6)\end{aligned}$$

trigonometric patterns

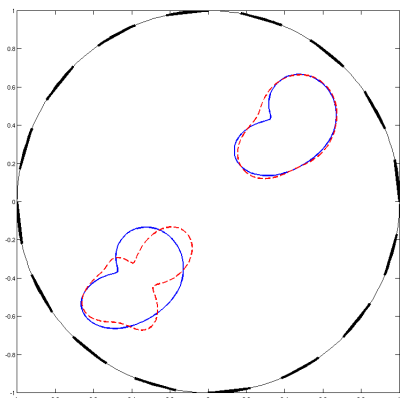


$$\begin{aligned}\sigma_1 &= 2.51 (2.5) \\ \sigma_2 &= 3.59 (3.6)\end{aligned}$$

Also for this example the results are not worse than for the continuum model.

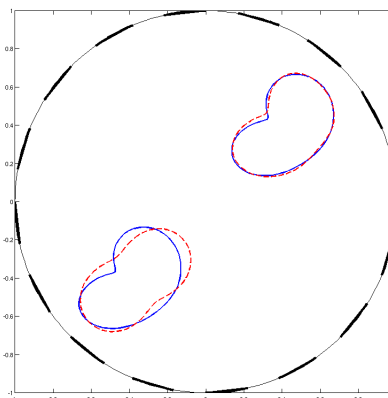
**Example 3:**

adjacent patterns



$$\begin{aligned}\sigma_1 &= 2.39 (2.5) \\ \sigma_2 &= 0.57 (0.6)\end{aligned}$$

trigonometric patterns



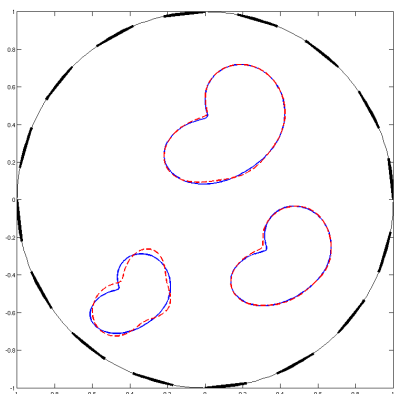
$$\begin{aligned}\sigma_1 &= 2.47 (2.5) \\ \sigma_2 &= 0.61 (0.6)\end{aligned}$$

Here the algorithm was not able anymore to find the correct shape of the subdomain with the negative contrast as it has been able for the continuum model. The number of

subdomains, their locations and the approximate conductivity have still been found.

**Example 4:**

adjacent patterns

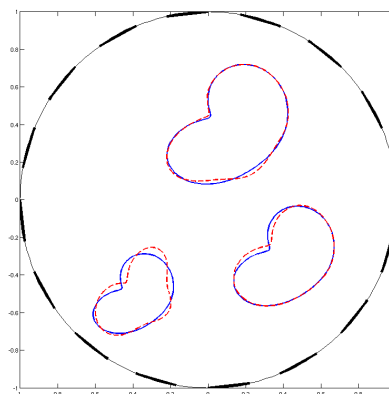


$$\sigma_1 = 3.97 (4.0)$$

$$\sigma_2 = 2.46 (2.5)$$

$$\sigma_3 = 1.96 (2.0)$$

trigonometric patterns



$$\sigma_1 = 3.95 (4.0)$$

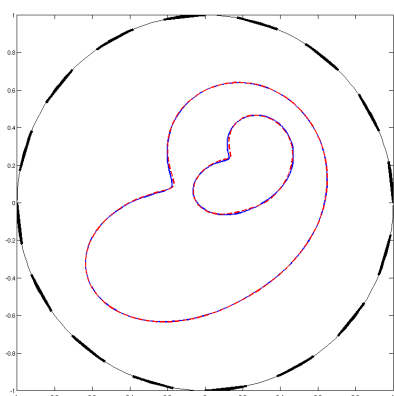
$$\sigma_2 = 2.48 (2.5)$$

$$\sigma_3 = 1.98 (2.0)$$

As for the continuum model the algorithm had also here some problems with the shape of the smallest subdomain. For the trigonometric current patterns it also seemed to have some problems with the area between the subdomains. However the reconstruction still seems to be satisfactory.

**Example 5:**

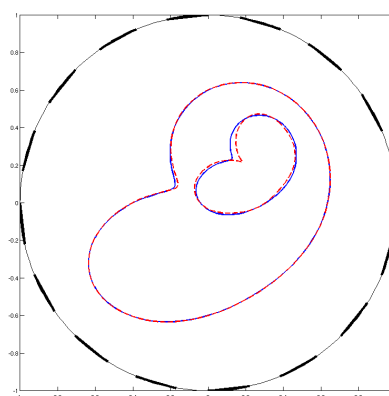
adjacent patterns



$$\sigma_1 = 2.50 (2.5)$$

$$\sigma_2 = 6.09 (6.0)$$

trigonometric patterns



$$\sigma_1 = 2.49 (2.5)$$

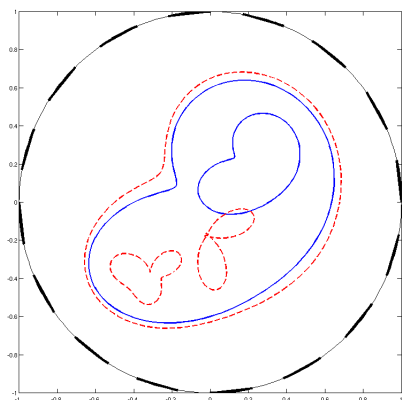
$$\sigma_2 = 6.35 (6.0)$$

As for the continuum model we had to set the maximum polynomial degree up to 8 to obtain the above results. After doing this the algorithm was able to find an exact reconstruction of the inner subdomain's shape for adjacent patterns and an almost perfect reconstruction for trigonometric patterns.

This example illustrates that the evolutionary algorithm for the complete electrode model is basically able to find inner inclusions exactly without any a-priori information.

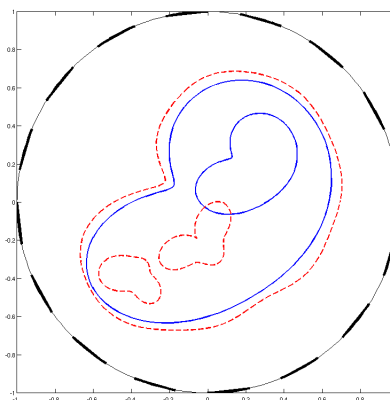
### Example 6:

adjacent patterns



$$\begin{aligned}\sigma_1 &= 1.73 (2.5) \\ \sigma_2 &= 8.98 (1.0) \\ \sigma_3 &= 5.25 (-)\end{aligned}$$

trigonometric patterns

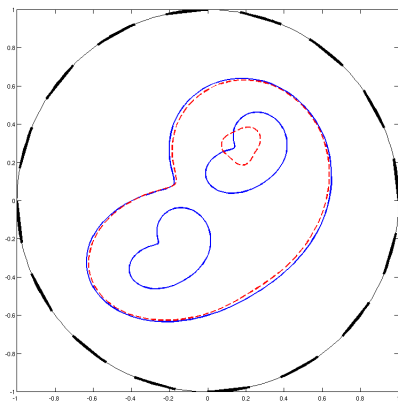


$$\begin{aligned}\sigma_1 &= 1.66 (2.5) \\ \sigma_2 &= 10.25 (1.0) \\ \sigma_3 &= 3.51 (-)\end{aligned}$$

With this example, however, the algorithm has obviously been pushed beyond its capabilities. The FEM was able to see that the conductivity was higher in the lower left half of the bean than in the upper right half, but the algorithm always put a subdomain with positive contrast in the lower left half. This did not even change when we explicitly fixed the shape and conductivity of the outer subdomain. Obviously without the right domain constellation the boundary-element method could only produce nonsense with the subdomains it was given.

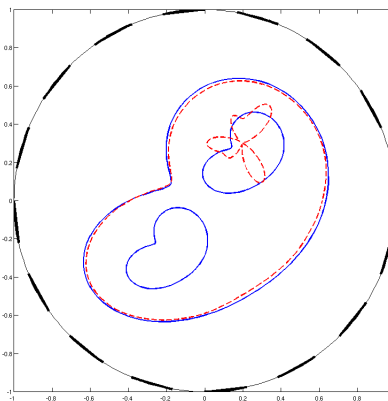
**Example 7:**

adjacent patterns



$$\begin{aligned}\sigma_1 &= 2.92 \text{ (2.5)} \\ \sigma_2 &= 22.01 \text{ (6.0)}\end{aligned}$$

trigonometric patterns

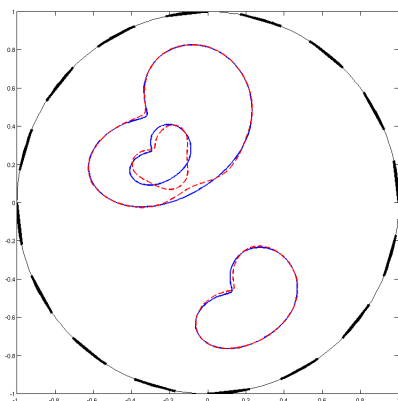


$$\begin{aligned}\sigma_1 &= 2.94 \text{ (2.5)} \\ \sigma_2 &= 6.75 \text{ (6.0)}\end{aligned}$$

Also for this example the algorithm was not able to find the correct domain constellation. This time the conductivity of the second inclusion was obviously too small to recognize the subdomain. As for the previous example the boundary-element method could not find a good approximation without knowing the correct domain constellation.

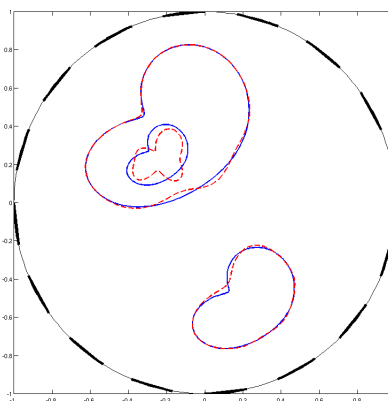
**Example 8:**

adjacent patterns



$$\begin{aligned}\sigma_1 &= 2.50 \text{ (2.5)} \\ \sigma_2 &= 3.59 \text{ (3.6)} \\ \sigma_3 &= 6.35 \text{ (6.0)}\end{aligned}$$

trigonometric patterns



$$\begin{aligned}\sigma_1 &= 2.50 \text{ (2.5)} \\ \sigma_2 &= 3.53 \text{ (3.6)} \\ \sigma_3 &= 8.45 \text{ (6.0)}\end{aligned}$$



Here the algorithm was again capable of finding the right domain constellation. The reconstruction of the subdomains at the first inclusion level is practically perfect, whereas there are some errors with the subdomain at the second inclusion level. This is probably due to the fact that the distinguishability of this subdomain is very small.

### 6.3.2 Results for noisy data

Also for the complete electrode model we tested the algorithm for noisy data. Unlike in the case of the continuum model the simulated noise was created such that for each noisy voltage value  $(U_\ell^n)^\delta$  we have

$$\frac{|(U_\ell^n)^\delta - U_\ell^n|}{|U_\ell^n|} \leq \hat{\delta} \quad (6.12)$$

i.e. we apply the noise limit criterion pointwise because taking an  $L^2$  criterion would not make sense for discrete data.

Unfortunately this means that the stochastical mean of  $\delta$  is not necessarily zero anymore, so that we cannot argue as for the continuum model and divide by  $\|U^\delta\|$  instead of  $\|U\|$  for the relative distinguishability. In fact, when we ran the algorithm this stopping criterion seemed to be too strong: In some cases for a noise level of 2% the algorithm already stopped with the initial population, although the approximation could be improved clearly in the next few generations after we turned off the criterion. Next, we tried to use the relative pointwise distinguishability

$$\frac{1}{ML} \sum_{i=1}^M \sum_{\ell=1}^L \frac{|(U_{\mathcal{I}}^i)_\ell - (U_\ell^i)^\delta|}{|(U_\ell^i)^\delta|}$$

instead, but this time the algorithm often seemed to run too long and it found many artifacts. Finally we tried the criterion

$$\max_{k=1, \dots, M} \frac{\|U_{\mathcal{I}}(I^k) - (U^k)^\delta\|_1}{\|(U^k)^\delta\|_1} \leq \hat{\delta}, \quad (6.13)$$

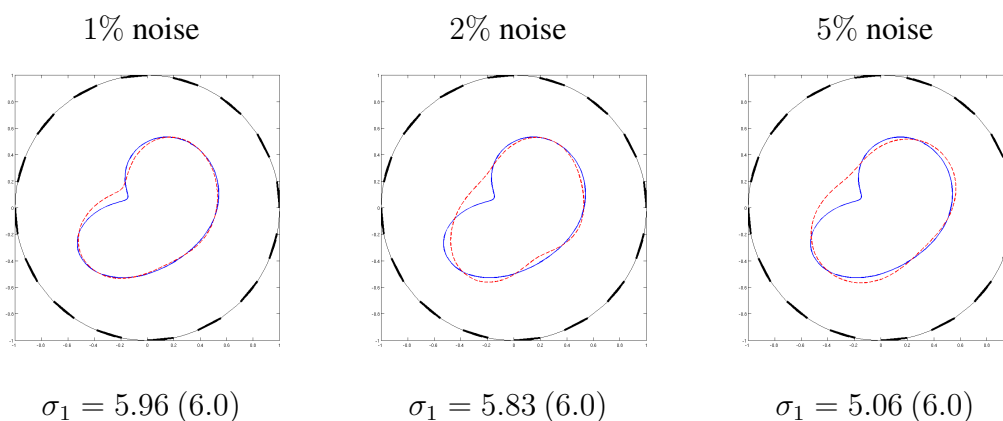
i.e. we stop the algorithm if the relative distinguishability with respect to each boundary data pair is lower than the noise level  $\hat{\delta}$ . The thought behind this was that if the configuration is still distinguishable for at least one boundary data pair the algorithm can still distinguish when using the partial fitness functional. For our test examples this criterion seemed to be the best of the three, so we kept it, although there is also no proper theoretical justification for it.

For the examples 6 and 7, where the algorithm already had problems for exact data we did not run it for noisy data. We also did not try it for example 8 because the results obtained for the continuum model already showed that the distinguishability of the inner subdomain is too small for noisy data. As for the continuum model we

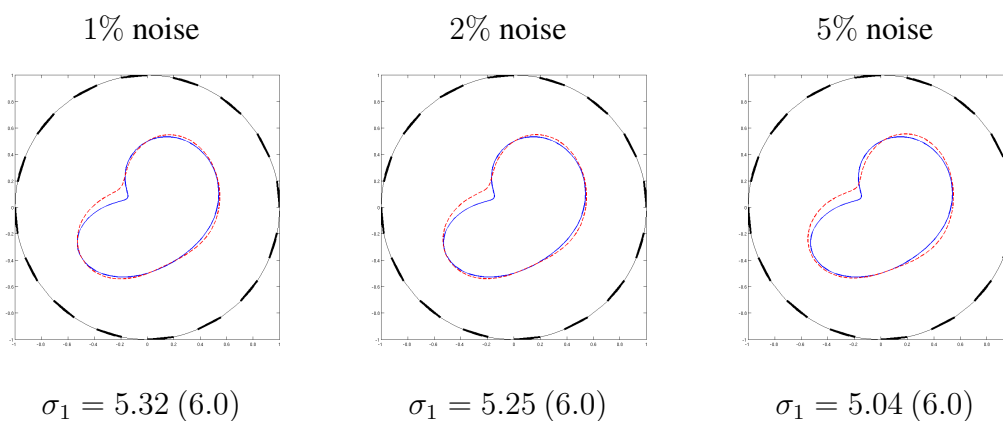
also ran the algorithm 6 times for each example and for each noise level where the noise values remained the same for each example and the random number generator was set to different initial values each time. We also ran each configuration for 10 discretization points per electrode as well as for 5, and for those examples for which this did not make a significant difference we will present the results for 5 points. As for the continuum model we will always present the result of the first run for each configuration.

### Example 1:

Results for adjacent current patterns:



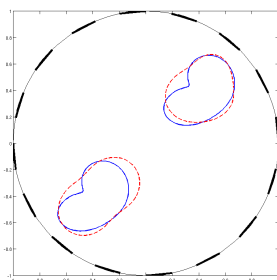
Results for trigonometric current patterns:



**Example 2:**

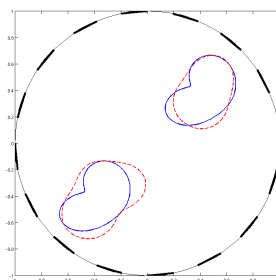
Results for adjacent current patterns:

1% noise



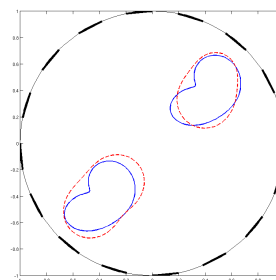
$$\begin{aligned}\sigma_1 &= 2.32 (2.5) \\ \sigma_2 &= 2.79 (3.6)\end{aligned}$$

2% noise



$$\begin{aligned}\sigma_1 &= 2.45 (2.5) \\ \sigma_2 &= 2.52 (3.6)\end{aligned}$$

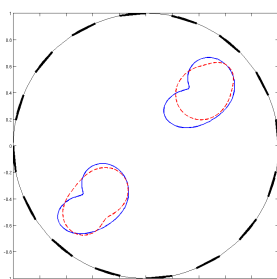
5% noise



$$\begin{aligned}\sigma_1 &= 2.18 (2.5) \\ \sigma_2 &= 2.48 (3.6)\end{aligned}$$

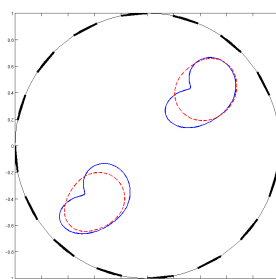
Results for trigonometric current patterns:

1% noise



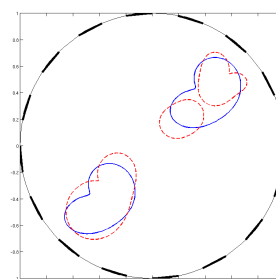
$$\begin{aligned}\sigma_1 &= 3.13 (2.5) \\ \sigma_2 &= 4.59 (3.6)\end{aligned}$$

2% noise



$$\begin{aligned}\sigma_1 &= 2.50 (2.5) \\ \sigma_2 &= 5.22 (3.6)\end{aligned}$$

5% noise

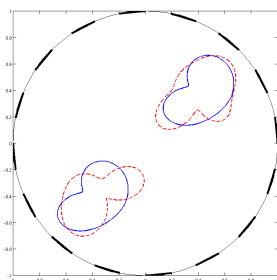


$$\begin{aligned}\sigma_1 &= 2.86 (2.5) \\ \sigma_2 &= 2.76 (3.6) \\ \sigma_3 &= 2.47 (-)\end{aligned}$$

**Example 3:**

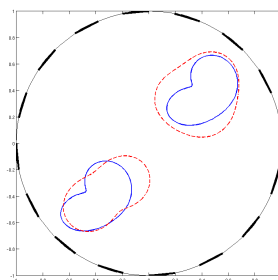
Results for adjacent current patterns:

1% noise



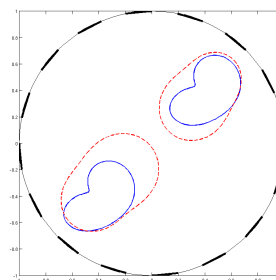
$$\begin{aligned}\sigma_1 &= 2.16 (2.5) \\ \sigma_2 &= 0.61 (0.6)\end{aligned}$$

2% noise



$$\begin{aligned}\sigma_1 &= 1.84 (2.5) \\ \sigma_2 &= 0.65 (0.6)\end{aligned}$$

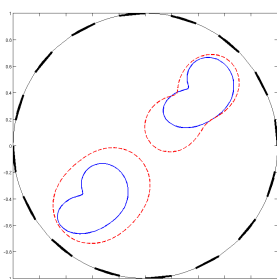
5% noise



$$\begin{aligned}\sigma_1 &= 1.90 (2.5) \\ \sigma_2 &= 0.73 (0.6)\end{aligned}$$

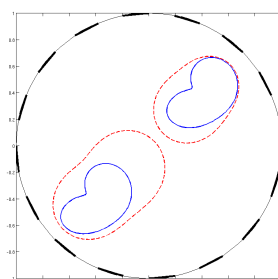
Results for trigonometric current patterns:

1% noise



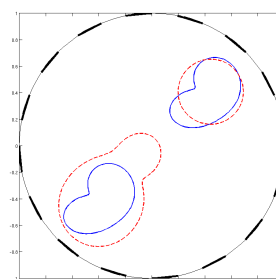
$$\begin{aligned}\sigma_1 &= 1.85 (2.5) \\ \sigma_2 &= 0.75 (0.6)\end{aligned}$$

2% noise



$$\begin{aligned}\sigma_1 &= 1.87 (2.5) \\ \sigma_2 &= 0.76 (0.6)\end{aligned}$$

5% noise

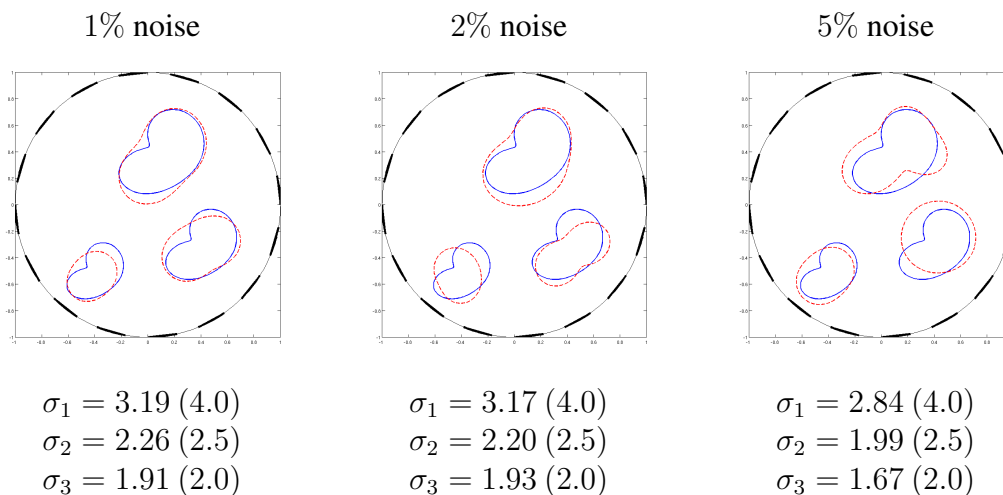


$$\begin{aligned}\sigma_1 &= 2.33 (2.5) \\ \sigma_2 &= 0.81 (0.6)\end{aligned}$$

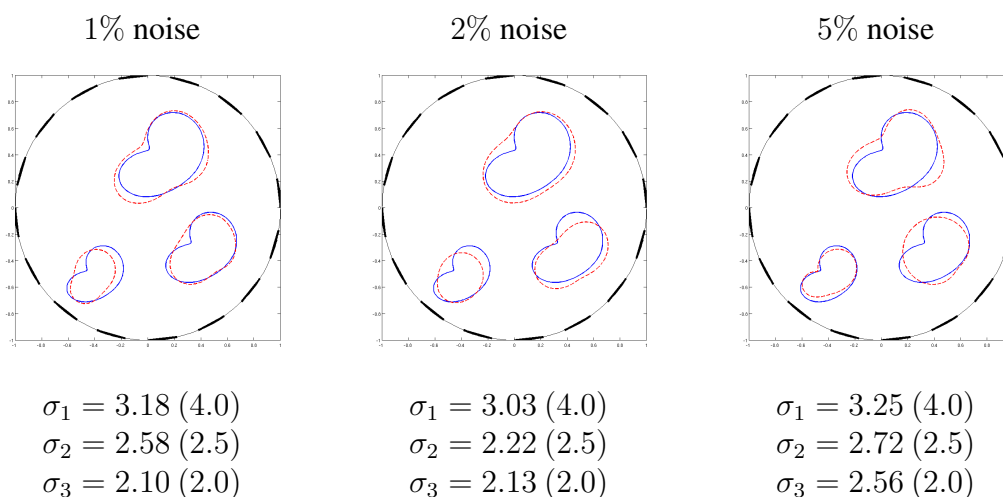
**Example 4:**

For this example we had to use 10 discretization points per electrode to obtain the presented results.

Results for adjacent current patterns:



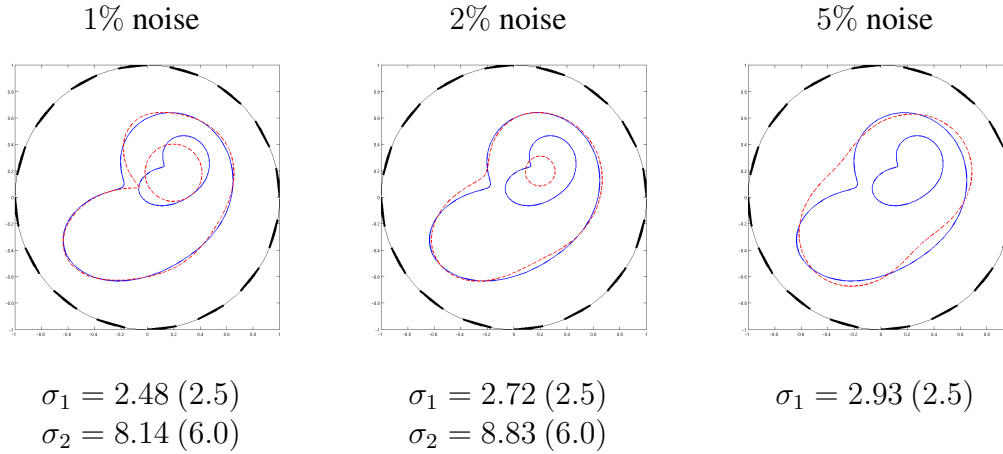
Results for trigonometric current patterns:



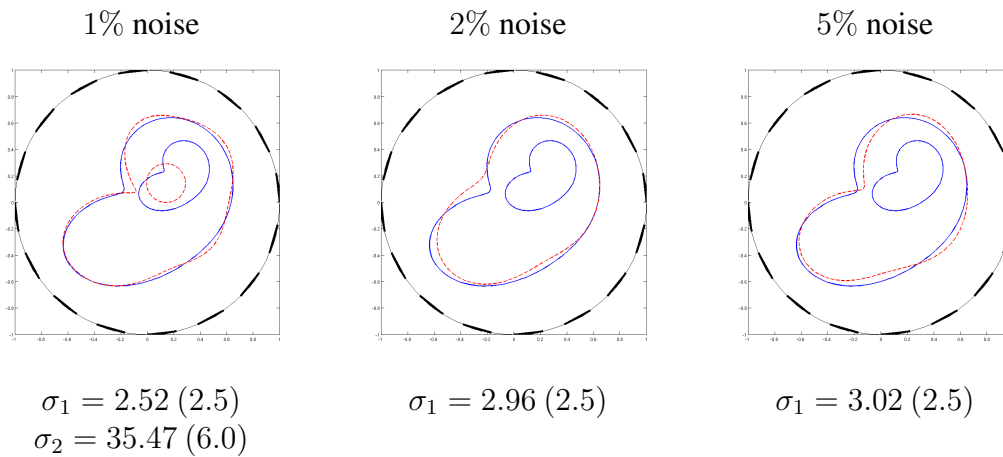
### Example 5:

For this example we had to use 10 discretization points per electrode to obtain the presented results.

Results for adjacent current patterns:



Results for trigonometric current patterns:



### 6.3.3 Summary and interpretation of the results

For exact data - as for the continuum model - the algorithm did not have significant problems to reconstruct the shape and the conductivity for the objects at one inclusion level and with the same contrast. However, for example 3 the boundary-element method was not able to find the correct shape of the subdomain with negative contrast - as it was still possible for the continuum model. In all these examples there was no significant difference between the results for adjacent and those for trigonometric current patterns.

For noisy data at one inclusion level the FEM was still able to find the right number of subdomains, especially for example 4 with 5% noise, where the splitting feature of the boundary-element method had failed. Also the shapes of the subdomains do not look worse than for the continuum model. So it seems that the boundary-element

method is appropriate also for the complete electrode model and that the FEM is less sensitive to noise than the factorization method..

For multiple inclusion levels the FEM - unlike the factorization method - was basically able to localize inner inclusions without exactly knowing the shape of the outer subdomain. This might be due to the fact that it does not proceed layerwise but is always executed on the whole grid. On the other hand knowing the exact shape of the inner subdomain does not improve its performance, because the subdomain has to be converted into an FEM-mesh representation - which discards the information about the form. This could be especially seen for the examples 6 and 7, where the inner inclusions could not even be localized when the exact shape and conductivity of the exterior subdomain were fixed.

Concerning the shape reconstruction for multiple inclusion levels the adjacent current patterns seem to be more appropriate than the trigonometric patterns. For example 5 as well as for example 8 the shape reconstruction is better in the adjacent case - which is especially remarkable if we consider the fact that the trigonometric patterns are theoretically the best ones.

For noisy data the algorithm was almost at its limits when we had multiple inclusion levels. For example 8 it was not at all able to localize the inner inclusion, and for example 5 already a noise level of 2% for trigonometric and 5% for adjacent patterns was enough to make it fail. Also for the other noise levels the results were not very good in this case. As for exact data also for noisy data using the adjacent patterns was leading to better results than using the trigonometric patterns when we had multiple inclusion levels. In any case it seems to be very difficult to find subdomains at several inclusion levels without any a-priori information.

## 6.4 Testing with real data

In this last section we finally want to describe the modifications to our method when applying the algorithm for real data. These real data have been provided by the Institute for Anesthesiology at the medical faculty of the University of Göttingen. They were obtained by placing electrodes around a saline tank and putting different objects into it. Then adjacent current patterns were applied on the electrodes. For this configuration some of the assumptions we made for the synthetic data are not valid anymore such that the program has to be modified. We will describe these modifications in the following subsections before finally presenting the results obtained for real data. To test the effects of the modifications we limited ourselves to the examples 2 and 3 as generic examples for one inclusion level.

### 6.4.1 Preparing for real data: Incomplete adjacent current patterns

The first difference to synthetic data is that with the device which generated the real data it is not possible to measure the potential at the current-carrying electrodes. So we only get *incomplete data sets*. Furthermore no absolute voltages but only *potential differences*  $\Upsilon_\ell^i$  can be measured.

These potential differences are modeled in the boundary-element method by the fact that also the voltages  $U_\ell^i$  are considered as unknowns and that the system (2.64)–(2.66) is extended for  $i = 1, \dots, M$  with the difference equations

$$U_{\ell+1}^i - U_\ell^i = \Upsilon_\ell^i \quad (6.14)$$

for  $\ell = 1, \dots, L - 1$  and

$$U_1^i - U_L^i = \Upsilon_L^i \quad (6.15)$$

for  $\ell = L$ . We note that the resulting system has range  $L - 1$ . So to guarantee uniqueness for  $U$ , we also include the grounding condition (1.2) into the system, such that we have for  $i = 1, \dots, M$

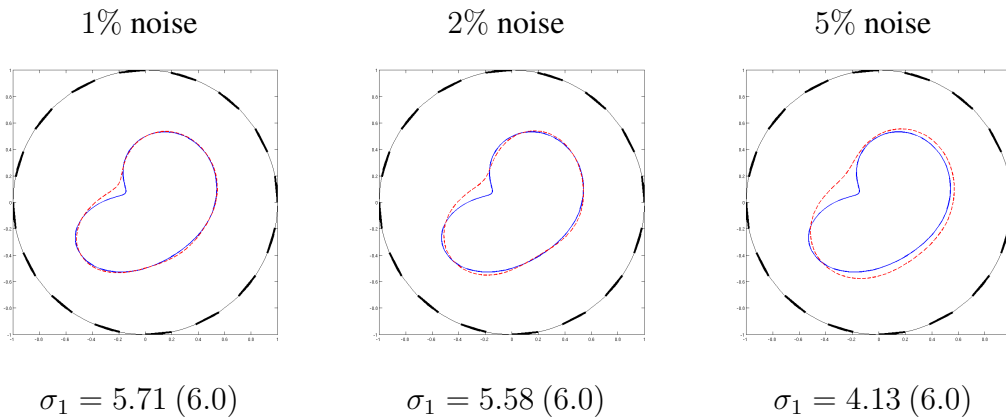
$$\sum_{\ell=1}^L U_\ell^i = 0. \quad (6.16)$$

For complete patterns the character of the system does not change this way, as the values for  $U$  can be directly computed from (6.14)–(6.16) in one single step. With incomplete data, however, the equations for the potential at the current-carrying electrodes are left out of the system, and so the values for the potential  $U$  really have to be considered as additional unknowns.

We tested the algorithm for this modification with exact data for the examples 1-5 and for noisy data for the examples 1-4. For exact data we omit the results, as this additional difficulty did not lead to significantly different results.

For noisy data the results looked as follows :

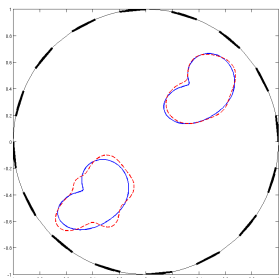
#### Example 1:





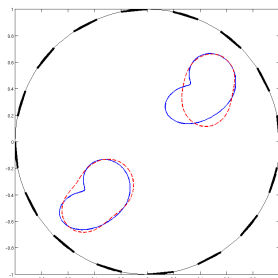
**Example 2:**

1% noise



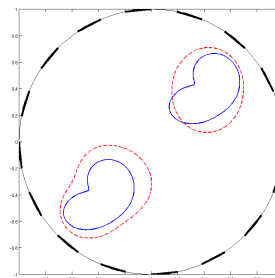
$$\begin{aligned}\sigma_1 &= 2.44 \text{ (2.5)} \\ \sigma_2 &= 3.09 \text{ (3.6)}\end{aligned}$$

2% noise



$$\begin{aligned}\sigma_1 &= 2.55 \text{ (2.5)} \\ \sigma_2 &= 3.27 \text{ (3.6)}\end{aligned}$$

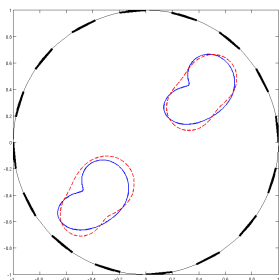
5% noise



$$\begin{aligned}\sigma_1 &= 1.86 \text{ (2.5)} \\ \sigma_2 &= 2.09 \text{ (3.6)}\end{aligned}$$

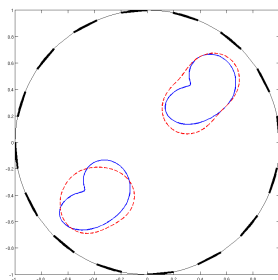
**Example 3:**

1% noise



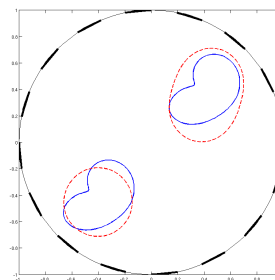
$$\begin{aligned}\sigma_1 &= 2.36 \text{ (2.5)} \\ \sigma_2 &= 0.65 \text{ (0.6)}\end{aligned}$$

2% noise

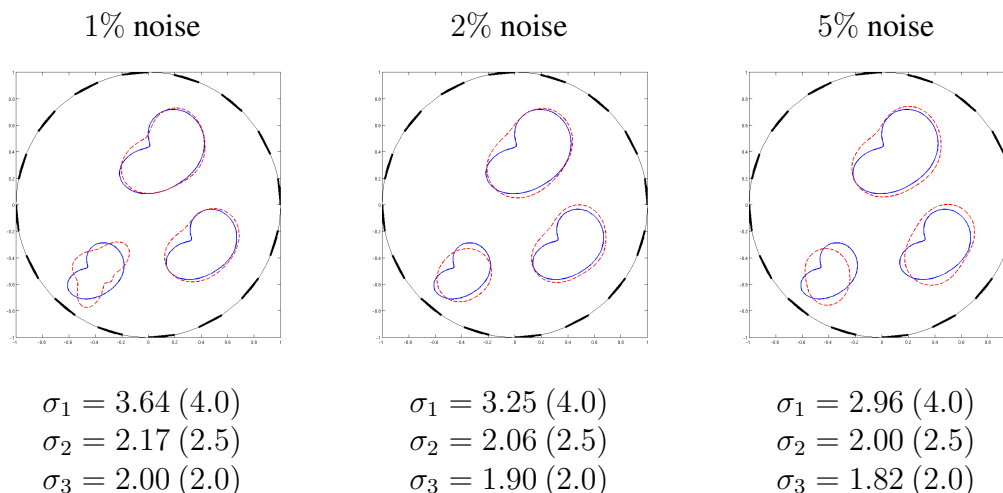


$$\begin{aligned}\sigma_1 &= 2.13 \text{ (2.5)} \\ \sigma_2 &= 0.65 \text{ (0.6)}\end{aligned}$$

5% noise



$$\begin{aligned}\sigma_1 &= 1.80 \text{ (2.5)} \\ \sigma_2 &= 0.69 \text{ (0.6)}\end{aligned}$$

**Example 4:**

So surprisingly also for noisy data there are no significant differences to the results obtained for complete data, and we obviously can omit the differences at the current-carrying electrodes without obtaining considerably worse results. Also in the following sections we will always use incomplete potential difference sets as input data.

### 6.4.2 Preparing for real data: Shrinking the electrode size

For our synthetic examples we always assumed the electrodes to cover 50% of the boundary surface. But for the device which was used to create the examples for real data they covered only about 10% of the surface. Therefore we had to examine what happens if we shrink the electrode size. For this we created synthetic data for an electrode coverage of 20% and 10% and then added **exactly** the same relative noise to the potential difference  $\Upsilon$  that we had added to the data in section 6.4.1.

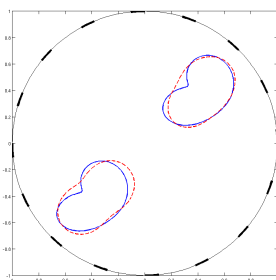
We further compared the computing time for one iteration step of the boundary-element method for example 2 with 50%, 20% and 10% electrode coverage. It was about 1 second for 50%, 4 seconds for 20% and 17 seconds for 10%. This is due to the fact that in all cases each electrode is discretized with 5 points, so that if we have less coverage we need more discretization points for the spaces between the electrodes.

For an electrode coverage of 20% and 10% we obtained the following results:

**Example 2:**

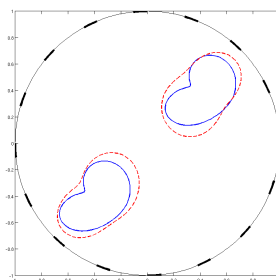
20 % electrode coverage

1% noise



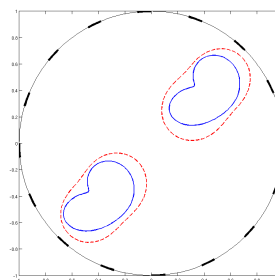
$$\begin{aligned}\sigma_1 &= 2.42 (2.5) \\ \sigma_2 &= 3.21 (3.6)\end{aligned}$$

2% noise



$$\begin{aligned}\sigma_1 &= 1.99 (2.5) \\ \sigma_2 &= 2.51 (3.6)\end{aligned}$$

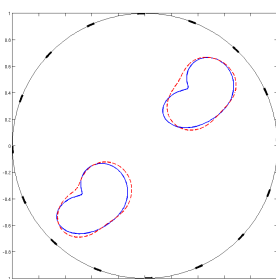
5% noise



$$\begin{aligned}\sigma_1 &= 1.78 (2.5) \\ \sigma_2 &= 2.28 (3.6)\end{aligned}$$

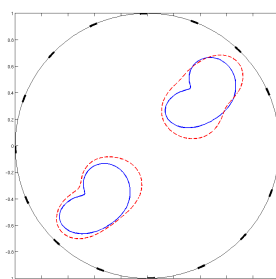
10 % electrode coverage

1% noise



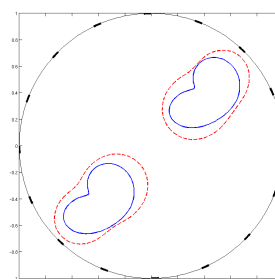
$$\begin{aligned}\sigma_1 &= 2.33 (2.5) \\ \sigma_2 &= 3.15 (3.6)\end{aligned}$$

2% noise



$$\begin{aligned}\sigma_1 &= 1.98 (2.5) \\ \sigma_2 &= 2.55 (3.6)\end{aligned}$$

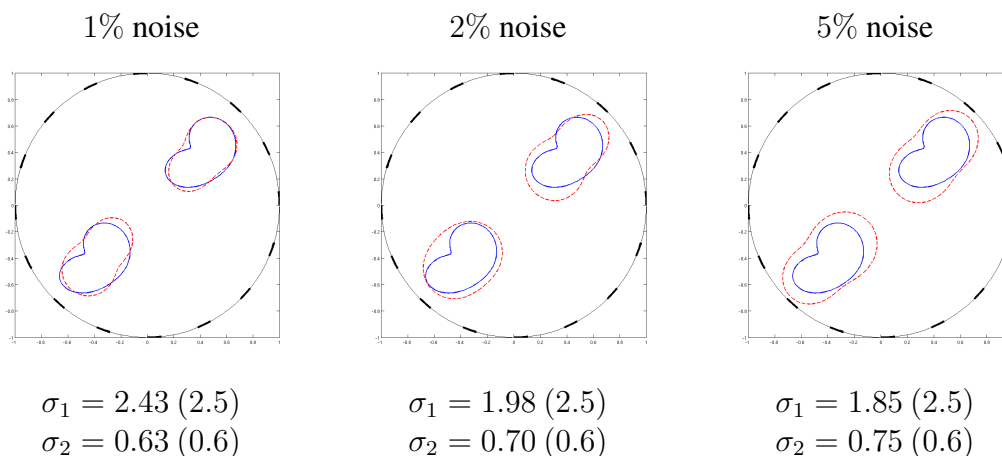
5% noise



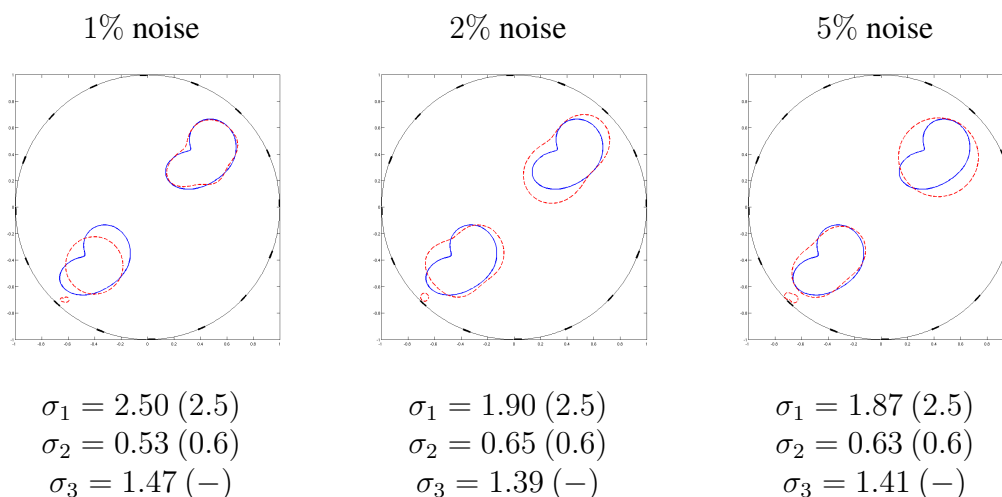
$$\begin{aligned}\sigma_1 &= 1.79 (2.5) \\ \sigma_2 &= 2.24 (3.6)\end{aligned}$$

**Example 3:**

20 % electrode coverage



10 % electrode coverage



We see that in three of the four cases the algorithm was still able to find the right number of subdomains. But although there are no large aggravations compared to the data for 50% coverage, in no case do we have a considerable improvement compared to a larger surface. Especially the results for 1% noise look worse than for a coverage of 50%. So summarizing it seems that taking smaller electrodes is certainly not an advantage for our method. Additionally it considerably increases the computational costs.

### 6.4.3 Preparing for real data: Systematic measurement errors

Although we already added pointwise noise to our synthetic data, the distribution of the relative noise was still assumed to be a uniform white noise. In reality this is

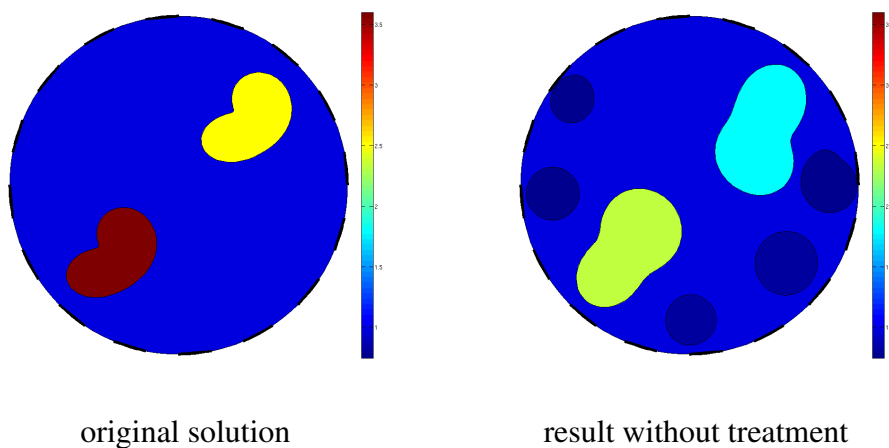
also not the case, but we rather have *systematic measurement errors* depending on the measuring device.

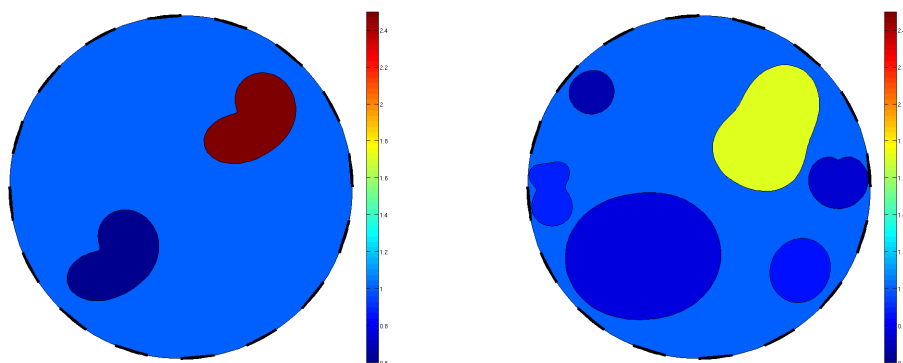
So as the next step we added some systematic relative noise to our data. This noise was obtained by using some phantom resistivity network described in [22]. It is constructed in the way that theoretically each measurement should be zero, such that the voltages which are measured can be interpreted as noise. The mean of this noise is 4.2%.

The most significant difference to the previously used white noise is that the sign of this noise is *always positive*, i.e. the measuring device usually measures a larger potential difference than there really is, but it never measures a smaller one. When we ran the algorithm it seemed that the stopping criterion with respect to (6.13) was too weak, so that we chose the stronger criterion with respect to (6.4) instead.

This time we will rather show the reconstruction as a colored plot without putting the exact solution into the same figure because in any case the shape reconstruction is very bad. So comparing the reconstructed shape to the given one within one figure would not bring any additional information. Instead we illustrate the true configuration again as a separate plot. When we applied the algorithm ‘as is’ to those data we obtained the following results:

### Example 2:



**Example 3:**

original solution

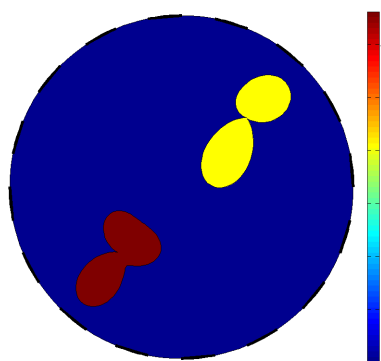
result without treatment

We see that in both cases the algorithm has created quite a lot of artifacts in addition to the real subdomains. Especially for example 3 the second subdomain with the negative contrast cannot be distinguished very well from the artifacts, as their conductivities are quite similar. A possible explanation for the artifacts is that for a noise whose stochastic mean is not zero the given background conductivity does not correspond anymore to the background conductivity we obtained via the FEM from section 3.2.

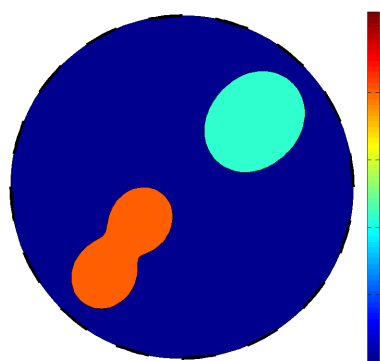
So we tried out two different modifications on the algorithm:

- During the creation of the starting population (as described in section 5.2) we compute the relative distinguishability (6.4) with respect to the *zero measurement* (no inclusions inside  $\Omega$ ) for each created subdomain. If it is lower than the average relative noise level, we delete the subdomain from the individual.
- We subtract the relative mean of the given noise from each value, such that now the mean of the new noise is close to zero.

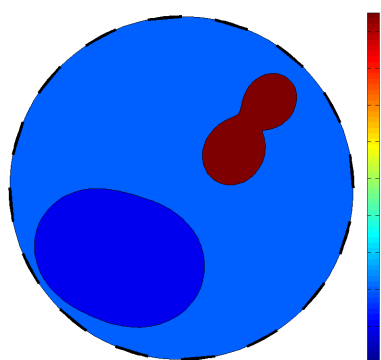
When applying each of these two modifications **separately** we obtained the following results:

**Example 2:**

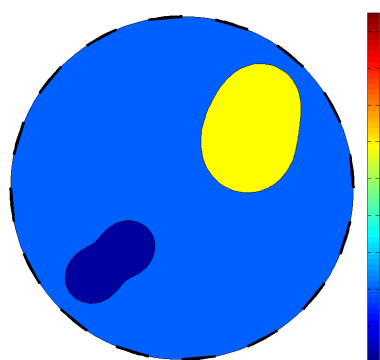
testing for distinguishability



subtraction of mean value

**Example 3:**

testing for distinguishability



subtraction of mean value

So we clearly see that both modifications have brought an improvement to the previous results, as in both cases the configuration does not contain any more artifacts. If we keep the mean of the noise as it is we see that for example 2 the conductivities are much larger than in reality, and for example 3 the second subdomain is much larger than it should be. So in this case the noise obviously leads to a stronger distinction than we really have.

#### 6.4.4 Preparing for real data: Approximate conversion from 3D data to 2D data

Another problem with real data is that the model we use is a two-dimensional model, whereas real measurements are always in three dimensions. This means that we cannot

directly use the measured values, but we have to convert them in some adequate way. Following [28] we looked at the case of a cylindrical tank with no objects inside (the so-called *zero measurement*) and at the synthetically created values for  $\Upsilon$ , once in a 2D and once in a 3D FEM model. For the first adjacent pattern the factors, with which the 2D values for  $\Upsilon$  have to be multiplied to obtain the 3D values, are the following:

electrodes	factor
3 – 4	5.13
4 – 5	3.14
5 – 6	2.31
6 – 7	1.80
7 – 8	1.53
8 – 9	1.37
9 – 10	1.33
10 – 11	1.37
11 – 12	1.52
12 – 13	1.82
13 – 14	2.29
14 – 15	3.14
15 – 16	5.14

For the zero measurement (as well as for a ball or a pole at the center of the tank) they are translation-invariant, so that these factors can also be used for all other measurements. Of course if the geometry is not translation-invariant - which is usually the case - we cannot take these values anymore to have an exact conversion. But as we do not assume to have any a-priori information, we cannot compute better values for our special cases. Therefore we have to take the above ones anyway. This creates an additional error in our data, which we call *dimension-conversion error*.

One of its effects seems to be that the value of the background conductivity  $\sigma_0$  is different for the 2D case. So although we suppose to know it beforehand we have to consider it as an additional unknown. For this we extend the equations (2.64) in an adequate way. Note that alternatively we could also have set  $f := \sigma_0 u$ , keep the equations (2.64) unchanged and compute  $1/\sigma_0$  from the equations (2.65) instead.

The fact that we do not know  $\sigma_0$  beforehand also creates problems in the domain conversion from section 5.2, as the subdomains are created by starting from the conductivity change (5.3) with respect to  $\sigma_0$ . So to find a good starting value for  $\sigma_0$  when using real data, we first started with the 3D value for  $\sigma_0$ . Then if the best individual that we found contained one big circle with a conductivity close to the initial  $\sigma_0$ , we created a new starting population with  $\sigma_0$  set to the new value instead.

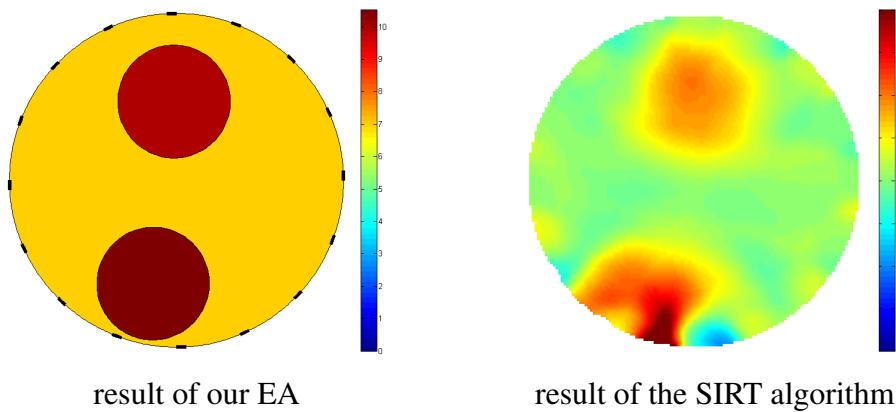


### 6.4.5 Results for real data

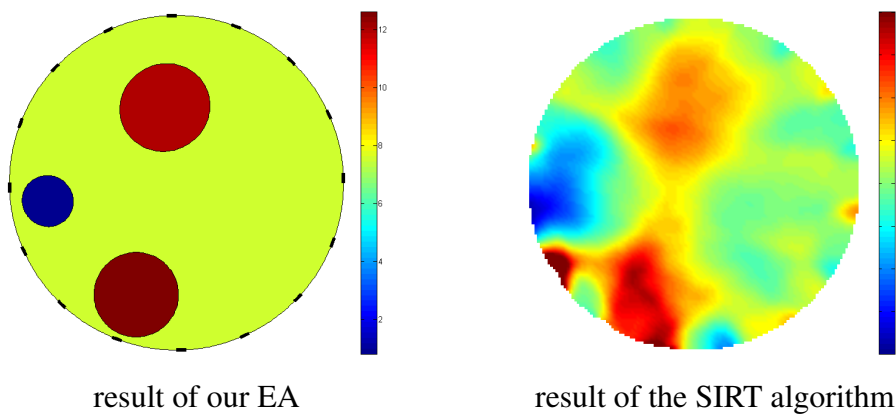
In the presented examples two sponges were put into the saline tank. In the second example a steel wire clew was additionally put in. For the algorithm we used we put all previously described modifications together: We used incomplete adjacent current patterns for electrode surfaces of 10% and divided the given data by the factors from section 6.4.4. These values for  $\Upsilon$  were considered as systematic measurement errors, and we subtracted the admitted mean value of 5% from each  $\Upsilon_\ell^i$ . This was done although we do not know the exact noise level. During the creation of our starting population we computed the relative distinguishability with respect to  $\sigma_0$  for all subdomains and deleted those subdomains whose distinguishability was smaller than the admitted noise level of 5%.

The results we obtained are compared to those obtained by the SIRT algorithm described in [31]. So on the left-hand side of the figures below we present the result which our evolutionary algorithm obtained, and on the right-hand side we present the result of the SIRT algorithm for the same input data. Instead of the conductivity  $\sigma$  we plot the *resistivity*  $1/\sigma$  because this is also done in the reference images.

The resistivity of the saline water was  $4.9 \Omega m$ , the resistivity of the sponges  $8 \Omega m$  and of the wire clew  $10^{-6} \Omega m$ . The result without the wire clew was the following:



and with the wire clew we got the following result:



In both cases the algorithm has identified the right number of subdomains at the approximately correct position and with the correct contrasts. For the initial background conductivity we started in both cases with  $1/4.9$ , but we finally ran the algorithm for  $\sigma_0 = 0.15$ . When running the algorithm it became even lower, i.e. the background resistivity increased. Also the resistivities we found for the sponges are higher than in reality.

Nevertheless if we take into account all the previously described difficulties we see that the algorithm is basically able to deal also with real data. Compared to the grid-based SIRT algorithm we can make a better distinction of different objects and eliminate all small accidental conductivity changes and artifacts.

## Conclusion and outlook

We introduced a novel boundary-element method to solve the inverse 2D electrical impedance tomography problem for a piecewise constant conductivity distribution. Then we incorporated this method into an evolutionary algorithm, which was used to couple this method together with an already existing grid-based method. This grid-based method was especially used to determine the number and geometry of the inhomogeneities, which was supposed to be unknown. Additionally, by using an idea from level-set methods, the boundary-element method was extended with a splitting and merging feature within the evolutionary algorithm.

The algorithm was then tested on various examples, for the easy-to-compute continuum model as well as for the more realistic complete electrode model. These tests illustrated that for exact data the boundary-element method is able to reconstruct the shape and conductivity of one unknown subdomain. By coupling it together with a grid-based method and extending it with a splitting feature we were also able to determine correctly the number and geometry of an unknown number of subdomains. Especially if we only have subdomains at one inclusion level and all with the same contrast, either the grid-based method or the splitting feature alone were sufficient to do that.

Also for noisy data the algorithm basically still worked, but especially for multiple inclusion levels the results were less satisfactory in most cases. For one inclusion level, however, the algorithm in most cases still found the correct number of subdomains as well as their approximate size and conductivity.

Finally we also applied the algorithm on real data. There some of the assumptions we made for synthetic data were not valid anymore. The changes to our algorithm to deal with these additional difficulties were described in detail, and at the end we successfully found the correct number and the approximative size and conductivity of two and three objects within a saline tank. So our evolutionary algorithm is basically able to be used also for real data.

Further investigations could be done in several areas:

- First, the three-dimensional case could be examined.
- On the theoretical level one could try to prove some convergence result for the boundary-element method for some special cases where the uniqueness has already been shown (see [19, 30, 1, 2]).

- Concerning the data, the next step would be a proper stochastic modeling of the noise, including a stochastic formulation of the fitness functional. Especially the stopping criterion for pointwise noise should be reconsidered, as in our examples it only has been chosen by trial and error.
- For the numerics a further investigation based on the results from [12] on how to treat the singularities of  $g$  at the edges of the electrodes might be useful, as in our implementation the exterior boundary has to be discretized very highly, which leads to considerable computational costs.
- Finally, on the algorithmical level, one could try to incorporate more methods into the evolutionary algorithm. Then they would be randomly chosen separately for each individual for each mutation step. It might be interesting to investigate if this leads to an addition of the advantages of each method or if the loss of information that occurs when converting the data from one representation to another is too high.

Summarizing we have shown that our boundary-element method is able to solve the inverse transmission problem quite well and that the coupling of different methods can considerably extend the possibilities of finding a solution without any a-priori information.

# Bibliography

- [1] Alessandrini, G., Isakov, V. and Powell, J.: *Local uniqueness in the inverse conductivity problem with one measurement*. Transaction of the American Mathematical Society **347**, No. 8, 3031–3041 (1995)
- [2] Astala, K. and Päivärinta, L.: *Calderón’s inverse conductivity problem in the plane*. Annals of Mathematics **163**, 265–299 (2006).
- [3] Aykroyd, R. and Cattle, B.: *A boundary-element approach for the complete-electrode model of EIT illustrated using simulated and real data*. Inverse Problems in Science and Engineering **15**, No.5, 441–461 (2007).
- [4] Babaeizadeh, S., Brooks, D. , Isaacson, D. and Newell, J.: *Electrode Boundary Conditions and Experimental Validation for BEM-based EIT Forward and Inverse Solutions*. IEEE Transactions on Medical imaging **25**, No.9, 1180–1188 (2006).
- [5] Barber, D.C. and Brown, B.H.: *Progress in Electrical Impedance Tomography*. Inverse Problems in Partial Differential Equations, SIAM, Philadelphia, 1990.
- [6] Borcea, L.: *A nonlinear multigrid for imaging electrical conductivity and permittivity at low frequency*. Inverse Problems **17**, 329–359 (2001).
- [7] Borcea, L.: *Electrical impedance tomography*. Inverse Problems **18**, R99–R136 (2002).
- [8] Brühl, M.: *Explicit characterization of inclusions in electrical impedance tomography*. SIAM Journal of Mathematical Analysis **32**, 1327–1341 (2001).
- [9] Brühl, M. and Hanke, M.: *Numerical implementation of two noniterative methods for locating inclusions by impedance tomography*. Inverse Problems **16**, 1029–1042 (2000).
- [10] Cheney, M, and Isaacson, D.: *Distinguishability in Impedance Imaging*. IEEE Transactions on Biomedical engineering **39**, 852–1040 (1992).
- [11] Cheney, M, Isaacson, D., Newell, J.C., Simske, S. and Goble, J.: *NOSER: An algorithm for solving the inverse conductivity problem*. International Journal of Imaging Systems and Technology **2**, 66–75 (1990).
- [12] Ciulli, S. and Ispas, S.: *Anomalous thresholds and edge singularities in electrical impedance tomography*. Journal of Mathematical Physics **37**, 4388–4417 (1996).

- [13] Dorn, O. and Lesselier, D.: *Level set methods for inverse scattering*. Inverse Problems **22**, R67-R131 (2006).
- [14] Eckel, H. and Kress, R.: *Nonlinear integral equations for the inverse electrical impedance problem*. Inverse Problems **23**, 475–491 (2007).
- [15] Eckel, H. and Potthast, R.: *An evolutionary Newton method for shape reconstruction*. Journal of Integral Equations and Applications **19**, 261–287 (2007).
- [16] Engl, H.W., Hanke, M. and Neubauer, A.: *Regularization of Inverse Problems*. Kluwer, Dordrecht, 1996.
- [17] Fogel, D.B.: *An Introduction to Simulated Evolutionary Optimization*. IEEE Transactions on Neural Networks: Special Issue on Evolutionary Computation **5**, 3–14 (1994).
- [18] Fogel, L. J. and Owens, A. J. and Walsh, M. J. : *Artificial Intelligence through Simulated Evolution*. John Wiley, New York, 1966.
- [19] Friedman, A. and Isakov, V.: *On the uniqueness of the inverse conductivity problem with one measurement*. Indiana University Mathematics Journal **38**, No. 3, 563–579 (1989).
- [20] Gebauer, B.: *The Factorization Method for Real Elliptic Problems*. Journal for Analysis and its applications **25**, 81–102 (2006).
- [21] Gebauer, B.: *Private communication*, 2007.
- [22] Hahn, G., Just, A. and Hellige, G.: *Determination of the dynamic measurement error of EIT systems*. ICEBI conference Graz 2007, IFMBE Proceedings **17**, 320–323 (2007).
- [23] Hähner, P.: *An inverse problem in electrostatics*. Inverse Problems **15**, 961–975 (1999).
- [24] Hanke, M. and Brühl, M.: *Recent progress in electrical impedance tomography*. Inverse Problems **19**, S65–S90 (2003).
- [25] Hofmann, B.: *Approximation of the inverse electrical impedance tomography problem by an inverse transmission problem*. Inverse Problems **14**, 1171–1187 (1998).
- [26] Hofmann, B.: *Das inverse Problem in der elektrischen Impedanztomographie*. Dissertation, Göttingen, 1997.
- [27] Holland, J. H : *Adaptation in natural and artificial systems* The University of Michigan Press, Ann Arbor, 1975.
- [28] Ider, Y. Z., Gencer, N. G., Atalar, E. and Tosun, H. *Electrical impedance tomography of translationally uniform cylindrical objects with general cross-sectional boundaries*. IEEE Transactions on Medical Imaging **9**, 49–59 (1990).
- [29] Ikehata, M. and Siltanen, S.: *Electrical impedance tomography and Mittag-Leffler's function*. Inverse Problems **20**, 1325–1348 (2004).

- [30] Isakov, V. and Powell, J.: *On the inverse conductivity problem with one measurement*. Inverse Problems **6**, 311–318 (1990).
- [31] Just, A., Küpper, T., Kürschner, D. and Jacobs, F.: *Core imaging by electrical resistivity tomography*. Proceeding SEGJ international symposium Tokyo 1998 **4**, 231–237 (1998).
- [32] Kaltenbacher, B.: *On the regularizing properties of a full multigrid method for ill-posed problems*. Inverse Problems **17**, 767–788 (2001).
- [33] Kirsch, A.: *Characterization of the shape of the scattering obstacle by the spectral data of the far field operator*. Inverse Problems **14**, 1489–1512 (1998).
- [34] Knudsen, K., Mueller, J. and Siltanen, S.: *Numerical solution method for the dbar-equation in the plane*. Journal of Computational Physics **198**, 500–517 (2004).
- [35] Koza, J. R.: *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, 1992.
- [36] Kress, R.: *Numerical analysis*. Springer-Verlag, Berlin Heidelberg New York, 1997.
- [37] Kress, R.: *Linear Integral Equations*. 2nd. ed. Springer-Verlag, Berlin Heidelberg New York, 1999.
- [38] Kress, R.: *On the numerical solution of a hypersingular integral equation in scattering theory*. Journal of Computational and applied mathematics **61**, 345–360 (1994).
- [39] Kress, R. and Rundell, W.: *Nonlinear integral equations and iterative solution for an inverse boundary value problem*. Inverse Problems **21**, 1207–1223 (2005).
- [40] Lackner, A.: *Dynamische Tourenplanung mit ausgewählten Metaheuristiken*. Dissertation Göttinger Wirtschaftsinformatik, Cuvillier Verlag, Göttingen, 2004.
- [41] Lube, G.: *Numerik elliptischer Randwertprobleme*. Lecture notes, Göttingen, 2002 (not published).
- [42] Maxwell, J. C.: *A Treatise on Electricity and Magnetism*. Reprint, Dover, 1954.
- [43] Pohlheim, H.: *Evolutionäre Algorithmen – Verfahren, Operatoren, Hinweise aus der Praxis*. Springer-Verlag, Berlin, 1999.
- [44] Potthast, R.: *Fréchet differentiability of boundary integral operators in inverse acoustic scattering*. Inverse Problems **10**, 431–447 (1994).
- [45] Rauch, J.: *Partial differential equations*. Springer-Verlag, New York, 1991.
- [46] Rechenberg, I.: *Evolutionstrategie - Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Frommann-Holzboog, Stuttgart, 1973.
- [47] Schwefel, H.-P.: *Evolutionstrategie und numerische Optimierung*. Technische Universität Berlin, 1975.

- [48] Somersalo, E., Cheney, M, and Isaacson, D.: *Existence and uniqueness for electrode models for electric current computed tomography*. SIAM Journal of Applied Mathematics **52**, 1023–1040 (1992).
- [49] Somersalo, E., Cheney, M, Isaacson, D. and Isaacson, E.: *Layer stripping: A direct numerical method for impedance imaging*. Inverse Problems **7**, 899–926 (1991).
- [50] Vauhkonen, M., Kaipio, J. P., Somersalo, E. and Karjalainen, P. A.: *Electrical impedance tomography with basis constraints*. Inverse Problems **13**, 523–530 (1997).
- [51] Vauhkonen, M.: *Electrical impedance tomography and prior information*. Dissertation, Kuopio University Publications, 1997.
- [52] Vauhkonen, M., Vadasz, D., Karjalainen, P. A., Somersalo, E. and Kaipio, J. P.: *Tikhonov regularization and prior information in electrical impedance tomography*. IEEE Transactions on medical imaging **17**, No.2, 285–293 (1998).
- [53] Vauhkonen, P.: *Image Reconstruction in Three-Dimensional Electrical Impedance Tomography*. Dissertation, Kuopio University Publications, 2004.
- [54] Wexler, A., Fry, B. and Neuman, M. R.: *Impedance computed tomography algorithm and system*. Applied Optics **24**, 3985–3992 (1985).
- [55] Woo, E. J., Hua, P., Webster, G. and Tompkins, W. J.: *A robust image reconstruction algorithm and its parallel implementation in electrical impedance tomography*. IEEE Transactions on medical imaging **12**, 137–146 (1993).
- [56] Yorkey, T. J. and Webster, J. G.: *A comparison of impedance tomographic reconstruction algorithms*. Clinical Physics Physiological Measurements, Supplement A **8**, 55–62 (1987).



# Danksagung

Zum Abschluss meiner Arbeit möchte ich mich insbesondere bei der Deutschen Forschungsgemeinschaft (DFG) für die Unterstützung durch ein Stipendium im Rahmen des Graduiertenkollegs 1023 "Identifikation in mathematischen Modellen" bedanken. Weiterhin danke ich Herrn Prof. Kreß für die interessante Themenstellung und die gute fachliche Betreuung und Herrn Prof. Potthast für die Übernahme des Zweitgutachtens und die Zusammenarbeit und Ideen bezüglich evolutionärer Algorithmen für inverse Probleme, die bereits mit der Betreuung meiner Diplomarbeit begonnen hat.

Ich danke auch den Mitarbeitern der Abteilung für Anästhesiologische Forschung des Universitätsklinikums Göttingen, insbesondere Frau Dr. Anita Just, für die Bereitstellung der realen Daten, sowie Dr. Bastian Gebauer für seine Idee der Faktorisierungsmethode mit mehreren Inklusionsebenen.



# Lebenslauf

Harry Eckel

geboren am 30.09.1977 in Kassel

Staatsangehörigkeit: deutsch

## Bildungsgang:

1984-1988	Grundschule Kassel
1988-1997	Wilhelmsgymnasium Kassel
20.06.1997	Abitur
1997-1999	Mathematisch-naturwissenschaftliches Grundstudium an der Université de Reims, Frankreich
1999-2004	Diplomstudium Mathematik an der Georg-August Universität Göttingen
13.04.2000	Vordiplom
12.11.2004	Diplom
seit Dezember 2004	Promotionsstudium im Graduiertenkolleg 1023 an der Georg-August Universität Göttingen

