


5-2014

# Inverse Problems Using Reduced Basis Method

Phil Gralla

Clemson University, pgralla@clemson.edu

Follow this and additional works at: [https://tigerprints.clemson.edu/all\\_theses](https://tigerprints.clemson.edu/all_theses)

 Part of the [Electrical and Computer Engineering Commons](#), and the [Materials Science and Engineering Commons](#)

---

## Recommended Citation

Gralla, Phil, "Inverse Problems Using Reduced Basis Method" (2014). *All Theses*. 2007.  
[https://tigerprints.clemson.edu/all\\_theses/2007](https://tigerprints.clemson.edu/all_theses/2007)

This Thesis is brought to you for free and open access by the Theses at TigerPrints. It has been accepted for inclusion in All Theses by an authorized administrator of TigerPrints. For more information, please contact [kokeefe@clemson.edu](mailto:kokeefe@clemson.edu).

# INVERSE PROBLEMS USING REDUCED BASIS METHOD

---

A Thesis  
Presented to  
the Graduate School of  
Clemson University

---

In Partial Fulfillment  
of the Requirements for the Degree  
Master of Science  
Mathematics

---

by  
Phil Gralla  
May 2014

---

Accepted by:  
Taufiqar Khan, Committee Chair  
James R Brannan  
Shitao Liu

# Abstract

Inverse Problems is a field of great interest for many applications, such as parameter identification and image reconstruction. The underlying models of inverse problems in many applications often involve Partial Differential Equations (PDEs). A Reduced Basis (RB) method for solving PDE based inverse problems is introduced in this thesis. The RB has been rigorously established as an efficient approach for solving PDEs in recent years. In this work, we investigate whether the RB method can be used as a regularization for solving ill-posed and nonlinear inverse problems using iterative methods. We rigorously analyze the RB method and prove convergence of the RB approximation to the exact solution. Furthermore, an iterative algorithm is proposed based on gradient method with RB regularization. We also implement the proposed method numerically and apply the algorithm to the inverse problem of Electrical Impedance Tomography (EIT) which is known to be a notoriously ill-posed and nonlinear. For the EIT example, we provide all necessary details and carefully explain each step of the RB method. We also investigate the limitations of the RB method for solving nonlinear inverse problems in general. We conclude that the RB method can be used to solve nonlinear inverse problems with appropriate assumptions however the assumptions are somewhat restrictive and may not be applicable for a wide range of problems.

# Table of Contents

<b>Title Page</b> . . . . .	<b>i</b>
<b>Abstract</b> . . . . .	<b>ii</b>
<b>List of Tables</b> . . . . .	<b>v</b>
<b>List of Figures</b> . . . . .	<b>vi</b>
<b>Abbreviations and Symbols</b> . . . . .	<b>vii</b>
<b>1 Introduction</b> . . . . .	<b>1</b>
<b>2 Mathematical Basics</b> . . . . .	<b>3</b>
2.1 Functional Analysis . . . . .	3
2.2 Gradient Method . . . . .	6
2.3 Partial Differential Equations . . . . .	7
2.4 Finite Element Method . . . . .	9
2.5 Gram-Schmidt Process . . . . .	12
2.6 Inverse Problems . . . . .	12
<b>3 Electrical Impedance Tomography</b> . . . . .	<b>15</b>
3.1 Mathematical Model . . . . .	16
3.2 Inverse Problem . . . . .	18
<b>4 Reduced Basis Method</b> . . . . .	<b>20</b>
4.1 Theory Basics . . . . .	21
4.2 A Priori Convergence Theory . . . . .	23
4.3 Dimension Reduction . . . . .	25
4.4 Orthogonalization . . . . .	29
4.5 Algorithm . . . . .	31
4.6 Application for Inverse Problems . . . . .	34
<b>5 Algorithm for Inverse Problems</b> . . . . .	<b>36</b>
5.1 Problem Description . . . . .	37
5.2 Decomposition of the weak Problem . . . . .	38
5.3 Numerical Implementation . . . . .	40
5.4 Results . . . . .	41
<b>6 Conclusion and Discussion</b> . . . . .	<b>43</b>
6.1 Theoretical Implications and Recommendations for future Research . . . . .	43
<b>Appendices</b> . . . . .	<b>44</b>

A	Tables . . . . .	45
B	Code . . . . .	46
	<b>Bibliography . . . . .</b>	<b>55</b>

# List of Tables

3.1	Electrical properties of biological tissue . . . . .	15
3.2	Resistivity of rocks and fluids . . . . .	16
5.1	Decomposition Example 2 . . . . .	40
5.2	Numerical Results for the EIT Example . . . . .	41
A1	Decomposition Example . . . . .	45

# List of Figures

2.1	Example for Triangulation . . . . .	10
4.1	Condition number of reduced-stifness basis . . . . .	31
5.1	Setup of the Example . . . . .	38
5.2	Domain of the Parameter $D$ . . . . .	39
5.3	Segmentation of $\Omega$ . . . . .	39
5.4	FE solution of the reference set $\mu_0$ . . . . .	40

# Abbreviations and Symbols

---



---

## Symbols

$a(\cdot, \cdot)$	bilinear form
$\alpha$	Lower bound for $\alpha(\mu)$ on a subset of $D$
$\alpha(\mu)$	coercivity constant for $a(\cdot, \cdot; \mu)$
$\alpha_0$	Lower bound for the $\alpha(\mu)$
$B_r(x)$	Open ball around $x$ with radius $r$
$\beta(\mu)$	Continuity constant for $a(\cdot, \cdot; \mu)$
$D$	Domain of the parameter $\mu$
$e$	Error, i.e. $u(\mu) - u_N(\mu)$
$\epsilon$	electric permittivity
$\exists$	there exists
$\forall$	for all
$\gamma(x, \omega)$	complex admittivity
$\nabla$	gradient
$\mathbb{K}$	field
$l$	linear functional
$\mu$	Parameter of the input of the PDE
$\langle \cdot, \cdot \rangle$	inner product
$\  \cdot \ $	norm
$\Omega$	physical domain of a given PDE
$\partial$	partial
$\perp$	perpendicular
$\mathbb{R}^n$	Realm of the reals with dimension $n$
$\sigma$	conductivity
$u(\mu)$	FE solution of the PDE for the input $\mu$
$u^e(\mu)$	True solution of the PDE for the input $\mu$
$u_N(\mu)$	RBM solution with $N$ ... for the nput $\mu$
$V$	vector space
$V^*$	dual space
$X$	Hilbert space of the FE solutions
$X^e$	Hilbert space of the truth solution

---



---

## Abbreviations

EIT	Electrical Impedance Tomography
FE	Finite Element (Method)
NtD	Neumann to Dirichlet map
PDE	Partial differential equation
RBM	Reduced Basis Method



# Chapter 1

## Introduction

*The best scientist is open to experience and begins with romance - the idea that anything is possible.*

- Ray Bradbury

Scientists often believe that if the answer is given, finding the question is trivial. Unfortunately, this is not the case for many questions. The mathematical field that involves finding a question or a cause for a known answer is the field of Inverse Problems. As the name implies, the problem we try to solve is inverse, meaning that the answer is known however the cause or the source of the answer is unknown.

Mathematical methods have been developed and analyzed for solving inverse problems with intense research over the past few decades. Many problems arise when trying to reconstruct sources or cause for known answers. This is due to the fact that the known answers, measurements in general, are not precise and involve noise. They might differ only slightly from the true solution, for example due to the method of measurement (measurement error) or non-captured physical relations by the mathematical model (model error). Unfortunately even small errors may lead to large discrepancies between reconstructed source and real source for ill-posed problems in the sense of Hadamard. A problem is considered well posed, according to Hadamard, if

1. A solution exists
2. The solution is unique

### 3. The solution depends continuously on the data

A number of methods have been developed to address the ill-posedness of an inverse problem via regularization. However, to obtain good results, we need appropriate assumptions and/or high computational cost. This is particularly true if the mathematical model for the inverse problem is nonlinear. This thesis proposes RB method as a regularization to solve nonlinear ill-posed inverse problems. The RB method has been introduced in [4] and [3]. The method is a fast way to solve parameterized partial differential equations (PDEs) for multiple inputs. Based on the properties of RB method, an algorithm is developed to solve nonlinear inverse problems. Then the proposed method is applied to solve the nonlinear ill-posed inverse problem in electrical impedance tomography.

The thesis is divided into three parts. In the first part, an outline of the mathematical background needed for understanding this thesis is given in Chapter 2 and an introduction to the electrical impedance tomography is provided in Chapter 3. In the second part, the RB method is introduced with proofs of the useful properties used in this thesis. Based on this theoretical background the implementation of the RB method and possible applications for inverse problems is discussed in Chapter 4. Then an algorithm is developed and applied to the EIT inverse problem which is detailed in Chapter 5. Finally, based on the numerical results for the EIT inverse problem and based on the theory explained in chapter 4 and 5, some conclusions about the applicability and limitations of the RB method for solving inverse problems are discussed in Chapter 6.

# Chapter 2

## Mathematical Basics

This chapter provides a short overview on the mathematical basics for this paper. The information presented here is based on [12], [15] and [2]. These references are in German however there are English translations for the first two references except for [2]. Nonetheless most graduate level textbooks for functional analysis, inverse problems, and optimal control/ numerical optimization include further explanation on the presented topics.

### 2.1 Functional Analysis

#### 2.1.1 Linear Functional

Let  $V$  be a vector space over the field  $\mathbb{K}$ . The functional  $l : V \rightarrow \mathbb{K}$  is said to be linear if and only if

$$l(c_1v + c_2u) = c_1l(v) + c_2l(u), \quad \forall u, v \in V, \forall c_1, c_2 \in \mathbb{K}$$

is being satisfied.

#### 2.1.2 Bilinear Form

Let  $V$  and  $Y$  be two vector spaces over the field  $\mathbb{K}$ . In this paper  $\mathbb{K}$  is either real  $\mathbb{R}$  or complex  $\mathbb{C}$ . An operator  $a : V \times Y \rightarrow \mathbb{K}$  is called bilinear form if the following properties hold

1.  $a(v_1 + v_2, y) = a(v_1, y) + a(v_2, y), \quad v_1, v_2 \in V, y \in Y$

2.  $a(v, y_1 + y_2) = a(v, y_1) + a(v, y_2), \quad v \in V, y_1, y_2 \in Y$
3.  $a(c \cdot v, y) = c \cdot a(v, y), \quad v \in V, y \in Y, c \in \mathbb{K}$
4.  $a(v, c \cdot y) = \bar{c}a(v, y), \quad v \in V, y \in Y, c \in \mathbb{K}.$

A bilinear form can be further characterized using one or both of the following definitions.

**Definition 2.1.1** *If  $a : V \times V \rightarrow \mathbb{K}$  is a bilinear form and  $a(u, v) = \overline{a(v, u)}$ ,  $a(\cdot, \cdot)$  is said to be symmetric.*

**Definition 2.1.2** *A bilinear form on a normed vector space  $(V, \|\cdot\|_V)$  is coercive, if  $\exists c, c > 0$  and constant, such that*

$$a(v, v) \geq c\|v\|_V.$$

*It can also be referred to as being an elliptic bilinear form.*

### 2.1.3 Hilbert Space

One of the most important spaces in the functional analysis are the Hilbert spaces. This space is a vector space with an inner product  $\langle \cdot, \cdot \rangle$ . Each inner product induces a norm by

$$\|f\|_X := \sqrt{\langle f, f \rangle}.$$

In addition every Hilbert space is complete in its induced norm. An inner product space, that is not complete is referred to as pre Hilbert space. Some examples for Hilbert spaces are

1.  $\mathbb{R}^n$  with the dot product as the inner product.
2.  $L^2$ , which includes all functions  $f : \mathbb{R} \rightarrow \mathbb{R}$  with  $\int_{-\infty}^{\infty} f^2 < \infty$ . The inner product is defined as

$$\langle f, g \rangle := \int_{-\infty}^{\infty} f(x) \cdot g(x) dx.$$

This space is an example for an infinite space, which are of particular interest in functional analysis.

## 2.1.4 Dual Space

The dual space of a given vector space,  $V$ , is the set of all linear functionals from  $V$  into  $\mathbb{R}$  and is denoted as  $V^*$ . In short

$$V^* = \{f \mid f : V \rightarrow \mathbb{R}, f \text{ is a linear functional}\}.$$

Note that this definition applies only for real vector spaces, since any linear functional is a linear map from the vector space onto its field, thus in the case above the vector space needs to be real. Equivalent definitions of dual spaces exist for complex vector fields.

The norm of a dual space is given by

$$\|l\|_{V^*} := \sup_{v \in V, v \neq 0} \frac{|l(f)|}{\|f\|_V}.$$

## 2.1.5 Weak Derivative

In some cases functions, such as solutions to PDEs, are not differentiable over some set  $U$ . Motivated by the integration by parts a more general form of derivative was introduced, the so called weak derivative.

**Definition 2.1.3** Let  $u$  and  $v$  be in  $L_{loc}(U)$ , the space of all locally integrable functions and  $\alpha$  is a multi index. If  $\forall \phi \in C_c^\infty(U)$

$$\int_U u D^\alpha \phi = (-1)^{|\alpha|} \int_U v \phi$$

$v$  is said to be the  $\alpha^{th}$  weak derivative of  $u$  in  $U$ .

The solution  $v$  is unique almost everywhere, i.e. it is unique except for a measure of zero. Note that this definition is based on the Lebesgue definition of integration.

## 2.1.6 Fundamental Inequalities

**Triangle** Let  $\|\cdot\|_Y$  be a norm on the space  $Y$ . Then  $\|\cdot\|_Y$  satisfies the triangle inequality

$$\|x - y\|_Y \leq \|x - z\|_Y + \|z - y\|_Y, \quad \forall x, y, z \in Y \tag{2.1}$$

**Cauchy-Schwarz** Let  $a : X \times X \rightarrow \mathbb{K}$  be a symmetric semi-definite bilinear form. Then  $a$  satisfies the following inequality

$$|a(u, v)| \leq \sqrt{a(u, u)}\sqrt{a(v, v)}. \quad (2.2)$$

This is called the Cauchy-Schwarz inequality.

## 2.2 Gradient Method

Finding a global minimum of any given function numerically is impossible. For differentiable functions however many established algorithms to approximate a local minimum exist. In this section the gradient method with adaptive step width is introduced. Later this method is used to solve the inverse problems for nonlinear operators numerically. The gradient method is based on the linearization of a function via Taylor expansion. For the point  $x_0$  the linearization of  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is

$$f(y) = f(x_0) + \nabla f(x_0)(x_0 - y) + E(\nabla f(x_0), y)$$

where  $E(\nabla f(x_0), y)$  denotes the linearization error, which can be dropped for a certain neighborhood of  $x_0$ <sup>1</sup>. Without the linearization error,  $f(y)$  is approximated by

$$\hat{f}(y) = f(x_0) + \nabla f(x_0)(x_0 - y)$$

at  $x_0$ . Since,  $\hat{f}(y)$  is linear now, it does not have a maximum on  $\mathbb{R}^n$ . If the linearization is constricted to the convergence radius 1, the constraint

$$\|x_0 - y\| \leq 1$$

is added. Therefore the linear function  $f(y)$  has its minimum and maximum on the border. This leads to the approximated minimum at

$$x_0 - d = x_0 - \nabla f(x_0) / \|\nabla f(x_0)\|.$$

---

<sup>1</sup>see Taylor expansion as explained in [10]

The direction  $d$  is normed and in the direction of the gradient. Because of this it is denoted as the normed gradient method. Since the linearisation is not perfect, the same step has to be repeated until certain criteria are met. For example the function values don't differ much and it is assumed that the approximation is good. Since convergence of the gradient method is sometimes not achieved an maximum number of iterations should be set, in order to end the algorithm in any case.

Additional improvements can be made if an adaptive step size is being implemented. In this thesis the Armijo rule is being used [2]. It is an heuristic iterative method to find a good step size for each iteration of the gradient method.

One issue is that this method finds only minima on unrestricted areas. To compensate this a so called penalty term is added to the function, i.e.

$$\min_{x \in D} f(x) + \lambda \cdot d(x, D)^2.$$

This penalty function is active, when an point of evaluation is outside of  $D$  and thus restricts the minimum on  $D$ .

## 2.3 Partial Differential Equations

Many models used in real world describing physical phenomena are based on so called partial differential equations (PDEs). As the name suggest PDEs are related to ordinary differential equations. They are a more general form and include multi-variable functions and their partial derivatives. Examples of appearance are description of fluid movement, chemical reactions, electrical fields and heat expansion. They are usually defined on multidimensional systems. A well known example is the heat conduction equation

$$\frac{\partial T}{\partial t} = \kappa \nabla^2 T.$$

Linear, second-order PDEs are classified into three categories, elliptic, hyperbolic and parabolic. They can be identified writing the PDE in the form of

$$Au_{xx} + 2Bu_{xy} + Cu_{yy} + Du_x + Eu_y + F = 0$$

and the matrix

$$Z = \begin{pmatrix} A & B \\ B & C \end{pmatrix}$$

defined. Then three cases are possible.

1.  $Z$  is positive definite, then the PDE is called *elliptic*.
2.  $Z$  is negative definite, then the PDE is called *hyperbolic*.
3.  $Z$  is singular, i.e.  $\det(Z) = 0$ , then the PDE is called *parabolic*.

It is possible that depending on  $x, y$  the coefficients  $A$ ,  $B$  and  $C$  change, i.e. they are no constants. Thus it is possible that the category of the PDE changes. In general PDEs are defined on a "physical" field  $\Omega$ . The category of the PDE is defined on the same field. This means a PDE may be hyperbolic on one field  $\Omega_1$  and elliptic on a second field  $\Omega_2$ . On  $\Omega = \Omega_1 \cup \Omega_2$  the PDE is a so called mixed type and would be referred to as elliptic-hyperbolic. The category of an PDE is essential for some numerical algorithm. To solve these equations additional conditions, so called boundary equations are required.

### 2.3.1 Boundary Conditions

There are two main types of boundary conditions, the Dirichlet and the Neumann boundary condition. The Dirichlet boundary condition is also referred to as a fixed condition. For this kind of boundary condition a fixed value is assigned to the points on  $\partial\Omega$ . For example for the heat equation it is

$$u(x, y) = c, \quad \forall (x, y) \in \partial\Omega.$$

The second type of boundary conditions describes a stream or flux on the boundary in mathematical terms, this means the deviation in the direction perpendicular to the boundary is given. This second type of boundary condition is called Neumann and is named after Carl Neumann. In the example for the heat equations the Neumann boundary is defined as

$$\frac{\partial u}{\partial n} = f(x, y).$$



Since  $n$  is not well defined we assume  $n$  to be directed inward towards  $\Omega$  and thus it becomes well defined.

## 2.4 Finite Element Method

Analytical solution of PDEs is often difficult and time consuming if not intractable. Numerical methods provide a convenient way to solve PDEs using a PC. As it is usually the case for numerical algorithm, only an approximation of the real solution is found. In this thesis, the finite element method is used to solve the relevant PDEs. It is a widely used and analyzed method for which additional information can be found in many standard books [8].

The finite element method is based on the weak formulation of the PDE. This means that a more general solutions space is considered and includes the natural boundary and continuity conditions of the problem. The weak formulations is stated as, find  $u^e(\mu) \in X^e$  such that

$$a(u^e(\mu), v; \mu) = f(v), \forall v \in X^e$$

where  $a(\cdot, \cdot; \mu)$  is a  $\mu$ -parametrized bilinear form,  $f$  is a linear functional and  $X^e$  is an appropriated Hilbert space. This bilinear form is equivalent to the weak formulation of the PDE.

To solve this problem numerical, the (physical) domain  $\Omega$  of the PDE is discretised using so called triangulation. The discrete domain is notated as  $\mathbb{T}_h$  with the property

$$\overline{\Omega} = \cup_{T_h \in \mathbb{T}_h} \overline{T_h}.$$

Since  $\Omega$  is bounded in a finite space, we have finite elements  $T_h$ . The nodes of  $\mathbb{T}_h$  are denoted by  $x_i$ ,  $i = 1, 2, \dots, m$ . The subscript  $h$  of  $\mathbb{T}_h$  and  $T_h$  stands for the longest age of all elements in  $\mathbb{T}_h$ . In this thesis the elements of  $\mathbb{T}_h$  are triangles, an example is given in figure 2.1.

The new solution space for the discrete domain is

$$X = \{v \in X^e \mid v|_{T_h} \in \mathbb{P}_p(\mathbb{T}_h), \forall T_h \in \mathbb{T}_h\}, \quad (2.3)$$

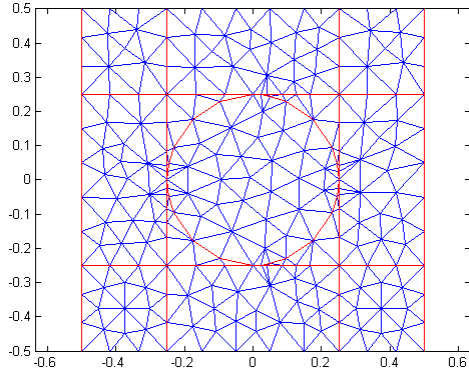


Figure 2.1: Example for Triangulation

where  $\mathbb{P}_p(T_h)$  is the space of  $p^{th}$  degree polynomials over the  $T_h$ . We now seek the solution  $u(\mu)$  of

$$a(u(\mu), \phi; \mu) = f(\phi), \quad \forall \phi \in X$$

for the input  $\mu$ . To compute this solution let  $\phi_i \in X$  be the nodal basic function, i.e.  $\phi_i(x_j) = \delta_{ij}$ . Then the  $\phi_i$  form a basis of  $X$ , i.e.

$$X = \text{span}\{\phi_1, \dots, \phi_m\}$$

and therefore the solution  $u(\mu)$  can be written as

$$u(\mu) = \sum_{i=1}^m u_i(\mu) \phi_i \tag{2.4}$$

where  $u_i(\mu)$  is the nodal value of  $u(\mu)$  at node  $x_i$ .

By using Galerkin projection onto  $X$  the approximation of  $u^e(\mu)$  in  $X$  is obtained from

$$a(u(\mu), \phi; \mu) = f(\phi), \quad \forall \phi \in X.$$

Since the  $\phi_i$  form a basis of  $X$  it is enough to solve

$$\begin{aligned}
 & a(u(\mu), \phi_i; \mu) = f(\phi_i) && i = 1, \dots, m \\
 \xrightarrow{2.4} & a\left(\sum_{j=1}^m u_j(\mu) \phi_j, \phi_i; \mu\right) = f(\phi_i) && i = 1, \dots, m \\
 \Rightarrow & \sum_{j=1}^m u_j(\mu) a(\phi_j, \phi_i; \mu) = f(\phi_i) && i = 1, \dots, m.
 \end{aligned}$$

This system can be written in matrix form as

$$A(\mu) \cdot u(\mu) = F$$

where  $A$  is a  $m \times m$  matrix and  $u(\mu), F$  are vectors of size  $m$ . The coefficients are determined as

$$a_{i,j} = a(\phi_j, \phi_i; \mu) \tag{2.5}$$

$$u(\mu)_i = u(x_i; \mu) \tag{2.6}$$

$$f_i = f(\phi_i). \tag{2.7}$$

In general,  $m$  is rather large, but  $A$  is a sparse matrix. Therefore the system can be solved by various methods, such as the CG-Algorithm.

Further information about the FE method can be found in many books, for example see [8] for convergence theory, computational complexity, properties of  $A(\mu)$  and example algorithms.

## Numerical Implementation

In this paper the *MATLAB* pde toolbox is used to find a FE approximation. This toolbox is an extension for *MATLAB* and can be tested for a few days for free. It is usually distributed together with *Simulink* but can be purchased separately. For the implementation used in this thesis the free version of the toolbox is sufficient.

## 2.5 Gram-Schmidt Process

The Gram-Schmidt Process is used to find an orthogonal basis for a given set of basis vectors. Let  $M = \{v_1, v_2, \dots, v_n\}$  be a basis for  $V$  which is not necessarily orthogonal. To find an orthogonal basis for  $V$  based on  $M$ , an iterative method is used. For each step a new vector is projected onto the perpendicular space of the already created one, i.e. let  $V_i = \text{span}\{u_1, \dots, u_i\}$  the space of  $i$  orthogonal vectors, then  $v_{i+1}$  is projected on  $V_i^T$  where  $V_i \perp V_i^T$  and  $V_i^T \cup V_i = V_{i+1} = \text{span}\{V_i, v_{i+1}\}$  to find  $u_{i+1}$  and thus an orthogonal basis for  $V_{i+1}$  is created. Summarized the algorithm is

### Gram-Schmidt Algorithm

Given  $M = \{v_1, v_2, \dots, v_n\}$

Set  $u_1 = \frac{v_1}{\|v_1\|_V}$

For  $i = 2 : n$

$$u = v_i - \sum_{k=1}^{i-1} \frac{\langle u_k, v_i \rangle}{\langle u_k, u_k \rangle} u_k$$

$$u_i = \frac{u}{\|u\|_V}$$

End

In addition to the orthogonalization, the new basis is also ortho-normal, due to the second step in the iteration.

## 2.6 Inverse Problems

The field of inverse problems is based on functional analysis and involves finding a cause/-source for a known solution and an operator. For finite spaces and linear models it is equivalent to solving an algebraic systems of the form

$$A \cdot x = b$$

for a known matrix  $A$  and solution vector  $b$ . However, in general the solution space may be infinite and operators may not be linear. Typical fields of application are parameter identification, image processing and image reconstruction. One example is computer tomography (CT) where methods of inverse problems are used to reconstruct the images of the inside of an patient that are non invasive [12].

In this section a small overview is provided. For further information, we refer to [12]. The format of this section is adapted from [12]. The emphasis is on basic definitions and the *Tikhonov-*

*Regularization.*

Let  $g$  be the known data and  $A : X \rightarrow Y$  be an operator. The inverse problem can be formulated as a minimization problem

$$f_{sol} = \operatorname{argmin} \|A(f) - g\|$$

where in the absence of noise,  $f_{sol}$  should be the true solution  $f_0$  mainly

$$\|f_{sol} - f_0\| = 0.$$

An important criteria is to estimate the distance between computed solution and the real solution to characterize whether the operator  $A$  is ill posed or well posed. There is more then one definition for ill posed problems, in this thesis the Hadamard's definition is used.

**Definition 2.6.1** *A problem is called well-posed if there exists a unique solution to the problem that continuously depends on its data. A problem is said to be ill-posed if it is not well posed.*

A few examples for well-posed and ill-posed problems,

well-posed problem	ill-posed problem
Multiplication by a small number $c \ll 1$ $c \cdot x = y$	division by a small number $x = c^{-1}y$
integration $F(x) = F(0) + \int_0^x f(\tau)d\tau$	differentiation $f(x) = F'(x)$

As can be seen in these examples, a forward problem may be well-posed but the inverse problem may not be, i.e. integration as a forward model is well-posed but its inverse problem, differentiation is ill-posed. Special emphasis is on the continuous dependence of the problem, since in reality not the exact solution  $g$  is known but rather a noisy version denoted with  $g^\varepsilon$ . Possible causes are noise in the measurement methods (measurement error) and un-captured physical phenomena in the mathematical model (model error). In such a case so called regularization of the problem is needed. One of the first methods was introduced by Tikhonov and Phillips. They introduced the Tikhonov functional

$$J_t(f) = \|A(f) - g^\varepsilon\|_Y^2 + t\|f - f^*\|_X^2, \quad t > 0 \tag{2.8}$$

which is being minimized over the domain of  $A$ . This functional takes known information  $f^*$  about the solution into account via a penalty term. Many variations of this penalty function exist and maybe more suitable for different problems.

## Chapter 3

# Electrical Impedance Tomography

Due to electrical properties, conductivity  $\sigma$  and electric permittivity  $\epsilon$  of a material differs under influence of external electric fields. Electrical impedance tomography (EIT) uses these different properties to generate images of an object by reproducing the map of  $\sigma(x)$  and  $\epsilon(x)$  for  $x \in \Omega$  from measured voltage and currents at the boundary  $\partial\Omega$ . Comparing this map to a table of known material properties (see Tables 3.1 and 3.2) the material in a probe can be determined.

The advantages of EIT over other imaging modalities are that EIT is non-invasive and EIT is widely used with applications ranging from geophysical sub-surface imaging for oil exploration to medical imaging for cancer detection. For example, EIT applications in geophysics and environmental science include locating underground mineral deposits, detection of leaks in underground storage tanks and for monitoring flows of injected fluids into the earth, used for oil extraction or environmental cleaning. EIT applications in medical imaging include detection of pulmonary emboli, monitoring of aponea, monitoring of heart functional blood flow and breast cancer detection. Contrary to computer tomography the patient doesn't need to be in a machine, since electrodes on his/her body are sufficient for necessary measurement. EIT is especially useful in geophysics, because

Tissue	$1/\sigma(\Omega cm)$	$\epsilon(\mu F m^{-1})$
Lung	950	0.22
Muscle	760	0.49
Liver	685	0.49
Hearth	685	0.88
Fat	> 1000	0.18

Table 3.1: Electrical properties of biological tissue measured at frequency 10 kHz, [13] and [14]

Rock or fluid	$1/\sigma(\Omega cm)$
Marine sand, shale	1-10
Terrestrial sands, claystone	15-50
Volcanic rocks, basalt	10-200
Granite	500-2000
Limestone dolomite, anhydrite	50-5000
Chloride water from oil fields	0.16
Sulfate water from oil fields	1.2

Table 3.2: Resistivity of rocks and fluids [9]

the measurements for the image reconstruction can be made outside and putting the electrodes on the surface of the earth is enough.

In the following sections the mathematical model is derived and the inverse problem described.

### 3.1 Mathematical Model

If the alternating current is used, the electric permittivity and electrical conductivity in EIT exhibit the complex admittivity function

$$\gamma(x, \omega) = \sigma(x) + i\omega\epsilon(x) \quad (3.1)$$

where  $i = \sqrt{-1}$  and  $\omega$  is the frequency of the alternating current. It is assumed that  $\omega$  is fixed and known. For the following problem description  $\gamma(x) := \gamma(x, \omega)$ . Since  $\gamma(x)$  is usually unknown,  $\gamma(x)$  is to be determined on a bounded region  $\Omega$  with a smooth boundary  $\partial\Omega$ . It is assumed that  $\gamma$  is strictly positive, isotropic and bounded. In addition, we assume that there is no current sources in  $\Omega$ .

If  $u(x)$  is the electric potential at point  $x$ , from Ohm's law we have,

$$-\text{div}((\gamma(x)\nabla u(x))) = 0, \quad x \in \Omega, \quad (3.2)$$

where the electric potential  $u$  can be modeled as a solution to this elliptic partial differential equation. The current source injected at the boundary can be modeled as the Neumann boundary condition:

$$\gamma(x) \frac{\partial u}{\partial n} = I(x), \quad x \in \partial\Omega \quad (3.3)$$



and the measurement of the voltage at the detector on the boundary can be modeled as the Dirichlet boundary condition:

$$u(x) = V(x), \quad x \in \partial\Omega. \quad (3.4)$$

From the Maxwell equations and given that  $\Omega$  is source free we get

$$\int_{\partial\Omega} I(x) ds(x) = 0,$$

which must be applied for the Neumann boundary condition. For the Dirichlet boundary value problem and  $\gamma(x) \in L^\infty(\Omega)$  and arbitrary  $V \in H^{\frac{1}{2}}(\partial\Omega)$ , there exist a unique solution  $u(x) \in H^1(\Omega)$  [5]. The Neumann boundary value problem has a unique solution  $u(x) \in H^1(\Omega)$  if a ground level is chosen [5]. One possible ground level is

$$\int_{\partial\Omega} \gamma(x) ds = 0.$$

In summary, we have two models for the EIT, the Neumann Boundary Model given the currents at the boundary

$$-\text{div}((\gamma(x)\nabla u(x))) = 0, \quad x \in \Omega \quad (3.5)$$

$$\gamma(x) \frac{\partial u}{\partial n} = I(x), \quad x \in \partial\Omega \quad (3.6)$$

and the Dirichlet Boundary Model for the measurement at the boundary

$$-\text{div}((\gamma(x)\nabla u(x))) = 0, \quad x \in \Omega \quad (3.7)$$

$$u(x) = V(x), \quad x \in \partial\Omega. \quad (3.8)$$

A typical EIT data collection experiment consists of applying an electrical current (Neumann data)  $I(x)$  on the boundary of  $\Omega$  and measuring the resulting electrical potential (Dirichlet data)  $V(x)$ , again on the boundary. Therefore, we have information about the so called Neumann-to-Dirichlet (NtD) map. We note that in practice voltage data is collected at a finite number of the electrodes on the boundary. In [1] a complete electrode model is presented and discussed to

account for the finite electrodes on the boundary. In this thesis, we assume knowledge of the voltage everywhere on the boundary and we do not consider the complete electrode model.

Similar to many inverse coefficient problems based on elliptic partial differential equations, the EIT suffers from a high-degree of nonlinearity and severe ill-posedness [7, 6]. However, due to the broad range of potential applications, designing numerical techniques for its efficient solution is still desirable. A large number of reconstruction methods have been developed in the literature mostly deterministic. The deterministic approaches include variational type methods of minimizing a certain discrepancy functional, e.g., the least-square fitting for the linearized or fully nonlinear model, either of Tikhonov type or iterative type regularization methods [7, 6]. In this thesis, we propose a RB method based reconstruction algorithm for EIT.

## 3.2 Inverse Problem

Contrary to the forward problem, for the inverse problem in EIT,  $\gamma(x)$  is unknown. The inverse problem in EIT is to reconstruct  $\gamma(x)$  given the measured voltage on the boundary. For the mathematical model of EIT, the Dirichlet trace operator, denoted by  $\Gamma_D u$ , is introduced, i.e. the restriction of  $u$  to the boundary

$$\begin{aligned}\Gamma_D : Y &\rightarrow Z \\ u &\mapsto \Gamma_D u.\end{aligned}$$

In addition the associated linear forward operator of the Neumann problem, which maps an input current  $I$  to the solution  $u$ , is denoted by

$$\begin{aligned}F_N^\gamma : X &\rightarrow Y \\ I &\mapsto u \text{ solves 3.5.}\end{aligned}$$

Combining these solutions form the so called NtD map (Neumann to Dirichlet map) and can be written as  $\Gamma_D F_N^\gamma$ . The Inverse problem then becomes:

Find  $\gamma$  such that

$$g(I) = g \tag{3.9}$$

$$\gamma = \gamma_0 \tag{3.10}$$

where  $g$  is the known data,  $g(I) = \Gamma_D F_N^\gamma(I)$  and  $\gamma_0$  is the real conductivity for the measured voltage  $g$ . In practice, only a noisy version of the data  $g^\varepsilon$  is known. The inverse problem can be formulated as the minimum of the functional

$$J_t(\gamma) = \|g(I) - g^\varepsilon\|_Z + t\Phi$$

where  $\Phi$  is defined by the chosen method of regularization, for example for the Tikhonov regularization

$$\Phi = \|\gamma - \tilde{\gamma}\|^2.$$

For the rest of this thesis only the real part of  $\gamma(x)$  is considered, i.e.  $\gamma(x) = \sigma(x)$ . This was done for a better representation of the solutions and does not change the results. All simulation in this paper are possible for complex numbers. Additionally,  $I$  and  $V$  are assumed real as well.

## Chapter 4

# Reduced Basis Method

Solving parameter dependent PDEs using a really fine mesh can be computationally very expensive particularly for inverse problems where one needs to compute the forward solution hundreds or thousands of times to estimate the parameters. To reduce the computational time the Reduced Basis (RB) method was introduced in the late 1970s ([4] p 1526). Based on an approximation of a finite element solution of the PDE,  $u(\mu)$ , the reduced basis method is an approximation of this finite solution by identifying and exploiting properties of the solution space. It can be shown that the solution itself is only defined on a low dimensional smooth manifold induced by parametric dependence. However in order to perform the desired basis reduction, the problem must be solved, using the FE-method, first. The reduced basis method is therefore used to compute a given PDE multiple times after it was analyzed. Since it is much faster solving the problem with the reduced basis, in fact it can be used for real time computation, it is used in areas where a given problem has to be solved multiple times, such as monitoring cracks in a surface (see example in [4] p. 1538-41 ).

Since for solving non-linear inverse problems numerically, solving a PDE multiple times is needed, the reduced basis method may be a good approach to solve inverse problems. Moreover, since it exploits properties of the parameter space it may regularize the inverse problem. In this chapter the theory of the RB Method is described and a possible way of applying this method for solving inverse problems is provided.

## 4.1 Theory Basics

The RBM method is based on properties of the solution for a given PDE if the input  $\mu$  can be parametrized, which means that  $\mu \in \mathbb{R}^p$  where  $p \in \mathbb{N}$ . The PDE is given in its weak form

$$a(u^e(\mu), v; \mu) = f(v), \quad \forall v \in X^e \quad (4.1)$$

where  $u^e(\mu) \in X^e$  is the solution for the  $p$ -dimensional input vector  $\mu \in D \subset \mathbb{R}^p$ . In addition,  $X^e$  is an appropriate Hilber space over the physical domain  $\Omega \subset \mathbb{R}^d$ . Since the dimension  $d$  is induced by the observed physic, it is  $d = 2, 3$  for the EIT in general. In this paper  $d = 2$  is assumed. Denote the set of all possible solutions as

$$M^e = \{u^e(\mu) \mid \mu \in D\}.$$

Since the dimension of  $D$  is equal to  $p$ ,  $M^e$  describes a manifold with dimension  $p$  where the bilinear form  $a(\cdot, \cdot; \mu)$  is the map of  $D \subset \mathbb{R}^p$  to  $X^e$ . Instead of approximating the whole space  $X^e$  we wish to approximate only  $M^e$  and find solutions based on this new approximation. Since the manifold is described by solutions of equation 4.1 the new basis space is

$$W_n^e = \text{span}\{u^e(\mu_i) \mid 1 \leq N\},$$

where  $\mu_i \in D$  are fixed nested samples. Depending on the smoothness of  $M$ ,  $u_n(\mu) \rightarrow u^e(\mu)$ , where the rate of convergence depends on the smoothness of  $M$ . In fact in [4] it is shown that in certain simple cases, exponential convergence rate can be proven. Since  $u^e(\mu_i)$  is generally unknown, it is approximated by finite element method where the solution  $u(\mu_i)$  satisfies

$$a(u(\mu), v; \mu) = f(v), \quad \forall v \in X \quad (4.2)$$

where  $X$  is the solution space of the FE method. As a result the RB method approximates the manifold

$$M = \{u(\mu) \mid \mu \in D\}.$$

and therefore

$$W_N = \text{span}\{u(\mu_i) \mid 1 \leq N\}$$

is the new basis space. The RB approximation of a new  $\mu$  will be

$$u_N(\mu) = \sum_{i=1}^N \lambda_i(\mu) u(\mu_i)$$

where the coefficients  $\lambda_i(\mu)$  need to be determined. To find the coefficients, Galerkin projection onto  $W_N$  is used, i.e. find the solution with respect to  $\lambda_i(\mu)$  for

$$\begin{aligned} a\left(\sum_{i=1}^N \lambda_i(\mu) u(\mu_i), w; \mu\right) &= f(w), \quad \forall w \in W_N. \\ \sum_{i=1}^N \lambda_i(\mu) a(u(\mu_i), w; \mu) &= f(w) \end{aligned}$$

Since  $W_N = \text{span}\{u(\mu_i) \mid 1 \leq i \leq N\}$ ,  $a(\cdot, \cdot)$  is bilinear and  $f(\cdot)$  is linear, it is enough to solve

$$\sum_{i=1}^N \lambda_i(\mu) a(u(\mu_i), u(\mu_j); \mu) = f(u(\mu_j)), \quad \forall j \in \{1, 2, \dots, N\} \quad (4.3)$$

For the following the RB approximation is compared to the FE method solution. By triangle inequality,

$$\|u^e(\mu) - u_N(\mu)\|_X \leq \|u^e(\mu) - u(\mu)\|_X + \|u(\mu) - u_N(\mu)\|_X$$

and therefore if the RBM approximation converges towards the FE solution it also converges to the exact solution by the convergence of the FE solution. For the rest of the paper the RB solution is compared to the FE solution only.

So far no computational effort reduction was made since the elements of  $W_N$  need to be computed using FE method. The RB reduces computational effort only if precomputed solutions of  $u(\mu_i)$  are made. The dimension of the linear system now depends on  $N$  and not on the dimension of the triangulation which, depending on the parametrization, is much larger than  $N$ . Additional information about the actual dimension reduction is provided later in this chapter. Therefore if the elements of  $W_N$  are known, i.e. precomputed, the computational effort is reduced in fact. However the bilinear form  $a(u(\mu_i), u(\mu_j); \mu)$  still depends on  $\mu$  and needs to be calculated for each pair of  $(u(\mu_i), u(\mu_j))$ . In [3] and [4] the additional assumption that the bilinear form can be decomposed into

$$a(w, v; \mu) = \sum_{q=1}^Q \Theta^q(\mu) a^q(w, v) \quad (4.4)$$

where  $\Theta^q(\mu) : D \rightarrow \mathbb{R}$  is a differentiable parameter-dependent coefficient function and the  $a^q(w, v) : X \times X \rightarrow \mathbb{R}$  are parameter-independent. If such a decomposition exist, the equation 4.3 can be specified into

$$\sum_{i=1}^N \sum_{p=1}^Q \lambda_i(\mu) \Theta^q(\mu) a^q(u(\mu_i), u(\mu_j)) = f(u(\mu_j)), \quad \forall j \in \{1, 2, \dots, N\}.$$

This form has the advantage, that the bilinear form is independent on  $\mu$  and thus can be precomputed and stored. This leads to further reduction of computational effort and time, however this decomposition is not always possible and tends to be difficult to achieve in reality. An example is given in chapter 5.

In some cases not the whole solution of the PDE is of interest but rather an output, such as pressure at certain points, maximum difference of heat or measurement at some selected points. This input-output behaviour is represented by

$$s(\mu) = l(u(\mu)) \tag{4.5}$$

in the case of EIT, for example, this would represent the measurement of voltage at known current at the sensors depending on the parameter  $\mu$ .

The solution is approximated by the finite element method, using triangulation, in this thesis. Other numerical methods such as finite volumes are also possible but are not addressed here. In the next section a *A Priori* convergence theory is presented to ensure convergence of the RB method.

## 4.2 A Priori Convergence Theory

Since the approximated solution is not calculated using the projection theorem we need to ensure convergence. In this section the following Lemma is proven, which ensures convergence in special cases using properties of the underlying PDE.

**Theorem 4.2.1** *Given 4.1 where,  $a(\cdot, \cdot; \mu)$ , is symmetric and satisfies a coercivity and continuity*

condition

$$0 < \alpha_0 \leq \alpha(\mu) \equiv \inf_{v \in X} \frac{a(v, v; \mu)}{\|v\|_X^2}, \quad \forall \mu \in D \quad (4.6)$$

$$\sup_{v \in X} \frac{a(v, v; \mu)}{\|v\|_X^2} \equiv \beta(\mu) < \infty \quad \forall \mu \in D. \quad (4.7)$$

The reduced basis approximation then is optimal with regard to the norm of  $X$  in the sense of

$$\|u(\mu) - u_N(\mu)\|_X \leq \sqrt{\frac{\beta(\mu)}{\alpha(\mu)}} \min_{w \in W_N} \|u(\mu) - w(\mu)\|_X.$$

**Proof 4.2.1** Since  $u(\mu)$  is a solution to 4.2 it is also true that

$$a(u(\mu), w; \mu) = f(w), \quad \forall w \in W_N$$

and therefore  $\forall w \in W_N$

$$\begin{aligned} 0 &= f(w) - f(w) \\ &= a(u(\mu), w; \mu) - a(u_N(\mu), w; \mu) \\ &= a(u(\mu) - u_N(\mu), w; \mu) \end{aligned}$$

It follows that  $\forall w_N = u_N + v \in W_N$ , where  $v \neq 0$ ,

$$a(u - w_N, u - w_N; \mu) = a(u - u_N - v, u - u_N - v; \mu) \quad (4.8)$$

$$= a(u - u_N, u - u_N; \mu) - \underbrace{2a(u - u_N, v; \mu)}_{=0} + a(v, v; \mu) \quad (4.9)$$

$$= a(u - u_N, u - u_N; \mu) + \underbrace{a(v, v; \mu)}_{\geq 0} \quad (4.10)$$

$$\geq a(u - u_N, u - u_N; \mu). \quad (4.11)$$

By coercivity it follows

$$\alpha(\mu) \|u(\mu) - u_N(\mu)\|^2 \leq a(u(\mu) - u_N(\mu), u(\mu) - u_N(\mu); \mu)$$



using 4.8

$$a(u(\mu) - w_N(\mu), u(\mu) - w_N(\mu); \mu) \leq a(u(\mu) - w_N(\mu), u(\mu) - w_N(\mu); \mu)$$

after applying the continuity condition

$$a(u(\mu) - w_N(\mu), u(\mu) - w_N(\mu); \mu) \leq \beta(\mu) \min_{w \in W_N} \|u(\mu) - w(\mu)\|^2$$

and therefore overall

$$\|u(\mu) - u_N(\mu)\|_X \leq \sqrt{\frac{\beta(\mu)}{\alpha(\mu)}} \min_{w \in W_N} \|u(\mu) - w(\mu)\|_X.$$

To proof convergence of  $u_N(\mu) \rightarrow u(\mu)$  we use the property of  $W_N$

$$\lim_{n \rightarrow 0} \min_{w \in W_N} \|u(\mu) - w(\mu)\|_X = 0.$$

Combing this theorem with theorem 4.2.1 convergence is shown. Note that the constants in 4.6 depend on  $\mu$ . Having shown convergence, we need to find an error estimation depending on  $N$  to find a good dimension of  $W_N$ . This means  $N$  should be small but the error should even smaller.

### 4.3 Dimension Reduction

To find an error estimation a few definitions are needed. The residual is defined as

$$r(v; \mu) = f(v) - a(u_N(\mu), v; \mu), \quad \forall v \in X. \quad (4.12)$$

the dual norm of this residual is

$$\varepsilon_N(\mu) = \sup_{v \in X} \frac{r(v; \mu)}{\|v\|_X}. \quad (4.13)$$

Let  $\alpha$  be a positive lower bound of  $\alpha(\mu)$  for all  $\mu \in D$ , i.e.

$$0 < \alpha_0 \leq \alpha \leq \alpha(\mu), \quad \forall \mu \in X,$$

then the energy error bound can be defined as

$$\Delta_N(\mu) = \frac{\varepsilon_N(\mu)}{\alpha} \quad (4.14)$$

and its associated effectivity as

$$\eta_N(\mu) = \frac{\Delta_N(\mu)}{\|e\|_X} \quad (4.15)$$

where  $e$  is the error  $u(\mu) - u_N(\mu)$ . Based on these definitions the following theorem is introduced which provides an error estimation.

**Theorem 4.3.1** *Let the bilinear form  $a(\cdot, \cdot; \mu)$  satisfies a coercivity and continuity condition as in theorem 4.2.1. Then the energy error bound,  $\Delta_N(\mu)$ , is an upper bound to the normed error,  $\|e\|_X$ , i.e.*

$$\Delta_N(\mu) \geq \|e\|_X. \quad (4.16)$$

The associated effectivity is bounded by

$$1 \leq \eta_N(\mu) \leq \frac{\beta(\mu)}{\alpha}. \quad (4.17)$$

**Proof 4.3.1** *From the definition of the residual 4.12 follows that  $\forall v \in X$ ,*

$$\begin{aligned} a(e, v; \mu) &= a(u(\mu), v; \mu) - a(u_N(\mu), v; \mu) \\ &= f(v) - a(u_N(\mu), v; \mu) \\ &= r(v; \mu) \end{aligned}$$

and note that by standard duality arguments, there exist  $\varepsilon(\mu) \in X$ , s.t.

$$\varepsilon_N(\mu) = \|r(v; \mu)\|_{X^*} = \|\varepsilon(\mu)\|_X$$

and

$$(\varepsilon(\mu), v)_X = r(v; \mu), \quad \forall v \in X.$$

Then using the coercivity of the bilinear form provides

$$\alpha \|e(\mu)\|_X^2 \leq a(e(\mu), e(\mu); \mu) \quad (4.18)$$

$$= r(e(\mu); \mu) \quad (4.19)$$

$$= (\varepsilon(\mu), e(\mu))_X. \quad (4.20)$$

By Cauchy-Schwarz inequality it follows

$$(\varepsilon(\mu), e(\mu))_X \leq \|\varepsilon(\mu)\|_X \|e(\mu)\|_X$$

and therefore

$$\begin{aligned} \alpha \|e(\mu)\|_X &\leq \|\varepsilon(\mu)\|_X \\ \Rightarrow \|e(\mu)\|_X &\leq \frac{\|\varepsilon(\mu)\|_X}{\alpha} = \Delta_N(\mu) \end{aligned}$$

With this the equation 4.16 for the error bound is proven. Now the bound on the associated effectivity needs to be shown. By the definition

$$\begin{aligned} \|\varepsilon(\mu)\|^2 &= a(e(\mu), \varepsilon(\mu); \mu) \\ &\stackrel{C.S.}{\leq} a(e(\mu), e(\mu); \mu)^{1/2} a(\varepsilon(\mu), \varepsilon(\mu); \mu)^{1/2} \\ &\leq \beta(\mu) \|\varepsilon(\mu)\|_X \|e(\mu)\|_X. \end{aligned}$$

It follows that

$$\alpha \leq \alpha(\mu) \leq \frac{\|\varepsilon(\mu)\|_X}{\|e(\mu)\|_X} \leq \beta(\mu). \quad (4.21)$$

To find the desired inequality rewrite  $\eta_N(\mu)$  as

$$\begin{aligned}\eta_N(\mu) &= \frac{\Delta_N(\mu)}{\|u(\mu) - u_N(\mu)\|_X} \\ &= \frac{\varepsilon_N(\mu)}{\alpha \Delta_N(\mu)} \\ &= \frac{\|\varepsilon(\mu)\|_X}{\alpha \|e(\mu)\|_X} \\ &\stackrel{4.21}{\leq} \frac{\beta(\mu)}{\alpha}\end{aligned}$$

and thus the upper bound is shown. Now for the lower bound we use that

$$\eta_N(\mu) = \frac{\|\varepsilon(\mu)\|_X}{\alpha \|e(\mu)\|_X}$$

and by 4.18 follows

$$\begin{aligned}\Rightarrow & \quad \alpha(\mu) \|e(\mu)\|_X \leq \|\varepsilon(\mu)\|_X \\ \Rightarrow & \quad 1 \leq \frac{\|\varepsilon(\mu)\|_X}{\alpha \|e(\mu)\|_X} = \eta_N(\mu)\end{aligned}$$

This theorem provides an estimation for the error and thus can be used for a stopping criteria to find  $N$  and  $W_N$ . Note that if a decomposition of  $a(\cdot, \cdot; \mu)$ , as provided in 4.4, is possible the  $\alpha$  can be determined via the minimum of  $\Theta^q(\mu)$  over  $\mu \in D$  and  $q = 1, \dots, Q$  since

$$\begin{aligned}a(w, w; \mu) &= \sum_{q=1}^Q \Theta^q(\mu) a^q(w, w) \\ &\geq \min_{q=1, \dots, Q} \{\Theta^q(\mu)\} \cdot \sum_{q=1}^Q a^q(w, w) \\ &= \min_{q=1, \dots, Q} \{\Theta^q(\mu)\} \cdot a(w, w) \\ &\geq \min_{q=1, \dots, Q} \{\Theta^q(\mu)\} \alpha_0 \|w\|_X\end{aligned}$$

if  $\Theta^q(\mu) > 0$  for all  $q = 1, \dots, Q$ ,  $\alpha = \alpha_0 \min_{q=1, \dots, Q} \{\Theta^q(\mu)\}$  can be chosen. In case,  $\alpha_0$  is unknown, the global minimum of  $\alpha(\mu)$  for  $\mu \in D$  can be chosen. Analog  $\beta$  can be approximated using  $\sup_{\mu \in D} \{\beta(\mu) \sup_{q=1, \dots, Q} \Theta^q(\mu)\}$ .  $\Theta^q$  is continuous and  $D \subset \mathbb{R}^d$  bounded in general and thus can be closed. Therefore, we can find a maximum/minimum for each  $q$  and therefore a maximum and

minimum overall  $q = \{1, 2, \dots, Q\}$ .

This approach is used in [3] to find an error estimation for the output-input behavior. In this thesis the approach is not pursued, since the assumption of a linear relation of input and output is not of particular interest for many inverse problems. Due to the fact, that a linear inverse problem can be solved directly using the FE method without many calculations of the underlying PDE, the advantage of the RB is lost.

## 4.4 Orthogonalization

So far the convergence of the reduced basis approximation towards the FE solution was shown as well as an error estimation provided to find an appropriated reduction dimension. However, to create the necessary  $N \times N$  equations for the reduced basis approximation. As given in 4.3. The calculation of

$$a(u(\mu_i), u(\mu_j) : \mu)$$

is necessary. This is usually done beforehand when  $W_N$  is being created and pre-stored to save computational time. The chosen  $u(\mu_i)$  may not be optimal to solve the algebraic system for the  $\lambda_i(\mu)$  however. In fact, since the solutions reside on a low dimensional manifold which is assumed to be smooth, the basis vectors tend to have similar directions, thus the algebraic system 4.3 is very ill conditioned. Using Gram-Schmidt to generate an orthogonal basis of  $W_N$  with respect to the inner product associated with the Hilbert Space  $X$ , will help to reduce the condition number of the system. The following theorem shows, that the change of the basis does not inflict conditioning properties, coercivity and continuity.

**Theorem 4.4.1** *Let  $w_n$  be a basis of  $W_N$  and  $\zeta_n$  an orthonormal basis of  $W_N$ . The algebraic system 4.3 inherits coercivity and continuity constants,  $\alpha(\mu)$  and  $\beta$ , i.e. the condition number has a upper bound  $\beta(\mu)/\alpha(\mu)$ .*

**Proof 4.4.1** *First note that  $w_n$  can be expressed in the new basis as*

$$w_n = \sum_{k=1}^N c_k^n \zeta_k$$

and thus using the coercivity property

$$\begin{aligned}
\sum_{k=1}^N \sum_{l=1}^N c_k^i c_l^j a(\zeta_k, \zeta_l; \mu) &= a(w_i, w_j; \mu) \\
&\geq \alpha(\mu)(w_i, w_j)_X \\
&= \alpha(\mu) \sum_{k=1}^N \sum_{l=1}^N c_k^i c_l^j (\zeta_k, \zeta_l)_X.
\end{aligned}$$

Since the  $\zeta$  are orthogonal and normal we have

$$\sum_{k=1}^N \sum_{l=1}^N c_k^i c_l^j (\zeta_k, \zeta_l)_X = \sum_{k=1}^N (c_k^i)^2$$

and thus

$$\sum_{k=1}^N \sum_{l=1}^N c_k^i c_l^j a(\zeta_k, \zeta_l; \mu) \geq \alpha(\mu) \sum_{k=1}^N (c_k^i)^2. \tag{4.22}$$

It follows that the coercivity property still hold for the new base. Analog we have

$$\sum_{k=1}^N \sum_{l=1}^N c_k^i c_l^j a(\zeta_k, \zeta_l; \mu) = a(w_i, w_j; \mu) \tag{4.23}$$

$$\leq \beta(\mu)(w_i, w_j)_X \tag{4.24}$$

$$= \beta(\mu) \sum_{k=1}^N \sum_{l=1}^N c_k^i c_l^j (\zeta_k, \zeta_l)_X \tag{4.25}$$

$$= \beta(\mu) \sum_{k=1}^N (c_k^i)^2 \tag{4.26}$$

From 4.22 and 4.23 follows the inequality

$$\alpha(\mu) \leq \frac{\sum_{k=1}^N \sum_{l=1}^N c_k^i c_l^j a(\zeta_k, \zeta_l; \mu)}{\sum_{k=1}^N (c_k^i)^2} \leq \beta(\mu) \tag{4.27}$$

and therefore the theorem is proven.

In [3] an example is given with the benefits in orthogonalizing the basis, see figure 4.1. In this example for the orthogonal basis the upper bound for the condition given by  $\frac{\beta(\mu)}{\alpha(\mu)}$  is reached at  $N = 4$ . The original basis displays an exponential growth (Data from [3]).

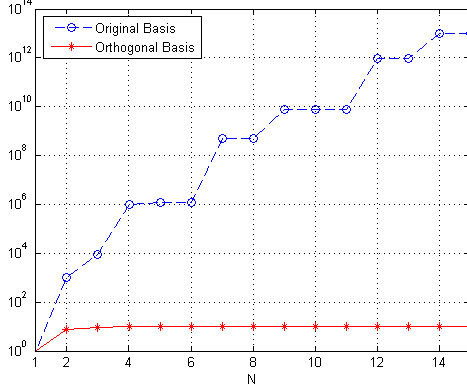


Figure 4.1: Condition number of reduced-stiffness basis in the original and orthogonalized basis as a function of  $N$ . The test point is  $\mu_t = (0.1, 1)$  for the thermal fin problem. [3]

## 4.5 Algorithm

The reduced basis method is separated into two steps, the off-line and the online stage. The off-line stage is performed once to generate  $W_N$  and  $S_N$ . In addition, if  $a(\cdot, \cdot; \mu)$  can be decomposed the  $a^q(\zeta_i, \zeta_j)$  are stored for all possible  $q, i$  and  $j$ . Note that the  $\zeta_i$  are an orthogonal basis for  $W_N$  as defined earlier. The online stage is based on the reduced basis found in the off-line stage to solve the given PDE by solving

$$B(\mu)u_n(\mu) = F$$

where the coefficients of  $B(\mu)$  and  $F$  are

$$B(\mu)_{i,j} = \sum_{q=1}^Q \Theta^q(\mu) a^q(\zeta_i, \zeta_j)$$

$$f_i = f(\zeta_i)$$

which depend on  $\mu$ .

First, we will have a closer look on the off-line stage before proceeding towards the online stage. To generate  $W_N$ , a greedy algorithm is used. This algorithm ensures to find a small  $N$  but still with a good estimation of the FE solution for a new  $\mu$  contrary to randomly selecting points in  $D$  to generate  $W_N$ . Let  $\Xi_h$  be a fine grid over  $D$  with the distance  $h$  between two adjacent grid points. In addition  $h \ll 1$  should be chosen. Based on this grid the  $\mu_i$  are chosen iterative for

$N = 0, 1, \dots, N_{max}$  via

$$\mu_{i+1} = \operatorname{argmax}_{\mu \in \Xi_h} \|u(\mu) - u_i \mu\|_X$$

and

$$W_{N+1} = W_N \cup \operatorname{span}\{u(\mu_{i+1})\}.$$

This iteration stops either if a  $N_{max}$  is reached or if  $\|u(\mu) - u_i \mu\|_X \leq \varepsilon_{tol,min}$ , where  $\varepsilon_{tol,min}$  is a prefixed error tolerance. The maximum dimension of  $W_N$  denoted as  $N_{max}$  also needs to be defined. This number should not be too small. Assuming the ideal case, where manifold  $M$  of the solutions is a linear subspace, its dimension is not larger than the dimension of  $D$ , the parameter domain. However, it can be equal and therefore,  $N \geq p$  should be chosen, when  $p = \dim(D)$ . Moreover,  $N$  should be chosen relatively small compared to the dimension of  $X$  to gain an computational advantage. We can write down a first version of the off-line step now.

---

### Greedy Algorithm

---

For  $N = 1 : N_{max}$

$$\mu_{N+1} = \operatorname{argmax}_{\mu \in \Xi_h} \|u(\mu) - u_N \mu\|_X$$

$$W_{N+1} = W_N \cup \operatorname{span}\{u(\mu_{i+1})\}$$

If  $\|u(\mu_{N+1}) - u_N \mu\|_X \leq \varepsilon_{tol,min}$

return;

end

End

Note that the minimum is found over  $\Xi_h$ , this means as proposed in [4] all points in  $\Xi_h$  are being evaluated. For obvious reasons, this is computationally very expensive. Instead of solving the PDE for each point, an iterative method can save expensive operations. This also has the advantage of not being restricted to the finite grid over  $D$ . Since the found minimum is only a local minimum and depends on the starting point, we have used different starting positions for the iterative method. Good starting points are shown to be the corner points of the parameter domain  $D$  as well as the center of  $D$ . If the domain is not restricted by polygons, opposite points on the boarder works well. The enhanced version of the greedy algorithm is



### Offline Algorithm (b)

---

---

For  $N = 1 : N_{max}$

$$\mu_{N+1} = \operatorname{argmax}_{\mu \in D} \|u(\mu) - u_N \mu\|_X$$

$$W_{N+1} = W_N \cup \operatorname{span}\{u(\mu_{i+1})\}$$

$$\text{If } \|u(\mu_{N+1}) - u_N \mu\|_X \leq \varepsilon_{tol, min}$$

return;

end

End

The only obvious change is that the maximum is now being taken over all  $D$ , however as an iterative method, i.e. gradient method, is being used, only local minima are found and  $N$  may be larger in this case than for the greedy algorithm. Furthermore, the conditioning of the boarder needs to be handled carefully. In general, a penalty term is added to find a minimum or maximum in a bounded area (see section 2.2). Nonetheless, the bilinear form may not be solvable for points outside of  $D$  and would lead to a computational error. To handle this problem a subset of  $D$  notated as  $D_g$  is being used. It has the following property

$$\forall x \in D_g : \min_{\mu \in \partial D} \|\mu - x\| \leq 2 * h$$

where  $h$  is defined by the approximation of derivatives, i.e.  $f'(x) \approx \frac{1}{2h}(f(x-h) - 2f(x) + f(x+h))$ . Now the gradient can be determined everywhere in  $D_g$  and additional points at the boarder of  $D_g$  are also evaluated and the penalty function can be used to restrict the solution on  $D_g$ . In both cases, if a decomposition of the bilinear form  $a(\cdot, \cdot; \mu)$  was possible where the  $a^q(\zeta_i, \zeta_j)$  can be computed and stored. This will make the online stage more efficient.

So far, we completed the off-line stage and found the desired reduced basis. In the next stage the reduced basis is used to find solutions for new  $\mu$ . In this stage the stiffness matrix  $B(\mu)$  with the coefficients are given as,

$$b(\mu)_{i,j} = a(\zeta_i, \zeta_j; \mu)$$

and the solution vector  $F$  defined as

$$f_j = f(\zeta_i)$$

needs to be computed and subsequently the linear system

$$B(\mu)u_N(\mu) = F$$

must be solved. Since the stiffness matrix  $B(\mu)$  has dimension  $N \times N$  and  $N \ll m$  is assumed, the computation is supposed to be much faster compared to solving the FE solution directly, which includes solving an  $m \times m$  linear system. Contrary to the matrix  $A(\mu)$  of the FE method, which is sparse in general, see 2.4, the matrix  $R(\mu)$  is full. The matrix is invertible if  $a(\cdot, \cdot; \mu)$  is well posed for  $\mu \in D$  otherwise we use methods such as the CG algorithm.

## 4.6 Application for Inverse Problems

Based on the theory of RB method, we now want to discuss its possible application to inverse problems. In the next chapter we formulate EIT as an application of the proposed approach. Solving the inverse problem numerically requires finding (global) minima, either solving the normal equations or by using minimizing a functional  $J$ . If the inverse problem is nonlinear, linearization is needed, solving the linear model for a new iterative step. Except for the solution of the normal equations, iterative procedures require many function evaluations. The RB method in combination with the Newton algorithm, can be used to solve non linear inverse problems based on fast RB solution of PDEs, because it reduces the needed computational time to evaluate the PDE. It does not provide a direct solution of a linear model, contrary to solving for example the normal equations.

Overall, a good application for the RB method is the minimization of the functional

$$J(\mu) = \|s(\mu) - g^\delta\|_X.$$

This functional does not include any regularization. In the next chapter is shown that the RB method includes a form of regularization. This follows directly from the assumption that the parameter domain  $\mathbb{R}^p$  has a small dimension. The possible solutions are restricted to this parameter domain

and thus a type of regularization exists. However, depending on the problem one may need to add additional regularization. One way is the Tikohnov functional

$$J_t(\mu) = \|s(\mu) - g^\delta\|_X + t \cdot \|\mu\|_V$$

introduced in 2.6, where  $s : V \rightarrow X$  is an Operator,  $\mu \in V$  and  $g^\delta$  is the known noised data. Other functionals are also possible and may lead to better results, depending on the Inverse Problem. The minimum of  $J(\mu)$  can now be determined using the gradient method if  $D$  is convex or unrestricted. In the case of EIT the Tikohnov Function is replaced by the TV functional

$$J_t(\mu) = \|s(\mu) - g^\delta\|_X + t \cdot \|\nabla \mu\|$$

as introduced in chapter 3. In the next chapter an algorithm is introduced and applied on the Inverse EIT problem.

## Chapter 5

# Algorithm for Inverse Problems

Because of the advantage of the Reduced Basis (RB) method in solving a given PDE fast multiple times for different parameters, the RB method is a good candidate for an algorithm to solve the inverse problem by minimizing functionals over a parameter domain  $D$ . The functional consist of the distance between reconstructed data and known noisy data. It can be extended with a regularization term. One example with regularization is the Tikohnov Functional

$$J_t(\mu) = \|s(\mu) - g^\delta\|_X + t \cdot \|\mu\|_V.$$

To apply RB to solve the inverse problem by minimizing the given functional  $J_t(\mu)$  three stages need to be considered.

**Stage 1** In the first stage the PDE needs to be analyzed and an output-input operator needs to be defined. By analyzing the PDE, mainly parameterization has to be done as well as a decomposition. The decomposition may not be possible but the RB method can still be applied as long as the PDE can be parameterized.

Additionally, the 'physical' domain  $\Omega$  of the PDE has to be discretized via triangulation for the FE approximation. This step can be done in the offline stage of the RB method, but to ensure a comparable solution it is advised to consider a fine discretization when analyzing the PDE. An example for this is given later in this section for the EIT.

**Stage 2** The second stage is the offline stage of the RB method. This means,  $W_N$  needs to be found and if the PDE is decomposed the  $a^q(\chi_i, \chi_j)$  should be saved. Note that the triangulation for the FE approximation is fixed in stage 1 and thus the computed solutions  $u$  for different  $\mu$  are comparable. However, if the bilinear form of the PDE cannot be decomposed, a new generation of the triangulation may be necessary for each  $\mu$ . Therefore, the parameter for the discretization needs to be similar in order to guarantee comparability of the different solutions  $u(\mu)$ .

**Stage 3** In this stage the output-input operator on the FE solution in combination with the parameterization being used to transform the inverse problem into a finite optimization problem. Using a gradient method a local minimum can be identified. The online stage of the RB method is used to guarantee a fast but still exact evaluation of the output-input behavior needed for the Newton step.

The three stages taken together form an algorithm to solve the inverse problem.

Algorithm
Set up PDE
Decompose weak formulation
Generate FE mesh
Offline stage of reduced basis method
Solve inverse problem
Using newton method to approximate input $\mu$
Use online stage of RBM to solve PDE

This general algorithm is now applied on an EIT problem. Each step of the algorithm is explained in its own section. The numerical results can be found in the next chapter, where they are compared to a solution found by simply solving the PDE in each step using the *MATLAB* PDE toolbox.

## 5.1 Problem Description

In our simulations, a sphere with center  $(x, y)$ , radius  $r$  and resistance  $\rho$  is to be identified by the EIT inverse problem. See the figure 5.1 for more explanation. The area  $\Omega$ , in which the sphere is located, is assumed to be a square with length  $1cm$  and midpoint at  $(0, 0)$ . The input  $\mu$  is

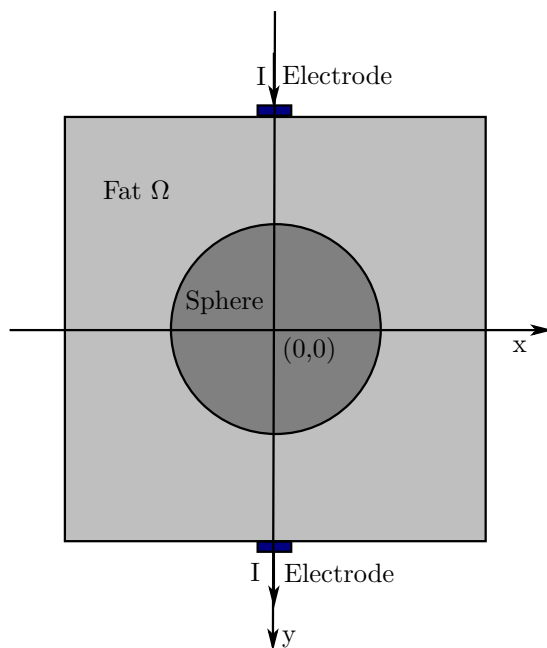


Figure 5.1: Setup of the example. Shown is the physical domain  $\Omega$ . In the middle is the sphere with center  $(x, y)$ . In the displayed case  $(x, y) = (0, 0)$ . The electrodes mark the place for current, but measurement is made on the whole boarder.

therefore

$$\mu = (x, y, r, \rho)^T.$$

For every fixed  $\rho$  the admissible  $(x, y, r)$  are shown in 5.2.

An electrical current is injected on the top and bottom boarder. Measurements are made on all four sides of  $\Omega$  for the Inverse Problem. Outside of the sphere, fat is assumed and thus  $\sigma(x) = 10^{-4}$  for  $x \notin B_r((x, y))$ . Note that the described PDE is an elliptic PDE, and can be solved using numerical algorithm.

## 5.2 Decomposition of the weak Problem

The assumed geometry of the underlying problem can be used to decompose the weak form of the problem in parameter independent sections, which multiplied by a scalar add up to the original problem. Let  $\mu_0 = (0, 0, 2.5, \rho)$  be a reference domain shown in figure 5.3. Denote this partition of  $\Omega$  as  $\Omega^0$ . To find the decomposition of the bilinear form we map  $\Omega(\mu) \rightarrow \Omega^0$  via continuous, piecewise-affine transformation, i.e. each sub domain is being mapped on its related part in the

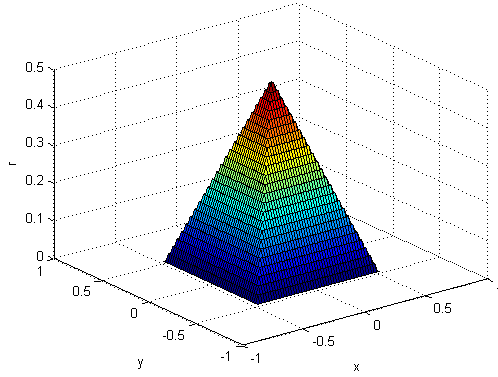


Figure 5.2: Domain of the Parameter  $D$  with respect to  $r, x, y$  for the parameterized EIT problem. The fourth dimension of  $D$  depending on  $\rho$  is left out since  $\rho$  is independent of  $x, y, r$ .

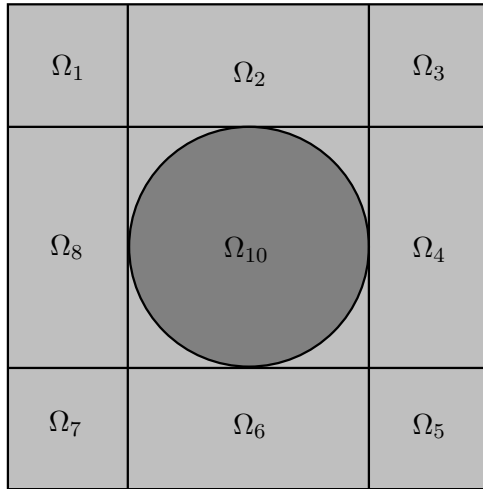


Figure 5.3: Segmentation of  $\Omega$  for the case  $\mu_0 = (0, 0, 2.5, \rho)$ .

reference domain. The result is presented in table A1, attached in the appendix.

Based on this decomposition, only  $\Omega^0$  needs to be discretized. This was done using the *MATLAB* build in routine for generating a mesh, which was refined twice. Based on the description of this function given in [11] a Delaunay triangulation algorithm is used to generate the initial mesh. Since this mesh proved to be too inaccurate and leading towards near singular matrices in the FEM a much finer mesh was used by applying the *MATLAB* build in function *refinemesh()* twice. The FE solution of this reference is shown in 5.4.

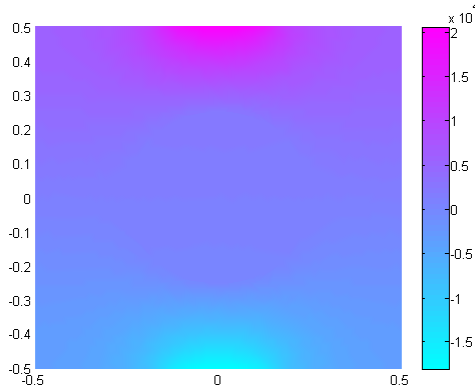


Figure 5.4: FE solution of the reference set  $\mu_0$  used to decompose the bilinear form for the EIT in  $\Omega$ .

q	$\Omega_q$	$\Theta^q(\mu)$
1	$[-0.5, -r] \times [r, 0.5]$	$\frac{(0.5-r)^2}{0.0675}$
2	$[-r, r] \times [r, 0.5]$	$\frac{2r(0.5-r)}{0.125}$
3	$[r, 0.5] \times [r, 0.5]$	$\frac{(0.5-r)^2}{0.0675}$
4	$[r, 0.5] \times [-r, r]$	$\frac{2r(0.5-r)}{0.125}$
5	$[r, 0.5] \times [-0.5, -r]$	$\frac{(0.5-r)^2}{0.0675}$
6	$[-r, r] \times [-0.5, -r]$	$\frac{2r(0.5-r)}{0.125}$
7	$[-0.5, -r] \times [-0.5, -r]$	$\frac{(0.5-r)^2}{0.0675}$
8	$[-0.5, -r] \times [-r, r]$	$\frac{2r(0.5-r)}{0.125}$
9	$[-r, r] \times [-r, r]/B_r((x, y))$	$\frac{(2*r)^2 - \pi*r^2}{0.5^2 - 0.25^2\pi}$
10	$B_r((x, y))$	$\frac{r^2}{0.25^2}$

Table 5.1: Decomposition used for the numerical implementation. This is the simplified case where  $(x, y) = (0, 0)$  is fixed.

### 5.3 Numerical Implementation

We made a few more assumptions for the problem described above for ease of computation. Due to computational ease,  $(x, y) = (0, 0)$  is assumed. As a result the dimension of  $\mathbb{R}^p$  is reduced to 2, which results in faster computation in the offline step and a reduced dimension of the RB  $W_N$ . The new parameter domain is  $D = [0, 0.5] \times (0, 100]$ . The decomposition of the bilinear form is unchanged but can be simplified as seen in 5.1. In addition,  $Q$  can be reduced to 4 by combining  $\Omega_1, \Omega_3, \Omega_5$  with  $\Omega_7$  and  $\Omega_2, \Omega_4, \Omega_6$  with  $\Omega_8$ , since their  $\Theta^q(\mu)$  is equal.

The last two stages are numerical implementation. For all calculations *MATLAB* is used. The used code is attached in the appendix. With the changes the offline stage of the RB method



$N$	$\ \mu - \mu_0\ _X$		$\mu_0$	noise level
	no regularization	regularization		
20	$1.35 \cdot 10^{-4}$	$1.02 \cdot 10^{-4}$	$(1.1 \cdot 10^{-3}, 0.3)$	1%
20	$2.76 \cdot 10^{-2}$	$1.54 \cdot 10^{-2}$	$(1.1 \cdot 10^{-3}, 0.3)$	10%
4	1.22		$(1.3 \cdot 10^{-3}, 0.2)$	5%

Table 5.2: Numerical results for the EIT example. The distance between reconstructed parameter  $\mu$  and the true source  $\mu_0$  are shown. In addition it is performed for multiple  $N$  and  $\mu_0$ .

can be performed. This stage is very time consuming, even if the enhanced algorithm is used. For this example  $N_{max} = 30$  is chosen and all solution  $u_N(\mu)$  are saved, thus it is possible to compare the solutions of the inverse problem for different  $N \in \{1, 2, \dots, 30\}$  with only one offline stage. In [4] and [3] this offline stage is performed using a super computer. For this thesis a dual core pc was used and could perform the offline stage.

Based on the  $W_N$  found in the offline stage the matrix  $A^q$  is generated and stored. To perform the last step for different  $N$  only the first  $N$  rows and columns are used. As the dimension of the parameter space is reduced no further regularization is required.

## 5.4 Results

The numerical results are shown in table 5.2. Since for one offline stage of the RBM, multiple test for different  $N$  can be made. The solution for large and small  $N$  are being compared. As can be seen good approximations of the underlying source can be made with less than  $N = 30$ .

The numerical results are promising and demonstrates the advantage of using the RB method in inverse problems, however there are some limitations in using the RB method. First of all, it is restricted for solving PDE only and thus cannot provide a general theory for solving nonlinear inverse problems. This is not a major drawback since many models in application are based on PDEs. Even if only some of the applications are PDE based, the RB method could still be valuable for these problems. A more serious concern is that a decomposition of the bilinear form is needed to be efficient in RB method. As it can be seen from the simple example given in this chapter that the decomposition of the PDE bilinear form may not possible for a general PDE based inverse problem. Furthermore, the proposed RB method for inverse problems depends heavily on the parameterization of the underlying PDE. This may lead to more restrictions on the geometry of the problem.

Besides these drawbacks that are direct consequences of the RB method, there is other

indirect drawbacks. For example, the basic assumption of the RB method is based on a low dimensional manifold. The dimension of the manifold follows directly from the dimension of the parameter space  $\mathbb{R}^n$ . Therefore, the RB method can only be applied on PDEs with a finite parameter space. Therefore,  $n$  needs to be small in order to gain computational advantage with the RB.

Nevertheless, the RB method has several advantages for solving inverse problems. The separation into offline and online stage allows for fast computation on user PCs, or even smart phones. Therefore it may be very useful to make initial screening whether a probe should be further examined or not. Likewise, since the PDE needs to be parameterized a regularization is already applied, since known information is not only taken into account via penalty function but the solution space is reduced. Depending on the problem this regularization may be sufficient to solve the inverse problem without traditional regularization.

## Chapter 6

# Conclusion and Discussion

It was shown that the RBM can be used to solve inverse problems that are based on PDE's. Both linear and nonlinear inverse problems can be tackled. Based on the result in chapter 4, it displays a fast and good approximation of the real solution. In addition, due to its nature to depend on special parameterized PDE's it includes a form of regularization already. However it suffer under strong limitations and only a small group of PDE's can be solved using this method. Moreover, the parameter space needs to have a small dimension. Its full potential embraces the RBM if the bilinear form of the PDE can be decomposed. This is not always possible and may lead to a more strict bound on the parameter. Overall, the disadvantages seem to be too sever to allow a wider use of the RBM in the field of inverse problems.

### 6.1 Theoretical Implications and Recommendations for future Research

The main restriction for the RBM is its assumption of a low dimensional manifold. Some issues can be addressed if the manifold does not need to be low dimensional. This would make the RBM more valuable for inverse problems. Even with all its restriction some applications can be assumed, such as regularization based on parameter identification or quick test for engines where it is enough to find out if further repairs need to be made.

# Appendices

## Appendix A Tables

q	$\Omega_q$		$\Theta^q(\mu)$
1	$[-0.5, -r+x]$	$\times [r+y, 0.5]$	$\frac{ -r+x+0.5  \cdot  0.5-r-y }{0.25^2}$
2	$[-r+x, x+r]$	$\times [r+y, 0.5]$	$\frac{2r \cdot  0.5-r-y }{0.25 \cdot 0.5}$
3	$[x+r, 0.5]$	$\times [r+y, 0.5]$	$\frac{ 0.5-x-r  \cdot  0.5-r-x }{0.25^2}$
4	$[x+r, 0.5]$	$\times [-r+y, r+y]$	$\frac{2r \cdot  0.5-r-x }{0.25 \cdot 0.5}$
5	$[x+r, 0.5]$	$\times [-0.5, -r+y]$	$\frac{ -r-x+0.5  \cdot  -0.5-r+y }{0.25^2}$
6	$[x-r, x+r]$	$\times [-0.5, -r+y]$	$\frac{2r \cdot  0.5-r+y }{0.25 \cdot 0.5}$
7	$[-0.5, x-r]$	$\times [-0.5, -r+y]$	$\frac{ -r+x+0.5  \cdot  -0.5-r+y }{0.25^2}$
8	$[-0.5, x-r]$	$\times [-r+y, r+y]$	$\frac{2r \cdot  0.5-r+x }{0.25 \cdot 0.5}$
9	$[x-r, x+r]$	$\times [y-r, r+y]/B_r((x, y))$	$\frac{4 \cdot 0.25^2 - \pi r^2}{4r^2 - \pi r^2}$
10		$B_r((x, y))$	$\frac{\pi r^2}{\pi 0.25^2}$

Table A1: Decomposition of the described PDE in Chapter 4.

## Appendix B Code

```
1 %Test script
2 clear
3 % Load data created with pdetool box app
4 % geometry
5 load('geomat.mat'); %includes gd,ns,fs
6
7 % Mesh information
8 load('meshmat.mat'); % includes p,e,t
9
10 % plot loaded data
11 %pdemesh(p,e,t)
12 %axis equal
13
14 % solve the pde
15 u = assempde(@(p,e,u,time)pdebound(p,e,u,time, 0.25),...
16     p,e,t,genc(1e-4,1.3e-3,0.25),0,0);
17
18 % plot solution
19 figure;
20 pdeplot(p,e,t,'xydata', u)
21 figure;
22 pdeplot(p,e,t,'xydata',log(u))
23
24 %% start offline stage
25 rbmoffline
26
27
28 %% Test
29 % This part can be used for testing the RBM method in general
30 %
31 Nmax = 3;
32 urbm = rbmonline([0.25,1.1e-3],f(1:Nmax),a1(1:Nmax,1:Nmax),...
33     a2(1:Nmax,1:Nmax),a3(1:Nmax,1:Nmax),a4(1:Nmax,1:Nmax),un(:,1:Nmax));
34 udirect = minimizer(a1,a2,a3,a4,f,[0.25,1.1e-3],p,e,t);
35 udirect = tri2grid(p,t,udirect,x,y);
36 udirect = reshape(udirect,500*500,1);
37 disp(norm(urbm-udirect)/250000)
38 %% genrate noisy data
39 % A multiplicative error is used
40 g = pdesolver(mu);
41 level = 0.1;
42 I = level*rand(size(g,1),size(g,2))+ones(size(g,1),size(g,2));
43 g = I.*g;
44 % solve problem
```

```

45 muo = [0.25; 1.3e-3];
46 munew = onlinestep(a1,a2,a3,a4, mu0);

```

matlab/test.m

```

1 function [q, g, h ,r] = pdebound(p,e,u,time, rad)
2 % Funtion to create boundary matrix
3 % The generated bound is used if the PDE is mapped onto the standard case
4 % where r=0.25, (x,y) = (0,0)
5 %
6 % Template given in pde handbook is used
7 % In addition to p,e,u, time, the parameter r is needed to
8
9 ne = size(e,2); % number of edges
10 % The matrices are defined for boudary contions. All four matrices are
11 % needed to be assembled.
12 %
13 % Dirichlet condition
14 % hu = r
15 %
16 % Neumann condition
17 % n(cVu) + qu = g
18 q = zeros(1,ne);
19 g = q;
20 h = zeros(1,2*ne);
21 r = h;
22
23 % Fill matrices
24 for k=1:ne
25     % Find general points to use for conditions, depending on x and y
26     x1 = p(1,e(1,k));
27     x2 = p(1,e(2,k));
28     xm = (x1+x2)/2;
29     y1 = p(2,e(2,k));
30     y2 = p(2,e(2,k));
31     ym = (y1+y2)/2;
32
33     % Need to adjust the boundary for each new input mu. Since only \(\r\)
34     % effects the boundary geometry, it is the only input considered to
35     % adjust the boundary.
36     if abs(xm) <= 0.1*0.25/rad
37         g(k) = rad/0.25*sign(ym)*10;
38     end
39     % The case if the geometry is not effected by the input mu.
40     %switch e(5,k)
41     % case {4}

```

```

42     %     g(k) = -10;
43     %     case {17}
44     %     g(k) = 10;
45     %     otherwise
46         %since all points are already zero no further steps needed.
47     %end
48
49 end

```

matlab/pdebound.m

```

1 function [q, g, h ,r] = pdebound2(p,e,u,time, rad)
2 % Funtion to create boundary matrix
3 % The generated bound is used if the PDE is not mapped onto the standard
4 % case
5 %
6 % Template given in pde handbook is used
7 % In addition to p,e,u, time, the parameter r is needed to
8
9 ne = size(e,2); % number of edges
10 % The matrices are defined for boudary contions. All four matrices are
11 % needed to be assembled.
12 %
13 % Dirichlet condition
14 % hu = r
15 %
16 % Neumann condition
17 % n(cVu) + qu = g
18 q = zeros(1,ne);
19 g = q;
20 h = zeros(1,2*ne);
21 r = h;
22
23 % Fill matrices
24 for k=1:ne
25     % The case if the geometry is not effected by the input mu.
26     switch e(5,k)
27         case {4}
28             g(k) = -10;
29         case {17}
30             g(k) = 10;
31         otherwise
32             %since all points are already zero no further steps needed.
33     end
34
35 end

```



matlab/pdebound2.m

```
1 function c = ccoeff(x,y,sd,t)
2 % Function generates the coefficients for the PDE. It also converts the
3 % solution into an accepted form for the pde solver.
4 c1 = 0.01;
5 c2 = 400;
6
7 %%
8 % This part can be used for different type of mapping
9 %r = sqrt(x(1,:).^2+x(2,:).^2);
10
11 %c = c1* ones(size(x,1),size(x,2));
12 %c(r<=0.25)=c2;
13 %%
14 %
15 % This case is used
16 if sd == 13
17     c = c2;%[c2, 0; 0, c2];
18 else
19     c = c1;%[c1,0;0,c1];
20 end
21
22 end
```

matlab/ccoeff.m

```
1 %
2 % Script for reproducing pictures
3 %
4 % This script was used for the thesis and the presentation/defense.
5 % Reconstruction of Condition Number. Data only apprixmated, since a
6 % slightly different model was used to create them.
7 n = 1:15;
8 c1 = [1 10^-3 0.9*10^-4 10^-6 1.25*10^-6 1.25*10^-6 5*10^-8 5*10^-8 8*10^-9 ...
9       8*10^-9 8*10^-9 9*10^-11 9*10^-11 10^-13 10^-13];
10 c2 = [1 7.5 9 10 10 10 10 10 10 10 10 10 10 10];
11 % Plot data using a logarithmic scale for y axis
12 semilogy(n,c1,'b--o')
13 hold all
14 semilogy(n,c2,'r-*')
15 grid on
16 set(gca,'XLim',[1 15])
17 set(gca,'XTick',[1 2:2:14])
```

```

18 legend('Original Basis', 'Orthogonal Basis','Location','NorthWest')
19 xlabel 'N'
20 %%
21 % Domain of the decomposition for x,y, and r
22 figure
23 x = linspace(-1,1,100);
24 y = linspace(-1,1,100);
25 [X,Y] = meshgrid(x,y);
26 Z = 0.5-max(abs(X),abs(Y));
27 Z(Z<=0) = NaN;
28 surf(X,Y,Z)
29 xlabel 'x'
30 ylabel 'y'
31 zlabel 'r'

```

matlab/recreatepictures.m

```

1 function u = gramschmidt(v)
2 % Function generates a orthonormal basis for a given basis v based on the
3 % Gram-Schmidt Process
4 % Input:
5 % - v, Matrix with basis vectors in Columns
6 %
7 % Output:
8 % - u, Matrix of new basis with vectors in columns
9
10 u = zeros(size(v,1),size(v,2));
11 u(:,1) = v(:,1)/norm(v(:,1));
12 for k = 2: size(v,2)
13     temp = zeros(size(v,1),1);
14     for l = 1 : k-1
15         temp = temp + v(:,k)'*u(:,l)*u(:,l);
16         % Since norm(u(:,l))=1, (u(:,l),u(:,l))=1
17     end
18     % Generate new basis vektor, by projection
19     u(:,k) = v(:,k)-temp;
20     % Normalize new vector
21     u(:,k) = u(:,k)/norm(u(:,k));
22 end

```

matlab/gramschmidt.m

```

1 function [x,iter] = grad_verfahren(func, x0, epsi)
2 % Gradient mehod
3 % input:
4 % - func, function for which minimum is being seeked.

```

```

5 % - x0,    starting point
6 % - epsi,  minimum function difference found in between two steps, is used
7 %         to generate a stop criterion
8 %
9 % output:
10 % - x,    found point of minimum
11 % - iter, number of needed iterations
12 %
13 %
14 hepsi = 1e-6; % needed for approximation of the gradient
15 maxIt = 5000; % maximum number of iterations
16 x=x0;
17 fold = func(x);
18 iter = 1;
19 while iter < maxIt;
20     % count number of iterations
21     iter = iter+1;
22     % calculate gradient
23     grad = zeros(1,size(x0,2));
24     for l = 1:size(x0,2)
25         d = [zeros(1,l-1) 1 zeros(1,size(x0,2)-l)];
26         grad(l) = (func(x+hepsi*d)-func(x-hepsi*d))/(2*hepsi);
27     end
28     s= -grad./norm(grad); % direction of search
29     % sadjusted increment
30     beti = 0.2; % needs to be in (0,1)
31     omi = 0.8; % needs to be in (0,1)
32     alphi = 1;
33     counter = 1;
34     con = 1; % condition met true or false
35     fcurrent = func(x);
36     while con && counter < 20; % no more then 20 steps
37         counter = counter +1;
38         if func(x+alphi*s) > fcurrent+omi*alphi*grad*s'
39             alphi = beti*alphi;
40         else
41             while func(x+alphi*s) <= fcurrent+omi*alphi*grad*s'
42                 alphi = alphi / beti;
43             end
44             con = 0;
45         end
46     end
47     % generate new point of evaluation
48     x = x + alphi*s;
49     % see if stop criterion is met
50     if norm(fcurrent-fold)<epsi

```

```

51         break;
52     end
53 end
54 end

```

matlab/grad\_verfahren.m

```

1 %
2 %             rbm_offline
3 %
4 % Make changes here
5
6 N = 5; %Square will be the maximum number of points for WN
7 points = 500; %
8 %
9 % No changes from here on
10 %
11 [r,sig] = meshgrid(linspace(0.25,0.25,N),linspace(1e-3,2e-3,N));
12 N = N^2;
13 %
14 % Generate starting points for the offline stage of the RBM
15 % starting points for gradient method, must have N elements
16 SN = [reshape(r,1,N);reshape(sig,1,N)];
17 %SN = [linspace(0.25,0.25,N);linspace(1e-3,1.2e-3,N)];
18 h = 1/points;
19 % Square grid for the evaluation of solution u
20 [x,y] = meshgrid(linspace(-0.5,0.5,points),linspace(-0.5,0.5,points));
21 f = zeros(1,N);
22 mint = [1 2*ones(1,4*points-2) 1];
23 un = zeros(points^2,N);
24 gux = un;
25 guy = un;
26 a1 = zeros(N,N);
27 a2 = zeros(N,N);
28 a3 = zeros(N,N);
29 a4 = zeros(N,N);
30 % Find area of the decomposition
31 I1 = x.^2+y.^2<=0.25^2;%speher in center
32 I2 = abs(x)>=0.25;
33 temp = abs(y)>=0.25;
34 I2 = I2.*temp; % Corners
35 temp = max(abs(x),abs(y))<=0.25;
36 I3 = (1-I2).*(1-temp);% Sides
37 I4 = x.^2+y.^2>=0.25^2;
38 I4 = I4.*temp;
39 %

```

```

40 %
41 % Generate the basis vectors for Wn
42 % No stop criteria is needed, all un are safed to compare later
43 for k = 1:N
44     % Find the next un
45     u = minimizer(a1,a2,a3,a4,f,SN(:,k),p,e,t);
46     % safe bilinear forms
47     u = tri2grid(p,t,u,x,y);
48     temp = u;
49     [ux,uy] = gradient(u,h,h);
50     gux(:,k) = reshape(ux,points*points,1);
51     guy(:,k) = reshape(uy,points*points,1);
52     un(:,k) = reshape(u,points*points,1);
53     % safe f, which is the integration over the partial of Omega
54     temp = [temp(:,1); temp(:,end); temp(1,:)'; temp(end,:)'];
55     f(k) = h*mint*temp;
56     % Fill new stiffnesmatrix
57     for l =1:k
58         bilin = reshape(gux(:,k),points,points)...
59             .*reshape(gux(:,l),points,points)+...
60             reshape(guy(:,k),points,points).*reshape(guy(:,l),points,points);
61         a1(1,k) = integrate(bilin(I1),h,h);
62         a1(k,1) = a1(1,k);
63         a2(1,k) = integrate(bilin(I2==1),h,h);
64         a2(k,1) = a2(1,k);
65         a3(1,k) = integrate(bilin(I3==1),h,h);
66         a3(k,1) = a3(1,k);
67         a4(1,k) = integrate(bilin(I4==1),h,h);
68         a4(k,1) = a4(1,k);
69     end
70 end
71
72
73 %[un,a1,a2,a3,a4,f]

```

matlab/rbmoffline.m

```

1 function c = genc(c1,c2,r)
2 %c1 = num2str(c1);
3 %c2 = num2str(c2);
4 c = '';
5 for k= 1:12
6     switch k
7         case {1,2,4,7}
8             temp = c1*2*r*(0.5-r)/(0.125);
9             c = [c num2str(temp) '!'];

```

```
10     case {3,5,6,8}
11         temp = c1*(0.5-r)^2/(0.0625);
12         c = [c num2str(temp) '!'];
13     otherwise
14         temp = c1*(4*r^2-pi*r^2)/(4*0.25^2-pi*0.25^2);
15         c = [c num2str(temp) '!'];
16     end
17 end
18 c = [c num2str(c2*r^2/(0.25^2))];
19
20 end
```

matlab/genc.m

# Bibliography

- [1] Liliana Borcea. Electrical impedance tomography. *Inverse Problems*, 18(6):R99, 2002.
- [2] Prof. Dr. Christof Büskens. *Numerik der Optimierung*. Fachbereich 3, University of Bremen, 2011/2012. Skript zur Vorlesung.
- [3] Nguyen Ngoc Cuong. *Reduced-Basis Approximation and A Posteriori Error Bounds for Non-affine and Nonlinear Partial Differential Equations: Application to Inverse Analysis*. PhD thesis, Singapore-MIT Alliance, 2005.
- [4] Nguyen Ngoc Cuong, Karen Veroy, and Anthony T. Patera. Certified real-time solution of parametrized partial differential equations.
- [5] G. Folland. *Introduction to Partial Differential Equations*. Princeton, NJ: Princeton University Press, 1995.
- [6] Bangti Jin, Taufiqar Khan, and Peter Maass. A reconstruction algorithm for electrical impedance tomography based on sparsity regularization. *International Journal for Numerical Methods in Engineering*, 89(3):337–353, 2012.
- [7] Bangti Jin and Peter Maass. Sparsity regularization for parameter identification problems. *Inverse Problems*, 28(12):123001, 2012.
- [8] C. Johnson. *Numerical Solution of Partial Differential Equations by the Finite Element Method*. Dover, 2009.
- [9] G. Keller. *Electrical Properties of Rocks and Minerals, Handbook of Physical Constants*. New York: Geological Society of America, 1988.
- [10] Königsberger. *Analysis 1*. Springer-Verlag Berlin Heidelberg New York, 2004.
- [11] MathWorks. Documentation center, 2013.
- [12] Andreas Rieder. *Keine Probleme mit Inversen Problemen. Eine Einführung in ihre stabile Lösung*. Vieweg & Sohn Verlag, 2003.
- [13] L. Rondi and F. Santosa. Enhanced electrical impedance tomography via the mumford-shah functional. *ESAIM: Control, Optimisation and Calculus of Variations*, 2001.
- [14] H. Schwan and C. Kay. The conductivity of living tissues. *NY Acad. Sci.*, 1957.
- [15] Dirk Werner. *Funktionalanalysis*. Springer Verlag, 2011.