Multiscale Optimization via Multilevel PCA-based Control Space Reduction
in Applications to Electrical Impedance Tomography

by

Maria Minhee Felicia Monica Chun

Bachelor of Science
Biomedical Engineering
Florida Institute of Technology
2020

A thesis
submitted to the College of Engineering and Science
at Florida Institute of Technology
in partial fulfillment of the requirements
for the degree of

Master of Science
in
Operations Research

Melbourne, Florida
May, 2022
The author grants permission to make single copies.
We the undersigned committee
hereby approve the attached thesis

Multiscale Optimization via Multilevel PCA-based Control Space Reduction
in Applications to Electrical Impedance Tomography

by Maria Minhee Felicia Monica Chun

Vladislav Bukshtynov, Ph.D.
Assistant Professor
Mathematical Sciences
Major Advisor

Christopher Bashur, Ph.D.
Associate Professor
Biomedical and Chemical Engineering and
Sciences

Ryan T. White, Ph.D.
Assistant Professor
Mathematical Sciences

Gnana Bhaskar Tenali, Ph.D.
Professor and Department Head
Mathematical Sciences
ABSTRACT

Title: Multiscale Optimization via Multilevel PCA-based Control Space Reduction in Applications to Electrical Impedance Tomography

Author: Maria Minhee Felicia Monica Chun

Major Advisor: Vladislav Bukshtynov, Ph.D.

A fully developed computational framework for the optimal reconstruction of binary-type images suitable for various models seen in biological and medical applications is developed and validated. This framework enables solutions to the inverse electrical impedance tomography (EIT) problems of cancer detection at different levels of complexity with multiple cancer-affected regions of different sizes based on available measurements usually affected by noise. A new spatial partitioning methodology and efficient scheme for switching between fine and coarse scales are developed to allow higher variations in the geometry of reconstructed binary images with superior performance confirmed computationally on various models. A nominal number of input parameters makes the approach simple for practical implementation in diverse settings. An easy-to-follow design of the entire framework allows extending its functionality to new models in many other applications and further enhancing its performance. The complete computational framework is tested in applications to 2D inverse EIT problems and demonstrates its high potential for improving the overall quality of EIT-based procedures.
Table of Contents

Abstract iii

List of Figures vi

Nomenclature viii

Acknowledgments x

Dedication xi

1 Introduction 1

2 Mathematical Models and Computational Algorithms 6

2.1 Inverse EIT Problem: Mathematical Model . . . . . . . . . . . . 6

2.2 Multiscale Control Space Reduction . . . . . . . . . . . . . . . 9

2.2.1 PCA-based Fine Scale Optimization . . . . . . . . . . . . 9

2.2.2 Coarse Scale Optimization via Partitioning . . . . . . . . . 10

2.3 Switching Between Scales . . . . . . . . . . . . . . . . . . . . 12

3 Partitioning via Spatial Element Grouping 18

3.1 Motivation and Criteria . . . . . . . . . . . . . . . . . . . . . . 18

3.2 Algorithm Description . . . . . . . . . . . . . . . . . . . . . . . 19

3.2.1 Input Data . . . . . . . . . . . . . . . . . . . . . . . . . . 19
List of Figures

2.1 Schematic illustrating the general concept of the multiscale optimization framework ............................................................................. 13

3.1 Model A: a 10 × 10 rectangular model with three high-value spots ................................................................. 26

3.2 Model B: a 10 × 10 rectangular model with one high-value spot of non-trivial shape ............................................... 29

3.3 Model C: a 10 × 10 rectangular model with six high-value spots of various sizes ............................................................... 30

3.4 Model D: a sample model in FreeFemm++ containing 16 spatial elements (triangles) .................................................. 32

4.1 2D geometry of electrodes and electrical potentials ................................................. 37

4.2 Spatial discretization of domain Ω ................................................................. 38

4.3 EIT Model #1: comparison of optimization results for $N_{\text{max}} = 1$ and $N_{\text{max}} = 5$ ............................................................... 41

4.4 EIT Model #1: comparison of optimization results for different parameters .................................................................................. 42

4.5 EIT Model #1: optimization results with increased ($K = 4$) data ...................................................................................... 43

4.6 EIT Model #1: optimization results with increased ($K = 4$) data and added noise ........................................................................ 44
4.7 **EIT Model #2**: comparison of optimization results for $N_{\text{max}} = 1$ and $N_{\text{max}} = 5$ .......................................................... 46

4.8 **EIT Model #2**: comparison of optimization results for different parameters .......................................................... 47
Nomenclature

\( \Omega \)  physical domain
\( n \)  space dimension
\( x \)  spatial variable (independent)
\( \sigma(x) \)  electrical conductivity
\( U \)  electrical potential at electrode \( E \)
\( \ell \)  electrode number
\( m \)  total number of electrodes
\( E \)  boundary electrode
\( Z \)  contact impedance
\( I \)  electrical current
\( \partial \Omega \)  boundary of domain \( \Omega \)
\( u(x) \)  electrical potential
\( n \)  external unit normal vector on \( \partial \Omega \)
\( I^* \)  measurement of electrical current \( I \)
\( J(\sigma) \)  objective function
\( \beta \)  scalarization weights in objective \( J(\sigma) \)
\( K \)  number of permutations of potentials within set \( U \)
\( \hat{\sigma}(x) \)  optimal solution for \( \sigma(x) \)
\( \psi(x) \)  adjoint PDE solution
\( \xi \)  fine scale reduced dimensional control vector
\( \Phi \)  PCA linear transformation matrix
\( \bar{\sigma}(x) \)  prior mean for PCA transform
\( N_r \)  number of realizations to construct matrix \( \Phi \)
\( N_\xi \)  number of principal components in truncated PCA
\( r_\xi \)  energy of full set ob basis vectors in PCA
\( \zeta \)  coarse scale reduced dimensional control vector
\( N_\zeta \)  size of control \( \zeta \)
\( \Delta \)  area of fine mesh elements
\( N \)  number of fine mesh elements
\( C \)  partitioning subsets
\( M \)  partitioning map
\( P \) partitioning (indicator) function
\( n_s \) number of iterations between switching scales
\( k_s \) count for switching cycles
\( \chi_c \) coarse-scale indicator function
\( \varepsilon_f \) termination tolerance for fine scale
\( \varepsilon_c \) termination tolerance for coarse scale
\( \alpha_{c\rightarrow f} \) coarse-to-fine projection parameter
\( N_{\text{max}} \) maximum number of expected cancer-affected regions
\( \delta_\zeta \) finite difference scheme parameter
\( r_\Omega \) radius
\( w \) electrode half-width
\( \sigma_{\text{true}}(x) \) true electrical conductivity
\( \Omega_c \) cancer affected region of \( \Omega \)
\( \sigma_c \) true electrical conductivity for \( \Omega_c \)
\( \Omega_h \) healthy tissue part of \( \Omega \)
\( \sigma_h \) true electrical conductivity for \( \Omega_h \)
\( J_p \) Tikhonov-type penalty term
\( J_f \) penalization term for fine scale optimization
\( \beta_f \) coefficient for \( J_f \)
\( J_c \) penalization term for coarse scale optimization
\( \beta_c \) coefficient for \( J_c \)
\( i, j, k \) summation indices
\( k \) also a number of current major iteration
Acknowledgements

I want to acknowledge and thank my research advisor Dr. Vladislav Bukshtynov for guiding me with my research. I would also like to thank my committee members for letting my defense be an enjoyable moment and for the helpful comments and suggestions for improvement. I would also like to thank Dr. Ryan T. White personally for helping me during difficult times and always being there to assist in any way and form. In addition, I would like to appreciate Priscilla Koolman for her related research. Thank You.
I dedicate my thesis work to my beloved family. To my loving parents Ogi Chun and Felicia Chun for always loving, supporting, and believing in me. They made me the person I am today, and I am beyond grateful for everything they’ve done for me. To my brother Daniel who has always looked after me and been by my side through thick and thin.

I also dedicate this thesis to my grandparents, Moon-Hye Kim, Kyoung-Ho Chun, Soon-Hee Kim, and You-Hee Chyu, for their unconditional love. I hope I have made you proud.
Chapter 1

Introduction

The description of cancer has been existing since the days in Egypt around 3000 BC. Despite decades of innovative clinical research and trials of new therapies, cancer is a threat to the world and is causing major morbidity and mortality cases. In 2020, an estimation of around 18.1 million cases of cancer was reported around the world [2]. Consequently, the importance of finding new and innovative ways of detecting cancer is in high demand. Both computed tomography (CT) and magnetic resonance imaging (MRI) have a powerful impact on the practice of medicine. Both techniques are frequently used to detect and monitor cancer within the body. However, they both come with adverse effects. CTs use ionizing radiation, meaning they could damage DNA and raise the risks of developing cancers and leukemia [24]. An MRI uses the gadolinium-based contrast dye (GBCD), a substance injected into the body through a vein to activate multiple sclerosis (MS) to monitor the MS progression. Studies demonstrate that gadolinium accumulates in tissues, including the brain, bone, and kidneys. Patients exposed to GBCAs during MRIs get harmful side effects since the dye accumulates within the body when used frequently over time [25].
Additionally, MRIs require more time to conduct in a closed tube design, which causes some individuals to experience anxiety or claustrophobia. Also, this procedure is highly noisy and therefore requires ear protection [3]. Additionally, MRIs are expensive and this, therefore, limits their availability to individuals with financial difficulties. Ultimately, MRI and CT techniques come with risks and uncertainties, raising multiple concerns among patients and healthcare professionals.

Electrical Impedance Tomography (EIT) is a non-invasive medical imaging method where surface electrodes are attached to the skin above the tissue examined with an application of small constant or alternating currents [23]. In this work, we direct our efforts to improve the EIT procedures used to obtain equivalently accurate and effective results compared to those of MRIs and CTs. Since EIT techniques provide a non-invasive imaging method, it does not cause damage to the body and includes benefits such as portability, low cost, and, most importantly, safety. It makes EIT a justifiable competitor with MRI and CT machines. Developing the use of EIT for more advanced non-invasive procedures in medicine is a field that will bloom since it can provide imaging of the interior of the human body in real-time without causing or putting the body at risk. We aim at improving this methodology by enabling the detection and measurements of cancerous tissues and their growths in real-time before they become too aggressive and potentially lethal. Early detection of certain cancers increases the chance of curing them and minimizing damaging effects on the patient’s body [22].

An acclaimed fact is that electrical properties, such as electrical conductivity or permittivity, change in the tissues of a body when they transform from healthy to malignant tissue [9,14,18]. For example, studies revealed that the
cancerous breast tissues show a different conductivity scope when compared to non-malignant tissues, an effect explained as a result of the increased density of the tumor stroma [20]. Differences in the electrical conductivity scope between different types of tissues are the contrast source in the EIT images. This physical phenomenon supports EIT techniques to produce images of biological tissues by interpreting their response to applied voltages or currents. The inverse EIT problem deals with reconstructing the electrical conductivity or permittivity by measuring voltages or currents at electrodes placed on the test volume surface. This so-called Calderon type inverse problem [13] is highly ill-posed; refer to the topical review paper [8]. Since the 1980s, various computational techniques have been suggested to solve this inverse problem computationally. We also refer to recent papers [5, 7, 27] to review the current state of the art and the existing open problems associated with EIT and its applications.

In this work, we propose and validate an efficient computational approach for the optimal reconstructing of biomedical images based on measurements obtained with noise. This approach is advantageous in various applications for medical practices dealing with different models characterized by near binary distributions such as electrical conductivity. The prototype computational framework suggested in [21] was built around gradient-based multiscale optimization techniques supplied with multilevel control space reduction over fine and coarse scales used interchangeably. The proper intercommunication between the different scales authenticates computational efficiency and the quality of the obtained result. To enhance the performance of this approach, we developed an algorithm for multiscale optimization using various techniques for control space reduction and assuming the identification of multiple regions as cancerous spots. The impact of the noise present in measurements was also systematically analyzed.
As proven in practical applications, fine-scale optimization performed on fine meshes provides high-resolution images. Fine meshes also contribute enormously towards increased sensitivity by enforcing accuracy in computed gradients of objectives if in use. The size of the control space defined over fine scales will significantly reduce after applying parameterization, for instance, by using linear transformations based on available sample solutions (realizations) when applying principal component analysis (PCA). However, fine-scale optimization may still suffer from over-parameterization if the problem is under-determined, i.e., the number of controls overweighs the size of available data (measurements). On the other hand, optimization performed on coarse meshes could arrive at a solution much faster due to the size of the control space. Usually, the solutions obtained at coarse meshes are of low quality and less accurate due to the lack of sensitivity naturally “coarsened”. In addition to that, coarse-scale optimization may suffer from being over-determined if the sizes of the control space and data in use are imbalanced.

The proposed multiscale optimization framework utilizes all advantages mentioned above while using fine and coarse scales. Moreover, using them both in one process helps mitigate their side effects. In addition, the penalization (based on the Tikhonov-type regularization) approach is added to the framework to help “synchronize” optimization at both scales and deal with noise. For example, fine-scale solution images may not provide clear boundaries between regions identified by different physical properties in space. As a result, a smooth transition cannot provide an accurate recognition of shapes (e.g., of cancer-affected regions) while solving the inverse problem of cancer detection (IPCD). In our computations, fine-scale optimization approximates the locations characterized by high and low electrical conductivity. Projecting solutions onto the
coarse scales provides dynamical (sharp-edge) filtering to the fine-scale images optimized further for better matching the available data. The filtered images projected back onto the fine scales preserve some information on recent changes obtained at the coarse scales. In our work, we extended the computational efficiency of the procedure for automated scale shifting to accumulate optimally progress obtained at both scales. Here, we keep the main focus on applying this computational approach to IPCD by the Electrical Impedance Tomography technique. However, there are no known restraints for employing the same methodology to a broad range of problems in biomedical sciences, physics, geology, chemistry, and other fields.

The structure of the thesis proceeds as follows. In Chapter 2, we present the mathematical description of the inverse EIT problem formulated in terms of PDE-constrained optimization. Here, we also describe the multiscale PCA-based control space reduction, switching between fine and coarse scales, and penalization for the improved performance. In Chapter 3, we present and analyze the structure of our partitioning algorithm by spatial grouping applied to different elements (cells) within the computational domain. Extensive computational results, including computational models in 2D, framework validation, and validation with biomedical models, are presented in Chapter 4. Lastly, concluding remarks and the discussion for future expansions are provided in Chapter 5.
Chapter 2

Mathematical Models and Computational Algorithms

2.1 Inverse EIT Problem: Mathematical Model

In the recent papers [4,6,21], the inverse EIT problem is formulated as a PDE-constrained optimization problem with extensive numerical analysis for 2D models by implementing various methods for solution space re-parameterization including the PCA coupled with dynamical control space upscaling. In our current discussion of the inverse EIT model we use the same notations as established in [21].

Let \( \Omega \subset \mathbb{R}^n \), \( n = 2, 3 \), be an open and bounded set (domain) representing the medium of a particular interest. We assume that function \( \sigma(x) : \Omega \to \mathbb{R}_+ \) represents (isotropic) electrical conductivity at point \( x \in \Omega \). We also assume that \( m \) electrodes \((E_l)_{l=1}^m\) with contact impedances \((Z_l)_{l=1}^m \in \mathbb{R}_+^m\) are attached to the periphery of the domain \( \partial \Omega \). We accept using the so-called “voltage-to–
current” model where constant voltages (electrical potentials) $U = (U_\ell)_{\ell=1}^m \in \mathbb{R}^m$ are applied to the electrodes $E_\ell$ to initiate electrical currents $(I_\ell)_{\ell=1}^m \in \mathbb{R}^m$ through the same electrodes. It is assumed that both electrical currents and voltages satisfy the conservation of charge and ground (zero potential) conditions, respectively

$$\sum_{\ell=1}^m I_\ell = 0, \quad \sum_{\ell=1}^m U_\ell = 0. \quad (2.1)$$

We formulate the inverse EIT (conductivity) problem [13] as a PDE-constrained optimization problem by considering minimization of the following objective/objective function

$$J(\sigma) = \sum_{\ell=1}^m (I_\ell - I^*_\ell)^2, \quad (2.2)$$

where $(I^*_\ell)_{\ell=1}^m \in \mathbb{R}^m$ are measurements made for electrical currents $I_\ell$ computed as

$$I_\ell = \int_{E_\ell} \sigma(x) \frac{\partial u(x)}{\partial n} \, ds, \quad \ell = 1, \ldots, m \quad (2.3)$$

based on conductivity field $\sigma(x)$ set here as a control/control variable. A distribution of electrical potential $u(x) : \Omega \to \mathbb{R}$ is obtained as a solution of the following PDE (elliptic) problem

$$\nabla \cdot [\sigma(x) \nabla u(x)] = 0, \quad x \in \Omega \quad (2.4a)$$

$$\frac{\partial u(x)}{\partial n} = 0, \quad x \in \partial \Omega - \bigcup_{\ell=1}^m E_\ell, \quad \ell = 1, \ldots, m \quad (2.4b)$$

$$u(x) + Z_\ell \sigma(x) \frac{\partial u(x)}{\partial n} = U_\ell, \quad x \in E_\ell, \quad \ell = 1, \ldots, m \quad (2.4c)$$

where $n$ is an external unit normal vector on $\partial \Omega$. A well-known fact is that the inverse EIT problem to identify electrical conductivity $\sigma(x)$ over the dis-
cretized domain Ω with available input data \((I^*_\ell)_{\ell=1}^m\) of size \(m\) is highly ill-posed. Therefore, the formulation of our optimization problem has to be adapted to the situation when the size of input data is increased through additional measurements while keeping the size of the unknown parameters, i.e. elements in the discretized description for \(\sigma(x)\), fixed. As detailed in [4], we use a “rotation scheme” by setting \(U^1 = U, I^1 = I\) and considering \(m - 1\) new permutations of boundary voltages

\[
U^j = (U_j, \ldots, U_m, U_1, \ldots, U_{j-1}), \quad j = 2, \ldots, m \tag{2.5}
\]

applied respectively to electrodes \(E_1, E_2, \ldots, E_m\). Using the “voltage–to–current” model allows us to measure associated currents \(I^{j*} = (I^{j*}_1, \ldots, I^{j*}_m)\) and further increase the total number of available measurements from \(m^2\) up to \(Km^2\) by applying (2.5) to \(K\) different permutations of potentials within set \(U\). Having a new set of \(Km\) input data \((I^{j*})_{j=1}^{Km}\) and the Robin condition (2.4c) used together with (2.3), we finally consider an updated optimization problem of minimizing a new objective function

\[
\mathcal{J}(\sigma) = \sum_{j=1}^{Km} \sum_{\ell=1}^m \beta^j_\ell \left[ \int_{E_\ell} \frac{U^j_\ell - u^j(x; \sigma)}{Z^j_\ell} ds - I^{j*}_\ell \right]^2, \tag{2.6}
\]

where each function \(u^j(\cdot; \sigma), j = 1, \ldots, Km\), solves problem (2.4a)–(2.4c).

Added weights \(\beta^j_\ell \geq 0\) in (2.6) in general allow setting the level of importance for measurements \(I^{j*}_\ell\) (when \(\beta^j_\ell > 0\)) or excluding those measurements (\(\beta^j_\ell = 0\)) from all computations related to objectives and associated gradients. We also note that the forward EIT problem (2.4a)–(2.4c) together with (2.3) may be used to generate various model examples and obtain synthetic data for inverse
EIT problems to adequately mimic cancer related diagnoses seen in reality.

Finally, as proposed in [4], the solution of the optimization problem

\[ \hat{\sigma}(x) = \arg \min_{\sigma} J(\sigma) \]  \hspace{1cm} (2.7)

to minimize objective function (2.6) subject to PDE constraint (2.4) could be obtained by an iterative algorithm utilizing adjoint gradients with respect to control \( \sigma \)

\[ \nabla_\sigma J = -\sum_{j=1}^{Km} \nabla \psi^j(x) \cdot \nabla u^j(x) \]  \hspace{1cm} (2.8)

computed based on solutions \( \psi^j(\cdot; \sigma) : \Omega \to \mathbb{R}, j = 1, \ldots, Km \), of the adjoint PDE problem

\[ \nabla \cdot [\sigma(x) \nabla \psi(x)] = 0, \hspace{1cm} x \in \Omega \]
\[ \frac{\partial \psi(x)}{\partial n} = 0, \hspace{1cm} x \in \partial \Omega - \bigcup_{\ell=1}^{m} E_\ell \]
\[ \psi(x) + Z_\ell \frac{\partial \psi(x)}{\partial n} = 2\beta_\ell \left[ \int_{E_\ell} \frac{u(x) - U_\ell}{Z_\ell} \right] ds + I_\ell^*, \hspace{1cm} x \in E_\ell, \ell = 1, \ldots, m \]  \hspace{1cm} (2.9)

2.2 Multiscale Control Space Reduction

2.2.1 PCA-based Fine Scale Optimization

Optimization problem (2.6)–(2.7), as discussed in Section 2.1 and when solved for spatially discretized state variable \( u(x) \) and control \( \sigma(x) \), is usually over-parameterized. To resolve the problem of ill-posedness associated with this issue we apply commonly used re-parameterization of the control space based on principal component analysis to represent \( \sigma(x) \) in terms of uncorrelated
variables (components of vector $\xi$) mapping $\sigma(x)$ and $\xi$ by

$$\sigma = \Phi \xi + \bar{\sigma}, \quad (2.10a)$$

$$\xi = \hat{\Phi}^{-1}(\sigma - \bar{\sigma}), \quad (2.10b)$$

where $\Phi$ is the linear transformation matrix constructed by using $N_r$ sample solutions (realizations) $(\sigma^*_n)_{n=1}^{N_r}$ as its columns with $\hat{\Phi}^{-1}$ denoting the pseudo-inverse of $\Phi$. The prior mean $\bar{\sigma}$ is given by $\bar{\sigma} = (1/N_r) \sum_{n=1}^{N_r} \sigma^*_n$, see [11,19] for details on constructing a complete PCA representation. Optimization problem (2.7) is now restated in terms of new model parameters $\xi \in \mathbb{R}^{N_\xi}$ used in place of control $\sigma(x)$ as follows

$$\hat{\xi} = \arg\min_{\xi} J(\xi) \quad (2.11)$$

subject to discretized PDE model (2.4) with control mapping (2.10) used for computing $J(\xi) = J(\sigma(\xi))$. For solving problem (2.11), gradients $\nabla_\xi J$ of objective $J(\sigma)$ with respect to new control $\xi$ can be expressed as

$$\nabla_\xi J = \Phi^T \nabla_\sigma J \quad (2.12)$$

to project gradients $\nabla_\sigma J$ obtained by (2.8) from initial (physical) $\sigma$-space onto the reduced-dimensional $\xi$-space.

### 2.2.2 Coarse Scale Optimization via Partitioning

Here, we briefly review the methodology of the control space upscaling via partitioning introduced in [21] and significantly upgraded as proposed in the current work. Generally speaking, the upscaling algorithm applies re-parameterization and defines a new control space with a reasonably small number of parame-
ners (controls) \((\zeta_j)_{j=1}^{N_\zeta} \in \mathbb{R}_+^{N_\zeta}\). In Chapter 3, we describe the general concept and practical algorithms by which spatial controls are grouped (partitioned) in our proposed approach. The coarse scale phase of the optimization framework includes the following steps.

1. Discretize control \(\sigma(x)\) in problem (2.7) over the fine mesh with \(N\) elements each of area (or volume in 3D) \(\Delta_i\).

2. Represent \(\sigma(x)\) by a finite set of controls \((\sigma_i)_{i=1}^{N} \in \mathbb{R}^{N}_+\).

3. Partition this set into \(N_\zeta\) subsets \(C_j, j = 1, \ldots, N_\zeta\), by selecting (with no repetition) \(N_j\) controls for \(j\)-th subset and defining a map (fine–to-coarse partition)

\[
\mathcal{M} : (\sigma_i)_{i=1}^{N} \rightarrow \bigcup_{j=1}^{N_\zeta} C_j, \quad C_j = \{\sigma_i : P_{i,j} = 1, i = 1, \ldots, N\},
\]

\[
\sum_{j=1}^{N_\zeta} |C_j| = \sum_{j=1}^{N_\zeta} N_j = N,
\]

supplied with the partition (indicator) function

\[
P_{i,j} = \begin{cases} 
1, & \sigma_i \in C_j, \\
0, & \sigma_i \notin C_j.
\end{cases}
\]

4. Compute new (upscaled) gradients \(\nabla_\zeta J\) by summing up those components \(\frac{\partial J}{\partial \sigma_i}\) of discretized gradients \(\nabla_\sigma J\) that are related to controls \(\sigma_i \in C_j\), i.e.,

\[
\frac{\partial J}{\partial \zeta_j} = \sum_{i=1}^{N} P_{i,j} \frac{\partial J}{\partial \sigma_i} \Delta_i.
\]

Here, we note that a practical application of this method to solving the in-
verse EIT problem presented in [21] assumes $N_\zeta = 2$. This assumption uses a simplistic approach for partitioning (subsets $C_1$ and $C_2$) based on the current values of control $\sigma(x)$ to distinguish regions with low ($C_1$) and high ($C_2$) conductivity throughout the entire domain $\Omega$. It may cause poor computational performance in cases featuring several cancer-affected regions of different sizes. As such, a new partitioning methodology and updated scheme for switching between fine and coarse scales are developed to allow higher variations in the geometry of reconstructed binary images discussed in the next section.

### 2.3 Switching Between Scales

In the same fashion, as introduced in [21], the proposed computational approach obtains the optimal solution $\hat{\sigma}(x)$ of a required binary type by employing multiscale optimization at both fine and coarse scales each with its own sets of controls used interchangeably. We still consider a simplified scheme for switching between scales: they are changed after completing $n_s$ optimization iterations. The coarse-scale indicator function helps to account for these switches:

$$\chi_c(k) = \begin{cases} 
0, & (2k_s - 2)n_s < k \leq (2k_s - 1)n_s, \text{ (fine scale)} \\
1, & (2k_s - 1)n_s < k \leq 2k_sn_s. \text{ (coarse scale)} 
\end{cases} \quad (2.16)$$

In (2.16), $k_s = 1, 2, \ldots$ and $k = 0, 1, 2, \ldots$ denote respectively the counts for switching cycles and optimization iterations. We also use this function to define the termination condition

$$\left| \frac{\mathcal{J}(\sigma^k) - \mathcal{J}(\sigma^{k-1})}{\mathcal{J}(\sigma^k)} \right| < (1 - \chi_c)\epsilon_f + \chi_c\epsilon_c, \quad k \neq k_sn_s + 1 \quad (2.17)$$
subject to chosen tolerances $\epsilon_f, \epsilon_c \in \mathbb{R}_+$. The general concept of switching between scales is shown schematically in Figure 2.1, including also multiple controls linked to solutions associated with different cancer-affected (with higher electrical conductivity) regions.

**Figure 2.1:** Schematic illustrating the general concept of the multiscale optimization framework, adopted from [21]. In (a-c), $\sigma_l$ and $\sigma_h$ values represent two modes associated respectively with healthy and cancer-affected regions within domain $\Omega$. (a) A typical histogram representing binary distribution of true electrical conductivity $\sigma(x)$ used in EIT. (b) An example of the Gaussian-type histogram typical for solution $\sigma^k(x)$ obtained after the $k$ iterations at a fine scale. (c) A binary histogram for solution $\sigma^k(x)$ obtained after $k$th iteration at a coarse scale. Positions of blue and red bars are associated with current values of $\sigma^k_{\text{low}}$ and $\sigma^k_{\text{high},n}$ ($1 \leq n \leq N_{\text{max}}$) controls, while their heights are computed based on the fine scale representation $\sigma(\xi^k)$ cut off by the current values of the coarse scale separation threshold controls $\sigma^k_{\text{th},n}$. See Section 2.3 for more details. Coarse–to–fine and fine–to–coarse projections are defined respectively by (2.18)–(2.20) and (2.23)–(2.27).

While performing the fine scale ($\chi_c(k) = 0$) optimization phase, control $\sigma^k = \sigma^k(x)$ obtained after the $k$th iteration as $\sigma(\xi^k)$ is updated by solving optimization problem (2.11) in the reduced-dimensional $\xi$-space and by using map (2.10a) as described in Section 2.2.1, i.e. $\sigma^k = \sigma(\xi^k)$. During the coarse scale ($\chi_c(k) = 1$) optimization phase, $\sigma(\xi^k)$ updated last time at the end of the
fine scale phase is used in partitioning discussed in Section 2.2.2. It is obvious that updates made for fine scale controls $\xi^k$ should ensure the receipt of as much as possible information related to recent changes in $\sigma^k$ during the coarse scale phase, not worsening the results $\sigma(\xi^k)$ previously obtained at the fine scale. [21] proposes to project solution $\sigma^k$ obtained at the end of the coarse scale phase onto $\xi$-space by using a convex combination of $\sigma(\xi^k)$ and $\sigma^k$, i.e.,

$$\bar{\sigma}(\xi^k) = \alpha_{c\rightarrow f} \sigma(\xi^k) + (1 - \alpha_{c\rightarrow f}) \sigma^k, \quad \alpha_{c\rightarrow f} \in [0, 1] \quad (2.18)$$

followed by re-initializing control $\xi^k$ from $\bar{\sigma}(\xi^k)$ using map (2.10b). As $\bar{\sigma}(\xi^k)$ and $\sigma^k$ have different (Gaussian and binary) distributions, the coarse–to–fine projection (2.18) also includes an extra step for projecting $\sigma^k$ to its PCA equivalent

$$\sigma^{k}_{PCA} = \Phi \hat{\Phi}^{-1}(\sigma^k - \bar{\sigma}) + \bar{\sigma}, \quad (2.19)$$

before using it in (2.18), see [11,26] for details. An optimal value of relaxation parameter $\alpha_{c\rightarrow f}$ is obtained by solving an additional 1D optimization problem

$$\alpha_{c\rightarrow f} = \hat{\alpha} = \arg\min \alpha \quad (2.20)$$

$$0 \leq \alpha \leq 1$$

$$J(\bar{\sigma}(\xi^k)) \leq J(\sigma(\xi^k))$$

appeared to be highly nonlinear due to the inequality constraint to control the quality of fine scale solutions $\sigma(\xi^k)$ in transition between subsequent switching cycles.

The proposed procedure for running optimization at the coarse scale has been substantially modified by assigning different controls to be responsible for reconstructing $\sigma(x)$ at locations associated with different cancer-affected
regions. The image quality (both shape and associated values of $\sigma(x)$ inside) for each region depends on its size and the distance to the measuring electrodes. As such, the (cumulative) sensitivity of the objective $\mathcal{J}(\sigma)$ (with respect to changes in the part of control $\sigma(x)$ that is related to the location of this region) varies between the regions, sometimes by several orders of the magnitude.

First, we formally specify the maximum number of expected cancer-affected (high conductivity) regions $N_{\text{max}}$. By considering the healthy part (low conductivity region) of domain $\Omega$ as a single region, partitioning (2.13)-(2.14) will attempt to create $N_\zeta = N_{\text{max}} + 1$ subsets $C_j$, $j = 1, \ldots, N_{\text{max}} + 1$. At a coarse scale, we define a new control vector $\zeta = (\zeta_j)_{j=1}^{2N_{\text{max}}+1}$ in which the first entry is the low value of (binary) electrical conductivity $\sigma(x)$ associated with a healthy region in domain $\Omega$. The next $N_{\text{max}}$ controls are the high values of $\sigma(x)$ related to the cancer-affected regions, i.e.,

$$
\zeta_1 = \sigma_{\text{low}}, \quad \zeta_2 = \sigma_{\text{high},1}, \quad \zeta_3 = \sigma_{\text{high},2}, \quad \ldots, \quad \zeta_{N_{\text{max}}+1} = \sigma_{\text{high},N_{\text{max}}}. \quad (2.21)
$$

These controls are shown schematically as respectively blue and red bars in Figure 2.1(c). The rest $N_{\text{max}}$ components

$$
\zeta_{N_{\text{max}}+1} = \sigma_{\text{th},1}, \quad \zeta_{N_{\text{max}}+2} = \sigma_{\text{th},2}, \quad \ldots, \quad \zeta_{2N_{\text{max}}+1} = \sigma_{\text{th},N_{\text{max}}}. \quad (2.22)
$$

take responsibility for the shape of those $N_{\text{max}}$ cancerous regions. They are set as separation thresholds to define a boundary between the low and high conductivity regions, as shown in green in Figure 2.1(c). Such a structure of control $\zeta$ allows us to create a systematic representation of the coarse-scale solution $\zeta^k$ for control $\sigma^k$ at the $k$th iteration based on the current fine-scale
representation $\sigma(\xi^k) = (\sigma_i(\xi^k))_{i=1}^N$, i.e.,

$$
\sigma_i^k = \begin{cases} 
\sigma_{low}^k, & \sigma_i(\xi^k) < \sigma_{th,n}^k, \\
\sigma_{high,n}^k, & \sigma_i(\xi^k) \geq \sigma_{th,n}^k,
\end{cases} \quad i = 1, \ldots, N, \quad 1 \leq n \leq N_{\text{max}}.
$$

(2.23)

Here, $n = n(i)$ denotes the number of a particular cancer-affected region defined subject to the partitioning map $M^k$ currently established and used for the $k$th iteration. We also note that

$$
0 < \sigma_{low}^k < \min_{1 \leq n \leq N_{\text{max}}} \sigma_{high,n}^k,
$$

$$
\min_{1 \leq i \leq N} \sigma_i(\xi^k) < \sigma_{th,n}^k < \max_{1 \leq i \leq N} \sigma_i(\xi^k), \quad n = 1, \ldots, N_{\text{max}}.
$$

(2.24)

Simply, (2.23) provides a rule for creating fine–to–coarse partition $M^k$ in (2.13) when $N_\zeta = N_{\text{max}} + 1$ based on the current state of control $\zeta^k$ (at the $k$th iteration). During the coarse scale ($\chi_c(k) = 1$) optimization phase, control $\sigma^k$ is updated by solving a $(2N_{\text{max}} + 1)$-dimensional optimization problem in the $\zeta$-space

$$
\hat{\zeta} = \arg\min_{\zeta} J(\zeta)
$$

(2.25)

subject to constraints (bounds) provided in (2.24), and then $\sigma^k = \sigma(\zeta^k)$. When solving problem (2.25) during the first switching cycle, $k = n_s$, $\zeta^k$ could be initially approximated by some constants, for example

$$
\sigma_{th,n}^k = \sigma_{ini} = \frac{1}{2} \left[ \max_{1 \leq i \leq N} \sigma_i(\xi^k) + \min_{1 \leq i \leq N} \sigma_i(\xi^k) \right],
$$

$$
\sigma_{low}^k = \text{mean}_{1 \leq i \leq N} \left\{ \sigma_i(\xi^k) : \sigma_i(\xi^k) < \sigma_{ini} \right\},
$$

$$
\sigma_{high,n}^k = \text{mean}_{1 \leq i \leq N} \left\{ \sigma_i(\xi^k) : P_{i,n+1} = 1, \sigma_i(\xi^k) \geq \sigma_{ini} \right\}, \quad n = 1, \ldots, N_{\text{max}}.
$$

(2.26)

This initialization procedure for coarse-scale controls $\zeta^k$ is due to the practical
approach used for creating and updating maps $\mathcal{M}^k$ (2.13); refer to Chapter 3 for more details. Switching from fine scale to coarse one when $k = (2k_s - 1)n_s, \ k_s > 1$, could be performed by utilizing the corresponding values obtained at the end of the previous coarse scale phase, i.e.,

$$
\zeta^k = \zeta^{k-2n_s}.
$$

While solving (2.25) presumably by approaches that require computing gradients, their first $N_{\max} + 1$ components

$$
\frac{\partial J(\zeta)}{\partial \zeta_1} = \frac{\partial J}{\partial \sigma_{\text{low}}}, \quad \frac{\partial J(\zeta)}{\partial \zeta_{n+1}} = \frac{\partial J}{\partial \sigma_{\text{high},n}}, \quad n = 1, \ldots N_{\max}
$$

could be easily obtained by using gradient summation formula (2.15) after completing partitioning map $\mathcal{M}^k$ (2.13)–(2.14) with employed (2.23). On the other hand, the rest components may be approximated by a finite difference scheme, e.g., of the first order,

$$
\frac{\partial J(\zeta)}{\partial \zeta_j} = \frac{\partial J}{\partial \sigma_{\text{th},n}}
\quad = \frac{1}{\delta_\zeta} \left[ J\left(\sigma^k(\ldots, \zeta_j + \delta\zeta, \ldots)\right) - J\left(\sigma^k(\ldots, \zeta_j, \ldots)\right) \right] + \mathcal{O}(\delta_\zeta), \quad (2.28)
\quad n = 1, \ldots, N_{\max}, \quad j = n + N_{\max} + 1.
$$

Parameter $\delta_\zeta$ in (2.28) is to be set experimentally, pursuing a trade-off between being reasonably small to ensure accuracy and large enough to protect the gradient components from being zero. In fact, formulas (2.23)–(2.27) provide a complete description of the fine–to–coarse projection for control $\sigma(x)$ used in our approach. A computational workflow to perform the described optimization over multiple scales is provided in Algorithm 6 of Appendix A.
Chapter 3

Partitioning via Spatial Element Grouping

3.1 Motivation and Criteria

In this chapter, we present and analyze the structure of our partitioning algorithm developed to construct maps \( M^k \) discussed in Chapter 2 by spatial grouping applied to different elements (cells) created after discretizing domain \( \Omega \). We assume that these \( N \) elements may be associated with some real-valued physical property (e.g., electrical conductivity \( \sigma(x) \)) that is changing throughout \( \Omega \). We also make several comments regarding the criteria used to perform this grouping and the partitioning procedure itself.

- **Primary grouping rule** is the physical property under consideration. The separation threshold value should be identified prior to executing the algorithm to differentiate between “low” and “high” values.

- Cells selected under the “low” values of the associated property are moved
into group #1 (default group).

- **Secondary grouping rule** applies to the cells with “high” values to re-group them according to their position in space.

- **Tertiary grouping rule** defines the maximum number of groups with “high” values.

We discuss the selected procedures to govern all three mentioned above rules and their practical implementation later in this chapter.

The partitioning algorithm has been implemented in MATLAB (see Appendix B containing getMap.m file) with a structure that allows an easy application (adjustment) to various problems featuring the same set of partitioning conditions and parameters. We will analyze its structure and make comments with references to the optimization problem discussed in Chapter 2.

### 3.2 Algorithm Description

#### 3.2.1 Input Data

The **input data** consists of the following objects:

- **thresh**: a positive real constant to specify a separation threshold for differentiating between low and high values of the provided physical property (in our case, electrical conductivity $\sigma$);

- **maxNumSpot**: a positive integer constant to set the maximum number of spatial regions (spots) characterized by the high values; if there are more spots identified compared to the `maxNumSpot` value then the smallest (according to their spatial size estimated by the number of included elements)
spots will be re-grouped (“glued” together) to satisfy this condition (tertiary grouping rule);

- **fileMap**: a string variable specifying the file (in this version, an MS Excel .xlsx-file) containing a matrix with the description of the neighbor cells (ith row contains the numbers associated with the cells that are neighbors of the ith cell);

- **fileModel**: a string variable specifying the file (in this version, also an MS Excel .xlsx-file) containing a vector whose components are the physical property discretized over domain Ω.

For practical use of the partitioning algorithm as a function within the entire optimization framework, **fileMap** and **fileModel** may serve as identifiers to data of any structures (formats) to provide respectively a map of the neighboring cells and the physical property under consideration.

### 3.2.2 Initialization

The initialization phase for the partitioning algorithm consists of creating the following five objects.

- **map**: the map of neighbors created from reading the Excel file identified by **fileMap**;

- **property**: an $N$-component vector of the discretized property meaning $(\sigma_i)_{i=1}^N$ in (2.13) created from reading the Excel file identified by **fileModel**. Here, we note that the components of **property** should be converted to doubles which is the default numeric data type (class) in MATLAB;
• countGroup = 1: the active group counter;

• V = [ ]: initializing an empty vector to keep cells with high values;

• G{1} = [ ]: initializing a vector of cells G{} to keep the grouping information; its first cell (component) will keep spatial elements with low values in a vector (default group).

3.2.3 Primary grouping rule

A procedure for Step #1 (refer to Algorithm 1) implements the primary grouping rule to initially assign all cells to two groups according to their “low” or “high” values (respectively to vectors G{1} and V). Practically, we assume that the healthy and cancerous tissues have respectively low and high electrical conductivity, i.e., property = 0.2 or property = 0.4, with the thresh parameter set to 0.3.

**Algorithm 1 Primary grouping rule – Step #1**

```plaintext
for i ← 1 to length(property) do
    if property(i) ≥ thresh then
        V ← i
    else
        G{1} ← i
    end if
end for
```

3.2.4 Secondary grouping rule

Our next Steps #2 and #3 complete the secondary grouping rule by respectively completing other groups (in the vector of cells G{}) with cells of high values
according to their neighbor information and updating these groups by merging ones containing the same members. Algorithm 2 illustrates the procedure for Step #2. Here, all cells with the high values recorded in $V$ are checked, and a Boolean variable $flagFound$ raises a flag if any neighbors are found in the existing groups. Alternatively, a new group (new vector) will be added to the vector of cells $G\{}$. 

**Algorithm 2** Secondary grouping rule – Step #2

```plaintext
def for i ← 1 to length(V) do
    flagFound ← FALSE
    currNeigh ← [ V(i) map(V(i), :) ]
    currNeigh ← INTERSECT(currNeigh, V)
    for k ← 2 to countGroup do
        if INTERSECT(currNeigh, G{k}) then
            G{k} ← UNION(G{k}, currNeigh)
            flagFound ← TRUE
            break
        end if
    end for
    if ~flagFound then
        countGroup ← countGroup + 1
        G{countGroup} ← currNeigh
    end if
end for
```

Algorithm 3, in its turn demonstrates the completion of the secondary grouping rule by checking the created groups to find any duplicated members. If
groups $G\{i\}$ and $G\{j\}$ ($i \neq j$) have common values in both vectors, then the found values are left in $G\{j\}$ and $G\{i\}$ becomes an empty set.

**Algorithm 3** Secondary grouping rule – Step #3

```
for $i \leftarrow 2$ to countGroup do
    for $j \leftarrow i + 1$ to countGroup do
        if INTERSECT($G\{i\}, G\{j\}$) then
            $G\{j\} \leftarrow UNION(G\{i\}, G\{j\})$
            $G\{i\} \leftarrow []$
        end if
    end for
end for
```

### 3.2.5 Tertiary grouping rule

The last steps (#4 and #5) implement the tertiary grouping rule to enforce the condition for the maximum number of regions with the high values allowed after the grouping process is complete. Algorithm 4 illustrates Step #4 for deleting empty groups left after Step #3 by creating a new vector of cells $F$ and assigning the default group $G\{1\}$ to a new default group $F\{1\}$. An additional vector $F_n$ is created to keep the number of cells in each group moved to $F\{}$ from $G\{}$. 

23
Algorithm 4 Tertiary grouping rule – Step #4

\begin{align*}
F\{1\} & \leftarrow G\{1\} \\
Fn(1) & \leftarrow \text{length}(F\{1\}) \\
countF & \leftarrow 1 \\
\text{for } i & \leftarrow 2 \text{ to countGroup do} \\
\quad \text{if } \neg \text{IsEmpty}(G\{i\}) \text{ then} \\
\quad \quad countF & \leftarrow countF + 1 \\
\quad \quad F\{countF\} & \leftarrow G\{i\} \\
\quad \quad Fn(countF) & \leftarrow \text{length}(F\{countF\}) \\
\quad \text{end if} \\
\text{end for}
\end{align*}

Applying the condition itself for the maximum number of spots defined by parameter $\text{maxNumSpot}$ within the new vector of cells $K\{}$ and sending all remaining cells to the last group as Step #5 are shown in Algorithm 5. Finally, at the end of Step #5, the vector of cells $K\{}$ is renamed to $G\{}$, and the latter serves as the output data for the entire algorithm containing the number of spatial elements (cells) subject to all rules established for this partitioning.
Algorithm 5  Tertiary grouping rule – Step #5

K\{1\} ← F\{1\}

for \(j \leftarrow 2\) to \(\text{MIN}(\text{maxNumSpot}, \text{countF})\) do
  \(\text{currMax} \leftarrow \text{FIND}(\text{Fn} = \text{MAX}(\text{Fn}(2:\text{end})))\)
  \(K\{j\} \leftarrow F\{\text{currMax}\}\)
  \(\text{currMax} \leftarrow [\]\)
  \(\text{Fn}(\text{currMax}) \leftarrow 0\)
end for

\(K\{\text{maxNumSpot}+1\} \leftarrow [\]\)

for \(j \leftarrow 2\) countF do
  if \(\text{Fn}(j)\) then
    \(K\{\text{maxNumSpot}+1\} \leftarrow [K\{\text{maxNumSpot}+1\} F\{j\}]\)
  end if
end for

end for

G\{} ← K\{}

3.3 Partitioning Applications

In this section, we provide the results of applying the developed partitioning algorithm discussed in Sections 3.1 and 3.2 (employing Algorithms 1–5) to illustrate the main concept of the cell (element) grouping techniques used to construct maps \(M^k\) mentioned in Chapter 2.

3.3.1 Applications to Rectangular Models

As a first example, we start with a simple rectangular model (model A) shown in Figure 3.1. We set variable fileMap to point out the Excel file containing
a $100 \times 8$ matrix $M$ with the description of the neighbor cells (maximum 8 possible neighbors for each cell). As mentioned previously in Section 3.2.1, the $i$th row of matrix $M$ contains the numbers associated with the cells that are neighbors of the $i$th cell. To identify neighbors, we use the principle of their common vertices (corner points). We also set variable `fileModel` to point out the Excel file containing a 100-component vector $\sigma$ with the description of the physical property (e.g., electrical conductivity $\sigma(x)$) discretized to assign numerical values to all 100 elements within the entire model. In pursuit of simplicity but without loss of generality, we use only two distinct values of $\sigma(x)$, namely 0.4 (red squares) and 0.2 (green squares), to identify respectively high and low values. Table 3.1 shows the combined structure (the first 15 rows) of the input data $[M|\sigma]$ (matrix $M$ augmented by the vector $\sigma$). The separation threshold `thresh` sets correspondingly a value of 0.3.

![Image of Model A](image.png)

**Figure 3.1:** Model A (with three high-value spots): a $10 \times 10$ rectangular model with 100 square cells (elements) labeled by shown numbers 1, 2, …, 100 and assigned values of physical property (electrical conductivity): (red) high value $\sigma(x) = 0.4$ and (green) low value $\sigma(x) = 0.2$. 

26
Table 3.1: Combined structure (the first 15 rows) of the input data $[M|\sigma]$ (matrix $M$ augmented by the vector $\sigma$) used for Model A.

<table>
<thead>
<tr>
<th>Map $M$:</th>
<th>$\sigma(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>11 12 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1</td>
<td>3 11 12 13 0 0 0 0</td>
</tr>
<tr>
<td>2</td>
<td>4 12 13 14 0 0 0 0</td>
</tr>
<tr>
<td>3</td>
<td>5 13 14 15 0 0 0 0</td>
</tr>
<tr>
<td>4</td>
<td>6 14 15 16 0 0 0 0</td>
</tr>
<tr>
<td>5</td>
<td>7 15 16 17 0 0 0 0</td>
</tr>
<tr>
<td>6</td>
<td>8 16 17 18 0 0 0 0</td>
</tr>
<tr>
<td>7</td>
<td>9 17 18 19 0 0 0 0</td>
</tr>
<tr>
<td>8</td>
<td>10 18 19 20 0 0 0 0</td>
</tr>
<tr>
<td>9</td>
<td>19 20 0 0 0 0 0 0</td>
</tr>
<tr>
<td>1</td>
<td>2 12 21 22 0 0 0 0</td>
</tr>
<tr>
<td>1</td>
<td>2 3 11 13 21 22 23</td>
</tr>
<tr>
<td>2</td>
<td>3 4 12 14 22 23 24</td>
</tr>
<tr>
<td>3</td>
<td>4 5 13 15 23 24 25</td>
</tr>
<tr>
<td>4</td>
<td>5 6 14 16 24 25 26</td>
</tr>
</tbody>
</table>

For checking the main functionality of the partitioning algorithm, we first run the MATLAB code `getMap.m` provided in Appendix B for model A with `maxNumSpot` parameter set to 1 and then to 3. The first run ($\text{maxNumSpot} = 1$) mimics the situation without recognition of any spatial connectivity being able to distinguish between low and high electrical conductivity $\sigma(x)$ by returning a map consisting of two groups only:

```
>> Default group: 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 15, 16, 17, 24, 25, 26, 28, 29, 31, 32, 34, 35, 36, 37, 41, 42, 43, 44, 45, 46, 47, 51, 52, 53, 54, 55, 56, 57, 58, 59, 61, 62, 64, 65, 66, 67, 68, 69, 70, 72, 75, 77, 78, 79, 80, 86, 88, 89, 90, 92, 96, 97, 98, 99, 100
```

```
>> Spot #1: 1, 11, 12, 13, 21, 22, 23, 33, 18, 19, 20, 27, 30, 38, 39, 40, 48, 49, 50, 60, 63, 71, 73, 74, 76, 82, 83, 84, 85, 87, 91, 93, 94, 95
```
The second run (\texttt{maxNumSpot} = 5) activates the full functionality of the partitioning algorithm by taking into account the recognition of spatial patterns subject to the provided “map of neighbors” and limiting this recognition to only the five largest spots containing cells connected by common corner points:

>> **Default group:** 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 15, 16, 17, 24, 25, 26, 28, 29, 31, 32, 34, 35, 36, 37, 41, 42, 43, 44, 45, 46, 47, 51, 52, 53, 54, 55, 56, 57, 58, 59, 61, 62, 64, 65, 66, 67, 68, 69, 70, 72, 75, 77, 78, 79, 80, 86, 88, 89, 90, 92, 96, 97, 98, 99, 100

>> **Spot #1:** 63, 71, 73, 74, 76, 82, 83, 84, 85, 87, 91, 93, 94, 95

>> **Spot #2:** 18, 19, 20, 27, 30, 38, 39, 40, 48, 49, 50, 60

>> **Spot #3:** 1, 11, 12, 13, 21, 22, 23, 33

>> **Spot #4:** N/A

>> **Spot #5:** N/A

The map above is consistent with the structure of model A and is seen as an expected outcome of applying Algorithms 1–5. Thus, we proceed with two more models to check the algorithm’s performance in applications to more complicated models.

As the next step, we repeat the last experiment with the separation threshold parameter (\texttt{maxNumSpot} = 5) in application to our model B, refer to Figure 3.2. We still use the same matrix \textit{M} as for model A (refer to Table 3.1) but a different description of \( \sigma(x) \) to carry out the structure of model B.

Despite the non-trivial structure of the model, the partitioning algorithm, as expected, returns the following map consistent with model B. Here, we note
Figure 3.2: Model B (with one high-value spot of non-trivial shape): a $10 \times 10$ rectangular model with 100 square cells (elements) labeled by shown numbers 1, 2, \ldots, 100 and assigned values of physical property (electrical conductivity): (red) high value $\sigma(x) = 0.4$ and (green) low value $\sigma(x) = 0.2$.

that the high conductivity region may not represent the shapes of the cancer-affected tissues naturally seen in medical practices. However, it may portray models found in other fields (e.g., geological applications: fractures filled with some fluids).

>>> Default group: 1, 3, 4, 5, 6, 9, 10, 11, 14, 15, 16, 18, 19, 20, 21, 22, 24, 25, 28, 29, 30, 31, 32, 37, 38, 39, 40, 41, 42, 43, 48, 49, 50, 51, 52, 55, 56, 59, 60, 61, 64, 65, 66, 67, 69, 70, 73, 74, 75, 76, 77, 80, 82, 83, 84, 85, 86, 87, 88, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100

>>> Spot #1: 2, 7, 8, 12, 13, 17, 23, 26, 27, 33, 34, 35, 36, 44, 45, 46, 47, 53, 54, 57, 58, 62, 63, 68, 71, 72, 78, 79, 81, 89, 90

>>> Spot #2: N/A

>>> Spot #3: N/A
The last numerical experiment in this section checks the ability of the partitioning algorithm to abide by the tertiary grouping rule (refer to Section 3.2.5 for details) for models containing many small (disconnected) spots. Our model C, shown in Figure 3.3, features six “cancerous” (high conductivity value) regions of various sizes. As previously, we use the same matrix $M$ as for models A and B (refer to Table 3.1) but a different description of $\sigma(x)$ to carry out the structure of model C.

![Figure 3.3: Model C (with six high-value spots of various sizes): a $10 \times 10$ rectangular model with 100 square cells (elements) labeled by shown numbers 1, 2, \ldots, 100 and assigned values of physical property (electrical conductivity): (red) high value $\sigma(x) = 0.4$ and (green) low value $\sigma(x) = 0.2$.](image)

First, we obtain the result computing the map with $\text{maxNumSpot} = 3$:

>> **Default group:** 6, 7, 8, 14, 15, 16, 17, 18, 19, 21, 22, 24, 25, 28, 29, 30,
   31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 42, 43, 44, 45, 46, 47, 51, 52, 53, 54,
55, 56, 61, 64, 65, 66, 67, 74, 75, 76, 77, 85, 86, 87, 88, 94, 95, 96, 97, 98, 99, 100

>> **Spot #1:** 48, 49, 50, 57, 58, 59, 60, 68, 69, 70, 78, 79, 80, 89, 90

>> **Spot #2:** 62, 63, 71, 72, 73, 81, 82, 83, 84, 91, 92, 93

>> **Spot #3:** 1, 2, 3, 4, 5, 11, 12, 13, 23, 9, 19, 20, 26, 27, 41

Then, we repeat this experiment with `maxNumSpot = 5`:

>> **Default group:** 6, 7, 8, 14, 15, 16, 17, 18, 19, 21, 22, 24, 25, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 42, 43, 44, 45, 46, 47, 51, 52, 53, 54, 55, 56, 61, 64, 65, 66, 67, 74, 75, 76, 77, 85, 86, 87, 88, 94, 95, 96, 97, 98, 99, 100

>> **Spot #1:** 48, 49, 50, 57, 58, 59, 60, 68, 69, 70, 78, 79, 80, 89, 90

>> **Spot #2:** 62, 63, 71, 72, 73, 81, 82, 83, 84, 91, 92, 93

>> **Spot #3:** 1, 2, 3, 4, 5, 11, 12, 13, 23,

>> **Spot #4:** 9, 10, 20

>> **Spot #5:** 26, 27, 41

We conclude that the tertiary grouping rule in Algorithms 4–5 is implemented correctly as the partitioning results seen for model C are in full agreement with discussions in Section 3.2.5. Finally, we conclude also on the correctness of the entire partitioning algorithm implementation as provided in Sections 3.1–3.2. Before applying this algorithm to the main EIT problem of Chapter 2, we have to check it in applications to non-rectangular models using various types of spatial discretization (e.g., the finite element method’s triangulation).
3.3.2 Application to Triangular Elements in FreeFem++

As discussed in detail in Chapter 4, all computations related to solving the main PDE problem (2.4) and the adjoint equation (2.9) are performed by using the computational facilities of the finite element-based software environment FreeFem++ [17]. Therefore, the partitioning algorithm described in this chapter and checked for correct work with rectangular models (with rectangle/square shapes of associated elements) must be validated for the same while using triangles as spatial elements in FEM's triangulation.

Figure 3.4: Model D (with two high-value spots of various sizes): a sample (circular-shaped) model with 16 triangular cells (elements) labeled by shown numbers (in blue) 1, 2, . . . , 16 and assigned values of physical property (electrical conductivity): (red) high value \( \sigma(x) = 0.4 \) and (white) low value \( \sigma(x) = 0.2 \). Numbers in black provides labels for vertices.

For the purpose of this validation, we created model D based on the example
Table 3.2: Combined structure of the input data $[M|\sigma]$ (matrix $M$ augmented by the vector $\sigma$) used for Model D.

<table>
<thead>
<tr>
<th>Map $M$:</th>
<th>$\sigma(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 5 7 9 11 13 14 15 16 0 0 0</td>
<td>0.4</td>
</tr>
<tr>
<td>4 5 6 7 8 9 0 0 0 0 0 0</td>
<td>0.2</td>
</tr>
<tr>
<td>1 4 5 7 9 10 11 12 13 14 15 16</td>
<td>0.2</td>
</tr>
<tr>
<td>2 3 5 6 7 8 10 12 14 0 0 0</td>
<td>0.2</td>
</tr>
<tr>
<td>1 2 3 4 6 7 8 9 11 0 0 0</td>
<td>0.2</td>
</tr>
<tr>
<td>2 4 5 7 8 12 0 0 0 0 0 0</td>
<td>0.4</td>
</tr>
<tr>
<td>1 2 3 4 5 6 8 9 10 11 12 14</td>
<td>0.2</td>
</tr>
<tr>
<td>2 4 5 6 7 0 0 0 0 0 0 0</td>
<td>0.4</td>
</tr>
<tr>
<td>1 2 3 5 7 11 0 0 0 0 0 0</td>
<td>0.4</td>
</tr>
<tr>
<td>3 4 7 12 13 14 0 0 0 0 0 0</td>
<td>0.2</td>
</tr>
<tr>
<td>1 3 5 7 9 16 0 0 0 0 0 0</td>
<td>0.4</td>
</tr>
<tr>
<td>3 4 6 7 10 14 0 0 0 0 0 0</td>
<td>0.2</td>
</tr>
<tr>
<td>1 3 10 14 15 16 0 0 0 0 0 0</td>
<td>0.2</td>
</tr>
<tr>
<td>1 3 4 7 10 12 13 15 16 0 0 0</td>
<td>0.2</td>
</tr>
<tr>
<td>1 3 13 14 16 0 0 0 0 0 0 0</td>
<td>0.4</td>
</tr>
<tr>
<td>1 3 11 13 14 15 0 0 0 0 0 0</td>
<td>0.2</td>
</tr>
</tbody>
</table>

provided in the FreeFem++ manual [1]. As shown in Figure 3.4, the spatial discretization of a circular-shaped domain is provided by 14 vertices (10 and 4 of them are respectively on the boundary and in the interior) and 15 triangular-shaped elements (cells). We also identify two regions with high values of the electrical conductivity provided by elements 6, 8 (region #1) and 1, 9, 11, and 15 (region #2). Table 3.2 shows the combined structure of the input data $[M|\sigma]$ (matrix $M$ augmented by the vector $\sigma$) where the neighbor map matrix $M$ is obtained by a simple computational procedure based on reading and analysis of the FreeFem++ mesh file mesh.msh.

We run again the MATLAB code getMap.m provided in Appendix B for our new model D with maxNumSpot parameter set to 3 which returns the following outcome.
This result confirms the correctness of the entire partitioning algorithm implementation as provided in Sections 3.1–3.2 which, in fact, is independent of the model dimensionality, the method of spatial discretization, and the structure (shape) of its spatial elements. As mentioned above, this algorithm is applied to the inverse EIT problem of Chapter 2 with a special treatment for the separation threshold $\text{thresh}$ discussed in Section 3.3.3.

### 3.3.3 Separation Thresholds as Controls

As discussed in Section 2.3, a computational algorithm for the partitioning applied within the multiscale optimization procedure requires more than one separation threshold to be treated as control (optimization) variables. In particular, by assuming a potential presence of $N_{\text{max}}$ cancerous regions ($N_{\text{max}} > 1$), we define the same number of separation thresholds $(\sigma_{\text{th},n})_{n=1}^{N_{\text{max}}}$ (2.22) subject to bounds (2.24) to establish proper boundaries between the low and high conductivity regions. It requires a few (minor) modifications to our partitioning algorithm to be practically applied at the $k$th iteration to solve the inverse EIT problem of Chapter 2.

1. Algorithms 1–5 are executed in their original form by setting the separation threshold $\text{thresh}$ to a numerical value of $\min_{1 \leq n \leq N_{\text{max}}} \sigma_{\text{th},n}^k.$
2. After initial map is established, some (non-empty) identified high-value groups (out of spots #1, #2, . . . , #N_{max}) may contain “extra” cells for which \( \sigma_i(\xi^k) < \sigma_{th,n}^k \). The search is run to identify all ”extra” cells.

3. All ”extra” cells are added to the default group with low-valued electrical conductivity.

All numerical results provided in Chapter 4 are obtained with these modifications enabled whenever \( N_{max} > 1 \).
Chapter 4

Computational Results

4.1 Computational Model in 2D

The computational part of our optimization framework integrates facilities for solving the EIT (forward PDE) problem (2.4), adjoint problem (2.9), and evaluation of the gradients according to (2.8), (2.12), and (2.28). These facilities are incorporated by using FreeFem++ [17], an open-source, high-level integrated development environment for obtaining numerical solutions for PDEs based on the finite element method (FEM). For solving numerically forward and adjoint PDE problems, spatial discretization is carried out by implementing FEM triangular finite elements: P2 piecewise quadratic (continuous) representation for electrical potential \( u(x) \) and P0 piecewise constant representation for conductivity field \( \sigma(x) \). Systems of algebraic equations obtained after such discretization are solved with UMFPACK, a solver for nonsymmetric sparse linear systems [15]. All computations are performed using 2D domain

\[ \Omega = \{ x \in \mathbb{R}^2 : x_1^2 + x_2^2 < r_\Omega^2 \} \]  

\[ (4.1) \]
which is a disc of radius $r_\Omega = 0.1$ with $m = 16$ equidistant electrodes $E_\ell$ with half-width $w = 0.12$ rad covering approximately 61% of boundary $\partial \Omega$ as shown in Figure 4.1(a). Electrical potentials $U_\ell$, see Figure 4.1(b), are applied to electrodes $E_\ell$ as seen in (2.5) following the “rotation scheme” discussed in Section 2.1. We also consider adding additional permutations within the set of potentials $U$, and, by choosing $K = 1$ or $K = 4$, we increase the total number of measurements from $m^2 = 256$ ($K = 1$) to $4m^2 = 1024$ ($K = 4$). The potentials are chosen to be consistent with the ground potential condition (2.1).

Determining the Robin part of the boundary conditions in (2.4c) we equally set the electrode contact impedance $Z_\ell = 0.1$. Figure 4.1(c) also shows an example of the distribution of flux $\sigma(x)\nabla u(x)$ of electrical potential $u$ in the interior of domain $\Omega$ during EIT.

Figure 4.1: (a) Equispaced geometry of electrodes $E_\ell$ placed over boundary $\partial \Omega$. (b) Electrical potentials $U_\ell$. (c) Electrical currents $I_\ell$ (positive in red, negative in blue) induced at electrodes $E_\ell$. Black arrows show the distribution of flux $\sigma(x)\nabla u(x)$ of electrical potential $u$ in the interior of domain $\Omega$.

Physical domain $\Omega$ is discretized using mesh totaling $N = 7726$ triangular FEM elements inside $\Omega$; refer to Figure 4.2. This mesh is then used to construct gradients $\nabla_\sigma J$, $\nabla_\xi J$, and $\nabla_\zeta J$ to perform the optimization procedure, as described in Algorithm 6, and compute maps $M^k$ using the partitioning
methodology discussed in Chapter 3. Unless stated otherwise, to solve problems (2.11) and (2.25) iteratively, our framework employs Sparse Nonlinear OPTimizer SNOPT, a software package for solving large-scale nonlinear optimization problems [16].

![Spatial discretization of domain Ω using 7726 triangular FEM elements.](image)

**Figure 4.2:** Spatial discretization of domain Ω using 7726 triangular FEM elements.

The actual (true) electrical conductivity $\sigma_{\text{true}}(x)$ we seek to reconstruct will be given analytically for each model by

$$
\sigma_{\text{true}}(x) = \begin{cases} 
\sigma_c, & x \in \Omega_c, \\
\sigma_h, & x \in \Omega_h,
\end{cases}
$$

and setting $\sigma_c = 0.4$ for cancer-affected regions of sub-domain $\Omega_c$ (various number of spots of different sizes and complexity of their geometry depending on the model) and $\sigma_h = 0.2$ to healthy tissue part $\Omega_h$. The initial guess for control $\sigma(x)$ uses a constant approximation to $\sigma_{\text{true}}$ given by $\sigma_0 = \frac{1}{2} (\sigma_h + \sigma_c) = 0.3$. Termination tolerances in (2.17) are set to $\epsilon_c = 0$ (to avoid early termination at
a coarse scale) and \( \epsilon_f = 10^{-10} \).

To enforce bounds established for coarse scale control \( \sigma_{th,n}^k \) in (2.24), in our all computations we used fine–to–coarse partition (2.23) redefined as

\[
\sigma_i^k = \begin{cases} 
\sigma_{low}, & \sigma_i(\xi^k) < (1 - \sigma_{th,n}^k) \min_i \sigma_i(\xi^k) + \sigma_{th,n}^k \max_i \sigma_i(\xi^k), \\
\sigma_{high,n}, & \text{otherwise},
\end{cases}
\]

\[
i = 1, \ldots N, \quad 1 \leq n \leq N_{\text{max}}
\]

while ensuring \( 0 < \sigma_{th,n}^k < 1 \).

For all computations mentioned in the rest of Chapter 4, we use a PCA-based map (2.10) between \( N \)-dimensional discretized control \( \sigma(x) \) and reduced-dimensional \( \xi \)-space as described in Section 2.2.1. A set of \( N_r = 1000 \) realizations \( (\sigma_i)_{i=1}^{1000} \) is created using a generator of uniformly distributed random numbers. Each realization \( \sigma_i \) “contains” from one to seven “cancer-affected” areas with \( \sigma_c = 0.4 \). Each area is located randomly within domain \( \Omega \) and represented by a circle of randomly chosen radius \( 0 < r \leq 0.3 r_\Omega \). We also apply the truncated singular value decomposition (TSVD) by choosing the number of principal components \( N_\xi \) by retaining 662 basis vectors in the PCA description. These values correspond to the preservation of respectively \( r_\xi = 99\% \) of the “energy” in the full set of basis vectors; see [11,21] for details.

### 4.2 Validation and Performance

We start with model #1 created to validate the applicability of our partitioning algorithm discussed in Chapter 3 and check the overall performance of the proposed computational framework to solve the inverse EIT problem. This model represents a typical situation for a cancer-affected biological tissue containing
several spots suspicious of tumor and, as such, having elevated electrical conductivity. Model #1, featuring three circular shaped cancerous spots of various sizes, is shown in Figure 4.3(a).

To evaluate the performance of the added partitioning algorithm, we first run optimization with the maximum number of expected cancer-affected regions $N_{\text{max}} = 1$; see the results in Figures 4.3(b,d,e,h). With the current computational functionality, this experiment mimics the basic situation when the partitioning algorithm has its secondary and tertiary grouping rules deactivated to work with all high conductivity spots as one (disconnected) region. Although the shapes of all three regions are captured (with some degree of accuracy), the values of $\sigma(x)$ in their interior are far from $\sigma_{\text{true}} = 0.4$. Mainly due to their sizes, these regions produce different sensitivity of the objective function with respect to any changes in conductivity within the regions. This difference keeps invisible in the case of being assigned to only one control, $\sigma_{\text{high}}$. By resetting $N_{\text{max}}$ to 5, we allow each region to utilize its own sensitivity information controlled by individual optimization control pairs, namely $(\sigma_{\text{high},n}, \sigma_{\text{th},n})_{n=1}^{N_{\text{max}}}$ (controls for “empty” groups are not updated during optimization). The new results, provided in Figures 4.3(c,f,i), show much better proximity of the numerical values $\sigma_{\text{high},n}$ to $\sigma_{\text{true}}$ and even better shape reconstruction for two bigger spots.

In the next numerical experiment, we try to improve the result in two different ways (still keeping $N_{\text{max}} = 5$). First, we increase the number of optimization iterations (for running both fine and coarse scales) $n_s$ from 5 to 10. Second, we allow the coarse-scale optimization using the same amount of available data as at the fine scale; refer respectively to Figures 4.4(a) and (b) for the outcomes. As shown by Figures 4.4(c,d), the same strategies are useless if applied to cases with $N_{\text{max}} = 1$. However, both enhancements with $N_{\text{max}} = 5$ bring side effects...
Figure 4.3: EIT Model #1: (a) true electrical conductivity $\sigma_{\text{true}}(x)$ and (b-i) optimization outcomes after applying multiscale framework by Algorithm 6 with (b,d,e,h) $N_{\text{max}} = 1$ and (c,f,i) $N_{\text{max}} = 5$ (both obtained with $K = 1$, $n_s = 5$, and limited data at the coarse scale). Plots in (b-d) show the obtained images at (b,c) coarse and (d) fine scales with added dashed circles to represent the location of cancer-affected regions taken from known $\sigma_{\text{true}}(x)$ in (a). Graphs in (e,f) present objective functions $J(\sigma^k)$ as functions of iteration count $k$ evaluated at fine (bottom edge) and coarse (top edge) scales. Changes in the coarse scale controls $\zeta^k = [\sigma_{\text{low}}^k \sigma_{\text{high,n}}^k \sigma_{\text{th,n}}^k]$ are shown in (h,i) with $\sigma_c = 0.4$ (red dashed line), $\sigma_{\text{low}}^k$ in blue, $\sigma_{\text{th,n}}^k$ in black, and $\sigma_{\text{high,n}}^k$ in various colors. (g) A histogram constructed for the coarse scale solution images (b,c).
in a form of over- or underestimating the electrical conductivity in some regions.

We suggest that these side effects may be caused by insufficient data provided by the measurements in the electrodes. This insufficiency may arise, e.g., by duplicated data obtained from different measurements. As such, we repeat optimization with total data size increased four times ($K = 4$) for four different cases: $N_{\text{max}} = 1$ vs. $N_{\text{max}} = 5$, with limited and full data used at the coarse scale, and $n_s = 5$ vs. $n_s = 10$. The results showing a significant improvement when $N_{\text{max}} = 5$ are provided in Figures 4.5(c,e,f). As noticed before, in the case

**Figure 4.4:** EIT Model #1: optimization outcomes after applying multiscale framework by Algorithm 6 with (a) $N_{\text{max}} = 5$, limited data at coarse scale, and $n_s = 10$, (b) $N_{\text{max}} = 5$, full data, and $n_s = 5$, (c) $N_{\text{max}} = 1$, limited data, and $n_s = 10$, and (d) $N_{\text{max}} = 1$, full data, and $n_s = 5$. All plots show the obtained images at the coarse scale with added dashed circles to represent the location of cancer-affected regions taken from known $\sigma_{\text{true}}(x)$ in Figure 4.3(a).
Figure 4.5: EIT Model #1: optimization outcomes obtained with increased $(K = 4)$ data when (a,b) $N_{\text{max}} = 1$, limited data at coarse scale, and $n_s = 5$, (c,d) $N_{\text{max}} = 5$, limited data, and $n_s = 5$, (e) $N_{\text{max}} = 5$, limited data, and $n_s = 10$, and (f) $N_{\text{max}} = 5$, full data at coarse scale, and $n_s = 5$. Plots in (a,c,e,f) show the obtained images at the coarse scale with added dashed circles to represent the location of cancer-affected regions taken from known $\sigma_{\text{true}}(x)$ in Figure 4.3(a). Graphs in (b,d) present objective functions $\mathcal{J}(\sigma^k)$ as functions of iteration count $k$ evaluated at fine (bottom edge) and coarse (top edge) scales.
with added data the optimization converges about three times faster; refer to Figures 4.5(b,d) for comparison.

Figure 4.6: EIT Model #1: optimization outcomes obtained with increased \( K = 4 \) data and added 0.5\% noise when (a) \( N_{\text{max}} = 1 \), limited data at coarse scale, and \( n_s = 5 \), (b) \( N_{\text{max}} = 5 \), limited data, and \( n_s = 5 \), (c) \( N_{\text{max}} = 5 \), limited data, and \( n_s = 10 \), and (d) \( N_{\text{max}} = 5 \), full data at coarse scale, and \( n_s = 5 \). All plots show the obtained images at the coarse scale with added dashed circles to represent the location of cancer-affected regions taken from known \( \sigma_{\text{true}}(x) \) in Figure 4.3(a).

Finally, we run the same four cases (see Figure 4.5) with 0.5\% normally distributed noise added to the measurement data. The results of this reconstruction is shown in Figure 4.6. As expected, numerical values \( \sigma(x) \) for some regions are affected. However, optimization provides the reconstruction of their shapes quite accurately.
4.3 Application to Complicated Models

Our model #2 is created to mimic a challenging case of applying the EIT techniques in medical practices for recognizing cancer at the early stages. The electrical conductivity $\sigma_{\text{true}}$ is shown in Figure 4.7(a). This model contains four (!) circular-shaped cancer-affected regions of very small sizes (the same size as the smallest region in model #1). The known complication comes from the fact that the order of difference in measurements generated by this model and “healthy tissue” ($\sigma(x) = \sigma_h, \forall x \in \Omega$) may be very close to the order of noise that appeared naturally in provided data. As for the previous model, we firstly compare the results obtained with $N_{\text{max}} = 1$ and $N_{\text{max}} = 5$; refer to Figures 4.7(b,d,e,h) and 4.7(c,f,i) respectively. The quality of both reconstructions is fairly comparable; however, the case supplied with the partitioning algorithm ($N_{\text{max}} = 5$) converges roughly two times faster.

Similar to what we did with model #1, we finally experiment with the increased data size allowed during the coarse scale and the number of iterations between scale switching $n_s$. The results obtained for cases with $N_{\text{max}} = 5$ vs. $N_{\text{max}} = 1$ (see Figures 4.8(b,c) and 4.8(e,f), respectively) show the significantly improved performance while using the partitioning algorithm and the extended set of controls assigned to particular spatial locations.

Here, we conclude that (subject to appropriate tuning procedures) the proposed computational framework supplied with the suggested partitioning algorithm shows a high potential for applications to cancer detection in cases where the increased sensitivity and accuracy are vital.
Figure 4.7: EIT Model #2: (a) true electrical conductivity $\sigma_{\text{true}}(x)$ and (b-i) optimization outcomes after applying multiscale framework by Algorithm 6 with (b,d,e,h) $N_{\text{max}} = 1$ and (c,f,i) $N_{\text{max}} = 5$ (both obtained with $K = 1$, $n_s = 5$, and limited data at the coarse scale). Plots in (b-d) show the obtained images at (b,c) coarse and (d) fine scales with added dashed circles to represent the location of cancer-affected regions taken from known $\sigma_{\text{true}}(x)$ in (a). Graphs in (e,f) present objective functions $J(\sigma^k)$ as functions of iteration count $k$ evaluated at fine (bottom edge) and coarse (top edge) scales. Changes in the coarse scale controls $\zeta^k = [\sigma^k_{\text{low}}, \sigma^k_{\text{high,n}}, \sigma^k_{\text{th,n}}]$ are shown in (h,i) with $\sigma_c = 0.4$ (red dashed line), $\sigma^k_{\text{low}}$ in blue, $\sigma^k_{\text{th,n}}$ in black, and $\sigma^k_{\text{high,n}}$ in various colors. (g) A histogram constructed for the coarse scale solution images (b,c).
Figure 4.8: EIT Model #2: (a) true electrical conductivity $\sigma_{true}(x)$ and (b-f) optimization outcomes after applying multiscale framework by Algorithm 6 with (b) $N_{max} = 5$, full data at coarse scale, and $n_s = 5$, (c) $N_{max} = 5$, limited data, and $n_s = 10$, (e) $N_{max} = 1$, full data, and $n_s = 5$, and (f) $N_{max} = 1$, limited data, and $n_s = 10$. Plots in (b,c,e,f) show the obtained images at the coarse scale with added dashed circles to represent the location of cancer-affected regions taken from known $\sigma_{true}(x)$ in (a). (d) A histogram constructed for the coarse scale solution images (b,c).
Chapter 5

Conclusion

5.1 Future expansion

The performance of the proposed computational framework to perform optimization at multiple scales could be further improved by investing in additional efforts to “accumulate” the progress made by optimization at both scales by “synchronizing” the optimization processes that run interchangeably. While performing optimization during the fine-scale phase for solving the problem (2.7) or (2.11), we consider penalizing the objective function $J(\sigma)$ with a Tikhonov-type penalty term, namely,

$$J_p = \beta \int_{\Omega} (\sigma(x) - \bar{\sigma}(x))^2 \, dx$$  \hspace{1cm} (5.1)$$

where $\beta \in \mathbb{R}^+$ is an adjustable parameter, and $\bar{\sigma}(x)$ represents the solution (also known as a reference solution) which our reconstruction $\sigma(x)$ should not differ too much from. By assuming that switching from the coarse scale to the fine one appears after completing the $k$th iteration we suggest the reference
solution be the best solution meaning the last one obtained at the coarse scale and projected to the fine scale by applying (2.19), i.e.,

$$\bar{\sigma}(x) = \sigma_{PCA}^k.$$  \hspace{1cm} (5.2)

With properly chosen parameter $\beta$, this penalization will “synchronize” the future fine scale solutions with those obtained previously at the coarse scale to prevent development of big gaps in their structures. Therefore, we will augment the (core) objective function $J$ given by (2.6) with a new term $J_f$ being active while performing optimization during the fine-scale phase only, i.e.,

$$J_f = (1 - \chi_c(k)) \beta_f \int_\Omega (\sigma(x) - \sigma_{PCA}^k)^2 \, dx.$$  \hspace{1cm} (5.3)

We may also assume that some prior knowledge exists for making predictions on the true values of controls $\sigma_{low}$ and $\sigma_{high}$ given respectively by two constant values $\bar{\sigma}_l < \bar{\sigma}_h$. Similar to (5.3), we define a penalization term activated during the coarse-scale phase optimization:

$$J_c = \chi_c(k) \beta_c \left[ (\zeta_1 - \bar{\sigma}_l)^2 + \sum_{n=1}^{N_{\text{max}}} (\zeta_{n+1} - \bar{\sigma}_h)^2 \right].$$  \hspace{1cm} (5.4)

The structure of the new objective function augmented by both penalization terms (5.3)–(5.4)

$$\bar{J} = J + J_f + J_c$$  \hspace{1cm} (5.5)
allows evaluation of corresponding gradients

\[ \chi_c(k) = 0 : \quad \nabla_\sigma \tilde{J} = \nabla_\sigma J + 2\beta_f(\sigma(x) - \sigma_{PCA}^k), \]

\[ \chi_c(k) = 1 : \quad \frac{\partial \tilde{J}(\zeta)}{\partial \zeta_1} = \frac{\partial J(\zeta)}{\partial \zeta_1} + 2\beta_c(\zeta_1 - \bar{\sigma}_1), \]

\[ \frac{\partial \tilde{J}(\zeta)}{\partial \zeta_{n+1}} = \frac{\partial J(\zeta)}{\partial \zeta_{n+1}} + 2\beta_c(\zeta_{n+1} - \bar{\sigma}_h), \quad n = 1, \ldots, N_{\text{max}} \]  

(5.6)

to support the same multiscale computational framework discussed in Chapter 2.

We also notice that Tikhonov-type penalization has been proven to have an additional effect of regularizing the reconstruction procedure against noise possibly contained in the measured data [10,12].

5.2 Final Comments

In this work, we propose a fully developed computational framework enabled to solve the inverse EIT problems for reconstructing binary images of different levels of complexity featuring multiple cancer-affected regions of different sizes based on available measurements usually affected by noise. The original methodology recently presented in [21] assumes a very simple approach for partitioning applied at the coarse scale causing limited computational performance in models presenting several cancer-affected regions of different sizes. Therefore, a new partitioning methodology and updated scheme for switching between the fine and coarse scales are developed to allow higher variations in the geometry of reconstructed binary images with superior performance confirmed computationally on various models of different complexity.
Bibliography


Appendix A

General Algorithm for Computational Framework
Algorithm 6 Multiscale optimization workflow (adopted from [21])

\[ k \leftarrow 0 \]
\[ \chi_c \leftarrow 0 \]
\[ \sigma^0 \leftarrow \text{initial guess } \sigma_0(x) \]
compute \( \xi^0 \) using \( \sigma^0 \) by (2.10b)

repeat
  compute \( u^k \) using \( \sigma^k \) by solving forward problem (2.4)
  compute \( \psi^k \) using \( u^k \) and \( \sigma^k \) by solving adjoint problem (2.9)
  compute \( \nabla_{\sigma} \mathcal{J}(\sigma^k) \) using \( u^k \) and \( \psi^k \) by (2.8)
  if \( \chi_c = 1 \) then
    compute \( \sigma(\xi^k) \) using \( \xi_k \) by (2.10a)
    compute \( \nabla_{\zeta} \mathcal{J}(\zeta^k) \) using \( \zeta^k, \sigma(\xi^k) \), and \( \nabla_{\sigma} \mathcal{J}(\sigma^k) \) by (2.15), (2.13)–(2.14), (2.23), and (2.28)
  else
    compute \( \nabla_{\xi} \mathcal{J}(\xi^k) \) using \( \nabla_{\sigma} \mathcal{J}(\sigma^k) \) by (2.12)
  end if
update \( \xi^{k+1} \) and \( \xi^{k+1} \) by using, depending on \( \chi_c(k) \), descent directions obtained respectively from \( \nabla_{\zeta} \mathcal{J}(\zeta^k) \) or \( \nabla_{\xi} \mathcal{J}(\xi^k) \)
if \( \chi_c = 1 \) then
  compute \( \sigma^{k+1} \) using \( \xi^{k+1} \) and \( \sigma(\xi^{k+1}) \) by (2.23)
else
  compute \( \sigma^{k+1} \) using \( \xi^{k+1} \) by (2.10a)
end if
\[ k \leftarrow k + 1 \]
update \( \chi_c \) using \( k \) by (2.16)
if \( \chi_c(k) \neq \chi_c(k - 1) \) then
  if \( \chi_c = 1 \) then
    update \( \sigma^k \) using \( \zeta^k \) and \( \sigma(\xi^k) \) by (2.23)
  else
    update \( \xi^k \) using \( \sigma^k \) and \( \sigma(\xi^k) \) by (2.18)–(2.20)
    update \( \sigma^k \) using \( \xi^k \) by (2.10a)
  end if
end if
until termination criterion (2.17) is satisfied to given tolerances \( \epsilon_f \) and \( \epsilon_c \)
Appendix B

MATLAB code for Chapter 3

% getMap.m
%
% MATLAB code for grouping cells (spatial elements)
% according to the provided physical property (low/high values)
% Maria Chun, FIT 2022
%

close all; clear; clc; tic;

% input data
thresh = 0.3; % separation threshold (low/high values)
maxNumSpot = 3; % max number of regions with high values
fileMap = 'neighandpoints10x10'; % neighbours by corner points
fileModel = 'modell';

% initialization
map = xlsread(fileMap,'A1:D100'); % reading map
[-, modelData] = xlsread(fileModel); % reading cell property
property = str2double(modelData); % assigning it to a vector
countGroup = 1; % active group counter
V = [];
G{1} = [];

% step #1: completing default group G{1} - low values
for i = 1:length(property)
    if (property(i) >= thresh)
        V = [V i];
    else
        G{1} = [G{1} i];
    end
end

% step #2: completing other groups G{2:countGroup} - high values
for i = 1:length(V)
    flagFound = false; % flag if neighbor found in the existing group
    currNeigh = [V(i) map(V(i),:)];
    currNeigh = intersect(currNeigh,V);
    for k = 2:countGroup
        if (intersect(currNeigh,G{k}))
            G{k} = union(G{k},currNeigh);
            flagFound = true;
            break;
        end
    end
    if (~flagFound)
        countGroup = countGroup + 1;
        G{countGroup} = currNeigh;
    end
% step #3: updating groups G{2:countGroup}
% by merging ones with common members
for i = 2:countGroup
    for j = i+1:countGroup
        if intersect(G{i},G{j})
            G{j} = union(G{i},G{j});
            G{i} = [];
        end
    end
end

% step #4: deleting empty groups & saving new groups in F
F{1} = G{1}; % default group F{1} - low values
Fn(1) = length(F{1}); % vector Fn to keep # of cells in each group
countF = 1;
for i = 2:countGroup
    if (~isempty(G{i}))
        countF = countF + 1;
        F{countF} = G{i};
        Fn(countF) = length(F{countF});
    end
end

% step #5a: applying condition for max # of spots
% and saving new groups in K
K{1} = F{1}; % default group K{1} - low values
for j = 2:min(maxNumSpot,countF)
    currMax = find(Fn==max(Fn(2:end)));
    currMax = currMax(1);
K{j} = F{currMax};
F{currMax} = [ ];
Fn(currMax) = 0;
end

% step #5b: finalizing step #5 by sending all remaining cells
% to the last group K{maxNumSpot+1}
K{maxNumSpot+1} = [ ];
for j = 2:countF
    if (Fn(j))
        K{maxNumSpot+1} = [K{maxNumSpot+1} F{j}];
    end
end

% step #5c: renaming groups back to G and visualizing results
clear G;
G = K;
disp(['Default group: ' num2str(G{1})]);
for i = 2:maxNumSpot+1
    disp(['Spot #' num2str(i-1) ': ' num2str(G{i})]);
end
B = reshape(property,10,10); B = B';
set(groot, 'DefaultFigureColormap', jet); colormap('default');
imagesc(B(end:-1:1,:)); hold on; axis xy; axis square; box off;
set(gca,'XColor','none','YColor','none'); caxis([0.0 0.5]);
for j = 1:10
    for i = 1:10
        text(i-0.2,11-j+0.1,num2str(10*(j-1)+i));
    end
end
disp(['Done! CPU elapsed time = ' num2str(toc) 's']);