

Determination of Cardiac Ejection Fraction by Electrical Impedance Tomography using a Hybrid Heuristic Approach, a Simulation Study

Marcos H. F. Ribeiro^{1,2}, Rodrigo Weber dos Santos², Luis Paulo S. Barra², and Franciane C. Peters²

¹ Departamento de Informática, Centro de Ciências Exatas, Campus da UFV, Viçosa, Brazil (marcosh.ribeiro@ufv.br)

² Programa de Pós-Graduação em Modelagem Computacional, Campus da UFJF, Juiz de Fora, Brazil
(rodrigo.weber@ufjf.edu.br, luis.barra@ufjf.edu.br, franciane.peters@engenharia.ufjf.br)

Abstract. *An important parameter to analyze the efficiency of the heart as a pump is Cardiac Ejection Fraction (EF), which is clinically highly correlated to the functional status of the heart. Diverse non invasive methods can be applied to measure EF, like Computer Tomography, Magnetic Resonance, Echocardiography, and others. Nevertheless, none of these techniques can be used to continuous monitoring of such parameter. On the other hand, electrical impedance tomography (EIT) may be applied to accomplish this goal. In addition, low cost and high portability are also EIT's features that justify the research for solutions involving such technique to monitor EF. EIT consists in reconstruct images of the conductivity distribution of the interior of a conductor domain by applying electric currents and measuring electrical potential on the boundary of the body. Mathematically, EIT can be classified as a non-linear inverse problem. This work proposes a method for the continuous estimation of cardiac ejection fraction, addressing it as an optimization problem. The models used in our approach assume that recent two-dimensional magnetic resonance images of the patient are available, and use them to reduce the search space. Another important feature is the parametrization of the geometry of internal inclusions inside the domain, which also reduces the cost of the method. This work proposes a Hybrid Iterated Local Search (ILS) heuristic for EIT inverse problem using Levenberg-Marquardt Method as local search. Experiments are performed on two-dimensional images with synthetically generated data for electric potentials. Two different protocols for current injection are tested in such experiments and preliminary results are presented.*

Keywords: *Cardiac Ejection Fraction, Electrical Impedance Tomography, Inverse Problem, Iterated Local Search.*

1. Introduction

Cardiac ejection fraction (EF) is an important parameter to analyze the efficiency of the heart as a pump. It indicates the amount of blood that is pumped from each ventricle

at each step of a heart cycle. In other words, EF is a measure of the blood fraction ejected from the ventricles in one heart cycle. Although it is possible to determine both left and right ventricles ejection fraction, clinically is more common to use only the ejection fraction of the left ventricle (EFLV), so the general term “ejection fraction” is often used to refer to the EFLV. By definition, the ejection fraction is calculated as follows:

$$EF = \frac{PV}{EDV} = \frac{EDV - ESV}{EDV}, \quad (1)$$

where PV denotes the volume of blood pumped, given by the difference between end-diastolic volume (EDV) and end-systolic volume (ESV). Diverse non-invasive techniques can be applied to determine EF, as echocardiography, cardiac magnetic resonance, and others. Although such techniques are able to produce high definition images for well-accurate diagnostics, they cannot be used for continuous monitoring, due specially to their high costs. To reach this goal, an alternative technique could be Electrical Impedance Tomography (EIT), which has advantages in terms of portability and in the fact of not using ionizing radiation, besides its low cost.

Electrical Impedance Tomography consists in reconstructing conductivity distribution images from the inside of a body, based on current injection and potential measurement protocols, where those potential measurements are taken on the boundary of the domain. This technique has been largely applied in different fields, as industrial monitoring [1], geophysics [2], and biomedical engineering [3, 4]. In the context of the latest field, recent work [5] has discussed viability of EIT to continuous monitoring of cardiac ejection fraction, and other related works [6–8] have shown preliminary results on the same subject. Such works deal with 2-D model of the human torso, contemplating internal inclusions for the heart ventricles and the lungs. The lungs are considered in the studies once their low conductivities work as barriers to the electrical currents used in the experiments. The mentioned 2-D model for human torso with cavities inclusions for the lungs and heart ventricles is used on the study presented here.

This work addresses the inverse problem of determining cardiac ejection fraction by means of EIT, from an optimization point of view. Recent work has shown that *Levenberg-Marquardt Method* (LMM) is well-suited for this purpose [9]. However, LMM is a local search method, and the quality of local optima obtained by that kind of strategy depends on the initial solution given to the technique. Global techniques, like Genetic Algorithms have also been tested for the problem [10], but LMM provided the best results, so far. Hence, a natural evolution of the research is to try methods using multiple local searches, fed with different initial solutions, like *Multistart Local Search* or *Iterated Local Search* (ILS), that has some advantages over the first, as will be further discussed in this work. Therefore, our work proposes the application of ILS heuristic to the EIT inverse problem, in order to investigate the impact of using this approach versus the classic version of Levenberg-Marquardt. The methods used in this study are presented next section, starting with the 2-D torso model.

2. Methods

2.1. Two-dimension Models for the Human Torso

The human torso model used in this work is the same used in [9]. It models the torso as a 2-D surface with five different regions. Two of them representing the lungs, one for each heart ventricle and the last one to represent the rest of the torso. The shape of the regions of interest are obtained by manual segmentation of magnetic resonance images, in two different phases of heart cycle: end of systole and end of diastole. For simplicity matters, the shape of the lungs and torso are considered constant during a heart cycle. Figure 1 illustrates the result of such manual segmentation.

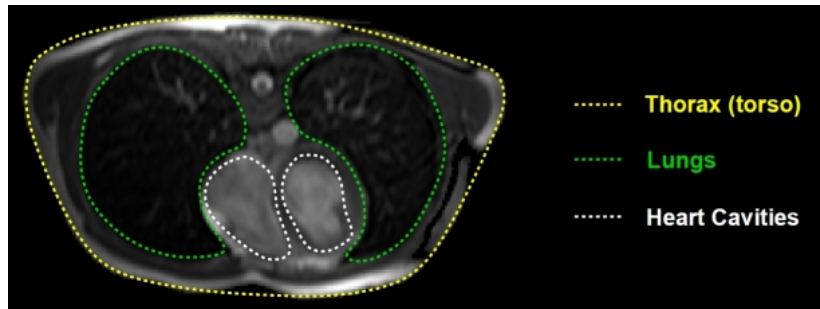


Figure 1. Manual segmentation of a magnetic resonance image

After the segmentation, an important step on modeling is to represent the boundary lines of the regions by means of extended x-splines [11] curves with minimum number of control points. There are 7 control points for left ventricle and 8 control points for right ventricle. As the goal of our method is to recover the ventricles shape from electric potential measurements taken on the boundary of the body, and with each control point represented by two coordinates, there are $7 \times 2 + 8 \times 2 = 30$ spline parameters (variables) to be estimated during the optimization process. In other words, the technique would have to find the best set of values of the parameters of these splines that minimizes the geometric errors in shape recovering. In addition, we apply a strategy to reduce the number of parameters. The main idea behind the strategy is to use only one parameter to define the position of each control point. Once the same control points were used in both systolic and diastolic phases, it is possible to connect their positions in each phase through a line. Then, for each control point i , a linear interpolation, parametrized by a scalar t_i , is performed to determine intermediate values between the two phases. In such convention, $t_i = 0, \forall i$, is relative to the position of spline control points i at the end of systole, while $t_i = 1, \forall i$, is relative to the position of spline control points i at the end of diastole. Provided that, the goal is redefined to recover the cavities shape by estimating 15 parameters $t_i, i = 1 \dots 15$, instead of the 30 original ones.

Besides geometrical issues, it is also important to discuss electrical issues for our model. The main feature to electrically identify a biological tissue is its conductivity. Main factors that influence the properties of biological tissues are presented by Grimnes [12]. The tissues can be classified in thirty different kinds, according to their electrical properties [13] and can be grouped in four major groups: epithelium, muscle, connective tissue and nervous tissue. The conductivity of a tissue can also be influenced by other “environment” issues, like

the frequency of electrical current, presence of water, temperature, etc.

In this work, we have taken some assumptions in order to simplify the problem. The first assumption is that the conductivity of a tissue is known, constant and isotropic. The last one has to do with the kinds of tissue themselves. We assume there are three different tissue conductivities in our model associated to: lungs, heart cavities and torso. Such assumptions are important since biological tissues are very difficult to characterize and even in the literature electrical properties values reported vary substantially.

For the tissues that composes the torso region, Bruder et al. [14] suggests to work with a mean resistivity value to represent such region. The resistivity of the air is $10^{20}\Omega cm$, but it is difficult to determine the resistivity of a lung filled with air. Rush et al. [15] propose a scheme to represent heart cavities filled with blood, which comprises a simplified resistivity distribution for the blood tissue surrounded by homogeneous material with resistivity ten times greater. Based on this, we have extended this scheme to represent lung regions filled with air. Table 1 present some resistivity values found in literature for our tissues of interest: lungs, blood, heart and torso.

Table 1. Resistivity values for biological tissues found in literature.

Tissue	Resistivity (Ωcm)	Reference(s)
Blood	150	Barber and Brown [16]
	150	Yang and Patterson [17]
	100	Schwan and Kay [18]
Heart	400	Patterson and Zhang [19]
	250	Yang and Patterson [17]
	400 - 800	Baysal and Eyuboglu [20]
Lung	727 - 2363	Barber and Brown [16]
	1400	Patterson and Zhang [19]
	600 - 2000	Baysal and Eyuboglu [20]
Torso	500	Bruder et al. [14]

In our experiments, we have taken the values of $1000\Omega cm$ for torso and of $100\Omega cm$ for blood. For the lungs, we used two different values of Ratio of Lung to Torso resistivity (RLT): $RLT = 20$, corresponding to $20000\Omega cm$ for lung resistivity and $RLT = 50$, corresponding to $50000\Omega cm$ for lung resistivity. Subsections 2.2 and 2.3, present, respectively, the forward and inverse problem of EIT using the model described here.

2.2. The Forward Problem

The forward problem of EIT consist of calculating electrical potentials on the external boundary of the torso generated by a current injection on a pair of electrodes. In the forward problem, the conductivity (or resistivity) distribution of the domain is known. In the model used in this work, the domain is divided in regions with different conductivities, as mentioned before. The electrical potential (ϕ) for every point must satisfy Laplace's equation:

$$\nabla^2 \phi = 0, \quad (2)$$

subject to the boundary conditions:

$$\begin{aligned}
\sigma_L \nabla \phi \cdot \eta &= \sigma_T \nabla \phi \cdot \eta, & x \in \Gamma_1 \\
\sigma_B \nabla \phi \cdot \eta &= \sigma_T \nabla \phi \cdot \eta, & x \in \Gamma_2 \\
\sigma_T \nabla \phi \cdot \eta &= J_i, & x \in \Gamma_3^{ie} \\
\nabla \phi \cdot \eta &= 0, & x \in (\Gamma_3 - \Gamma_3^{ie})
\end{aligned}$$

where Γ_1 is the interface between lung and torso region; Γ_2 is the interface between blood and torso region; Γ_3 is the external boundary of the body; Γ_3^{ie} is the portion of Γ_3 in which the i^{th} electrode is placed on; J_i is the electric current injected through the i^{th} electrode; and σ_T , σ_B and σ_L are, respectively, torso, blood and lung conductivities.

The present work uses the Boundary Elements Method (BEM) [21] to solve the forward problem, with implementation based on the one used in [22]. Next section describes the inverse problem associated with the forward problem presented here.

2.3. The Inverse Problem

From an electric point of view, the inverse problem associated to EIT aims on generating an image of the electrical resistivity from measures of electrical potential at the external boundary. From a geometric point of view, the aim is to recover the shape of the ventricular cavities via the estimation of vector \mathbf{t} , containing the parameters $t_i, i = 1 \dots 15$, as described in Section 2.1. This problem can be formulated as an optimization problem. Our goal is to minimize an objective function that computes the distance between measured electrical potential values (taken from a pair of electrodes on the external boundary of the body) and the computed ones. The computed potential values depend on the heart cavities shape, parametrized by vector \mathbf{t} and they are calculated as described in Section 2.1. Therefore, the goal is to find the best parameter vector \mathbf{t} that minimizes:

$$F = \frac{1}{2} \|R(\mathbf{t})\|^2 = \frac{1}{2} \sum_{j=1}^m (\phi(\mathbf{t})_j - \bar{\phi}_j)^2 \quad (3)$$

where $\bar{\phi}_j$ is the j^{th} measured electrical potential; $\phi(\mathbf{t})_j$ is the corresponding computed electrical potential that depends on the heart cavity shape parametrized by \mathbf{t} , calculated as described in Section 2.2; m is the number of measurements taken, and depends on the current injection pattern; and $R(\mathbf{t})$ is the so called residual vector. It is important to note that, in this work, the values of $\bar{\phi}_j$, which were supposed to be measured, are synthetically generated. The optimization problem presented this section is solved, in this work, using two different approaches, described in Subsections 2.3.2, 2.3.3 and 2.3.4.

Before discussing optimization methods, it is important to pay some attention to an important aspect though. The inverse problem is here defined in terms of minimizing the residual between measured and computed values. Thus, there must to be some criteria to determine such residuals, or errors. As we are interested in obtain images from the inside domain of a body, we used *geometric error*, presented next Subsection, to determine the accuracy of the implemented techniques. Nevertheless, although such metric is used inside

the optimization methods, for instance, to determine if a solution is better than a previous one, at the Section 3.3, which presents the computational results of our work, other kind of metric is used to compare those techniques.

2.3.1 Geometric Error

In the context of this work, we define *geometric error* as the difference between the geometry of the inclusion obtained by an optimization method and the “real”, known, geometry. In this work, the “real” geometries refer to the ones corresponding to the synthetically generated data. Intuitively, the geometric error can be evaluated by visual inspection of the generated images. However, when two or more images are too close to each other, such visual inspection cannot be precise enough to determine which one is more accurate. Besides, for the purpose of automatizing the process of calculating the geometric error, a visual inspection is ineffective. Thus, objective metrics are needed to accomplish such purposes.

The geometric error is calculated in the following way. The known mesh of target inclusion is refined by dividing each element into ten parts. For each node of the mesh of elements of the identified inclusion, the distance to the closest node of the target mesh is calculated. The obtained values are accumulated until all nodes of the identified inclusion are considered. The total value obtained is divided by the number of nodes of the boundary of the identified inclusion and by the target perimeter. The result is the geometric error. The obtained value is non-dimensional. An unitary value means that, in average, each node of the boundary of the identified inclusion is far from the target by the value of the target perimeter. The value of the error aims on reflecting the quality of the obtained images in the optimization process. Further studies could contemplate other geometric features, like the area of an inclusion. More on geometric error and some strategies to compute it in EIT problem can be found in [23]. Once presented the metric used to determine the quality (or the error) of a minimizing solution, next Subsections present the optimization techniques themselves.

2.3.2 Levenberg-Marquardt Method and Related Work

The problem represented by Equation 3 is a non-linear least square problem, and different methods can be applied to solve this optimization problem. Peters et al. [9] addressed such problem with Levenberg-Marquardt Method (LMM) [24], which is also in the basis for our approach. LMM can be viewed as a modification of Gauss-Newton method, with the model trust region approach. The minimizer of the non-linear least-square problem is obtained iteratively in the method. At each step, updates to \mathbf{t} approximation are given by the minimizer \mathbf{t}_+ of the following constrained linear least-squared problem:

$$\begin{aligned} &\text{minimizes} && \|R(\mathbf{t}_0) + \mathbf{J}(\mathbf{t}_0)(\mathbf{t}_+ - \mathbf{t}_0)\|_2 \\ &\text{subject to} && \|\mathbf{t}_+ - \mathbf{t}_0\|_2 \leq \delta_0 \\ &\text{where} && R(\mathbf{t}) = \phi(\mathbf{t}) - \bar{\phi} \\ &&& J \in \mathbb{R}^{m \times n}, J_{ij} = \frac{\partial R_i}{\partial t_j} \end{aligned}$$

Where $R(\mathbf{t})$ is the residual vector; \mathbf{t}_0 is the current value for the minimization parameters vector; \mathbf{t}_+ is the updated solution vector; J is the Jacobian matrix with the derivatives of each element of the residual vector with respect to the optimization variables; and δ_0 is the initial radius value for the trust region. The vector \mathbf{t}_+ , solution of the constrained minimization problem, is given by

$$\mathbf{t}_+ = \mathbf{t}_0 - (J(\mathbf{t}_0)^T J(\mathbf{t}_0) + \mu \mathbf{I})^{-1} J(\mathbf{t}_0)^T R(\mathbf{t}_0), \quad (4)$$

where \mathbf{I} is the identity matrix. The parameter μ is the parameter that provides the modification of Gauss-Newton method, mentioned earlier, and its value changes from one iteration to another. A more detailed description about Levenberg-Marquardt Method can be found in [24].

In this work, LMM is used in two different ways. First, a set of independent executions of LMM, each one using a different initial guess for \mathbf{t} that is randomly chosen, composes a traditional Multistart Local Search method based on LMM. In the second scheme LMM is used as the local search method for the heuristic called Iterated Local Search (ILS) [25].

A more detailed discussion about the experimental setup and other related issues is presented in the next sections. By now, we limit the discussion to the fact that Levenberg-Marquardt was the method chosen to implement the local search, in spite of other alternatives, due to its promising results shown in [9]. Other previous works have adopted different alternatives, such as Powell's method [26], Genetic Algorithms [27, 28] and Feasible Arc Interior Point Algorithm (FAIPA) [29], but LMM has been shown as the best alternative, so far.

In next subsection, we describe the ILS metaheuristic, used to compose the hybrid approach proposed in this work.

2.3.3 Iterated Local Search

The ILS metaheuristic is a template for the development of a heuristic, i.e. is a *meta-heuristic*. The ILS template defines that first a local search is applied to an initial solution. Then, at each iteration, a *perturbation* of the obtained local optimum is performed, followed by a local search method being applied to the perturbed solution, resulting in a new local optimum. Finally, this new local optimum is subjected to some acceptance criteria, and replaces the old one if it attends to certain predefined conditions. This process is repeated until some stopping criteria are attended. Such process is described by Algorithm 1.

Before presenting our implementation of the ILS template, is necessary, though, to bring a light to some details of ILS method.

- **Local search.** Any method, deterministic or stochastic, can be used as a local search in ILS metaheuristic. That method is treated from a black box point of view. It is important to note that population-based heuristics, like Genetic Algorithms, are not suitable for this purposes, in principle, once local search is usually a single-solution based method, like Levenberg-Marquardt.
- **Perturbation Method.** This should be a large move of the current solution, in order to provide diversification to ILS solutions and as an attempt of pushing the search to

Algorithm 1: Template for Iterated Local Search algorithm

input : An initial solution s_0

output: Best solution found

// Apply a predefined local search method

1 $s_* \leftarrow \text{local_search}(s_0)$;

2 **repeat**

 // Perturb local optimum obtained

3 $s' \leftarrow \text{Perturbation}(s_*, \text{search_history})$;

 // Apply local search on the new perturbed solution

4 $s'_* \leftarrow \text{local_search}(s')$;

 // Accepting criteria

5 $s_* \leftarrow \text{Acceptance}(s_*, s'_*, \text{search_history})$;

6 **until** Stopping criteria;

another basin of attraction. It is important that the perturbation method preserves some part of the given solution while strongly perturb the other. This is intended to keep some “good” information and at the same time to diversificate the obtained solutions.

- **Acceptance criteria.** It defines the conditions that a new solution (local optimum) have to satisfy to replace the current one.

Figure 2 illustrates the basic schema in which Iterated Local Search is based on. A last important observation about ILS is that it differs from a *Multistart Local Search* where initial guesses are usually chosen randomly. ILS presents an improvement, since it is an iterative process where local searches are performed using perturbed versions of previously found local optima.

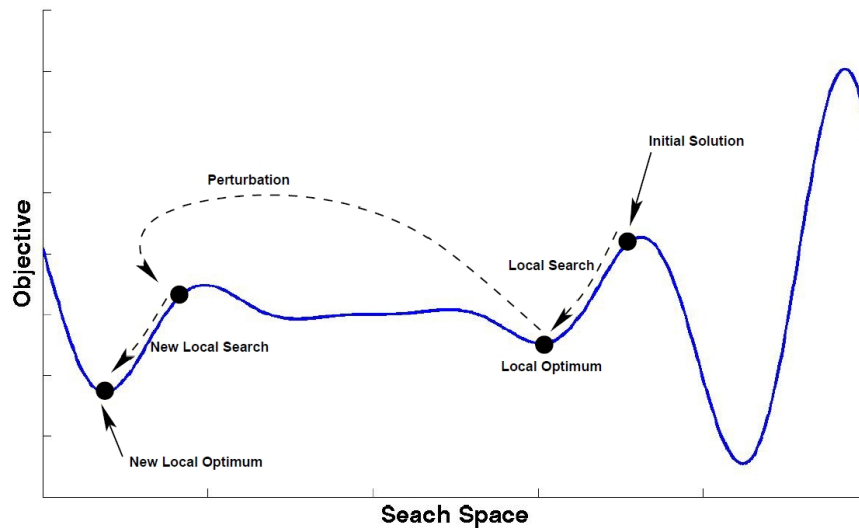


Figure 2. ILS basic schema

Next, we present our implementation of ILS using Levenberg-Marquardt as the local search method, as well as the perturbation method and acceptance criteria used. We call our implementation ILS-LM.

2.3.4 Hybrid ILS-LM Heuristic

To define our ILS implementation, we need to determine some important points. First one, and most important of all, we set LM method as the local search procedure. We also propose a perturbation method that guarantees a minimum of diversification of the local optima. This perturbation method, that we call *K-dPerturb*, will be better described next. Finally, we have to establish the acceptance and stopping criteria. Algorithm 2 illustrates the choices made in this work.

Algorithm 2: Hybrid ILS-LM proposed

input : *maxit*: maximum number of iterations to be executed
input : Initial solution s_0 : systolic or diastolic ventricle cavity of a normal heart
input : *perc*: the percentage of vector \mathbf{t} parameters (represented here by s_*) that are affected by random perturbation
input : ϵ_0 : precision used in expansion of a *k-d tree* node
output: Best solution found

- 1 $s_* \leftarrow \text{Levenberg-Marquardt}(s_0)$;
- 2 initialize *k-dTree* with s_* ;
- 3 **repeat**
- 4 $s' \leftarrow \text{K-dPerturb}(s_*, k\text{-dTree}, \text{perc}, \epsilon_0)$;
- 5 $s'_* \leftarrow \text{Levenberg-Marquardt}(s')$;
- 6 $s_* \leftarrow \text{SimpleAcceptance}(s_*, s'_*, k\text{-dTree})$;
- 7 **until** *maxit times*;

Before continuing, an important observation here is that for simplicity reasons, we omitted the passing of the “measured” values ($\bar{\phi}_j$ in the inverse problem formulation) as parameters to the procedures described by the algorithms presented in this section, but one should keep in mind they are needed to calculate the precision of the methods used. Observed this, we can continue with the analysis of the heuristic ILS-LM.

SimpleAcceptance implements a criterion in which the solution s'_* replaces s_* if its relative error is lower than the one of the current solution. Accepted or not, every considered solution s'_* is added to the search history, implemented with a *k-d tree* data structure [30, 31], described later. This approach focuses only on intensification, i.e., it just aims on the quality of solutions. In our proposal, diversification aspects are provided inside the perturbation method. Thus, *K-dPerturb* is a partially random method using the same *k-d tree* structure of *SimpleAcceptance* to avoid generating repeated or very similar solutions, which could lead to the same local optima. As already mentioned, the *k-d tree* structure also works, in this case, as a search history. High-level algorithms for *SimpleAcceptance* and *K-dPerturb* are presented, respectively, in Algorithms 3 and 4.

It is worth to observe that we have defined in the implementation that a node from *k-dTree* can be expanded when the distance between two solutions belonging to that node is greater than a given precision ϵ_0 , which was kept during our experiments with the value 10^{-2} . Also, we kept as constant the values of $\text{perc} = 0.15$ and $\text{maxit} = 11$, the maximum number of performed iterations. The choice of $\text{maxit} = 11$ was due to how we compared ILS-LM and LMM, and is discussed later.

Algorithm 3: Simple acceptance method

input : s_* : current best solution
input : s'_* : candidate to be the new current best solution
input : k - $dTree$ structure, with history of all s_* found
output: Solution s_* , updated current best solution

- 1 $n \leftarrow$ node of k - $dTree$ where s'_* should be placed on;
- 2 **if** n can be expanded **then**
- 3 expand n ;
- 4 insert s'_* in k - $dTree$;
- 5 **if** $error(s'_*) < error(s_*)$ **then**
- 6 $s_* \leftarrow s'_*$;

The steps related to the perturbation of a control point p , that is, the process comprised by lines from 6 to 16 in Algorithm 4, implement a way of controllably disturb such control point. The basic idea is simple and consists in produce a new value for the parameter t_i corresponding to that point. Concerning to the description of the perturbation method, the terms p or t_i are used interchangeably.

These controlled perturbations work as the following. After a control point p is selected for being perturbed, its two neighbors are identified (p_{left} and p_{right}). Then, a line segment, connecting such neighbors, is traced ($segRef$) and the orthogonal projection of p over $segRef$ is obtained (p_{orth}). These are the *preparation steps*. Next, the method calculates limits to the perturbation. Those limits are calculated in order to be dimensionless, just like the parameters t_i . Yet, they are intended to promote as much perturbation as possible, but without producing values too far from a feasible geometry, so the maximum normalized propotion between the distances $d1_p$, $d2_p$ and $d3_p$ is used as such limitation. This resulting factor has the desired features: is dimensionless and promote good perturbations, without too much relaxation. Figure 3 illustrates the distances used in perturbation factor calculation.

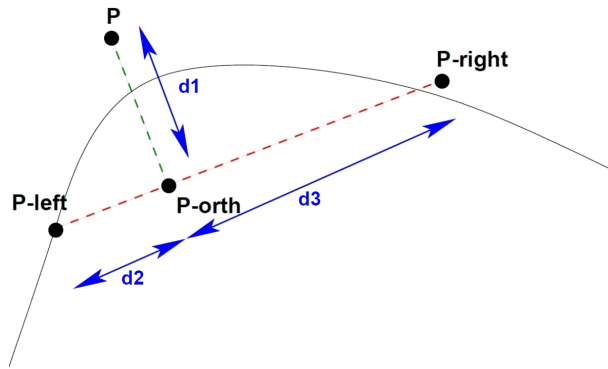


Figure 3. Distances used in perturbation factor calculation.

Besides the perturbation process, another key feature of our implementation is the use of a k - d tree to keep the search history of obtained local optima. Such data structure allows the algorithm to prioritize unexplored regions of search space, in an attempt to avoid redundant efforts on performing searches towards already found local optima. The k - d tree data structure

Algorithm 4: K-d Perturbation method

input : s_* : current best solution
input : k - $dTree$ structure, with history of all s_* found
input : $perc$ and ϵ_0 : the same of Algorithm 2
output: Solution s' to be provided as initial solution for Levenberg-Marquardt local search

```
1  $n \leftarrow k$ - $dTree$  node corresponding to  $s_*$ ;  
2  $accept \leftarrow \text{false}$ ;  
3 repeat  
    // Perturbation phase  
4    $\{ts\} \leftarrow$  randomly selected  $perc$  percent of  $t$  parameters from  $s_*$ ;  
5   foreach  $t \in \{ts\}$  do  
       // Preparation steps  
6        $p \leftarrow$  point corresponding to  $t$ ;  
7        $p_{left} \leftarrow$  neighbor of  $p$  in anti-clockwise direction;  
8        $p_{right} \leftarrow$  neighbor of  $p$  in clockwise direction;  
9        $refSeg \leftarrow$  line segment from  $p_{left}$  to  $p_{right}$ ;  
10       $p_{orth} \leftarrow$  orthogonal projection of  $p$  in  $refSeg$ ;  
       // Perturbation factor calculation steps  
11       $d1_p \leftarrow dist(p, p_{orth})$ ;  
12       $d2_p \leftarrow dist(p_{orth}, p_{left})$ ;  
13       $d3_p \leftarrow dist(p_{orth}, p_{right})$ ;  
14       $d_p \leftarrow \frac{\max(d1_p, d2_p, d3_p) - \min(d1_p, d2_p, d3_p)}{\min(d1_p, d2_p, d3_p)}$ ;  
15       $perturbFactor \leftarrow$  randomly select a value in  $[-d_p, d_p]$ ;  
       // Perturb: new value for  $t$  does not need to be in  
        $[0, 1]$   
16       $t \leftarrow t + perturbFactor$ ;  
  
    // Validation of diversification phase  
17     $n' \leftarrow k$ - $dTree$  node corresponding to  $s'$ ;  
18    if  $n \neq n'$  then  
19      insert  $s'$  in  $k$ - $dTree$ ;  
20       $accept \leftarrow \text{true}$ ;  
21    else  
22      if  $(n = n')$  and  $(n \text{ can be expanded})$  then  
23        expand  $n$ ;  
24        insert  $s'$  in  $k$ - $dTree$ ;  
25         $accept \leftarrow \text{true}$ ;  
26      else  
27        discard  $s'$ ;  
28 until  $accept = \text{true}$ ;
```

is presented next Subsection.

2.3.5 K-d trees

A *k-d tree* [30, 31] is a data structure for organizing points in a *k-dimensional* space, by means of establishing partitions in that space. In this work, the referred space is the search space of the optimization method, i.e. the space composed by the feasible values of vector \mathbf{t} . From a computational point of view, *k-d tree* can be implemented as a binary tree with good performance on retrieving information. Each node of the current state of the tree represents a point in the space. If the node is not a leaf, its geometric interpretation could be thought as implicitly generating a splitting hyperplane diving the space in two parts.

Every node in the tree is associated with one of the k dimensions. The direction of splitting hyperplane is orthogonal (perpendicular) to the axis corresponding to that dimension. This way, all of the points of the space with coordinate values, corresponding to the selected dimension, that are inferior to that one of the node lie on the left subtree, while the points with values greater than that one of the node lie on the right subtree. Therefore, the hyperplane would be the region of the space with that value for the selected dimension and its normal would be the corresponding axis.

As there are different possible ways of choosing the order of axis selection, also there are different possible ways to construct *k-d trees*. The most common one consists in choosing the axis in a cyclic manner, returning to the first axis after the last one has been chosen. The order of axis are usually predefined. For example, in \mathbb{R}^3 space, one could select $x - axis$ for root node, after $y - axis$ for the children nodes of root, and then $z - axis$ for the grandchildren of root, returning to $x - axis$ for next generation, and to $y - axis$ for next generation and so on. Figure 4 shows an example of both generated data structure and the corresponding space division, for a particular case of \mathbb{R}^2 where the points in Table 2 are inserted into the structure in the order they appear in referred Table.

Table 2. Sequence of points inserted into *kd-tree* example.

Order	Point	Order	Point
1	(4, 10)	7	(9, 6)
2	(7, 9)	8	(5, 3)
3	(1, 8)	9	(3, 1)
4	(8, 4)	10	(2, 5)
5	(5, 10)	11	(6, 1)
6	(7, 3)		

In our implementation, the use of a *k-d tree* provides a method for fast retrieving the spatial position of a stored local minimum (vector \mathbf{t} representation), as well as the possibility of prioritizing different regions of the space when perturbing a solution. If a perturbation generates a new solution that is not far enough (such criterion is determined by parameter ϵ_0) from an existing point in the tree, this perturbation is not considered and a new one is generated. In the other way, if a new solution is accepted in the distance criterion, it can be placed in the tree, splitting the search space into new smaller portions that can be explored later.

Summing up, the *k-d tree* structure consists in a log of all intermediate solutions generated during the execution of ILS-LM. That is, it keeps all local optima generated by each of

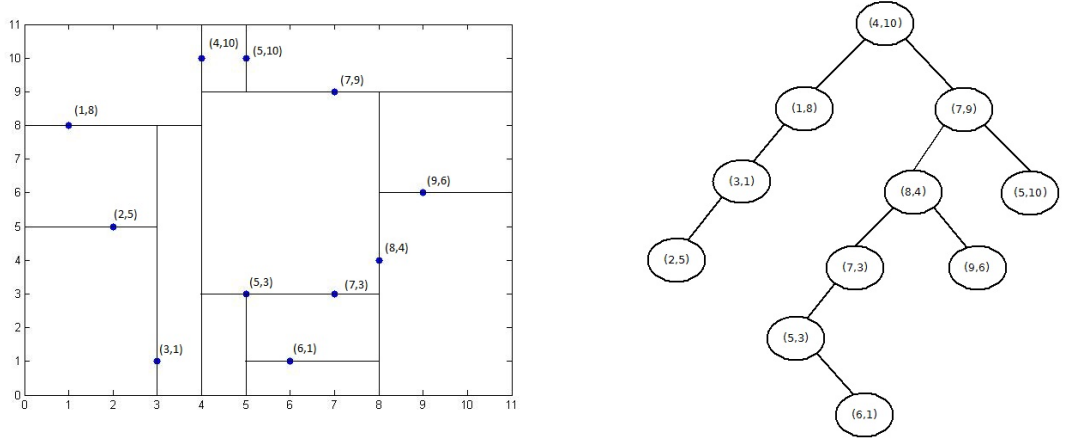


Figure 4. **Left:** space partitions and **right:** generated k - d tree for the point sequence in Table 2

Levenberg-Marquardt (local search) execution and also all the solutions returned (approved) by K - d Perturb. Once discussed this last aspect of our proposal, in Section 3, we discuss the experiments performed in this work, as well as the obtained results.

3. Results

Before taking a look into the experiments performed and their corresponding computational results, it is important to discuss a little about a factor that directly influences the experiments, the stimulation patterns.

3.1. Stimulation Patterns

The choice of current injection protocols and electrical potential measurements is an important aspect on studying EIT problem. The problem is ill-conditioned, so the image generated is very sensible to the choices made. Nevertheless, this work do not focuses on the study of such protocols and measurements. A deeper discussion on the subject can be found on [32]. We are limited here to test the same two patterns used in [9]. The first one is called diametrical. It is called this way because of an analogy with a circular domain, where the electrodes used to inject current would be diametrically opposed. In this pattern, eight different cases of current injection are taken. For each case, there are thirteen measures of potential. This yields $13 \times 8 = 104$ measures.

The other pattern, called alternative, consists of an attempt to explore the region of interest better than other regions. Hence, the electrodes used to inject current are taken near to the heart, in this pattern. These electrodes are also called driven electrodes. Therefore, there are six cases of current injection with thirteen measures, what yields $13 \times 6 = 78$ measures. Figure 5 presents diagrams for both patterns. Each doubled arrow line indicates a pair of driven electrodes in each case of current injection.

Once again, it is important to note that the terms “measure” or “measured” used, in the context of this study, refers to synthetically generated values, generated by numerical methods.

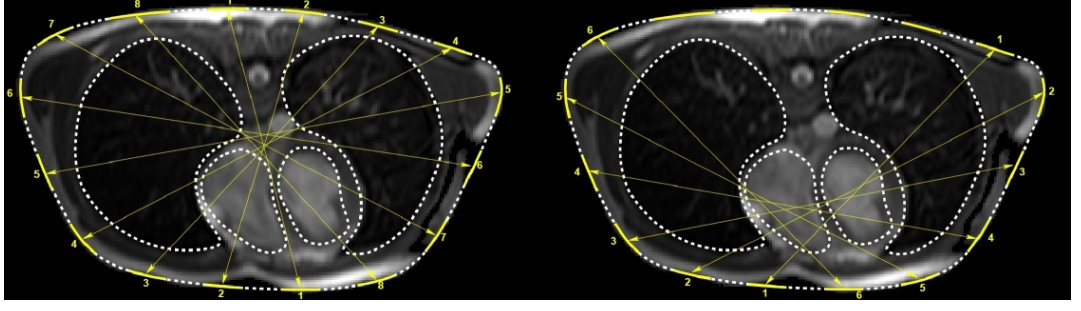


Figure 5. Stimulation patterns simulated. Left: diametrical pattern. Right: alternative pattern.

Next section presents the experimental setup for the tests performed.

3.2. Experimental Setup

As mentioned before, the experiments performed in this work aim at reducing geometric error, what directly has to do with minimizing the error while determining EF. To do so, we need to establish target values for EF that the tested techniques have to found. We call those values target values. Such target values were determined as described next.

For the two dimensional model used in the present work, the areas of the transversal section of heart cavities were assumed to be proportional to their volumes, that is, a cylindrical approximation is used, so that EF is calculate by

$$EF = \frac{EDA - ESA}{EDA}, \quad (5)$$

where EDA stands for the area of transversal section of the ventricle at the end of diastole, while ESA stands for the area of transversal section of the ventricle at the end of systole. Applying Equation 5 to values obtained from segmentation of MR images taken at the end of systole and at the end of diastole, we have calculated that EF of left ventricle is 59.24%, while EF of right ventricle is 29.95%.

An artificial cardiac disfunction was synthetically generated then. Such disfunction consists in altering cardiac cycle by making the end-systolic volume being greater than the normal, while the end-diastolic volume remains unaltered. This configures a new heart cycle, in which EF of left ventricle values 33.01% and EF of right ventricle values 16.19%. These are the target values to be estimated by the optimization methods used here.

Once defined the target values, we needed to determine initial guesses to the minimization techniques. Two different initial guesses for vector \mathbf{t} were used. The first one corresponds to the shape of the ventricles at the end of diastole ($t_i = 1, \forall i$) while the second corresponds to the shape at the end of systole ($t_i = 0, \forall i$). In both cases, they refer to the values of a normal heart. A comparison of target values and initial guesses can be viewed in Figure 6.

Altogether, eight different experimental setups were submitted to LM and ILS-LM. Each one of these experimental setups is composed by one of two possible values for Ratio of Lung to Torso resistivity ($RLT = 20$ or $RLT = 50$); one of two possible values for stimulus pattern (diametrical or alternative); and one of two possible values for initial guess ($t_i = 0$ or $t_i = 1$). Thus, there are $2 \times 2 \times 2 = 8$ different experimental setup combinations, concerning

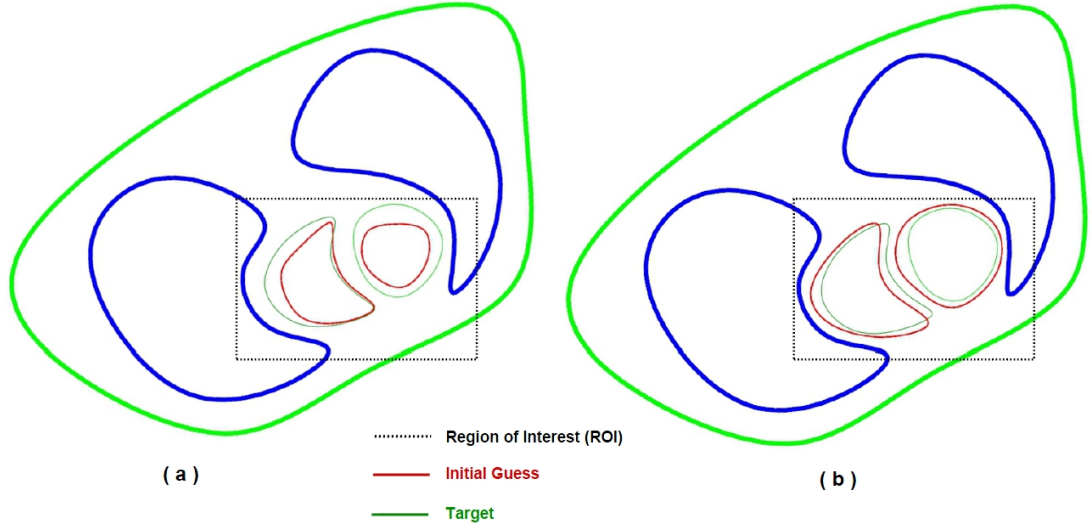


Figure 6. Targets (dark green) and initial guesses (red) given to the optimization procedures: (a) $t_i = 0, \forall i$; (b) $t_i = 1, \forall i$.

to RLT, stimulus patterns and initial guesses values. The results of such experiments, as well as more information about the executions, are shown in Section 3.3.

3.3. Computational Results

The experiments performed were set up the way described in the last subsection. In addition, other configurations used in the work were already mentioned, namely $\epsilon_0 = 10^{-2}$, $perc = 0.15$ and $n = 11$ for our ILS-LM method. One should remember that the first local search performed by ILS-LM is executed outside the loop, so we have a total of $1 + 11 = 12$ LM executions inside the metaheuristic. In order to make a fair comparison, we executed the LM algorithm 12 times, each one using a different and randomly chosen initial guess. One of the 12 Levenberg-Marquardt executions receives the same initial guess as the one used for the ILS approach. The rest of them receive randomly perturbed (in 15% of control points) versions of this initial approximation. The perturbation used to generate these initial approximations is the same used inside *K-dPerturb*, as described in Subsection 2.3.4.

As already mentioned, the metric used here to compare the optimization methods (relative errors to target EF values) is not the same used inside the implementations to measure the quality of solutions (geometric error). The first one is more adequate to the analysis performed in this section, while the second can be more easily applied inside the algorithm, when compared to the prior metric. As the two metrics are directly correlated, such change of measurements does not affect the conclusions made. The relative error to ejection fraction is computed as below:

$$\Delta\% = 100 \times \frac{\tilde{EF} - EF}{EF}, \quad (6)$$

where $\Delta\%$ is the relative error, \tilde{EF} is the ejection fraction calculated from the values obtained by the optimization techniques and EF is the target value for ejection fraction. Such target values are 16.19 for the right ventricle and 33.01 for the left ventricle, as mentioned in

Subsection 3.2. The value of \tilde{EF} is calculated according to Equation 5, using, for it, the values of area of the geometric shapes resulting from the solutions returned by each minimization technique tested.

As the hybrid technique returns only its best solution, we have made the same to the pack of executions of classic LM. To compare results, we consider only the best result of the twelve executions of LM. Table 3 summarizes those results. The values presented in the table are the relative errors obtained by each method.

Table 3. Relative errors obtained in the set of experiments.

Diametrical Pattern								
Initial Guess	LM (Best Solution Found)				ILS Approach			
	RLT=50		RLT=20		RLT=50		RLT=20	
	RV	LV	RV	LV	RV	LV	RV	LV
$t_i = 0$	-18.10%	4.15%	-5.19%	3.03%	-8.03%	2.76%	-6.42%	1.85%
$t_i = 1$	2.47%	-2.36%	-1.91%	0.03%	-1.98%	0.21%	-5.37%	-0.06%
Alternative Pattern								
Initial Guess	LM (Best Solution Found)				ILS Approach			
	RLT=50		RLT=20		RLT=50		RLT=20	
	RV	LV	RV	LV	RV	LV	RV	LV
$t_i = 0$	-19.33%	8.09%	25.51%	-9.18%	-8.71%	3.70%	15.69%	-3.45%
$t_i = 1$	15.26%	-0.15%	28.47%	-10.18%	10.75%	0.24%	17.23%	-4.03%

One can see that, in general, diametrical pattern presents better results than the alternative pattern. The only exception is the case of left ventricle with $RLT = 50$ for classic Levenberg-Marquardt. Once diametrical pattern uses more pair of electrodes, this may indicates that more studies should be made in the direction of finding the best injection protocol with minimum of measures taken. Another important observation is that, also with only one exception (right ventricle, $RLT = 20$, diametrical pattern), the hybrid approach has found better results than classic LM.

Concerning to initial guesses, for the diametrical pattern, all results using $t_i = 1, \forall i$ (end of diastole) were better than using $t_i = 0, \forall i$ (end of systole). In some cases, like right ventricle with $RLT = 50$, the improvement was significant. For the case of the alternative pattern, the end of diastole was a better initial guess only in three out of eight cases. Hence, the best initial guess may vary with the protocol of current injection and measurement used.

Concerning the Ratio of Lung to Torso resistivity, the value of $RLT = 20$ has shown better results for both techniques, once again, with only one exception (right ventricle, $t_i = 1$, ILS-LM), when considering diametrical pattern. The opposite occurred when considering the alternative pattern.

In terms of computational costs, each execution of classic LM took about 25 min for the diametrical pattern, and about 20 min for the alternative pattern, what yields near 5 hours of execution for the more expensive case (diametrical) and about 4 hours for the cheapest one (alternative). The execution of ILS-LM was a little faster: around 4 hours and 30 min for the diametrical case and around 3 hours and 45 minutes for the alternative pattern. Such tests were taken in a machine with Intel Core I5, 2.8GHz, 4GB of RAM and the methods were implemented in C/C++ and Fortran77 languages.

4. Conclusions

Comparing the different protocols, the behavior of the diametrical one seems to be more stable. The results of diametrical pattern were more accurate than the alternate one, but with higher costs and using more measures. However, the reduction of execution time in about 20% obtained by the alternate pattern, justifies further investigations on the topic of stimulation patterns.

The most important conclusion of this work is about the optimization technique used. The Iterative Local Search implementation improved the quality of the obtained solutions and reduced the execution time of the inverse problem. The results suggest that this new hybrid method explores the search space in a more efficient way, probably due to the perturbation procedure that allows diversification on the solutions prioritizing regions not yet explored (due to the use of k-d trees). In addition, the fact of using variations of local optima to feed local search procedures might have provided faster convergences of the other local searches, what would explain the the reason why this new hybrid method was faster than the traditional Multistart Local Search method using LM.

Therefore, the results suggest that this new hybrid method, the ILS-LM, is a promising technique for the solution of the inverse problem associated to Electrical Impedance Tomography. Considering also the fact that some of the aspects of ILS-LM were implemented in a very simple fashion, like the *SimpleAcceptance* criteria or the stopping condition, there is much to be done in the direction of increasingly improve the cost vs. benefit relation of the techniques used in EIT inverse problem. In addition, future work should also investigate more intelligent and sophisticated perturbation and acceptance methods than those proposed in this work.

Acknowledgements

The first author would like to thank Departamento de Informática of Universidade Federal de Viçosa. All authors would like to thank the Programa de Pós-graduação em Modelagem Computacional of Universidade Federal de Juiz de Fora, FAPEMIG, CNPq, CAPES and FINEP.

References

- [1] M. Kim, K. Kim, and S. Kim, "Phase boundary estimation in two-phase flows with electrical impedance tomography," *Int. Comm. Heat Transfer*, vol. 31, no. 8, pp. 1105–1114, 2004.
- [2] F. Trigo, R. Lima, and M. Amato, "Electrical impedance tomography using extended kalman filter," vol. 51, no. 1, pp. 72–81, 2004.
- [3] N. Polydorides, W. R. B. Lionheart, and H. McCann, "Krylov subspace iterative techniques: On the brain activity with electrical impedance tomography," vol. 21, no. 6, pp. 596–603, 2002.

- [4] J. Seo, O. Kwon, H. Ammari, and E. Woo, "A mathematical model for breast cancer lesion estimation: Electrical impedance technique using ts2000 commercial system," vol. 51, no. 11, pp. 1898–1906, 2004.
- [5] F. Peters, L. Barra, and R. Santos, "Determination of cardiac ejection fraction by electrical impedance tomography - numerical experiments and viability analysis," in *Computational Science - ICCS 2009*, ser. Lecture Notes in Computer Science, vol. 5544/2009. Springer Berlin / Heidelberg, 2009, pp. 819–828. [Online]. Available: <http://www.springerlink.com/content/ln752561546p5973>
- [6] L. P. S. Barra, F. C. Peters, C. P. Martins, and H. J. C. Barbosa, "Computational experiments in electrical impedance tomography," in *XXVII Iberian Latin American Congress on Computational Methods in Engineering*, Belém, Brazil, 2006.
- [7] L. P. S. Barra, R. W. Santos, F. Peters, E. P. Santos, and H. Barbosa, "Parallel computational experiments in electrical impedance tomography," in *18th Symposium on Computer Architecture and High Performance Computing*, vol. 1, Sociedade Brasileira de Computação. Ouro Preto, Brazil: High Performance Computing in the Life Sciences, 2006, pp. 7–13.
- [8] L. P. S. Barra, P. Mappa, S. Cardoso, and F. C. Peters, "Comparison of the computational performance of optimization algorithms in the solution of an inverse problem (in portuguese)," in *VIII Simpósio de Mecânica Computacional - SIMMEC*. Belo Horizonte, Brazil: PUC-Minas, junho 2008.
- [9] F. C. Peters, L. P. S. Barra, and R. W. Santos, "Determination of cardiac ejection fraction by electrical impedance tomography," in *Medical Imaging*, O. F. Erondy, Ed. InTech, 2011, pp. 253–270.
- [10] C. T. Hsiao, G. Chahine, and N. Gumerov, "Application of a hybrid genetic/powell algorithm and a boundary element method to electrical impedance tomography," vol. 173, pp. 433–454, 2001.
- [11] C. Blanc and C. Schlick, "X-splines: a spline model designed for the end-user," in *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, ser. SIGGRAPH '95. New York, NY, USA: ACM, 1995, pp. 377–386. [Online]. Available: <http://doi.acm.org/10.1145/218380.218488>
- [12] S. Grimnes and O. G. Martinsen, *Bioimpedance and Bioelectricity Basics, Second Edition*. Academic Press, 2008.
- [13] C. Gabriel, S. Gabriel, and E. Corthout, "The dielectric properties of biological tissue: I. literature survey," *Physics in Medicine and Biology*, vol. 41, no. 11, pp. 2231–2249, 1996.
- [14] H. Bruder, B. Scholz, and K. Abraham-Fuchs, "The influence of inhomogeneous volume conductor models on the ecg and the mcg," *Physics in Medicine and Biology*, vol. 39, no. 11, pp. 1949–1968, 1994.

- [15] S. Rush, J. A. Abildskov, and McFeer, "Resistivity of body tissues at low frequencies," *Circulation research*, vol. 12, pp. 40–50, 1963.
- [16] D. Barber and A. Seagar, "Applied potential tomography," *Clin. Phys. Physiol. Meas.*, vol. 8, pp. 47–54, 1984.
- [17] F. Yang and R. P. Patterson, "The contribution of the lungs to thoracic impedance measurements: a simulation study based on a high resolution finite difference model," *Physiological Measurement*, vol. 28, no. 7, pp. S153–S161, 2007.
- [18] H. Schwan and C. Kay, "Specific resistance of body tissues," *Circulation Research*, vol. 4, no. 6, pp. 664–670, 1956.
- [19] R. P. Patterson and J. Zhang, "Evaluation of an eit reconstruction algorithm using finite difference human thorax models as phantoms," *Physiological Measurement*, vol. 24, no. 2, p. 467475, 2003.
- [20] U. Baysal and E. B. M. , "Tissue resistivity estimation in the presence of positional and geometrical uncertainties," *Physics in Medicine and Biology*, vol. 45, no. 8, p. 23732388, 2000.
- [21] C. Brebbia, J. C. F. Telles, and L. C. Wrobel, *Boundary Elements Techniques: Theory and Applications in Engineering*. Springer-Verlag, 1984.
- [22] L. P. S. Barra, F. C. Peters, and R. W. Santos, "Numerical experiments for the viability analysis of the determination of the cardiac ejection fraction by the electrical impedance tomography (in portuguese)," in *XXIX CILAMCE - Congresso Ibero Latino Americano de Métodos Computacionais em Engenharia*, Maceió, Brasil, 2008.
- [23] F. C. Peters and L. P. S. Barra, "A strategy for parametrization refinement in the solution of a geometric inverse problem," in *3rd Southern Conference on Computational Modeling - MCSUL*, 2009, pp. 136–142.
- [24] K. Madsen, H. B. Nielsen, and O. Tingleff, *Methods for Non-Linear Least Squares Problems (2nd ed.)*, 2004.
- [25] H. R. Lourenco, O. C. Martin, and T. Stutzle, "Iterated local search," *Social Science Research Network*, vol. 57, no. 513, p. 49, 2001.
- [26] W. Press, S. Teukolsky, W. Vetterling, and Flannery, *Numerical Recipes in Fortran77*. Cambridge University Press, 1986.
- [27] J. H. Holland, *Hidden order: how adaptation builds complexity*. Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc., 1995.
- [28] ———, *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1978.

- [29] J. Herskovits, “Feasible direction interior-point technique for nonlinear optimization,” *J. Optim. Theory Appl.*, vol. 99, no. 1, pp. 121–146, 1998.
- [30] J. L. Bentley, “Multidimensional binary search trees used for associative searching,” *Commun. ACM*, vol. 18, no. 9, pp. 509–517, 1975.
- [31] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed. The MIT Press, 2001.
- [32] F. C. Peters and L. P. S. Barra, “Some numerical results on the influence of measurement strategies and load patterns in the eit inverse problem,” *Journal of Physics: Conference Series*, vol. 224, no. 1, p. 012145, 2010.