NEW OBJECTIVE FUNCTION FOR ELECTRICAL IMAGE TOMOGRAPHY RECONSTRUCTION

Gustavo Vilela Momenté
Bruno Henrique Lobo Netto Peixoto
Computational Geometry Lab, Escola Politécnica da USP
gustavo.momente@usp.br, bruno.peixoto@usp.br

Marcos Sales Guerra Tsuzuki
Computational Geometry Lab, Escola Politécnica da USP
mtsuzuki@usp.br

Thiago De Castro Martins
Computational Geometry Lab, Escola Politécnica da USP
thiago@usp.br

Abstract. *The Electrical Impedance Tomography (EIT) reconstruction can be solved as an optimization problem. where the objective is to minimize the Euclidean distance between measured surface potentials and the respective currents on parametrized simulated impedance domain. The optimization problem can be solved by a combination of Simulated Annealing (SA) for optimization and Finite Element Method (FEM) for simulation of the impedance domain, but at a high computational cost, since SA requires an elevated number of objective function evaluations. This cost has been mitigated by using partial evaluations of the objective function, but this approach has scalability problems with the growing number of nodes in the FEM mesh. A new objective function is proposed here, based on the total error for overdetermined linear systems that can be partially evaluated without the scalability problems of the Euclidean Distance. Reconstructions were performed with the new objective function using 1st and 2nd order FEM models, and their performance was compared with traditional methods.*

Keywords: *Electrical Impedance Tomography, Inverse Problems, Least Square Problems*

1. INTRODUCTION

Electrical Image Tomography (EIT) is a diffuse imaging technique that obtains the electrical conductivity distribution inside an object from surface measurements. To obtain these measurements a set of electrodes is attached to the object surface, e.g. a human body, then electrical current is injected through these electrodes in multiple patterns and their electrical potentials are measured.

The task of reconstructing EIT images is often performed as an optimization process, where one attempts to maximize the *likeness* of a parametric simulated model. This likeness is often measured as the difference between the physical surface measurements and the same measurements on the simulated model (Herrera *et al.*, 2007; Mello *et al.*, 2008; Martins *et al.*, 2012). A common drawback of this approach is that the underlying optimization problem is non-linear, ill-conditioned (Mello *et al.*, 2008) and the objective function is expensive to evaluate, requiring the solution of full Finite Element Model (FEM) problems (Martins *et al.*, 2012). In this paper is proposed a new definition of likeness, measured as the total least square error of a (FEM) problem where the surface conditions are imposed to be exactly the same as those of the physical measurements. This new measure is potentially cheaper to evaluate, particularly when partial evaluations of the objective function are allowed.

This paper is structured as follows. Section 2presents the mathematical formulation of the EIT problem as well the approach method adopted to represent the studied physical domain (FEM). In this same section, the FEM formulation is also explained. Section 3.defines a new Objective function, explains how to use it in the studied problem and how preconditioning may improve the performance of convergence.

2. FORMULATION OF THE FORWARD PROBLEM

To solve the EIT image reconstruction problem as an optimization problem involves the simulation of a parametrized model in order to obtain the expected surface measurements from a conductivity distribution. This process (obtaining surface measurements from conductivity distribution) is called the EIT *forward problem*, in contrast with the original *inverse problem* (obtaining conductivity distribution from surface measurements).

Let Ω be a domain with conductivity σ and current $\mathbf{J}$ injected through boundary electrodes, the typical forward problem in EIT is find the potential distribution ϕ within Ω and in particular the resulting potentials at the measurement electrodes

ϕ_m. Low frequencies are utilized so the inductive and capacitive effects can be ignored. So with these quasi-static conditions, the solution of the forward problem consists in finding the solution to the following equation (Herrera, 2007):

$$\nabla\left(\sigma\nabla\phi\right) = 0 \tag{1}$$

with boundary conditions given by the currents J at the domain's frontier

$$\sigma\frac{\partial\phi}{\partial\hat{n}} = J \tag{2}$$

where $\hat{n}$ is the external normal. Here is adopted that positive values for J means that current is *exiting* the domain. Boundary conditions as the ones in Eq. (2) are called *Neumann boundary conditions*

2.1 Finite Element Method

Given a generic domain Ω, analytical and exact solutions for the Eq. (1) with the boundary condition represented by the Eq. (2) are not trivial. So, a numerical method is required. The adopted method in this paper is the Finite Element Method (FEM) (Castro (2006); Bathe (1995)).

The main idea of the method is the use of an approximate solution $\tilde{\phi}$, restricting the possible solutions to a space of functions generated by the *linear combination* of *shape functions*.

With this in mind, given $\Psi = [\psi_1, \psi_2, \cdots, \psi_n]$ a finite vector of *n* linearly independent functions, with $\psi_i \in C(\mathbb{R})$, i.e., functions from $\mathbb{R}^2$ to $\mathbb{R}$. Thus, the domain $\mathcal{A}$ of the explored functions by the FEM is given by

$$\mathcal{A} = \left\{u : \mathbb{R}^2 \to \mathbb{R} \mid u = x^T \cdot \Psi, x \in \mathbb{R}^n\right\} \tag{3}$$

The approximate function $\tilde{\phi}$ may be written by

$$\tilde{\phi} = \Phi^T\Psi \tag{4}$$

given $\Phi = [\phi_1, \phi_2, \cdots, \phi_n]$ the vector of coefficients that defines ϕ.

The variational form of the problem may derive the equations to FEM (Martins and de Sales Guerra Tsuzuki, 2011). It leads to the definition of the *stiffness matrix* $K(\sigma)$ and the *load vector* C of the problem defined by

$$\boldsymbol{K}\left(\sigma\right) = \int_\Omega \sigma\left(\nabla\Psi \otimes \nabla\Psi\right)\,\mathrm{d}\Omega \tag{5}$$

$$\boldsymbol{C} = \oint_{\partial\Omega} \Psi J \,\mathrm{d}s \tag{6}$$

By definition, $\boldsymbol{K}\left(\sigma\right)_{ij} = \int_\Omega \sigma\left(\nabla\psi_i \otimes \nabla\psi_j\right)\,\mathrm{d}\Omega$ is a symmetric and positive-definite matrix (Martins and de Sales Guerra Tsuzuki (2011) deduced that $\Phi^T\boldsymbol{K}\left(\sigma\right)\Phi^T$ corresponds to the first term of the variational approach, which is necessarily positive). The resulting linear system to the FEM is given by

$$\boldsymbol{K}\left(\sigma\right)\Phi - \boldsymbol{C} = 0 \tag{7}$$

The linear system in Eq. (7) is efficiently solved by numerical methods.

2.2 Shape Functions Deduction

The shape functions are generally polynomials. Their degree are kept the lower as possible to avoid high computational cost. The polynomials which base is $\mathcal{B} = \{1,\ x\ , y\}$ are frequently used in FEM problems, conciliating performance and accuracy. However, this subsection aims to deduce the shape function of the nodes of an element based on an arbitrary base $\mathcal{B}$, since the dimension of the base $\mathcal{B}$ is equal to the number of nodes of the element (nodes not coincidental). A possible approach is described by Cook *et al.* (2002) and Zienkievicz and Taylor (1988).

Given $\psi_i \in \mathcal{V}$, $\mathcal{V} = \{m_1(x, y), \ldots, m_n(x, y)\}$ a vector space, the Eq. 4 may be also written as:

$$\tilde{\phi} = c^T.\mathcal{V}(x, y) = \sum_{i=1}^{n} c_i m_i \tag{8}$$

where $c^T = [c_1 \ \dots \ c_n]$ e $\mathcal{V}(x, y) = [m_1 \ \dots \ m_n]^T$.

The function ψ_i is unitary to (x_i, y_i), i.e., in his own node, and null for all the others, by definition. Substituting this fact in the Eq. (8), the following matrix is obtained:

$$\phi_i = c^T.\mathcal{V}(x_i, y_i) \tag{9}$$

So

$$\Phi^T = c^T.Q \Rightarrow c^T = \Phi^T.Q^{-1} \tag{10}$$

where

$$Q = \left[\ V(x_1, y_1) \mid \cdots \mid V(x_n, y_n) \ \right]$$

By the previous definition, $\mathcal{V}(x, y)$ is a column-vector. An other representation to Q is

$$Q = \begin{bmatrix} m_1(x_1, y_1) & \cdots & m_1(x_n, y_n) \\ \vdots & \ddots & \vdots \\ m_n(x_1, y_1) & \cdots & m_n(x_n, y_n) \end{bmatrix}$$

Substituting the Eq. 10 in the Eq. (8), the following is obtained

$$\tilde{\phi} = \Phi^T.Q^{-1}.\mathcal{V} \tag{11}$$

Comparing the Eq. (4) and the Eq. (11), it's possible to conclude that the shape functions may be obtained by the equation below, where Q is defined by the Eq. (11):

$$\Psi = Q^{-1}.\mathcal{V} \tag{12}$$

2.3 Stiffness and Load Matrices Formulation

With the shape functions, obtained in Sec. (2.2), it's possible to formulate a procedure to construct the matrices involved to solve the problem, defined by the Eq. (5) and Eq. (6).

The stiffness matrix K has dimension $n \times n$ and its element K_{ij} is given by the following equation

$$\mathbf{K_{ij}}\left(\sigma\right) = \int_{\Omega} \sigma \left(\nabla\psi_i \otimes \nabla\psi_j\right) \, \mathrm{d}\Omega \tag{13}$$

The formulation below uses a second order polynomial to construct the stiffness matrix. Previous studies (Martins and de Sales Guerra Tsuzuki, 2011) adopt the polynomials of first order to reconstruct the images generated by the TIE problem. The following steps are deduced to a second order function, but may be replicated to shape functions of any order.

It's intuitive to induce that if ψ_i,$\psi_j \in P_2(R^2)$, $\nabla\psi_i \otimes \nabla\psi_j \in P_2(R^2)$, as shown below

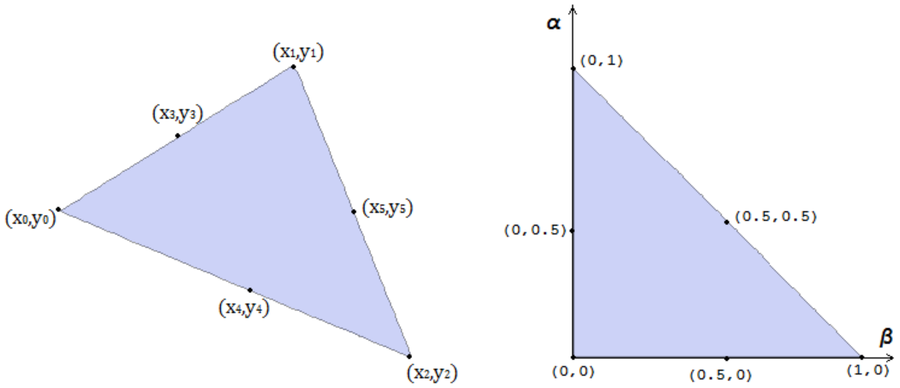
$$\nabla\psi_i \otimes \nabla\psi_j = c_0 + c_1.x + c_2.y + c_3.x^2 + c_4.xy + c_5.y^2 \tag{14}$$

with c_i values depending of the previous coefficients of the functions ψ_i and ψ_j.

The next step of the procedure consists in the calculation of the integrals of σ weighted by the monomials defined in the Eq. (14) and respective coefficients c_i.

$$\int_{\Omega} \sigma \left(\nabla\Psi_i \otimes \nabla\Psi_j\right) \, \mathrm{d}\Omega = \sum_{i=1}^{n} c_{i-1} \int_{\Omega} \sigma.m_i \, \mathrm{d}\Omega \tag{15}$$

Due to the generality of the domain (Fig. 1a), a linear transformation is necessary (Fig. 1b). The linear transformation may be defined by the equations below:



(a) Triangular element represented in the (x,y) space

(b) Triangular element represented in the (α, β) space

Figure 1: Elements represented by different coordinates, transformed by an linear transformation

$$x = (x_1 - x_0)\alpha + (x_2 - x_0)\beta + x_0 \tag{16}$$

$$y = (y_1 - y_0)\alpha + (y_2 - y_0)\beta + y_0 \tag{17}$$

The integrals in cartesian and local coordinates are correlated by the Eq. (18).

$$\int_{\Omega_{xy}} f(x,y)\,\mathrm{d}x\mathrm{d}y = \int_{\Omega_{\alpha\beta}} f(\alpha,\beta)\|J(x,y)\|\,\mathrm{d}\alpha\,\mathrm{d}\beta \tag{18}$$

$\|J(x,y)\| = 2A$ is the Jacobian of the relation $(x,y)/\,(\alpha,\beta)$ and A the area of the triangular element.

The parametrized domain $\Omega_{\alpha\beta}$ defines new upper and lower limits. The conductivity σ is parametrized by a first order function $\sigma = \sigma_0(1 - \alpha - \beta) + \sigma_1\alpha + \sigma_2\beta$.

The integrals are calculated at the beginning of the program, and the arising coefficients associated to the conductivity of the nodes are stored in a database structure (a linked list). Each integral of the pair i and j may be represented by

$$\mathbf{K_{ij}}\left(\sigma\right) = \int_{\Omega} \sigma\left(\nabla\psi_i \otimes \nabla\psi_j\right)\,\mathrm{d}\Omega = a_0^{ij}\sigma_0 + a_1^{ij}\sigma_1 + a_2^{ij}\sigma_2 \tag{19}$$

where

$$\begin{aligned} a_0^{ij} = A\Bigg[& \frac{c_0}{3} + \frac{1}{12}c_1\left(2x_0 + x_1 + x_2\right) + \frac{1}{12}c_2\left(2y_0 + y_1 + y_2\right) + \\ & \frac{1}{30}c_3\left(3x_0^2 + x_1^2 + x_1x_2 + x_2^2 + 2x_0\left(x_1 + x_2\right)\right) + \\ & \frac{1}{60}c_4\left(2x_0\left(3y_0 + y_1 + y_2\right) + x_1\left(2y_0 + 2y_1 + y_2\right) + x_2\left(2y_0 + y_1 + 2y_2\right)\right) + \\ & \frac{1}{30}c_5\left(3y_0^2 + y_1^2 + y_1y_2 + y_2^2 + 2y_0\left(y_1 + y_2\right)\right)\Bigg] \end{aligned} \tag{20}$$

$$\begin{aligned} a_1^{ij} = A\Bigg[& \frac{c_0}{3} + \frac{1}{12}c_1\left(x_0 + 2x_1 + x_2\right) + \frac{1}{12}c_2\left(y_0 + 2y_1 + y_2\right) + \\ & \frac{1}{30}c_3\left(x_0^2 + 3x_1^2 + 2x_1x_2 + x_2^2 + x_0\left(2x_1 + x_2\right)\right) + \\ & \frac{1}{60}c_4\left(x_0\left(2y_0 + 2y_1 + y_2\right) + 2x_1\left(y_0 + 3y_1 + y_2\right) + x_2\left(y_0 + 2\left(y_1 + y_2\right)\right)\right) + \\ & \frac{1}{30}c_5\left(y_0^2 + 3y_1^2 + 2y_1y_2 + y_2^2 + y_0\left(2y_1 + y_2\right)\right)\Bigg] \end{aligned} \tag{21}$$

$$\begin{aligned} a_2^{ij} = A \Bigg[& \frac{c_0}{3} + \frac{1}{12} c_1 \left(x_0 + x_1 + 2x_2\right) + \frac{1}{12} c_2 \left(y_0 + y_1 + 2y_2\right) + \\ & \frac{1}{30} c_3 \left(x_0^2 + x_1^2 + 2x_1x_2 + 3x_2^2 + x_0\left(x_1 + 2x_2\right)\right) + \\ & \frac{1}{30} c_5 \left(y_0^2 + y_1^2 + 2y_1y_2 + 3y_2^2 + y_0\left(y_1 + 2y_2\right)\right) + \\ & \frac{1}{60} c_4 \left(x_0\left(2y_0 + y_1 + 2y_2\right) + 2x_2\left(y_0 + y_1 + 3y_2\right) + x_1\left(y_0 + 2\left(y_1 + y_2\right)\right)\right) \Bigg] \end{aligned} \tag{22}$$

The procedure to build the whole matrix K consists in the following steps:

1. The matrix $K[n, n]$ is initialized entirely with null. In this project, this step is skipped due to the use of the library *Eigen* from *C++*;
2. For each element *e*, the 21 combinations are calculated[1], resulting in $21.3 = 63$ coefficients. They are stored in the database. If the coefficient associated to i and j already exists, it's added to the previous;
3. Calculate the coefficients for all the elements.

Each line *i* of the stiffness matrix represents a node and each column *j* is related to the influence of the node *j* on *i*. At the end of the process, the database contains the coefficients of each node of the matrix, which, in turn, is associated to the coefficients (and respective conductivities) of the nodes of the vertices that are linked to the analyzed line (node).

The load vector is easier to deduce. Based on the Eq. (6), the only terms that the integral are not null are those which the current of the experiment is passing through. These positions of the vector are equal to the associated current.

The solution of the problem depends on the definition of the ground electrode (Martins and de Sales Guerra Tsuzuki, 2011). Rearranging the lines and columns of the stiffness matrix, it's possible to eliminate it based on the fact that its voltage is the reference and null by definition.

3. TOTAL LEAST SQUARE ERROR AS AN OBJECTIVE FUNCTION

In early works by Martins and de Sales Guerra Tsuzuki (2011) was assumed that the total error of the CG algorithm was concentrated in the electrode nodes. This assumption is conservative as the ration between the number of electrodes and the total number of nodes tends to diminish.

Taking Eq. (7) and reordering the nodes to have the electrode nodes as the last elements in $\mathbf{\Phi}$, we have:

$$\begin{pmatrix} \mathbf{K}_{ii} & \mathbf{K}_{ic}^T \\ \mathbf{K}_{ic} & \mathbf{K}_{cc} \end{pmatrix} \begin{pmatrix} \mathbf{\Phi}_i \\ \mathbf{\Phi}_c \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{J}_c \end{pmatrix} \tag{23}$$

where $\mathbf{\Phi}_i$ is the internal tension vector, $\mathbf{\Phi}_c$ is the electrode tension vector, $\mathbf{K}_{ii}$, $\mathbf{K}_{ic}$ and $\mathbf{K}_{cc}$ are the blocks of the symmetric matrix $\mathbf{K}(\sigma)$ after the reordering and $\mathbf{J}_c$ the current injected in the electrodes.

Replacing $\mathbf{\Phi}_c$ by the measured tensions $\mathbf{\Phi}_m$ in Eq. (23) we have:

$$\hat{\mathbf{K}}\left(\sigma\right)\mathbf{\Phi} + \mathbf{e} = \hat{\mathbf{J}} \tag{24}$$

where

$$\begin{aligned} \hat{\mathbf{K}} &= \begin{pmatrix} \mathbf{K}_{ii} \\ \mathbf{K}_{ic} \end{pmatrix} \\ \hat{\mathbf{J}} &= \begin{pmatrix} -\mathbf{K}_{ic}^T\mathbf{\Phi}_m \\ \mathbf{J}_c - \mathbf{K}_{cc}\mathbf{\Phi}_m \end{pmatrix} \end{aligned} \tag{25}$$

and $\mathbf{e}$ an error vector added to the reduced system to make it consistent with the substitution $\mathbf{\Phi}_c \Rightarrow \mathbf{\Phi}_m$ (it can be noted that in general this substitution prompts to an overdetermined and inconsistent system). As this error theoretically goes to zero as $\mathbf{\Phi}_c$ approaches $\mathbf{\Phi}_m$, the minimum of his norm can be taken, respecting Eq. (24), as the measure of consistency between the simulated domain and the physical measures. So the new objective function is:

[1] As the stiffness matrix is symmetric, $K_{ij} = K_{ji}$ and the number of possibles combinations between the 6 shape functions in the node is $\frac{6!}{2!(6-2)!} = 21$

$$E(\sigma) = \sqrt{\sum_i D^i(\sigma)^2} \tag{26}$$

$$D^i = \min_{\mathbf{\Phi}} \sqrt{\mathbf{e}^T\mathbf{e}} : \hat{\mathbf{K}}\left(\sigma\right)\mathbf{\Phi} + \mathbf{e} = \hat{\mathbf{J}}_i. \tag{27}$$

The minimization problem in Eq. (27) is a typical least squares problem whose solution is given by:

$$D^2 = \hat{\mathbf{J}}^T\hat{\mathbf{J}} - \hat{\mathbf{J}}^T\hat{\mathbf{K}}\left(\hat{\mathbf{K}}^T\hat{\mathbf{K}}\right)^{-1}\hat{\mathbf{K}}^T\hat{\mathbf{J}}. \tag{28}$$

This is a measure of the error added to Eq. (24). It is observed that it depends only on $\hat{\mathbf{K}}(\sigma)$.

Golub and Meurant (1994) studied how to efficiently and iteratively calculate quadratic forms as $\mathbf{v}^T f(\mathbf{A})\mathbf{v}$, where $\mathbf{v}$ is a vector, $\mathbf{A}$ is a symmetric matrix and $f(x)$ is an analytic function.

In fact, if $\mathbf{A}$ is symmetric, positive and definite, so,$\mathbf{v}^T f(\mathbf{A})\mathbf{v}$ can be expressed as a Riemman-Stieltjes integral $\int_a^b f\left(\lambda\right)\,\mathrm{d}\alpha(\lambda)$ for a measure $\alpha(\lambda)$ obtained from $\mathbf{v}$ and the spectral decomposition of $\mathbf{A}$. That integral in turn can be approached by Gaussian Quadrature rules whose coefficients can be derived from Lanczos algorithm (Meurant, 2006) when the start vector is $\mathbf{v}$.

The result of this approach is an iterative algorithm that produces upper and lower boundaries to $\mathbf{v}^T f(\mathbf{A})\mathbf{v}$. This can be applied to the calculations of Eq. (28) and consequently in iteratively obtaining boundaries to the objective function Eq. (26)

Golub and Welsch (1969) shows that for $f(x) = x^{-1}$

$$\mathbf{v}^T f(\mathbf{A})\mathbf{v} \approx G_k = \|\mathbf{v}\|_2^2 \mathbf{e}_1^T \mathbf{T}_k^{-1} \mathbf{e}_1 \tag{29}$$

where G_k is the Gaussian quadrature and $\mathbf{T}_k$ are the matrices from Lanczos algorithm when the starting vector is $\mathbf{v}$. To obtain the Gauss-Radau quadrature rule one must modify $\mathbf{T}_k$ in order to add to its eigenvalues the lowest one of $\mathbf{A}$, obtaining this way $\tilde{\mathbf{T}}_k$. So the Gauss-Radau quadrature R_k can be expressed as:

$$\mathbf{v}^T f(\mathbf{A})\mathbf{v} \approx R_k = \|\mathbf{v}\|_2^2 \mathbf{e}_1^T \tilde{\mathbf{T}_k}^{-1} \mathbf{e}_1 \tag{30}$$

It's important to notice that for our $f(x)$, Gauss quadrature rule gives an upper boundary and Gauss-Radau quadrature rule gives a lower boundary for $\mathbf{v}^T f(\mathbf{A})\mathbf{v}$ (Golub and Meurant, 1994).

3.1 Taking Advantage of the New Objective Function Configuration

The general method to apply Gaussian quadrature rules was discussed in the previous section. But some modifications can be made to enhance the process's performance.

In (Golub and von Matt, 1979), some modifications were proposed to enhance the evaluation of form as the ones in Eq. (28), that diminish the precision losses that one have when directly evaluating $\hat{\mathbf{K}}\hat{\mathbf{K}}^T$. The new calculation is made with the use of the "Lanczos Bidiagonalization II" algorithm by (Golub and von Matt, 1979).

Algorithm 1: Lanczos Bidiagonalization II adapted to Eq. (28)

```
$p_0 := \hat{\mathbf{J}}/\|\hat{\mathbf{J}}\|_2$;
$\mathbf{s}_0 := \hat{\mathbf{K}}^T\mathbf{p}_0$ ;
$\gamma_1 := \|\mathbf{s}_0\|_2$;
$\mathbf{q}_0 := \mathbf{s}_0/\gamma_1$ ;
$\mathbf{r}_1 := \hat{\mathbf{K}}\mathbf{q}_0 - \gamma_1\mathbf{p}_0$;
$\delta_1 := \|\mathbf{r}_1\|_2$ ;
$\mathbf{p}_1 := \mathbf{r}_1/\delta_1$ ;
for $k := 2$ to $n$ do
    $\mathbf{s}_{k-1} := \hat{\mathbf{K}}^T\mathbf{p}_{k-1} - \delta_{k-1}\mathbf{q}_{k-2}$;
    $\gamma_k := \|\mathbf{s}_{k-1}\|_2$;
    $\mathbf{q}_{k-1} := \mathbf{s}_{k-1}/\gamma_k$;
    $\mathbf{r}_k := \hat{\mathbf{K}}\mathbf{q}_{k-1} - \gamma_k\mathbf{p}_{k-1}$ ;
    $\delta_k := \|\mathbf{r}_k\|_2$ ;
    $\mathbf{p}_k := \mathbf{r}_k/\delta_k$;
```

This algorithm builds matrices

$$\mathbf{B}_k = \begin{pmatrix} \gamma_1 & & \\ \delta_1 & \ddots & \\ & \ddots & \gamma_k \\ & & \delta_k \end{pmatrix} \tag{31}$$

It's easy to observe that $\mathbf{B}_k^T \mathbf{B}_k$ is equal to $\mathbf{T}_k$ in the previous section (Golub and von Matt, 1979).

Therefore, the final form of Gaussian quadrature is

$$G_k = \|\hat{\mathbf{K}}^T \hat{\mathbf{J}}\|_2^2 g_k \tag{32}$$

with $g_k = \mathbf{e}_1^T (\mathbf{B}_k^T \mathbf{B}_k)^{-1} \mathbf{e}_1$.

It's easy to see that g_k is the first element of $\left(\mathbf{B}_k^T \mathbf{B}_k\right)^{-1}$. Now, using the fact that $\mathbf{B}_k^T \mathbf{B}_k$ is a tridiagonal symmetric matrix, a Cholesky-like factorization can be written $\mathbf{B}_k^T \mathbf{B}_k = \mathbf{L}_k^D \mathbf{\Delta}_k^{-1} (\mathbf{L}_k^D)^T$ with

$$\mathbf{L}_k^D = \begin{pmatrix} \psi_1 & & & \\ \beta_1 & \psi_2 & & \\ & \ddots & \ddots & \\ & & \beta_{k-1} & \psi_k \end{pmatrix} \tag{33}$$

and $\mathbf{\Delta}_k$ a diagonal matrix with elements ψ_j. So we have the following algorithm adapted from (Golub and Meurant, 2010) to obtain the factorization and g_k.

Algorithm 2: Obtaining g_k

```
$\psi_1 := \gamma_1^2 + \delta_1^2$;
$\beta_1 := \gamma_2 \delta_1$;
$g_{k,1} := 1/\psi_1^2$ ;
$c_1 := 1$ ;
for $i := 2$ to $k$ do
    $\pi_i := 1/\psi_i$;
    $\beta_i = \gamma_{i+1} \delta_i$;
    $t := \beta_{i-1}^2 \pi_{i-1}$;
    $\psi_i := \gamma_i^2 + \delta_i^2 - t$;
    $c_i := c_{i-1} \pi_{i-1} t$ ;
    $g_{k,i} := g_{k,i-1} + c_i \pi_i$ ;
```

With this we can obtain the upper limit to our function $\hat{\mathbf{K}}\hat{\mathbf{K}}^T$. Now we will obtain the lower limit using the Gauss-Radau quadrature rule. To do so, one must first do a sequence of given rotations to transform $\mathbf{B}_k$ in a upper bidiagonal matrix with the following form:

$$\mathbf{G}^T \mathbf{B}_k = \mathbf{U}_k = \begin{pmatrix} \tau_1 & \theta_1 & & \\ & \ddots & \ddots & \\ & & \ddots & \theta_{k-1} \\ & & & \tau_k \end{pmatrix}. \tag{34}$$

After this, we must modify $\mathbf{U}_k$ to $\tilde{\mathbf{U}}_k$. To do this, we set τ_k to be equal to the lowest eigenvalue of $\mathbf{B}_k^T \mathbf{U}_k$ the we call a. So $\tilde{\mathbf{T}}_k$ is given by $\tilde{\mathbf{U}}_k^T \tilde{\mathbf{U}}_k$.

Therefore, the final form of Gauss-Radau quadrature is:

$$R_k = \|\hat{\mathbf{K}}^T \hat{\mathbf{J}}\|_2^2 r_k \tag{35}$$

with $r_k = \mathbf{e}_1^T (\tilde{\mathbf{U}}_k^T \tilde{\mathbf{U}}_k)^{-1} \mathbf{e}_1$.

We stay with the problem of obtaining r_k, initially on could think in applying Alg. (2) to $\tilde{\mathbf{U}}_\mathbf{k}$. But it can be shown that this can be incorporated in that algorithm with a simple change. First, let's consider

$$\mathbf{U}_k^T \mathbf{U}_k = \mathbf{B}_k^T \mathbf{G} \mathbf{G}^T \mathbf{B}_k \tag{36}$$

as Givens rotations are orthogonal (Golub and Van Loan, 1996), therefore, $\mathbf{G}\mathbf{G}^T = \mathbf{I}$. So,

$$\mathbf{U}_k^T \mathbf{U}_k = \mathbf{B}_k^T \mathbf{B}_k = \mathbf{L}_k^T \mathbf{\Delta}_k^{-1} \mathbf{L}_k \tag{37}$$

As the only difference between $\mathbf{U}_k$ and $\tilde{\mathbf{U}}_k$ is in the last element we won't need to apply all Alg. 2 to obtain r_k, but only add the following step to it in the end of the for loop:

$$r_i := g_{i-1} + c_i/(a - t); \tag{38}$$

The last problem that remains is obtain the first eigenvalue of $\hat{\mathbf{K}}^T \hat{\mathbf{K}}$. In (Golub and Meurant, 1996) is suggested that the lower eigenvalue of $\mathbf{T}_k$ can be used as an approximation to $\hat{\mathbf{K}}^T \hat{\mathbf{K}}$'s lower. So, one can use the Inverse Power Iteration algorithm to obtain the lower eigenvalue as described in (Panju, 2011).

Now we can compute upper and lower bounds to Eq. (28). These bounds get tighter as more iterations are done. So, they can be used as a stopping criteria to simulated annealing proposed by Martins *et al.* (2011).

3.2 Preconditioning

Lanczos algorithm is notoriously sensitive to the matrix condition number (Golub and Van Loan, 1996). Moreover, matrix like $\hat{\mathbf{K}}^T \hat{\mathbf{K}}$ doesn't have good conditioning.

Then, preconditioning can be used, as is done to enhance the convergence ratio in the resolution of Conjugated Gradients based on Lanczos Algorithm (Golub and Van Loan, 1996), to improve the condition number of the relevant matrices for our problem.

So, if we have a matrix

$$\mathbf{M} = \mathbf{P}^T \mathbf{P} \approx \hat{\mathbf{K}}^T \hat{\mathbf{K}} \tag{39}$$

that is an approximation of the original matrix, then, $\mathbf{P}^{-T} \hat{\mathbf{K}}^T \hat{\mathbf{K}} \mathbf{P}^{-1}$ has better conditioning than $\hat{\mathbf{K}}^T \hat{\mathbf{K}}$ (Saad, 2003). So, defining $\check{\mathbf{K}} = \hat{\mathbf{K}} \mathbf{P}^{-1}$, that can be rewritten as $\hat{\mathbf{K}} = \check{\mathbf{K}} \mathbf{P}$, we have from Eq. (28):

$$\begin{aligned}
D^2 &= \hat{\mathbf{J}}^T \hat{\mathbf{J}} - \hat{\mathbf{J}}^T \hat{\mathbf{K}} \left(\hat{\mathbf{K}}^T \hat{\mathbf{K}} \right)^{-1} \hat{\mathbf{K}}^T \hat{\mathbf{J}} \\
D^2 &= \hat{\mathbf{J}}^T \hat{\mathbf{J}} - \hat{\mathbf{J}}^T \check{\mathbf{K}} \mathbf{P} \left(\mathbf{P}^T \check{\mathbf{K}}^T \check{\mathbf{K}} \mathbf{P} \right)^{-1} \mathbf{P}^T \check{\mathbf{K}}^T \hat{\mathbf{J}} \\
D^2 &= \hat{\mathbf{J}}^T \hat{\mathbf{J}} - \hat{\mathbf{J}}^T \check{\mathbf{K}} \mathbf{P} \mathbf{P}^{-T} \left(\check{\mathbf{K}}^T \check{\mathbf{K}} \right)^{-1} \mathbf{P}^{-1} \mathbf{P}^T \check{\mathbf{K}}^T \hat{\mathbf{J}}
\end{aligned} \tag{40}$$

so, if we consider only symmetric $\mathbf{P}$ we have:

$$D^2 = \hat{\mathbf{J}}^T \hat{\mathbf{J}} - \hat{\mathbf{J}}^T \check{\mathbf{K}} \left(\check{\mathbf{K}}^T \check{\mathbf{K}} \right)^{-1} \check{\mathbf{K}}^T \hat{\mathbf{J}} \tag{41}$$

in other words, we obtain the same results applying Eq.(28) at $\hat{\mathbf{K}}$ or in $\check{\mathbf{K}}$. So nothing must be changed in the preview methods, only apply them in $\check{\mathbf{K}}$.

As $\hat{\mathbf{K}}$ was defined in section 3we have $\hat{\mathbf{K}}^T \hat{\mathbf{K}} = \mathbf{K}_{ii}^T \mathbf{K}_{ii} + \mathbf{K}_{ic}^T \mathbf{K}_{ic}$. As $\mathbf{K}_{ic}$ has only information about the electrodes nodes, it's smaller and sparser than $\mathbf{K}_{ii}$. And, as decompositions for non-square matrices as $\hat{\mathbf{K}}$, we can consider $\mathbf{K}_{ii}^T \mathbf{K}_{ii}$ as an approximation to $\hat{\mathbf{K}}^T \hat{\mathbf{K}}$ during the incomplete factorization and preconditioning process.

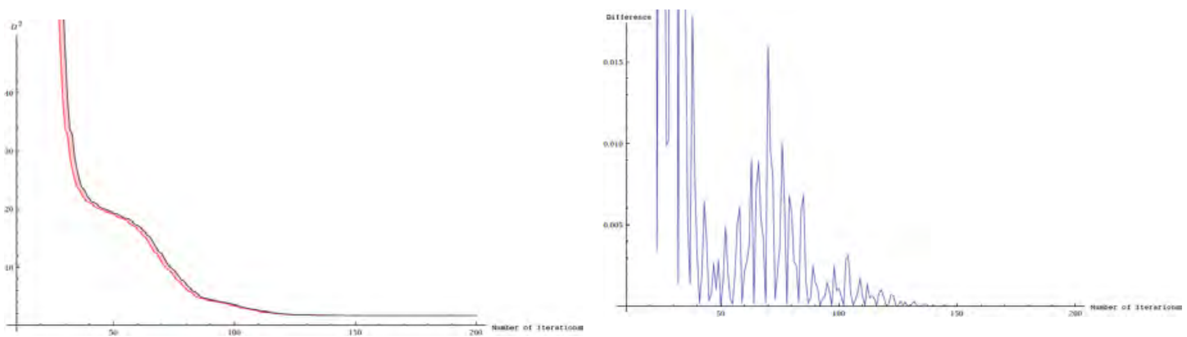
So, if we look for decompositions to $\mathbf{K}_{ii}$, for example, $\mathbf{K}_{ii} \approx \mathbf{L}\mathbf{L}^T = \mathbf{P}$ we can build the preconditioned matrix $\check{\mathbf{K}}$. We tested different incomplete factorization techniques and they will be discussed in Section 4.

4. RESULTS

The process discussed in the preview sections were first implemented using Wolfram Mathematica® Wolfram Research, Inc. (2011) software then incorporated to the code developed by Martins *et al.* (2011). First the convergence of the Gauss and Gauss-Radau quadratures rules was studied. To do this we used a random current pattern and the convergence with this data can be observed in Fig. 2a. As we can see they comportment was as expected Gauss quadrature provided a lower boundary and Gauss-Radau an upper.

Also, we didn't evaluate the lower eigenvalue a in all iterations, but first, we calculated $\mathbf{B}_5 0$, then we obtained the lower eigenvalue using Inverse Power Iteration as described in (Panju, 2011). This value was used to evaluate Gauss-Radau quadrature. This is valid because after a certain point of $\mathbf{B}_k$ growing the lower eigenvalue variation is so small the evaluation costs is not justified.

Also, as the lower eigenvalue is very small, we made some tests assuming $a = 0$, so reducing even more computational costs, the results are in Fig. 2b.



(a) Results obtained using Gauss quadrature(in red) and using Gauss-Radau quadrature(in black)

(b) Comparative between results obtained using Gauss-Radau quadrature rule with the evaluation of the lower eigenvalue and assuming it equal zero

Figure 2: Guassian quadrature rule results

After theses tests, we started to evaluate different incomplete factorization approaches. To evaluate the effectiveness of the incomplete factorization we considered the following criteria: the condition number and eigenvalue dispersion of $\check{\mathbf{K}}^T\check{\mathbf{K}}$ and the incomplete factorization density.

Initially, without any preconditioning $\hat{\mathbf{K}}^T\hat{\mathbf{K}}$ has the dispersion and condition number represented in Fig. 3a.

Initially we evaluated the Incomplete LQ factorization as proposed by Saad and Sosonkina (2001). In our implementation we adopted the $\mathbf{L}$ matrix zero patter to be equal as the one of $\mathbf{K}_{ii}$. And for $\mathbf{Q}$ we stored only the n greater elements in each line. To evaluate the influence of the number of stored elements we made tests for $n = 10^{s+1}$ with s ranging from zero to one. The obtained results are in Fig. 3b,3c,3b.

It is observed that the reconstruction present in Fig. 5c it's more loyal to reality than the one present in Fig. 5b. But it presents some reconstruction artefacts in the top where only liquid was present. Those artefacts come from the imposition of a continuum impedance domain that doesn't necessarily occurs, resulting, in artefacts like those.

As we can see in Fig. 3 the preconditioning can enhance the matrix condition number, but it generates matrix denser than the original ones. Also, as we applied the factorization to $\mathbf{K}_{ii}$ we expected that $\mathbf{P} = \mathbf{LQ}$ were symmetric or close to a symmetric matrix, but unfortunately this didn't happen, so we can conclude that this factorization don't give good results to our problem.

We also analysed the Balanced Incomplete factorization(BIF) proposed by Mas and TŪMA (2008).We used drop tolerance equal to unity to don't allow very small values. We also evaluated the influence of the shift parameter s in the range 0.1 to 1.1. As we can see in Fig. 3f the bigger the shift the better condition is the resulting matrix, but as we can observe in Fig. 3e it has low influence in the eigenvalue dispersion that turn out to be more disperse than in the original matrix.

We also, evaluated Incomplete Cholesky factorization as proposed in (Golub and Van Loan, 1996). To do so we applied it to a matrix $\hat{\mathbf{K}}_{ii} = \mathbf{K}_{ii} + \alpha\mathbf{I}$ with α ranging from zero to one. As we can see in Fig. 3h the values are more compacted around the origin, chiefly for $\alpha \geqslant 0.5$, that is yet a small value, then, the factorization maintains a much information about the original matrix. And generally the matrix has better condition number than the original as can be observed in Fig. 3i.

Then, after evaluating all this candidates we opted to use IC with a 0.5 shift, this was done considering the following aspects:

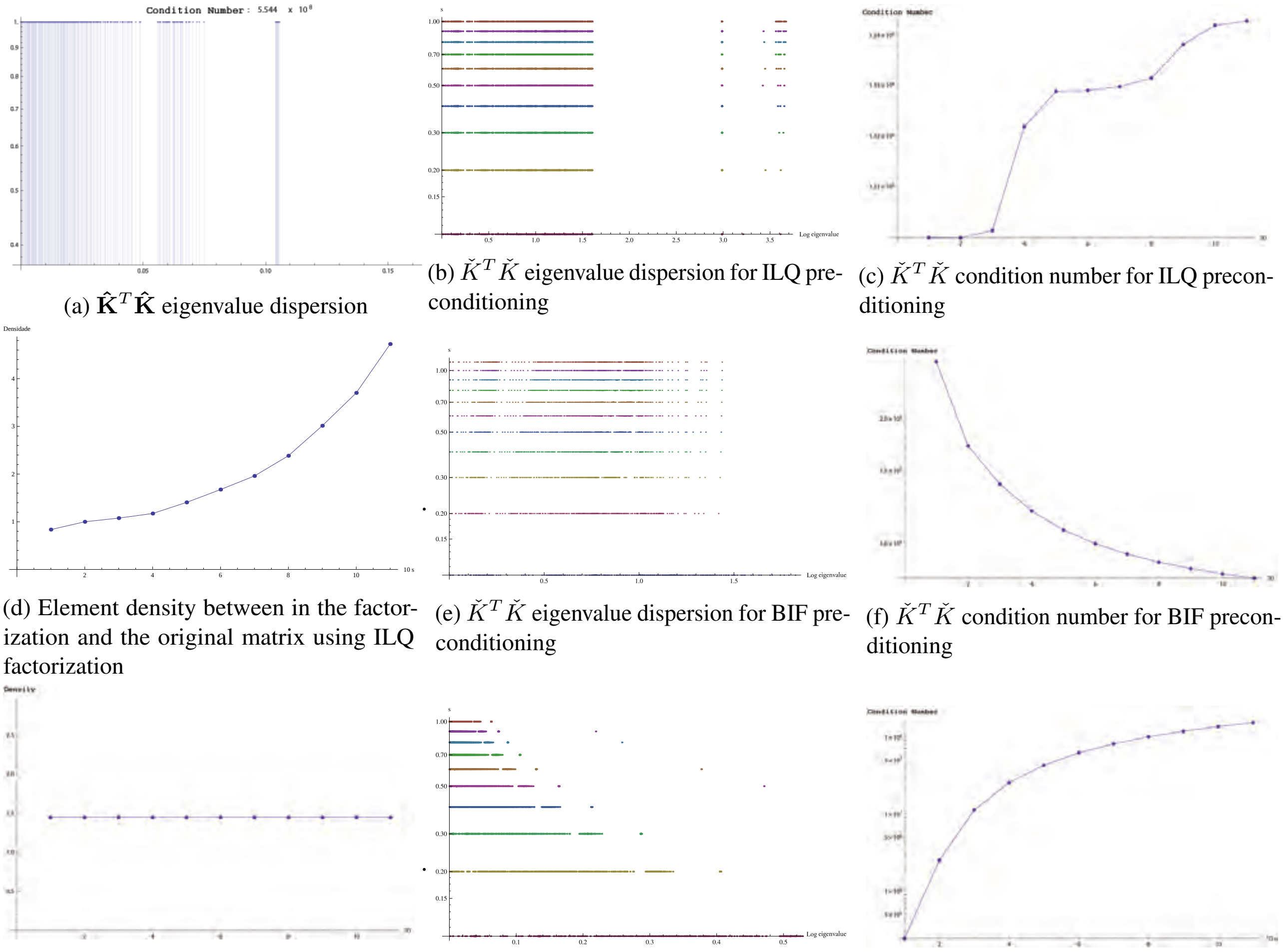
- This preconditioner let the eigenvalues closer to the origin, justifying even more using Gauss-Radau with the lower eigenvalue set as zero. This removes a considerable amount of calculations in the quadrature evaluation procedure.

- To factorize the matrix, we first store the lower triangular matrix structure of the matrix to be factorized in the $\mathbf{L}$ matrix. And, as new elements aren't inserted in places where there was a zero before, the access and modification can be implemented efficiently using C++ template library *Eigen*.

- This factorization effectively reduce the number of iterations in Gaussian quadrature rules to obtain convergence as can be seen in Fig. 4. And also has better conditioning that the original matrix.

The experimental data collected by Martins *et al.* in the arrangement show in Fig. 5a was used to perform reconstructions, both with the old and new method. Two meshes were used in the reconstruction, a first-order one with 1047 nodes and a second order mesh with 4289 nodes.

It is observed that the reconstruction present in Fig. 5c it's arguably as close to reality as the one present in Fig. 5b. Yet it presents some reconstruction artifacts in the top where only liquid was present. Those artifacts come from the imposition of a constant impedance domain across the outer ring that doesn't necessarily occurs, resulting, in artifacts like those.



(a) $\hat{\mathbf{K}}^T\hat{\mathbf{K}}$ eigenvalue dispersion

(b) $\check{K}^T\check{K}$ eigenvalue dispersion for ILQ preconditioning

(c) $\check{K}^T\check{K}$ condition number for ILQ preconditioning

(d) Element density between in the factorization and the original matrix using ILQ factorization

(e) $\check{K}^T\check{K}$ eigenvalue dispersion for BIF preconditioning

(f) $\check{K}^T\check{K}$ condition number for BIF preconditioning

(g) Element density between in the factorization and the original matrix using BIF factorization

(h) $\check{K}^T\check{K}$ eigenvalue dispersion for IC preconditioning

(i) $\check{K}^T\check{K}$ condition number for IC preconditioning

Figure 3: Preconditioning techniques comparative

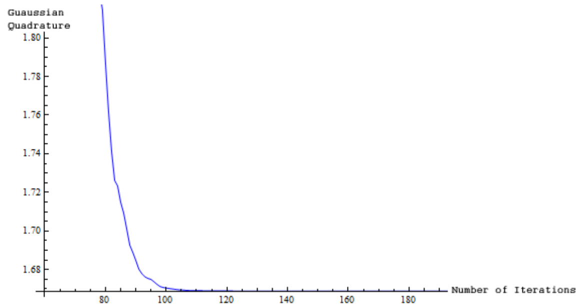
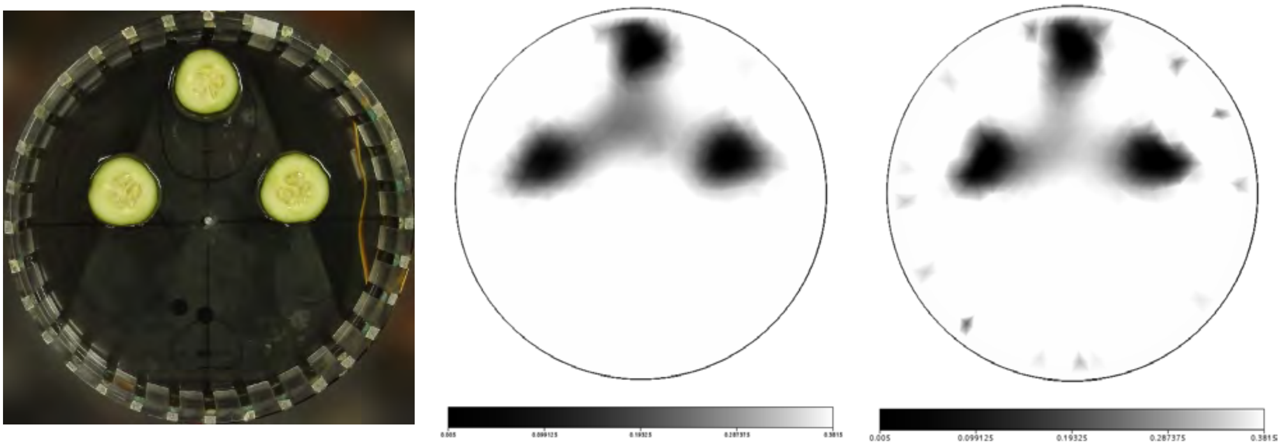


Figure 4: Gaussian quadrature convergence when applied to the preconditioned matrix



(a) Apparatus and configuration used to obtain the data

(b) Reconstruction using the preview method by Martins *et al.* (2012)

(c) Reconstruction obtained using the methos proposed here

Figure 5: Experimental apparatus and reconstructions using different methods

On the performance, the new algorithm obtains the image in roughly 2/3 of the time of the method in (Martins *et al.*, 2012) when using first order elements. For second order elements, the process was more than 20 times *slower* without any improvement in image quality, becoming inviable. Reasons for this performance loss are still being investigated.

5. CONCLUSION

This work presented a novel approach to EIT image reconstruction with partial evaluation using a new objective function. This new objective function, based on the total error of overdetermined linear systems, solves the scalability problems present in (Martins *et al.*, 2012). As the new method is based on a Lanczos algorithm, its performance is strongly dependent of the system matrix condition number, several preconditioning techniques were studied. Reconstructions were performed on EIT experimental data, both with the new method and the one proposed in (Martins *et al.*, 2012). Total reconstruction time was reduced in about 33% for meshes of first-order with similar image quality, confirming the hypothesis of better scalability of the new method.

6. ACKNOWLEDGEMENTS

This work was supported by FAPESP (Grants 2009/07173-2 and 2010/19380-0). Gustavo Vilela Momenté was supporte by FAPESP under Grant 2012;04902-6. Bruno Peixoto was sponsored by SESU under the PET program. Marcos de Sales Guerra Tsuzuki and Thiago de Castro Martins were partially supported by CNPq (Grants 06.415/20127 and 309.570/2010-7).

7. REFERENCES

Bathe, K.J., 1995. *Finite Element Procedures (Part 1-2)*. Prentice Hall. ISBN 0133014584.

Castro, Sobrinho, A.d.S., 2006. *Introdução ao Método dos Elementos Finitos*. Ltda., Editora Ciência Moderna, Rio de Janeiro. ISBN 85-7393-512-X.

Cook, R.D., Malkus, D.S., Plesha, M.E. and Witt, R.J., 2002. *Concepts and Applications of Finite Element Analysis*. John Wiley & Sons, Inc., 4th edition.

Golub, G.H. and Meurant, G., 1994. "Matrices, moments and quadrature". In *Numerical Analysis*, D.F Griffiths and G.A. Watson Eds., Vol. 303 of *Pitman Research Notes in Mathematics*, pp. 105–156.

Golub, G.H. and von Matt, U., 1979. "Generalized Cross-Validation for Large Scale Problems". In *Recent advances in total least squares techniques and errors-in-variable modelling*, S. van Huffel Ed., SIAM, Philadelphia, pp. 139–148.

Golub, G.H. and Welsch, J.H., 1969. "Calculation of gauss quadrature rules". Technical report.

Golub, G. and Meurant, G., 1996. "Matrices, moments and quadrature ii or how to compute the norm of the error in iterative methods". *BIT*, Vol. 37, pp. 687–705.

Golub, G. and Meurant, G., 2010. *Matrices, Moments and Quadrature with Applications*. Princeton Series in Applied Mathematics. Princeton University Press. ISBN 9780691143415.

Golub, G. and Van Loan, C., 1996. *Matrix Computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press. ISBN 9780801854149. URL `http://books.google.com.br/books?id=mlOa7wPX6OYC`.

Herrera, C.N.L., 2007. *Algoritmo de tomografia por impedância elétrica baseado em simulated annealing*. Master's thesis, Escola Politécnica da USP.

Herrera, C.N.L., Vallejo, M.F.M., Moura, F.S., Aya, J.C.C. and Lima, R.G., 2007. "Electrical impedance tomography algorithm using simulated annealing search method". In *Proceedings of International Congress Of Mechanical Engineering*. ABCM, Brasília.

Martins, T., Camargo, E., Lima, R.G., Amato, M.B.P. and Tsuzuki, M.S.G., 2011. "Electrical impedance tomography reconstruction through simulated annealing with incomplete evaluation of the objective function." In *Proceedings of the 33rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS 11)*. IEEE EMBS, Boston.

Martins, T.C., de Camargo, E.D.L.B., Lima, R.G., Amato, M.B.P. and de Sales Guerra Tsuzuki, M., 2012. "Image reconstruction using interval simulated annealing in electrical impedance tomography." *IEEE Trans. Biomed. Engineering*, Vol. 59, No. 7, pp. 1861–1870.

Martins, T.C. and de Sales Guerra Tsuzuki, M., 2011. "Interval simulated annealing applied to electrical impedance tomography". In ABCM, ed., *Proceedings of the 21th International Congress of Mechanical Engineering (COBEM2011*. Natal, Brazil.

Mas, Rafael Bru, J.M.J. and TŮMA, M., 2008. "Balanced incomplete factorization". *SIAM J. Sci. Comput.*, Vol. 30, No. 5, pp. 2302–2318. ISSN 1064-8275. doi:10.1137/070696088. URL `http://dx.doi.org/10.1137/070696088`.

Mello, L.A.M., de Lima, C.R., Amato, M.B.P., Lima, R.G. and Silva, E.C.N., 2008. "Three-dimensional electrical impedance tomography: a topology optimization approach." *IEEE transactions on bio-medical engineering*, Vol. 55, No. 2 Pt 1, pp. 531–40.

Meurant, G., 2006. *The Lanczos and Conjugate Gradient Algorithms: From Theory to Finite Precision Computations*. Society for Industrial and Applied Mathematics.

Panju, M., 2011. "Iterative methods for computing eigenvalues and eigenvectors". *The Waterloo Mathematics Review*, Vol. 1, pp. 9–18.

Saad, Y., 2003. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2nd edition. ISBN 0898715342.

Saad, Y. and Sosonkina, M., 2001. "Enhanced preconditioners for large sparse least squares problems". Technical Report Preprint umsi-2001-1, Minnesota Supercomputer Institute, University of Minnesota.

Wolfram Research, Inc., 2011. *Mathematica*. Wolfram Research, Inc., Champaign, Illinois EUA, version 8.0 edition.

Zienkievicz, O.C. and Taylor, R.L., 1988. *The Finite Element Method*. McGraw-Hill, 4th edition.

8. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.