NEW OBJECTIVE FUNCTION FOR ELECTRICAL IMAGE TOMOGRAPHY RECONSTRUCTION

Gustavo Vilela Momenté
Bruno Henrique Lobo Neto 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 2 presents 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 $\Omega$ be a domain with conductivity $\sigma$ and current $J$ injected through boundary electrodes, the typical forward problem in EIT is find the potential distribution $\phi$ within $\Omega$ and in particular the resulting potentials at the measurement electrodes.
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):

$$\nabla (\sigma \nabla \phi) = 0$$

(1)

with boundary conditions given by the currents \( J \) at the domain’s frontier

$$\sigma \frac{\partial \phi}{\partial n} = J$$

(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 \( \Omega \), 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

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

(3)

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

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

(4)

given \( \Phi = [\phi_1, \phi_2, \cdots, \phi_n] \) the vector of coefficients that defines \( \phi \).

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

$$K(\sigma) = \int_{\Omega} \sigma (\nabla \Psi \otimes \nabla \Psi) \, d\Omega$$

(5)

$$C = \oint_{\partial \Omega} \Psi J \, ds$$

(6)

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

$$K(\sigma) \Phi - C = 0$$

(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 Zienkiewicz 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:

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

(8)
where $c^T = [c_1 \ldots c_n]$ e $\mathbf{V}(x, y) = [m_1 \ldots m_n]^T$.

The function $\psi_i$ 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:

$$\phi_i = c^T \mathbf{V}(x_i, y_i)$$

So

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

where

$$Q = \begin{bmatrix} V(x_1, y_1) & \cdots & V(x_n, y_n) \end{bmatrix}$$

By the previous definition, $\mathbf{V}(x, y)$ 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

$$\tilde{\phi} = \Phi^T Q^{-1} \mathbf{V}$$

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):

$$\Psi = Q^{-1} \mathbf{V}$$

### 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

$$K_{ij}(\sigma) = \int_{\Omega} \sigma (\nabla \psi_i \otimes \nabla \psi_j) \, d\Omega$$

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 $\psi_i, \psi_j \in P_2(R^2)$, $\nabla \psi_i \otimes \nabla \psi_j \in P_2(R^2)$, as shown below

$$\nabla \psi_i \otimes \nabla \psi_j = c_0 + c_1 x + c_2 y + c_3 x^2 + c_4 xy + c_5 y^2$$

with $c_i$ values depending of the previous coefficients of the functions $\psi_i$ and $\psi_j$.

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

$$\int_{\Omega} \sigma (\nabla \psi_i \otimes \nabla \psi_j) \, d\Omega = \sum_{i=1}^{n} c_{i-1} \int_{\Omega} \sigma m_i \, d\Omega$$

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

$$\int_{\Omega_{xy}} f(x, y) \, dx \, dy = \int_{\Omega_{\alpha, \beta}} f(\alpha, \beta) \|J(x, y)\| \, d\alpha \, d\beta$$ (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 $\sigma$ 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

$$K_{ij}(\sigma) = \int_{\Omega} \sigma (\nabla \psi_i \otimes \nabla \psi_j) \, d\Omega = a_{ij}^{ij} \sigma_0 + a_{ij}^{ij} \sigma_1 + a_{ij}^{ij} \sigma_2$$ (19)

where

$$a_{ij}^{ij} = A \left[ \frac{c_0}{3} + \frac{1}{12} c_1 (2x_0 + x_1 + x_2) + \frac{1}{12} c_2 (2y_0 + y_1 + y_2) + \frac{1}{30} c_3 (3x_0^2 + x_1^2 + x_2^2 + 2x_0 (x_1 + x_2)) + \frac{1}{60} c_4 (2x_0 (3y_0 + y_1 + y_2) + x_1 (2y_0 + 2y_1 + 2y_2) + x_2 (2y_0 + y_1 + 2y_2)) + \frac{1}{30} c_5 (3y_0^2 + y_1^2 + y_2^2 + 2y_0 (y_1 + y_2)) \right]$$ (20)

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

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.

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.

\[ \Phi \text{ zero as } \]

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 
\[ \begin{pmatrix} 3 \times 3 \end{pmatrix} \times \begin{pmatrix} 3 \times 3 \end{pmatrix} = 21 \]

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

1. The matrix \( K'[m, n] \) 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 \( 21 \times 3 = 63 \) 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 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.

\[ \begin{align*} 
\Phi_i &= 21 \\
\Phi_m &= 30 \\
\end{align*} \]

\[ \begin{align*} 
\hat{\Phi} &= \Phi_i - \Phi_m \\
\end{align*} \]

\[ \begin{align*} 
\hat{\Phi} &= \Phi_i - \Phi_m \\
\end{align*} \]

\[ \begin{align*} 
\hat{\Phi} &= \Phi_i - \Phi_m \\
\end{align*} \]

\[ \begin{align*} 
\hat{\Phi} &= \Phi_i - \Phi_m \\
\end{align*} \]
\[ E(\sigma) = \sqrt{\sum_i D_i(\sigma)^2} \]  

\[ D^i = \min_{\Phi} \sqrt{e^T e : \tilde{K}(\sigma) \Phi + e = \tilde{J}_i}. \] (27)

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

\[ D^2 = \tilde{J}^T \tilde{J} - \tilde{J}^T \tilde{K} \left( \tilde{K}^T \tilde{K} \right)^{-1} \tilde{K}^T \tilde{J}. \] (28)

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

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

In fact, if \( A \) is symmetric, positive and definite, so, \( v^T f(A)v \) can be expressed as a Riemann-Stieltjes integral \( \int f(\lambda) \alpha(\lambda) \, d\lambda \) for a measure \( \alpha(\lambda) \) obtained from \( v \) and the spectral decomposition of \( A \). 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 \( v \).

The result of this approach is an iterative algorithm that produces upper and lower boundaries to \( v^T f(A)v \). 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 \( f(x) = x^{-1} \)

\[ v^T f(A)v \approx G_k = \|v\|^2 e^T_k T_k^{-1} e_1 \] (29)

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

\[ v^T f(A)v \approx R_k = \|v\|^2 e^T_k T_k^{-1} e_1 \] (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 \( v^T f(A)v \) (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 \( \tilde{K}\tilde{K}^T \). 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)

\[
\begin{align*}
p_0 &:= \tilde{J}/||\tilde{J}||_2; \\
s_0 &:= \tilde{K}^T p_0; \\
\gamma_1 &:= ||s_0||_2; \\
q_0 &:= s_0/\gamma_1; \\
r_1 &:= \tilde{K} q_0 - \gamma_1 p_0; \\
\delta_1 &:= ||r_1||_2; \\
p_1 &:= r_1/\delta_1; \\
\text{for } k := 2 \text{ to } n \text{ do} \\
& \quad s_{k-1} := \tilde{K}^T p_{k-1} - \delta_{k-1} q_{k-2}; \\
& \quad \gamma_k := ||s_{k-1}||_2; \\
& \quad q_{k-1} := s_{k-1}/\gamma_k; \\
& \quad r_k := \tilde{K} q_{k-1} - \gamma_k p_{k-1}; \\
& \quad \delta_k := ||r_k||_2; \\
& \quad p_k := r_k/\delta_k;
\end{align*}
\]
This algorithm builds matrices

\[ B_k = \begin{pmatrix} \gamma_1 & \delta_1 & \cdots & \gamma_k & \delta_k \\ \gamma_1 & \delta_1 & \cdots & \gamma_k & \delta_k \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \gamma_1 & \delta_1 & \cdots & \gamma_k & \delta_k \end{pmatrix} \]  

(31)

It’s easy to observe that \( B_k^T B_k \) is equal to \( T_k \) in the previous section (Golub and von Matt, 1979). Therefore, the final form of Gaussian quadrature is

\[ G_k = \| \tilde{K}^T \tilde{J} \|^2 g_k \]  

(32)

with \( g_k = e_1^T (B_k^T B_k)^{-1} e_1 \).

It’s easy to see that \( g_k \) is the first element of \((B_k^T B_k)^{-1}\). Now, using the fact that \( B_k^T B_k \) is a tridiagonal symmetric matrix, a Cholesky-like factorization can be written

\[ B_k^T B_k = L_k D_k \Delta_k^{-1} \]  

(33)

and \( \Delta_k \) a diagonal matrix with elements \( \psi_j \). So we have the following algorithm adapted from (Golub and Meurant, 2010) to obtain the factorization and \( g_k \).

**Algorithm 2: Obtaining \( g_k \)**

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

With this we can obtain the upper limit to our function \( \tilde{K} \tilde{K}^T \). 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 \( B_k \) in an upper bidiagonal matrix with the following form:

\[ G^T B_k = U_k = \begin{pmatrix} \tau_1 & \theta_1 & \cdots & \theta_{k-1} & \tau_k \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \tau_1 & \theta_1 & \cdots & \theta_{k-1} & \tau_k \end{pmatrix} \]  

(34)

After this, we must modify \( U_k \) to \( \tilde{U}_k \). To do this, we set \( \tau_k \) to be equal to the lowest eigenvalue of \( B_k^T U_k \) the we call \( \alpha \). So \( \tilde{T}_k \) is given by \( \tilde{U}_k^T \tilde{U}_k \).

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

\[ R_k = \| \tilde{K}^T \tilde{J} \|^2 r_k \]  

(35)

with \( r_k = e_1^T (\tilde{U}_k^T \tilde{U}_k)^{-1} e_1 \).

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

\[ \tilde{U}_k^T U_k = B_k^T G G^T B_k \]  

(36)
as Givens rotations are orthogonal (Golub and Van Loan, 1996), therefore, \(GG^T = I\). So,

\[
U_k^T U_k = B_k^T B_k = L_k^T \Delta_k^{-1} L_k
\]

As the only difference between \(U_k\) and \(\tilde{U}_k\) 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:

\[
r_i := g_{i-1} + c_i/(a - t);
\]

The last problem that remains is obtain the first eigenvalue of \(\hat{K}^T \hat{K}\). In (Golub and Meurant, 1996) is suggested that the lower eigenvalue of \(T_k\) can be used as an approximation to \(\hat{K}^T \hat{K}\)'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 \(\hat{K}^T \hat{K}\) 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

\[
M = P^T P \approx \hat{K}^T \hat{K}
\]

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

\[
D^2 = \hat{J}^T \hat{J} - \hat{J}^T \hat{K} \left(\hat{K}^T \hat{K}\right)^{-1} \hat{K}^T \hat{J}
\]

\[
D^2 = \hat{J}^T \hat{J} - \hat{J}^T \hat{K} \left(\hat{K}^T \hat{K} P\right)^{-1} P^T \hat{K}^T \hat{J}
\]

\[
D^2 = \hat{J}^T \hat{J} - \hat{J}^T \hat{K} P \left(P^T \hat{K}^T \hat{K} P\right)^{-1} P^T \hat{K}^T \hat{J}
\]

so, if we consider only symmetric \(P\) we have:

\[
D^2 = \hat{J}^T \hat{J} - \hat{J}^T \hat{K} \left(\hat{K}^T \hat{K}\right)^{-1} \hat{K}^T \hat{J}
\]

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

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

So, if we look for decompositions to \(K_{ic}\), for example, \(K_{ic} \approx LL^T = P\) we can build the preconditioned matrix \(\hat{K}\). 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 \(B_k0\), 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 \(B_k\) 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: Gaussian quadrature rule results

After these 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 $\tilde{K}^T\tilde{K}$ and the incomplete factorization density.

Initially, without any preconditioning $\tilde{K}^T\tilde{K}$ 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 $L$ matrix zero pattern to be equal as the one of $K_{ii}$. And for $Q$ 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 $K_{ii}$ we expected that $P = LQ$ 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 TUMA (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 $\tilde{K}_{ii} = K_{ii} + \alpha I$ with $\alpha$ ranging from zero to one. As we can see in Fig. 3h the values are more compacted around the origin, chiefly for $\alpha \geq 0.5$, 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 $L$ 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 shown 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.
Figure 3: Preconditioning techniques comparative

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


8. RESPONSIBILITY NOTICE

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