

An Efficient Multi-Threaded Newton-Raphson Algorithm for Strong Coupling Modeling of Multi-Physics Problems

Amir Akbari*, Dennis Giannacopoulos

McGill University, Department of Electrical and Computer Engineering, Montreal, Canada

ARTICLE INFO

Keywords:

Multi-physics problems
Newton-Raphson method
Gaussian belief propagation
Parallel scalability
Radiofrequency ablation

ABSTRACT

The Finite Element Gaussian Belief Propagation (FGaBP) method is an iterative algorithm with abundant parallelism making it an alternative for the traditional Finite Element Method (FEM), especially for large multi-physics problems. In this paper, we extend the FGaBP method to solve the coupled electrical-thermal problem that emerges in the modeling of radiofrequency ablation (RFA) of hepatic tumors. The strongest form of coupling algorithms, which is the Newton-Raphson (NR) method, is implemented in parallel using the localized computations of FGaBP. The parallel scalability of the FGaBP method is retained in the proposed algorithm by calculating local Jacobian matrices for each element and then updating the solutions for both electrical and thermal problems accordingly at each NR iteration.

1. Introduction

In general, finite element method (FEM) implementations consist of two computationally expensive stages which are the sparse matrix assembly stage, and the solving stage of the linear system using iterative solvers. These two stages are even more expensive for non-linear applications. While for certain cases, the assembly needs to be done only once and, therefore, its cost can be tolerated, for non-linear problems, such as multi-physics applications, the assembly stage can dominate the solving stage. This is especially so for a pure Newton-Raphson (NR) method, when the construction of a Jacobian matrix at each linearizing iteration could be prohibitively expensive for large scale problems.


The Jacobian-free Newton-Krylov (JFNK) method is the most widely used inexact solver for the NR algorithm. This method employs a Krylov-based iterative solver in which the Jacobian matrix does not have to be explicitly formed, and in this way, JFNK bypasses the main obstacle associated with the assembly stage of the NR method. On the other hand, in the solving stage, the efficiency of JFNK depends critically on preconditioning the inner Krylov subspace method. It is in this area that the Jacobian-free appeal of JFNK must yield to the construction and use of a preconditioning matrix which require the execution of a number of global algebraic operations in each JFNK iteration, such as a Sparse Matrix-Vector Multiplication (SMVM). The SMVM operation, in particular, can strongly limit the acceleration of the solving stage using parallel processing due to its dependency on the underlying sparse data-structure [10]. In addition, the approximation error associated with the Jacobian-vector multiplication represents the greatest disadvantage of JFNK, especially when variables associated with different physics being coupled in a multi-physics application differ by orders of magnitude.

The Gaussian belief propagation (GaBP) method is a recursive message passing algorithm that offers distributed computation providing potential to parallelize FEM applications [11]. This work presents a NR reformulation of GaBP in order to exploit its localized computations and message passing scheme for solving multi-physics problems in parallel. Similar to the JFNK, GaBP does not need to explicitly form a global Jacobian matrix, instead, local computations are performed to calculate local Jacobian matrices for each element in parallel. This provides a NR algorithm amicable to different parallel computing architectures. On the other hand, in contrast to the JFNK, there is no approximation associated with the local Jacobian matrices which makes the novel NR method more accurate than the JFNK method.

As a multi-physics application, the new method is applied to the coupled electrical-thermal phenomenon that appears in radiofrequency ablation (RFA) of liver tumors. RFA delivers electrical currents to biological tissues to thermally damage a tumor by raising its temperature for a certain duration. The purpose of an effective RFA is the entire ablation of the tumor with a safety margin of destroyed healthy tissue in its vicinity, which in turn relies on the extent of the ablation zone. However for treating tumors in the liver, the hepatic blood vessels dissipate heat and change the size and shape of the lesion zone [3]. Recently, computational models of RFA have gained attention as a tool for studying the heat distribution inside the tissue [15, 27]. To investigate the capabilities of our NR method, the coupled electrical-thermal phenomenon in a simplified model of RFA is simulated in this paper.

The paper is organized as follows. In Section 2, a summary of the coupled phenomena in RFA is provided. In Section 3, we present the GaBP formulation for the coupled electrical-thermal problem in RFA, and a parallel NR formulation based on the GaBP representation of the problem. Finally, in Sections 5 and 6, we present the results and concluding remarks, respectively.

*Corresponding author

 amir.akbari3@mail.mcgill.ca (A. Akbari);

dennis.giannacopoulos@mcgill.ca (D. Giannacopoulos)

ORCID(s):

2. The multi-physics phenomena in RFA

RFA probes operate between 460–550 kHz. At these frequencies, the wavelength is several orders of magnitude larger than the size of the ablation electrode. Thus, the displacement current is neglected and a quasi-static electrical conduction model is assumed, which allows us to solve the electric potential around the probe by using Laplace's equation:

$$\nabla \cdot (\sigma(T) \nabla v) = 0, \quad (1)$$

where $\sigma(T)$ is the temperature-dependent electrical conductivity (S/m), and v is the electric potential (V). In the literature, $\sigma(T)$ is usually introduced by mathematical functions which reflect its dependence on temperature. According to [28], the most commonly used mathematical function for $\sigma(T)$ is a piecewise function which uses different mathematical expressions according to the temperature range. At temperatures below 100 °C, $\sigma(T)$ increases with temperature with a rate from 1.5%/°C to 2%/°C, either linearly or exponentially [28]. At temperatures above 100 °C, a rapid drop of $\sigma(T)$ occurs due to vaporisation. However, the computer simulation of RFA could be ended when the maximum temperature in the tissue reaches 100 °C [8], considering the fact that the RF clinical power delivery protocol is usually based on varying the voltage applied between electrodes to keep the temperature at the electrode tip set to 80–90 °C [20].

Solving Laplace's equation has a high computational cost since it requires a fine volume mesh on the surface of the very thin probe tips. Thus, Audigier et al [3] model the electrical heating with a Dirichlet boundary condition on a sphere with a pre-determined radius around the probe tip. Another approach to approximate the Joule heating is proposed by [18] and used in later RFA simulation frameworks [17], [29] and [23]. In this approach, the power deposition is estimated using a Gaussian distribution around the probe tip. However, these approximations are, potentially, too much of a simplification as it has been shown by [14] that electrical parameters are critical in models of hepatic RFA. In addition, none of these approaches consider the temperature dependency of σ and the coupling between the electrical problem and the thermal problem. The most commonly used model for heat transfer in the tissue is the *Pennes* model [19]:

$$\rho_{ti} c_{ti} \frac{\partial T}{\partial t} = Q + \nabla \cdot (d \nabla T) + H(T_{bl} - T), \quad (2)$$

where T is the temperature (K), c_{ti} is the special heat capacity of tissue (J/kg·K), ρ_{ti} is the tissue density (kg/m³), d is the thermal conductivity (W/m·K), H is the convective transfer coefficient (W/m³/K) and T_{bl} is the baseline physiological blood temperature taken to be 310 K. On the right hand side of (2), $Q = \sigma |\nabla v|^2$ is the heat source (W/m³) which depends on the electric potential. On the other hand, in the electrical problem σ is changing with the temperature. As a result, the electrical and thermal problems are coupled to each other and a multi-physics approach is needed to model this phenomenon.

In the *Pennes* model, the blood temperature is assumed constant, which is only valid within and close to large vessels [3]. For this reason, Payne et al [18] proposes a multi-physics model for computing heat diffusion in the liver, where each elementary volume is assumed to comprise both tissue and blood with a certain fraction:

$$(1 - \epsilon) \rho_{ti} c_{ti} \frac{\partial T_{ti}}{\partial t} = (1 - \epsilon) Q + (1 - \epsilon) \nabla \cdot (d \nabla T_{ti}) + H(T_{bl} - T_{ti}), \quad (3a)$$

$$\epsilon \rho_{bl} c_{bl} \left(\frac{\partial T_{bl}}{\partial t} + \mathbf{v} \cdot \nabla T_{bl} \right) = \epsilon Q + \epsilon \nabla \cdot (d \nabla T_{bl}) - H(T_{bl} - T_{ti}). \quad (3b)$$

In these two coupled equations, ϵ stands for blood volume fraction (fraction of blood volume over total volume), \mathbf{v} is the blood velocity (m/s), and subscripts 'ti' and 'bl' refer to tissue and blood phase respectively. Most RFA theoretical models have used a constant value for d , probably due to the fact that changes in d with temperature are not so marked as in σ [25]. The numerical values of the electrical and thermal model parameters used in the literature are provided in Table 1. Because of its high computational cost, the electrical problem and its coupling to the thermal problem has often been over-simplified in the recent literature. This paper tries to develop an efficient multi-physics solver and applies it to the coupled electrical-thermal problem in RFA.

Table 1

Values of electrical and thermal model parameters used in the literature.

Parameter	Value	Ref
Baseline liver electrical conductivity, σ_{ti} (S/m)	0.33	[9]
Probe electrical conductivity, σ_{probe} (S/m)	10 ⁸	[9]
Convective transfer coefficient, H (W/(m ³ K) ⁻¹)	24.4×10 ⁵	[3]
Liver density, ρ_{ti} (kg/m ³)	1060	[18]
Liver heat capacity, c_{ti} (J/kg·K)	3600	[18]
Baseline liver thermal conductivity, d_{ti} (W/m·K)	0.512	[18]
Blood density, ρ_{bl} (kg/m ³)	1000	[9]
Blood heat capacity, c_{bl} (J/kg·K)	4180	[18]
Probe density, ρ_{probe} (kg/m ³)	6450	[9]
Probe heat capacity, c_{probe} (J/kg·K)	840	[9]
Probe thermal conductivity, d_{probe} (W/m·K)	18	[9]

3. Parallel formulation of the coupled electrical-thermal problem

As explained in Section 2, the focus of this paper is on the coupled electrical-thermal problem. For this reason, the coupling between the tissue temperature and the blood temperature, i.e., (3) is neglected and the thermal problem is modeled by (2). If the theta-scheme [26] is used for time discretization of (2), and similar FEM mesh for both (1) and (2), we will have the following set of coupled discrete equations:

$$[\mathbf{M}_v(T)] \{ \mathbf{v} \} = 0 \quad (4a)$$

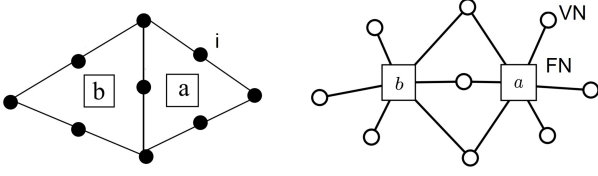


Figure 1: (Left): Sample FEM mesh of two second order triangles. (Right): Two types of nodes in the factor graph representation.

$$[\mathbf{M}_T + \delta t \theta \mathbf{S}_T] \{\mathbf{T}\}^{n+1} = [\mathbf{M}_T - \delta t (1 - \theta) \mathbf{S}_T] \{\mathbf{T}\}^n \quad (4b)$$

$$+ \delta t \{(1 - \theta)\{\mathbf{f}(v)\}^n + \theta\{\mathbf{f}(v)\}^{n+1}\}$$

where $M_{v,ij}(T) = \int \sigma(T) \bar{\nabla} N_i \cdot \bar{\nabla} N_j dV$ depends on the temperature, $f_j(v) = \int (HT_{bl} + \sigma(T)|\nabla v|^2) N_j dV$ depends on the potential, $M_{T,ij} = (\rho c_{ti} + \delta t H) \int N_i N_j dV$ and $S_{T,ij} = d \int \bar{\nabla} N_i \cdot \bar{\nabla} N_j dV$. Note that N_i and N_j are FEM scalar basis functions, n is the time step number, δt is the time step value and $0 \leq \theta \leq 1$ is the parameter of the theta-scheme time discretization. Because of straightforward implementation, successive substitution is often the first route to a multi-physics simulation. This approach, which is considered a weak-coupling model iterates over the uni-physics problems, solving the first equation for the first unknown, given the second unknown, and the second equation for the second unknown, given the first. The main advantage of successive substitution is making use of the existing codes for the uni-physics problems; however, it may ignore strong couplings between physics and give a false sense of completion [16]. If a solver code that goes beyond the weak-coupling is needed, the NR method is the simplest algorithm. This section provides an efficient FEM-based NR algorithm for solving the coupled electrical-thermal problem of (4) in a strong coupling manner. For this purpose, a parallel FEM formulation based on GaBP algorithm introduced by [10] is modified for solving multi-physics problems. The FEM-based GaBP (FGaBP) method was tested on simple electrostatic problems showing good parallel scalability compared to other parallel algorithms [10]. This paper extends FGaBP to deal with a multi-physics scheme.

The main idea of FGaBP is to reformulate FEM into an inference problem over a factor graph, which may be solved applying the Gaussian belief propagation rules. As shown in Fig. 1, FGaBP turns the FEM mesh into a factor graph. Two different types of nodes are then introduced in the factor graph: variable nodes (VN, nodes of unknowns) represented by circles, and factor nodes (FNs) represented by squares. The solution at each VN is considered a random variable with a Gaussian distribution whose shape is defined by two parameters, α and β , where α is the reciprocal of the variance and β/α is the mean. By passing messages between each FN and all its connected VNs, the GaBP algorithm tries to find the values of α and β for each VN. A message, m_{ai} , is sent from factor node a (FN_a) to the connected variable node i (VN_i) and represents the most probable solution value at i ,

as observed from FN_a. In return, VN_i sends a message back to FN_a representing observations from other connected FNs. The following is the formulation of the FGaBP algorithm update rules[10]:

1. $t = 0$: Initialize all messages $\beta^{(0)} = 0$ and $\alpha^{(0)} = 1$.
2. Iterate: $t = 1, 2, \dots$ and $t_* \leq t$.
 - (a) For each VN_i, compute messages α_{ia} and β_{ia} to each connected FN_a ($a \in \mathcal{N}(i)$) as follows:

$$\alpha_i^{(t_*)} = \sum_{k \in \mathcal{N}(i)} \alpha_{ki}^{(t_*)}, \quad \alpha_{ia}^{(t)} = \alpha_i^{(t_*)} - \alpha_{ai}^{(t_*)} \quad (5)$$

$$\beta_i^{(t_*)} = \sum_{k \in \mathcal{N}(i)} \beta_{ki}^{(t_*)}, \quad \beta_{ia}^{(t)} = \beta_i^{(t_*)} - \beta_{ai}^{(t_*)} \quad (6)$$

where $\mathcal{N}(i)$ is the neighborhood set of node i .

- (b) For each FN_a:
 - i. receive messages $\alpha_{ia}^{(t)}$ and $\beta_{ia}^{(t)}$ where $i \in \mathcal{N}(a)$.
 - ii. Assume $\mathcal{A}^{(t_*)}$ is a diagonal matrix of incoming $\alpha_{ia}^{(t_*)}$ messages, and $\mathcal{B}^{(t_*)}$ is a vector of incoming $\beta_{ia}^{(t_*)}$ messages, then define matrix \mathbf{W} and vector \mathbf{K} as follows:

$$\mathbf{W}^{(t_*)} = \mathbf{M} + \mathcal{A}^{(t_*)} \quad (7)$$

$$\mathbf{K}^{(t_*)} = \mathbf{B} + \mathcal{B}^{(t_*)} \quad (8)$$

where \mathbf{M} and \mathbf{B} are element a characteristic matrix and source vector respectively.

- iii. Partition $\mathbf{W}^{(t_*)}$ and $\mathbf{K}^{(t_*)}$ as follows:

$$\mathbf{W}^{(t_*)} = \begin{bmatrix} \mathbf{W}_{\mathcal{L}(i)}^{(t_*)} & \mathbf{V}^T \\ \mathbf{V} & \bar{\mathbf{W}}^{(t_*)} \end{bmatrix} \quad (9)$$

$$\mathbf{K}^{(t_*)} = \begin{bmatrix} \mathbf{K}_{\mathcal{L}(i)}^{(t_*)} \\ \bar{\mathbf{K}}^{(t_*)} \end{bmatrix} \quad (10)$$

where $\mathcal{L}(i)$ is the local index corresponding to the global variable node i .

- iv. Compute and partition $(\mathbf{W}^{(t_*)})^{-1}$ as follows:

$$(\mathbf{W}^{(t_*)})^{-1} = \begin{bmatrix} \tilde{\mathbf{W}}_{\mathcal{L}(i)}^{(t_*)} & \tilde{\mathbf{C}}^T \\ \tilde{\mathbf{C}} & \tilde{\mathbf{W}} \end{bmatrix}. \quad (11)$$

- v. Compute and send new FN_a messages $\alpha_{ai}^{(t)}$ and $\beta_{ai}^{(t)}$ to each VN_i as follows:

$$\alpha_{ai}^{(t)} = \frac{1}{\tilde{\mathbf{W}}_{\mathcal{L}(i)}^{(t_*)}} - \alpha_{ia}^{(t_*)}. \quad (12)$$

$$\beta_{ai}^{(t)} = \mathbf{B}_{\mathcal{L}(i)} + \frac{1}{\tilde{\mathbf{W}}_{\mathcal{L}(i)}^{(t_*)}} (\bar{\mathbf{K}}^{(t_*)})^T \tilde{\mathbf{C}}^T. \quad (13)$$

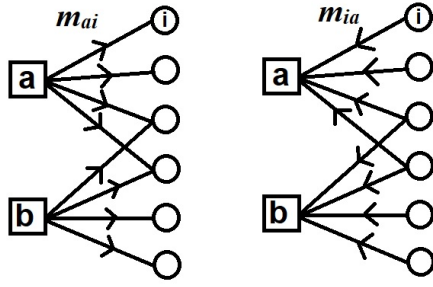


Figure 2: (Left): A message m_{ai} is sent from FN_a to VN_i . (Right): A message m_{ia} is sent back from VN_i to FN_a .

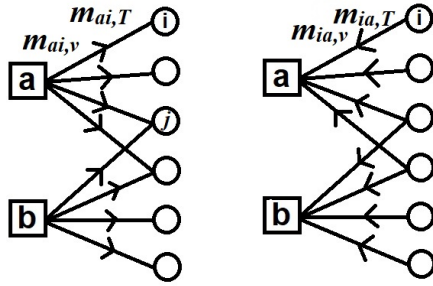


Figure 3: (Left): A message $m_{ai,v}$ ($m_{ai,T}$) is sent from FN_a to VN_i in the electrical (thermal) problem. (Right): A message $m_{ia,v}$ ($m_{ia,T}$) is sent back from VN_i to FN_a in the electrical (thermal) problem.

- At message convergence, the mean of the VNs, or solutions, can be obtained by:

$$u_i^{(t)} = \frac{\beta_i^{(t)}}{\alpha_i^{(t)}} \quad (14)$$

where

$$\beta_i = \sum_{k \in N(i)} \beta_{ki}, \quad \alpha_i = \sum_{k \in N(i)} \alpha_{ki}. \quad (15)$$

The message passing scheme of FGaBP is shown graphically by Fig. 2. In a uni-physics problem, each message contains information about the value of the unknown parameter at each node. For example, in Fig. 2, left, the message m_{ai} in the electrical problem carries information on the potential at node i , v_i . On the other hand, the different physics involved in a multi-physics problem need to exchange information with each other; if a strong coupling approach is needed, the exchanged messages between different physics must contain information about their sensitivity with respect to each other, besides information about their values. Figure 3 shows the message passing scheme in a multi-physics scenario. Here, a message $m_{ai,v}$ ($m_{ai,T}$) sent in the electrical (thermal) problem carries information about v_i (T_i) and also dv_i/dT_j (dT_i/dv_j).

When the update rules of (12) and (13) are applied to the coupled electrical-thermal problem, it should be noted that

according to (4), in the electrical problem, element a characteristic matrix, \mathbf{M}_v , depends on the temperature values at nodes i when $i \in \mathcal{N}(a)$. On the other hand, in the thermal problem, element a source vector, \mathbf{B}_T , depends on the electric potential values at nodes i where $i \in \mathcal{N}(a)$. As a result, equations (11) and (12) can be rewritten as:

$$\alpha_{ai,v}^{(t)} = \frac{1}{\tilde{W}_{\mathcal{L}(i),v}(T_j)} - \alpha_{ia,v}^{(t_*)}, \quad (16a)$$

$$\beta_{ai,v}^{(t)} = B_{\mathcal{L}(i),v} + \frac{1}{\tilde{W}_{\mathcal{L}(i),v}(T_j)} (\bar{\mathbf{K}}_v^{(t_*)}(v_j))^T \tilde{\mathbf{C}}_v(T_j), \quad (16b)$$

$$\alpha_{ai,T}^{(t)} = \frac{1}{\tilde{W}_{\mathcal{L}(i),T}} - \alpha_{ia,T}^{(t_*)}, \quad (16c)$$

$$\beta_{ai,T}^{(t)} = B_{\mathcal{L}(i),T}(v_j) + \frac{1}{\tilde{W}_{\mathcal{L}(i),T}} (\bar{\mathbf{K}}_T^{(t_*)}(v_j, T_j))^T \tilde{\mathbf{C}}_T, \quad (16d)$$

where $(\alpha_{ai,v}, \beta_{ai,v})$ are messages in the electrical problem, $(\alpha_{ai,T}, \beta_{ai,T})$ are messages in the thermal problem and $i, j \in \mathcal{N}(a)$. Equations (16a) to (16d) present a coupled system of equations at each FGaBP iteration inside FN_a . The rest of this section provides a parallel strong coupling algorithm to solve these coupled equations. According to (16c), α messages in the thermal problem are independent of the potential values; as a result we only consider the coupling between (16a), (16b) and (16d):

$$\begin{cases} \alpha_{ai,v}^{(t)} = \frac{1}{\tilde{W}_{\mathcal{L}(i),v}(T_j)} - \alpha_{ia,v}^{(t_*)}, \\ \beta_{ai,v}^{(t)} = B_{\mathcal{L}(i),v} + \frac{1}{\tilde{W}_{\mathcal{L}(i),v}(T_j)} (\bar{\mathbf{K}}_v^{(t_*)}(v_j))^T \tilde{\mathbf{C}}_v(T_j), \\ \beta_{ai,T}^{(t)} = B_{\mathcal{L}(i),T}(v_j) + \frac{1}{\tilde{W}_{\mathcal{L}(i),T}} (\bar{\mathbf{K}}_T^{(t_*)}(v_j, T_j))^T \tilde{\mathbf{C}}_T. \end{cases} \quad (17)$$

In general, a multi-dimensional zero-finding problem could be written as:

$$F(x) = 0, \quad (18)$$

where $F(x) = (f_1(x), f_2(x), \dots, f_n(x))^T$ is a system of n coupled equations and each $f_i(x)$ maps the vector of unknowns $x = (x_1, x_2, \dots, x_n)^T$ with dimension n into a scalar. The NR method solves such a non-linear system by solving linear systems successively, that is:

$$J(x^{(m)})\Delta x^{(m)} = -F(x^{(m)}), \quad (19)$$

where Δx is called the update vector, $J(x)$ is an n -by- n Jacobian matrix of $F(x)$, i.e., $J_{ij}(x) = \partial F_i / \partial x_j$, and m is the iteration number. The update vector is then used in order to obtain the solution vector x for the next iteration:

$$x^{(m+1)} = x^{(m)} + \Delta x^{(m)}. \quad (20)$$

Solving a linear system of equations at each iteration can make NR method prohibitively expensive for a large n . This section tries to alleviate this problem by applying the NR method at the element-level equations in (17). Suppose $k = \mathcal{L}(i)$ and $l = \mathcal{L}(j)$ are the local indices corresponding to global VNs i and j , respectively. Since both FGaBP and NR are iterative, we will have two nested iterations at each time step. If we are at iteration t (or $t_* \leq t$) of FGaBP and iteration m of NR, then we will have the following non-linear residuals for FN_a :

$$\begin{cases} \alpha_{ak,v}^{(m,t)} - f_k(v_l^{(m,t_*)}, T_l^{(m-1)}) = 0 \\ \beta_{ak,v}^{(m,t)} - g_k(v_l^{(m,t_*)}, T_l^{(m-1)}) = 0 \\ \beta_{ak,T}^{(m,t)} - h_k(v_l^{(m-1)}, T_l^{(m,t_*)}) = 0, \end{cases} \quad (21)$$

where f_k , g_k and h_k are the right-hand-sides of the equations in (17), respectively. At each FGaBP iteration, the sensitivities $\partial\alpha_{ak,v}/\partial T_l$, $\partial\beta_{ak,v}/\partial T_l$, $\partial\beta_{ak,v}/\partial v_l$, $\partial\beta_{ak,T}/\partial v_l$, and $\partial\beta_{ak,T}/\partial T_l$ are calculated and sent to the neighboring nodes of FN_a . After FGaBP iterations have converged, the non-linear system of (17) can be solved with a pure NR method, using local data for each FN. A local Jacobian matrix is constructed for FN_a based on (17) and the sensitivities calculated during FGaBP as follows:

$$\mathbf{J}_a = \begin{bmatrix} \mathbf{I}_{2nc \times 2nc} & \mathbf{J}_{a,v} \\ \mathbf{J}_{a,T} & \mathbf{I}_{nc \times nc} \end{bmatrix}_{3nc \times 3nc}, \quad (22)$$

where nc is the number of nodes per cell, and \mathbf{I} is the identity matrix. The details on the calculation of the off-diagonal elements of \mathbf{J}_a are provided in the appendix. The calculation of $\mathbf{J}_{a,v}$ and $\mathbf{J}_{a,T}$ entries needs the partial derivatives of local scalars and vectors with respect to the temperature values at the nodes in the neighborhood of FN_a ; As a result, the Jacobian matrix associated with FN_a , \mathbf{J}_a , is only dependent on the local data structure of FN_a , containing a dense matrix \mathbf{M}_a , vector \mathbf{B}_a and messages $(\alpha_{ai}, \beta_{ai})$. After FGaBP iterations converge and the local Jacobian is formed for FN_a , in accordance with (19), $\alpha_{ak,v}$, $\beta_{ak,v}$, and $\beta_{ak,T}$ updates are computed as:

$$\begin{Bmatrix} \{\Delta\alpha_{ak,v}\}_{nc \times 1} \\ \{\Delta\beta_{ak,v}\}_{nc \times 1} \\ \{\Delta\beta_{ak,T}\}_{nc \times 1} \end{Bmatrix}^{(m)} = [\mathbf{J}_a^{-1}]^{(m)} \cdot \begin{Bmatrix} \{f_k - \alpha_{ak,v}\}_{nc \times 1} \\ \{g_k - \beta_{ak,v}\}_{nc \times 1} \\ \{h_k - \beta_{ak,T}\}_{nc \times 1} \end{Bmatrix}^{(m)}. \quad (23)$$

Finally, at the end of the current NR iteration, the messages $\alpha_{ak,v}$, $\beta_{ak,v}$, and $\beta_{ak,T}$ are updated using the results from (23):

$$\begin{Bmatrix} \{\alpha_{ak,v}\} \\ \{\beta_{ak,v}\} \\ \{\beta_{ak,T}\} \end{Bmatrix}^{(m+1)} = \begin{Bmatrix} \{\alpha_{ak,v}\} \\ \{\beta_{ak,v}\} \\ \{\beta_{ak,T}\} \end{Bmatrix}^{(m)} + \begin{Bmatrix} \{\Delta\alpha_{ak,v}\} \\ \{\Delta\beta_{ak,v}\} \\ \{\Delta\beta_{ak,T}\} \end{Bmatrix}^{(m)}. \quad (24)$$

The updated messages are then used in the next NR iteration in the current time step. After the NR iterations have converged, we proceed to the next time step. Figure 4 shows these three nested iterations graphically.

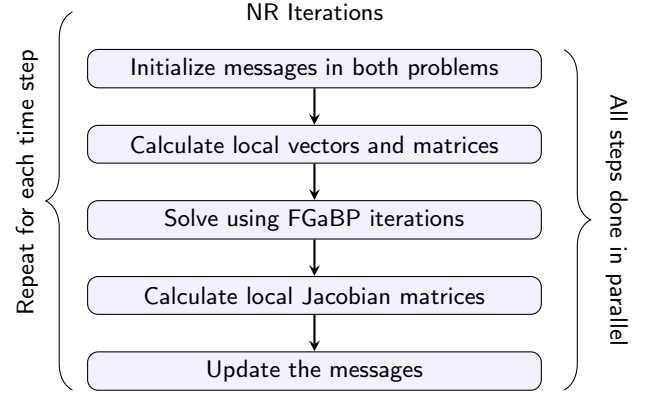


Figure 4: The parallel NR algorithm.

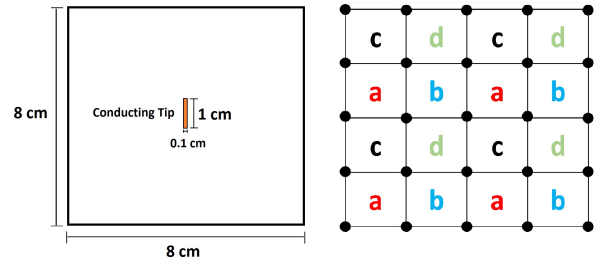


Figure 5: (Left): The geometry of the test case in two dimensions. (Right): Structured quadrilateral mesh containing four colors.

4. Implementation

In this section, the parallel NR algorithm is applied to the coupled equations in (1) and (2) in both two dimensional (2D) and three dimensional (3D) geometries. As shown in Fig. 5, left, tissue surrounding the probe tip is modeled by a 8.0 cm by 8.0 cm square (8.0 cm by 8.0 cm by 8.0 cm cubic) domain in 2D (3D), when the 16 V source voltage is modeled as a Dirichlet boundary condition on the probe located at the center of the domain. The outer boundaries of the domain serve as a return ground electrode. In the thermal problem, the normal body temperature, i.e., 37 °C is the initial value of the temperature, and a Neumann boundary condition is applied to the outer boundary. The electrical and thermal parameters in (1) and (2) are chosen according to Table 1. In addition, the electrical conductivity increases with the temperature with a linear rate of 2%/°C. The algorithm is implemented using the open-source FEM software deal.II [7].

An element-based coloring message schedule is implemented in order to avoid any race conditions when parallel processing is used. The race condition might happen when two different elements, e.g., elements a and b in Fig.3, left, try to update the same global node j . If elements a and b belong to two different threads, then we need to make sure that the messages $m_{aj,v}$ and $m_{bj,v}$ do not try to update the voltage value at node j at the same time. One solution is to schedule the messages based on element coloring. The mesh elements are colored so that no two adjacent elements

have the same color symbol. In this way, the messages in each color group can be computed and safely communicated in parallel, since elements that belong to the same group do not share any global nodes. A mesh coloring diagram in 2D is illustrated in Fig. 5, right, using a quadrilateral mesh.

```

1 Start timer;
2 Import problem geometry and mesh;
3 # pragma omp parallel;
4 for NR iteration  $m = 1, 2, \dots$  do
5   for each level in the mesh hierarchy do
6     # pragma omp for;
7     for cell  $a = 1, 2, \dots$  do
8       for node  $i \in \mathcal{N}(a)$  do
9         Initialize messages  $m_{ai,(v,T)}$ ;
10      end
11      Calculate local vector  $\mathbf{f}$  and local
        matrices  $\mathbf{M}_{v,T}$  inside cell  $a$ , i.e., (4).
12    end
13  end
14  for cycles = 1, 2, ... in the V-cycle do
15    for Mesh levels from fine to coarse do
16      Execute  $v_1$  iterations of Algorithm 2;
17      Restrict;
18    end
19    Execute Algorithm 2 on the coarsest level;
20    for Mesh levels from coarse to fine do
21      Execute  $v_2$  iterations of Algorithm 2;
22      Prolongate;
23    end
24    # pragma omp single;
25    if global tolerance < tolerance then
26      break;
27    end
28  end
29  Execute Algorithm 3 on the finest level;
30  # pragma omp single;
31  if global residual < NR tolerance then
32    break;
33  end
34 end
35 End timer;
36 Plot output;

```

Algorithm 1: Parallel NR pseudo-code in each time step.

Mesh refinement is conducted in 2D (3D) by splitting each quadrilateral (hexahedral) cell into four (eight) smaller cells successively. In this parent-child scheme, we start by a coarse mesh and continue mesh refinement until a fine mesh is achieved. In order to accelerate the FGaBP iterations, information from the coarse mesh (Parent) are transferred to the next fine mesh (Child). The transferred information are the local messages, $m_{ai,v}$ and $m_{ai,T}$ calculated for each element a and node $i \in \mathcal{N}(a)$. In our multi-physics scheme, these messages contain the so called *beliefs* regarding the temperature and potential values at each node, as well as the

```

1 for FGaBP iteration  $t = 1, 2, \dots$  do
2   for color  $c = 1, 2, \dots$  do
3     # pragma omp for;
4     for cell  $a$  in color  $c$  do
5       for node  $i \in \mathcal{N}(a)$  do
6         Calculate  $m_{ai,(v,T)}$ , i.e., (5) to (13);
7         Calculate  $J_a$ , i.e., (22);
8         Update  $v_i$  and  $T_i$ , i.e., (14);
9         Calculate local message tolerance;
10      end
11    end
12    # pragma omp critical;
13    Update global message tolerance;
14  end
15  # pragma omp single;
16  if global tolerance < FGaBP tolerance then
17    return(global tolerance);
18    break;
19  end
20 end

```

Algorithm 2: FGaBP with local Jacobian calculation

```

1 for color  $c = 1, 2, \dots$  do
2   # pragma omp for;
3   for cell  $a$  in color  $c$  do
4     Update  $m_{ai,(v,T)}$ , i.e., (23), (24);
5     Update  $v_i$  and  $T_i$ , i.e., (14);
6     Calculate local NR residual;
7     Calculate local vector  $\mathbf{f}$  and local matrices
         $\mathbf{M}_{v,T}$  inside cell  $a$ , i.e., (4).
8   end
9   # pragma omp critical;
10  Update global NR residual;
11 end

```

Algorithm 3: NR parallel update

sensitivity of these values with respect to their neighboring nodes. The transfer of information between different refinement levels is conducted in a multi-grid scheme. Algorithm 1 presents a pseudo-code for the parallel NR method with OpenMP directives. Lines 14 to 28 indicate the multi-grid scheme implemented as a V-Cycle. Algorithms 2 and 3 execute FGaBP iterations and the application of local Jacobian matrices to update the messages, respectively. Although the sensitivity information is carried on by the messages, only the potential and temperature values are considered in the prolongation and restriction; as a result, these two steps are implemented in parallel exactly as proposed by [12] and are not described here.

5. Results

The numerical results of the new NR formulation are verified using COMSOL Multiphysics software. The temperature obtained from COMSOL and the new multi-physics NR

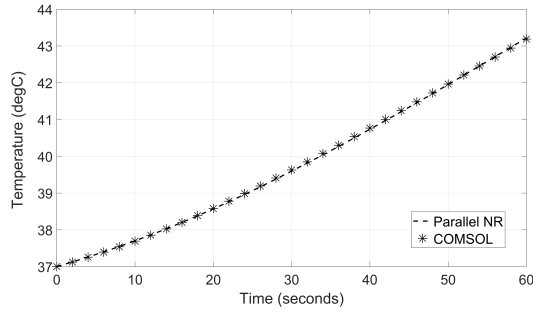


Figure 6: Temperature values over time at a specific point, i.e., $x = 1$ cm and $y = 1$ cm, when the probe tip is at the center of the domain. The time step size δt is equal to 0.5 s.

algorithm at a specific location during one minute of ablation are shown in Fig. 6. Quantitatively, the Root-Mean-Square (RMS) error between these two simulations is 0.028 °C, which validates our method. To test the parallel scalability properties of the method, a CPU implementation with multi-threading (OpenMP) is provided. As indicated in Algorithm 1, the CPU time calculation includes all the steps except for the output of results. All runs are executed on a Compute Canada cluster node. The node contains 2×20 -core Intel Gold 6148 Skylake 2.4 GHz CPUs with 186 GB DRAM. When the parameters $v_1 = 1$ and $v_2 = 5$ in Algorithm 1, the V-Cycle required 5 iterations for all 2D and 7 iterations for all 3D runs. This is independent of the number of unknowns in the finest level which is in agreement with the findings reported by [12]. Note that the number of iterations in the inner FGaBP execution, i.e., line 19 in Algorithm 1, remains proportional to the problem size in the coarsest level. In the electrical problem, we start by a mesh size of 400 cells in 2D (8000 cells in 3D) which takes 380 (430) inner iterations to reach a message residual of 10^{-8} . The thermal problem converges faster in both 2D and 3D implementations. Figure 7 shows the speedup for a fixed problem size of 4,173,281 unknowns in 3D with respect to the number of processors (strong scaling). A curve is fitted to the plot based on Amdah's law [24], from which the ratio of the serial part of the algorithm (s) is obtained as 0.032. Weak scaling is performed by running the algorithm with different numbers of threads and with a correspondingly scaled problem size in 2D. The problem size is changed from 410,881 to 13,148,192 unknowns. The scaled speedup data and a linear curve fitted to it based on Gustafson's law [13] are depicted in Fig. 8. The fitted value for s is 0.12 which is different from that given by Amdah's law and strong scaling. The discrepancy in s can be due to the approximations in both Amdah's and Gustafson's laws, i.e., the serial part is assumed to remain constant, and the parallel part is assumed to be scaled up in proportion to the number of threads.

In our strong and weak scaling studies, no scheduling type is specified for the parallel loops, i.e., OpenMP uses its default scheduling type. In Fig. 9 a comparison between static and dynamic scheduling with different chunk sizes is provided. The number of threads is chosen to be 12 and 16,

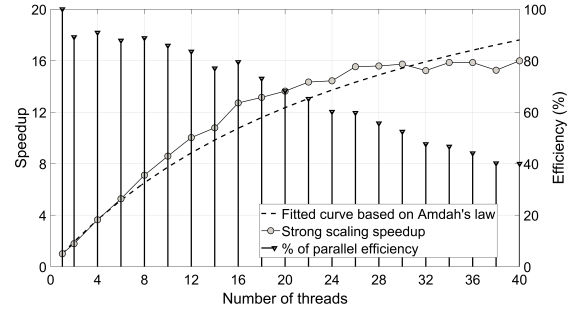


Figure 7: Strong scaling of the multi-physics NR method in terms of speedup with respect to 1 core implementation. The dashed line is the fitted curve based on Amdah's law.

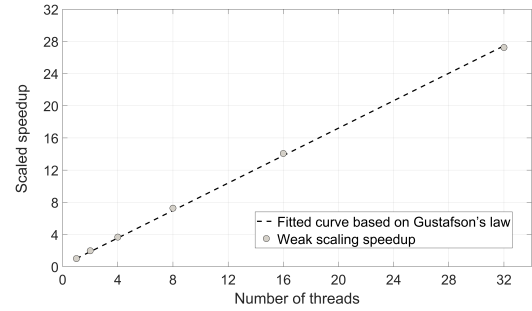


Figure 8: Weak scaling of the multi-physics NR method in terms of scaled speedup with respect to 1 core implementation. The dashed line is the fitted curve based on Gustafson's law

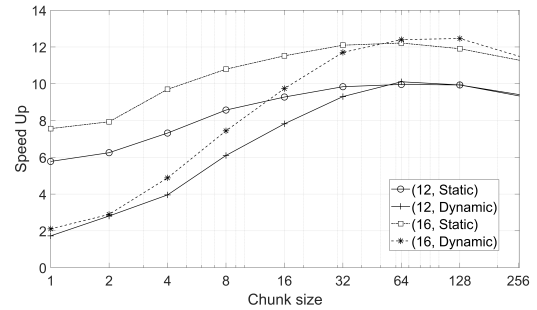


Figure 9: Speedup of the parallel NR method in terms of OpenMP for loop's scheduling type and chunk size when the number of threads = 12 and 16.

when the problem size is the same as that of strong scaling, i.e., 4,173,281 unknowns in 3D. According to Fig. 9, the best performance observed is similar to that delivered by the default scheduling in Fig. 7. Also, for reasonably large chunk sizes, the difference between static and dynamic scheduling is insignificant, which is due to the fact that the iterations of the parallel loops in the code have almost the same computational work.

We compare our OpenMP NR algorithm to a parallel implementation of the traditional NR method provided by the optimized library PETSc (Portable, Extensible Toolkit for Scientific Computation)[4, 5, 6]. PETSc employs the Message Passing Interface (MPI) standard for communica-

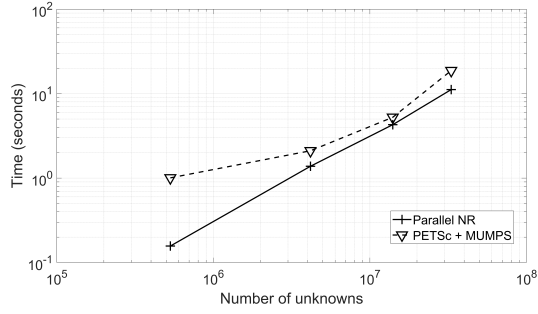


Figure 10: Execution times using 16 cores.

tion between parallel tasks. The PETSc implementation is tested on a single cluster node with total 40 number of cores, where one MPI task is defined per core. The Jacobian matrix is formed at each NR iteration, when the assembly of the Jacobian is done in parallel using deal.II's WorkStream shared-memory model. For the solving stage, PETSc provides an interface to a variety of iterative and direct solvers, from which the MULTifrontal Massively Parallel sparse direct Solver (MUMPS) [2, 1] is selected here. Figure 10 shows the average execution times per time step for our OpenMP NR code with 16 threads, and the PETSc implementation with 16 threads for the assembly and 16 MPI tasks for the solution, respectively. Problem sizes change from 500K to 33M unknowns in 3D. The Parallel NR demonstrated faster execution time while preserving linear scalability with the number of unknowns. As the problem size increases, the overhead due to PETSc's MPI calls reduces resulting in improved efficiency for larger problems.

Finally, the convergence plot of the parallel NR method is depicted and compared to that of a Gauss-Seidel algorithm when applied to the same electrical-thermal problem in Fig. 11. As for the NR approach, in the early iterations, changes of the residual norm are almost linear; however, when we get sufficiently close to the solution, the quadratic convergence can be observed in the last three iterations until the convergence criterion, i.e., a residual norm smaller than 10^{-10} , is achieved.

6. Conclusions

The message passing properties of the Belief Propagation algorithm on a graphical model are exploited for the first time to compute and distribute local sensitivities instead of forming a large Jacobian matrix in a multi-physics scenario. This approach provides a highly parallel NR algorithm for solving the coupled electrical-thermal problem in RFA modeling. This problem has often been over-simplified in the literature because of its high computational cost. A simplified model of the coupled problem that is solved with the new NR algorithm. In order to verify the correctness of the algorithm, obtained results are compared with COMSOL multiphysics simulation, showing good fidelity. The parallel scalability of the FGaBP method is retained in the new multi-physics version proposed here. Our results show

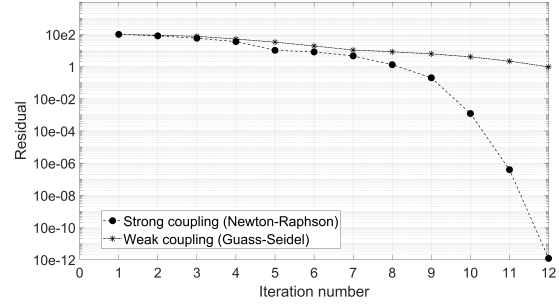


Figure 11: The convergence plot of the parallel NR algorithm compared to that of a Gauss-Seidel algorithm.

speedups of sixteen times with respect to one CPU core. Shared-memory implementation of our algorithms showed faster run time against highly optimized open-source library PETSc. Future work should include comparison against an optimized multi-threaded solver such as Parallel Sparse Direct Solver (PARDISO) [21, 22], distributed-memory implementation based on MPI on cluster architectures, comparison with parallel implementations of inexact Newton's methods including JFNK methods, and the application of the proposed method to more realistic models of RFA.

7. Acknowledgments

This research was enabled in part by support provided by Calcul Québec (www.calculquebec.ca), Compute Canada (www.computeCanada.ca), and also financial support from NSERC (The Natural Sciences and Engineering Research Council of Canada).

A. Appendix

A local Jacobian matrix is constructed for FN_a , based on the sensitivities calculated during FGaBP:

$$\mathbf{J}_a^{(m)} = \begin{bmatrix} \mathbf{I}_{2nc \times 2nc} & \mathbf{J}_{a,v}^{(m)} \\ \mathbf{J}_{a,T}^{(m)} & \mathbf{I}_{nc \times nc} \end{bmatrix} \quad (25)$$

The off-diagonal elements of the local Jacobin matrix are updated during FGaBP iterations; inside iteration m of Newton's method and iteration t of FGaBP we have:

$$\mathbf{J}_{a,v}^{(m,t)}(k, l) = \frac{\partial \alpha_{ak,v}^{(m,t)}}{\partial \beta_{al,T}^{(m-1)}} = \frac{\partial \alpha_{ak,v}^{(m,t)}}{\partial T_l^{(m-1)}} \frac{\partial T_l^{(m-1)}}{\partial \beta_{al,T}^{(m-1)}} = \frac{1}{\alpha_{l,T}^{(m-1)}} \frac{\partial \alpha_{ak,v}^{(m,t)}}{\partial T_l^{(m-1)}}, \quad (26)$$

for $0 \leq k < nc$, and:

$$\mathbf{J}_{a,v}^{(m,t)}(k, l) = \frac{\partial \beta_{ak,v}^{(m,t)}}{\partial \beta_{al,T}^{(m-1)}} = \frac{\partial \beta_{ak,v}^{(m,t)}}{\partial T_l^{(m-1)}} \frac{\partial T_l^{(m-1)}}{\partial \beta_{al,T}^{(m-1)}} = \frac{1}{\alpha_{l,T}^{(m-1)}} \frac{\partial \beta_{ak,v}^{(m,t)}}{\partial T_l^{(m-1)}}, \quad (27)$$

for $nc \leq k < 2nc$. Here the chain rule and also the fact that based on (13) we can write $\partial T_l / \partial \beta_{al,T} = 1/\alpha_{l,T}$ are used.

Now, based on (21) the elements of the sub-matrix $\mathbf{J}_{a,v}^{(m,t)}$ are updated as:

$$\mathbf{J}_{a,v}^{(m,t)}(k, l) = \frac{1}{\alpha_{l,T}^{(m-1)}} \frac{\partial f_k}{\partial T_l^{(m-1)}}, \quad (28)$$

for $0 \leq k < nc$, and:

$$\mathbf{J}_{a,v}^{(m,t)}(k, l) = \frac{1}{\alpha_{l,T}^{(m-1)}} \left(\frac{\partial g_k}{\partial T_l^{(m-1)}} + \sum_{l' \in \mathcal{N}(a)} \frac{\partial g_k}{\partial v_{l'}^{(m,t_*)}} \frac{\partial v_{l'}^{(m,t_*)}}{\partial T_l^{(m-1)}} \right), \quad (29)$$

for $nc \leq k < 2nc$, where:

$$\frac{\partial f_k}{\partial T_l^{(m-1)}} = \frac{-1}{\tilde{W}_{k,v}^2} \frac{\partial \tilde{W}_{k,v}}{\partial T_l^{(m-1)}}, \quad (30)$$

$$\begin{aligned} \frac{\partial g_k}{\partial T_l^{(m-1)}} &= \frac{-1}{\tilde{W}_{k,v}^2} (\bar{\mathbf{K}}_v^{(m,t_*)})^T \cdot \tilde{\mathbf{C}}_v \frac{\partial \tilde{W}_{k,v}}{\partial T_l^{(m-1)}} + \\ &\quad \frac{(\bar{\mathbf{K}}_v^{(m,t_*)})^T}{\tilde{W}_{k,v}} \cdot \frac{\partial \tilde{\mathbf{C}}_v}{\partial T_l^{(m-1)}}, \end{aligned} \quad (31)$$

and

$$\frac{\partial g_k}{\partial v_{l'}^{(m,t_*)}} = \begin{cases} 0 & k = l' \\ \frac{1}{\tilde{W}_{k,v}(T_l)} \frac{\partial (\bar{\mathbf{K}}_v^{(m,t_*)})^T}{\partial v_{l'}^{(m,t_*)}} \cdot \tilde{\mathbf{C}}_v(T_l) & k \neq l'. \end{cases} \quad (32)$$

Finally, the partial derivatives in 30 to 32 are calculated analytically based on (6) to (10).

Following a similar approach, the elements of the sub-matrix $\mathbf{J}_{a,T}^{(m,t)}$ can be calculated as follows:

$$\begin{aligned} \mathbf{J}_{a,T}^{(m,t)}(k, l) &= \frac{\partial \beta_{ak,T}^{(m,t)}}{\partial \beta_{al,v}^{(m-1)}} = \frac{\partial \beta_{ak,T}^{(m,t)}}{\partial v_l^{(m-1)}} \frac{\partial v_l^{(m-1)}}{\partial \beta_{al,v}^{(m-1)}} = \frac{1}{\alpha_{l,v}^{(m-1)}} \frac{\partial \beta_{ak,T}^{(m,t)}}{\partial v_l^{(m-1)}} \\ &= \frac{1}{\alpha_{l,v}^{(m-1)}} \left(\frac{\partial h_k}{\partial v_l^{(m-1)}} + \sum_{l' \in \mathcal{N}(a)} \frac{\partial h_k}{\partial T_{l'}^{(m)}} \frac{\partial T_{l'}^{(m)}}{\partial v_l^{(m-1)}} \right), \end{aligned} \quad (33)$$

for $0 \leq l < nc$, and:

$$\begin{aligned} \mathbf{J}_{a,T}^{(m,t)}(k, l) &= \frac{\partial \beta_{ak,T}^{(m,t)}}{\partial \alpha_{al,v}^{(m-1)}} = \frac{\partial \beta_{ak,T}^{(m,t)}}{\partial v_l^{(m-1)}} \frac{\partial v_l^{(m-1)}}{\partial \alpha_{al,v}^{(m-1)}} = \frac{-\beta_{l,v}^{(m-1)}}{[\alpha_{l,v}^{(m-1)}]^2} \frac{\partial \beta_{ak,T}^{(m,t)}}{\partial v_l^{(m-1)}} \\ &= \frac{-\beta_{l,v}^{(m-1)}}{[\alpha_{l,v}^{(m-1)}]^2} \left(\frac{\partial h_k}{\partial v_l^{(m-1)}} + \sum_{l' \in \mathcal{N}(a)} \frac{\partial h_k}{\partial T_{l'}^{(m)}} \frac{\partial T_{l'}^{(m)}}{\partial v_l^{(m-1)}} \right), \end{aligned} \quad (34)$$

for for $nc \leq l < 2nc$. Here, the chain rule and the following relation are used:

$$v_l = \frac{\beta_{l,v}}{\alpha_{l,v}} = \frac{1}{\alpha_{l,v}} \sum_{a' \in \mathcal{N}(l)} \beta_{a'l,v}, \quad (35)$$

which means $\partial v_l / \partial \beta_{al,v} = 1/\alpha_{l,v}$ and $\partial v_l / \partial \alpha_{al,v} = -\beta_{l,v}/\alpha_{l,v}^2$. Similar to 30 to 32, the sensitivities $\partial h_k / \partial T_{l'}$ and $\partial h_k / \partial v_l$ are calculated as follows:

$$\frac{\partial h_k}{\partial T_{l'}^{(m)}} = \begin{cases} 0 & k = l' \\ \frac{1}{\tilde{W}_{k,v}} \frac{\partial (\bar{\mathbf{K}}_T^{(m,t_*)})^T}{\partial T_{l'}^{(m)}} \cdot \tilde{\mathbf{C}}_T & k \neq l' \end{cases}, \quad (36)$$

$$\frac{\partial h_k}{\partial v_l^{(m-1)}} = \frac{\partial B_{k,T}}{\partial v_l^{(m-1)}} + \frac{1}{\tilde{W}_{k,v}} \frac{\partial (\bar{\mathbf{K}}_T^{(t_*)})^T}{\partial v_l^{(m-1)}} \tilde{\mathbf{C}}_T, \quad (37)$$

in which the partial derivatives are calculated analytically based on (6) to (10).

References

- [1] Amestoy, P., Buttari, A., L'Excellent, J.Y., Mary, T., 2019. Performance and Scalability of the Block Low-Rank Multifrontal Factorization on Multicore Architectures. *ACM Transactions on Mathematical Software* 45, 2:1–2:26.
- [2] Amestoy, P., Duff, I.S., Koster, J., L'Excellent, J.Y., 2001. A fully asynchronous multifrontal solver using distributed dynamic scheduling. *SIAM Journal on Matrix Analysis and Applications* 23, 15–41.
- [3] Audigier, C., Mansi, T., Delingette, H., Rapaka, S., Mihalef, V., Carnegie, D., Boctor, E., Choti, M., Kamen, A., Ayache, N., et al., 2015. Efficient lattice boltzmann solver for patient-specific radiofrequency ablation of hepatic tumors. *IEEE Transactions on Medical Imaging* 34, 1576–1589.
- [4] Balay, S., Abhyankar, S., Adams, M.F., Brown, J., Brune, P., Buschelman, K., Dalcin, L., Dener, A., Eijkhout, V., Gropp, W.D., Karpeyev, D., Kaushik, D., Knepley, M.G., May, D.A., McInnes, L.C., Mills, R.T., Munson, T., Rupp, K., Sanan, P., Smith, B.F., Zampini, S., Zhang, H., Zhang, H., 2020. PETSc Users Manual. Technical Report ANL-95/11 - Revision 3.13. Argonne National Laboratory. URL: <https://www.mcs.anl.gov/petsc>.
- [5] Balay, S., Abhyankar, S., Adams, M.F., Brown, J., Brune, P., Buschelman, K., Dalcin, L., Dener, A., Eijkhout, V., Gropp, W.D., Karpeyev, D., Kaushik, D., Knepley, M.G., May, D.A., McInnes, L.C., Mills, R.T., Munson, T., Rupp, K., Sanan, P., Smith, B.F., Zampini, S., Zhang, H., Zhang, H., 2019. PETSc Web page. <https://www.mcs.anl.gov/petsc>. URL: <https://www.mcs.anl.gov/petsc>.
- [6] Balay, S., Gropp, W.D., McInnes, L.C., Smith, B.F., 1997. Efficient management of parallelism in object oriented numerical software libraries, in: Arge, E., Bruaset, A.M., Langtangen, H.P. (Eds.), *Modern Software Tools in Scientific Computing*, Birkhäuser Press. pp. 163–202.
- [7] Bangerth, W., Hartmann, R., Kanschä, G., 2007. deal. ii—a general-purpose object-oriented finite element library. *ACM Transactions on Mathematical Software (TOMS)* 33, 24–es.
- [8] Berjano, E.J., 2006. Theoretical modeling for radiofrequency ablation: state-of-the-art and challenges for the future. *BioMed. Eng. Online* 5.
- [9] Cheong, J.K., Yap, S., Ooi, E.T., Ooi, E.H., 2019. A computational model to investigate the influence of electrode lengths on the single probe bipolar radiofrequency ablation of the liver. *Computer methods and programs in biomedicine* 176, 17–32.
- [10] El-Kurdi, Y., Dehnavi, M.M., Gross, W.J., Giannacopoulos, D., 2015. Parallel finite element technique using gaussian belief propagation. *Comput. Phys. Commun.* 193, 38–48.
- [11] El-Kurdi, Y., Gross, W.J., Giannacopoulos, D., 2012. Efficient implementation of gaussian belief propagation solver for large sparse diagonally dominant linear systems. *IEEE Trans. Magn.* 48, 471–474.
- [12] El-Kurdi, Y., Gross, W.J., Giannacopoulos, D., 2014. Parallel multi-grid acceleration for the finite-element gaussian belief propagation algorithm. *IEEE transactions on magnetics* 50, 581–584.

- [13] Gustafson, J.L., 1988. Reevaluating amdahl's law. *Communications of the ACM* 31, 532–533.
- [14] Hall, S.K., Ooi, E.H., Payne, S.J., 2015. Cell death, perfusion and electrical parameters are critical in models of hepatic radiofrequency ablation. *International Journal of Hyperthermia* 31, 538–550.
- [15] Ho, S., Niu, S., Fu, W., 2012. Design and analysis of novel focused hyperthermia devices. *IEEE Trans. Magn.* 48, 3254–3257.
- [16] Keyes, D.E., McInnes, L.C., Woodward, C., Gropp, W., Myra, E., Pernice, M., et al., 2013. Multiphysics simulations: Challenges and opportunities. *The International Journal of High Performance Computing Applications* 27, 4–83.
- [17] Mariappan, P., Weir, P., Flanagan, R., Voglreiter, P., Alhonnoro, T., Pollari, M., Moche, M., Busse, H., Futterer, J., Portugaller, H.R., et al., 2017. Gpu-based rfa simulation for minimally invasive cancer treatment of liver tumours. *International journal of computer assisted radiology and surgery* 12, 59–68.
- [18] Payne, S., Flanagan, R., Pollari, M., Alhonnoro, T., Bost, C., O'Neill, D., Peng, T., Stiegler, P., 2011. Image-based multi-scale modelling and validation of radio-frequency ablation in liver tumours. *Phil. Trans. R. Soc. A* 369, 4233–4254.
- [19] Pennes, H.H., 1998. Analysis of tissue and arterial blood temperatures in the resting human forearm. *Journal of applied physiology* 85, 5–34.
- [20] Pérez, J.J., Pérez-Cajaraville, J.J., Muñoz, V., Berjano, E., 2014. Computer modeling of electrical and thermal performance during bipolar pulsed radiofrequency for pain relief. *Medical physics* 41, 071708.
- [21] Petra, C.G., Schenk, O., Anitescu, M., 2014a. Real-time stochastic optimization of complex energy systems on high-performance computers. *Computing in Science & Engineering* 16, 32–42.
- [22] Petra, C.G., Schenk, O., Lubin, M., Gärtner, K., 2014b. An augmented incomplete factorization approach for computing the schur complement in stochastic optimization. *SIAM Journal on Scientific Computing* 36, C139–C162.
- [23] Reinhardt, M., Brandmaier, P., Seider, D., Kolesnik, M., Jenniskens, S., Sequeiros, R.B., Eibisberger, M., Voglreiter, P., Flanagan, R., Mariappan, P., et al., 2017. A prospective development study of software-guided radio-frequency ablation of primary and secondary liver tumors: Clinical intervention modelling, planning and proof for ablation cancer treatment (clinicimppact). *Contemporary clinical trials communications* 8, 25–32.
- [24] Rodgers, D.P., 1985. Improvements in multiprocessor system design. *ACM SIGARCH Computer Architecture News* 13, 225–231.
- [25] Solazzo, S.A., Liu, Z., Lobo, S.M., Ahmed, M., Hines-Peralta, A.U., Lenkinski, R.E., Goldberg, S.N., 2005. Radiofrequency ablation: importance of background tissue electrical conductivity—an agar phantom and computer modeling study. *Radiology* 236, 495–502.
- [26] Szabó, T., 2008. On the discretization time-step in the finite element theta-method of the discrete heat equation, in: *International Conference on Numerical Analysis and Its Applications*, Springer. pp. 564–571.
- [27] Tang, Y., Jin, T., Flesch, R.C., 2017. Numerical temperature analysis of magnetic hyperthermia considering nanoparticle clustering and blood vessels. *IEEE Trans. Magn.* 53, 1–6.
- [28] Trujillo, M., Berjano, E., 2013. Review of the mathematical functions used to model the temperature dependence of electrical and thermal conductivities of biological tissue in radiofrequency ablation. *International Journal of Hyperthermia* 29, 590–597.
- [29] Voglreiter, P., Mariappan, P., Pollari, M., Flanagan, R., Sequeiros, R.B., Portugaller, H.R., Futterer, J., Schmalstieg, D., Kolesnik, M., Moche, M., 2018. Rfa guardian: Comprehensive simulation of radiofrequency ablation treatment of liver tumors. *Scientific reports* 8, 787.