# Efficient Solver for Multi-Physics Modelling of Radiofrequency Tumour Ablation Using High Performance Computing

**Amir Akbari**

Presented in partial fulfillment of the requirements of the degree of Doctor of Philosophy

Department of Electrical & Computer Engineering, McGill University,
Monréal, Québec, Canada

August 2021

# Abstract

Radiofrequency ablation (RFA) is the preferred choice when minimally invasive methods are recommended for liver cancer treatment. Being categorized under local hyperthermia, RFA destroys tumours without surgery and instead uses high temperatures to *cook* the cancerous cells. This method works effectively when the target area is kept within an exact temperature range for a certain duration. However, this goal is not always easy to achieve as accurate measurement of the temperature inside a tumour is difficult. In addition, in particular for liver tumours, the outcome of RFA treatment is challenged by the presence of hepatic blood vessels dissipating heat and changing the size and shape of the ablation zone. Nonetheless, prediction of the ablation zone is feasible with the help of mathematical modelling and computer simulation. Such simulations must be fast and accurate to be practical in clinical procedures.

We first overview the mathematical models of RFA of hepatic tumours and the multi-physical aspects of these models. In order to reduce the simulation time, the multi-physics model must be implemented on high performance parallel computers. We explain the limitations of the finite element method (FEM) for solving multi-physics problems on parallel computers and then present the finite element Gaussian belief propagation (FGaBP) method as an alternative to FEM for parallel multi-physics simulations.

In a second step, we focus on parallel implementation of FGaBP for the solution of the multi-physics problems emerging in RFA models. Both weak and strong coupling approaches are considered. For the strong coupling, the Newton-Raphson (NR) method is implemented in parallel by calculating local Jacobian matrices for each finite element instead of explicitly forming a global Jacobian matrix. This provides an NR algorithm amenable to different parallel computing architectures. The algorithm is tested on shared-memory and distributed-memory architectures and compared to highly optimized parallel solvers, indicating faster solution for large-scale problems.

The outcome of RFA simulations depends on various physiological parameters of the liver, the exact values of which are unknown. Typically, RFA simulations use average values for these parameters taken from the literature, resulting in a deterministic ablation outcome. In contrast, one can implement a stochastic simulation model, where the input

parameters are modelled as random variables. The stochastic approach computes how the uncertainty in the parameters propagates to the simulation output. In the last chapter, a variant of the belief propagation algorithm called non-parametric belief propagation is utilized in order to develop a novel sample-based solver for uncertainty analysis of RFA.

# Résumé

L'ablation par radiofréquence (ARF) est le choix de préférence lorsque des méthodes minimalement invasives sont recommandées pour le traitement du cancer du foie. Classée dans la catégorie de l'hyperthermie locale, la ARF détruit les tumeurs sans chirurgie et utilise plutôt des températures élevées pour cuire les cellules cancéreuses. Cette méthode fonctionne efficacement lorsque la zone cible est maintenue dans une plage de température exacte pendant une certaine durée. Cependant, cette mission n'est pas toujours facile à réaliser tant la température à l'intérieur d'une tumeur est difficile à mesurer avec précision. De plus, pour les tumeurs du foie, le résultat du traitement par ARF est remis en cause par la présence de vaisseaux sanguins hépatiques dissipant la chaleur et modifiant la taille et la forme de la zone d'ablation. Néanmoins, la prédiction de la zone d'ablation est réalisable à l'aide de la modélisation mathématique et de la simulation informatique. De telles simulations doivent être rapides et précises pour être praticables dans les procédures cliniques.

Nous avons d'abord passé en revue les modèles mathématiques de ARF des tumeurs hépatiques et les aspects multi-physiques de ces modèles. Afin de réduire le temps de simulation, le modèle multi-physique doit être implémenté en utilisant le calcul parallèle. Nous expliquons les limites de la méthode des éléments finis (MEF) pour résoudre des problèmes multi-physiques sur des ordinateurs parallèles, puis présentons la méthode de Gaussian belief propagation (GaBP) comme alternative à la MEF pour les simulations multi-physiques parallèles.

Dans un deuxième temps, nous concentrons sur l'implémentation parallèle de la GaBP pour la solution des problèmes multi-physiques émergeant dans les modèles ARF. Les approches de couplage faible et fort sont considérées. Pour le couplage fort, la méthode de Newton-Raphson (NR) est mise en œuvre en parallèle en calculant des matrices jacobiennes locales pour chaque élément fini au lieu de former explicitement une matrice jacobienne globale. Cela fournit un algorithme NR compatible avec différentes architectures de calcul parallèles. L'algorithme est testé sur des architectures à mémoire partagée et à mémoire distribuée et comparé à des solveurs parallèles hautement optimisés, indiquant une solution plus rapide pour les problèmes à grande échelle.

Le résultat des simulations ARF dépend de divers paramètres physiologiques du foie dont les valeurs exactes sont inconnues. En règle générale, les simulations ARF utilisent des valeurs moyennes pour ces paramètres tirées de la littérature, ce qui entraîne un résultat déterministe d'ablation. En revanche, on peut implémenter un modèle de simulation stochastique où les paramètres d'entrée sont modélisés comme des variables aléatoires. L'approche stochastique calcule à quel point l'incertitude dans les paramètres se propage au résultat de la simulation. Dans le dernier chapitre, une variante de l'algorithme de belief propagation appelée non-parametric belief propagation est utilisée pour développer un nouveau solveur basé sur des échantillons pour l'analyse d'incertitude de ARF.

# Acknowledgements

I would like to express my sincere gratitude to my supervisor Prof. Dennis Giannacopoulos for his guidance and endless support during my PhD studies. He always encouraged me to pursue my own ideas and to conduct my own research, which helped me become a better researcher. I would also like to thank my supervisory PhD committee members, Prof. Maryam Mehri Dehnavi and Prof. Milica Popovich, for their valuable suggestions and constructive feedback on my research project. Lastly, my deepest gratitude goes to my family who have supported me in every way possible throughout my time at McGill and my entire life.

# Contents

# List of Figures

# Acronyms

**BP** Belief Propagation 30–32, 81, 82, 91

**CPU** Central Processing Unit 23, 29, 37, 49, 50, 63, 72

**DDM** Domain Decomposition Method 40, 41

**DRAM** Dynamic Random-access Memory 50, 63, 72

**FEM** Finite Element Method 7, 9, 23–26, 29, 30, 32, 33, 39, 42, 45, 54, 55, 61, 68, 78, 85, 87, 89, 90, 92

**FGaBP** Finite Element Gaussian Belief Propagation 9, 25, 26, 30, 32, 33, 35–39, 41, 43–59, 61–63, 68–72, 74, 92–94

**FMGaBP** Finite Element Multigrid Gaussian Belief Propagation 37, 38

**FN** Factor Node 32–35, 37, 45, 46, 56, 58, 69

**GaBP** Gaussian Belief Propagation 8, 31, 32, 82, 83, 86

**GMM** Gaussian Mixtures Models 87, 88

**GPU** Graphics Processing Unit 23, 24

**HCC** Hepatocellular Carcinoma 1

**HPC** High Performance Computing 7, 8, 30, 35

**JFNK** Jacobian-free Newton-Krylov 22, 27, 28, 53, 54, 93, 94

**K-L** Karhunen-Loeve 80

**LBM** Lattice-Boltzmann Method 23, 24

**MCS** Monte Carlo Simulation 78, 79, 89–91, 94

**MPI** Message Passing Interface 23, 66, 67, 70, 72, 73

**NBP** Non-parametric Belief Propagation 82–84, 87, 90, 91

**NKS** Newton-Krylov-Schwarz 28

**NR** Newton-Raphson 7, 9, 26, 27, 30, 35, 53–55, 57–63, 65–69, 71, 74, 93, 94

**ODEs** Ordinary Differential Equations 17

**PC** Polynomial Chaos 81

**PDE** Partial Differential Equation 10, 24, 26, 44, 45, 76, 77, 80, 81, 84

**PDF** Probability Density Function 31, 32, 86

**PETSc** Portable, Extensible Toolkit For Scientific Computation 66

**RF** Radio Frequency 5

**RFA** Radiofrequency Ablation 2–9, 11, 12, 14, 15, 17, 18, 22–26, 44, 53, 72, 75, 76, 84, 92, 94

**RMS** Root-Mean-Square 44, 49, 63, 72

**SFEM** Stochastic Finite Element Method 9, 10, 24, 25, 76–79, 81, 84, 94

**SoS** Speed Of Sound 4

**SpMV** Sparse Matrix-vector Multiplication 27–30, 39, 53

**SSFEM** Spectral Stochastic Finite Element Method 25, 80

**VN** Variable Node 32–35, 45, 46, 56, 57

**WK** Wulff-Klinger 13

# Chapter 1

# Introduction

## 1.1 Motivation

### 1.1.1 Clinical context

Liver cancer is ranked by the world health organization (WHO) as the fourth deadliest cancer worldwide. Hepatocellular carcinoma (HCC) is the most common type of primary liver cancer (more than 1 million cases per year) and a leading cause of cancer-related death in many parts of the world, with an increasing rate in Western countries [1]. Although the disease burden [1] of many other major cancers is decreasing, the overall burden of liver cancer worldwide is increasing over time. In fact, the incidence rates of HCC in the United States have increased twofold to threefold over the past three decades [2].

Different treatment strategies for HCC have evolved significantly over the past few decades, among which surgical resection, liver transplantation, and local ablation are considered the most effective. Surgical resection removes part of the tissue that is known to be cancerous, while transplantation replaces the whole organ with a healthy one from another body. Local ablation induces tumour necrosis by delivering heat directly into tumours and is potentially effective for patients with early-stage HCC [1].

[1] The impact of a disease as measured by various indicators such as mortality, morbidity or financial cost.

Even though surgical resection and liver transplantation provide the best long-term survival, only less than 25% of patients with liver cancer qualify for them [3]. This could be due to different factors such as health condition and age of the patient, or the location, size, and number of tumours. Regarding tumour location for instance, proximity to vital organs like the vascular structure is a problem in resection. In addition, patients with multiple tumours might have inadequate hepatic reserve to tolerate resection [3]. On the other hand, liver transplantation is limited by the number of donors. Consequently, minimally invasive ablative therapies such as radiofrequency ablation (radiofrequency ablation (RFA)), high intensity focused ultrasound (HIFU), cryotherapy, microwave ablation, and laser ablation have gained growing interest for the treatment of liver tumours. Ablation therapies can also be used during the waiting time prior to transplantation to prevent tumour progression [1].

Among the minimally invasive treatments for liver cancer, RFA has become the preferred treatment option for patients who are not candidates for standard surgery or are in an early tumour stage. There is clinical evidence that RFA could provide equivalent outcome as compared to resection for treatment of tumours with a diameter smaller than 2-2.5 cm [4]. Moreover, patient recovery after RFA is faster and post-procedure quality of life is higher after RFA than after surgical resection [4]. Repeat treatment is also easier in RFA compared to resection [5]. Compared to microwave ablation which is typically performed at frequencies of either 915 MHz or 2.45 GHz [6], RFA is the more established thermal technique, specially for lesions with a diameter smaller than 2–2.5 cm, and is the safer option because of the less aggressive heat production and better ablation zone predictability [7].

The differences between RFA and microwave ablation arise from the heating mechanism. RFA relies on resistive heating sourced by electrical current passing through the surrounding tissue. On the other hand, microwave ablation heats the tissue through dielectric polarization by applying an electromagnetic field of either 915 or 2450 MHz to the tissue around the ablation probe. Due to its direct heating mechanism, the microwave ablation leads to larger ablation zones than RFA, that are created more quickly.

RFA is a minimally (compared to surgery) invasive therapy with application in various medical fields, such as the elimination of cardiac arrhythmias, or the destruction of tumours in different locations including liver, kidney, lung, bone, prostate, and breast. It uses devices operating between 460-550 kHz delivering electrical currents to biological tissues to thermally damage a tumour by raising its temperature to approximately 100 °C for a period of 10-15 minutes. RFA generally uses a pair of electrodes: an active electrode with a small surface area that is placed in the target zone, and a larger dispersive electrode to close the electrical circuit [8]. Different types of active electrodes can be used, such as the cool-tip single electrode or the RITA probe (see Fig. 1.1) with three, four or six umbrella shaped prongs which can be deployed within the tumour. During RFA, the active probe is inserted into the target zone, usually under ultrasound guidance, as shown by Fig. 1.2.

RFA is the most widely used minimally invasive approach for liver cancer treatment; however, there is still a need for training tools for the less-experienced physicians to improve the treatment outcome. The survival rates significantly depend on the experience of the physician, i.e., experience of 0–2 years brings 46%-69% survival, while rates of 3–4 years experience corresponds to 89%-92% [4]. Such a learning curve exists since effective RFA is expected to entirely destroy the tumour with a safety margin of damaged healthy tissue in its vicinity, which in turn relies on the extent of the ablation zone. Consequently, the temperature needs to be carefully monitored, leading to the halt of radio frequency current when the clinical endpoint is reached. However, for treating tumours in the liver, the ablation zone is difficult to control as the hepatic blood vessels dissipate heat and change the size and shape of the lesion zone.

This challenge can result in an imperfect outcome of the treatment. Indeed, based on clinical evidence, a noteworthy mismatch between expected and observed ablation zones exists, which leads to lower survival rates due to over-treatment with severe injuries (up to 9%) or under-treatment with tumour recurrence (up to 40%)[9]. In addition, treatment of tumours larger than 3 cm needs a larger lesion zone whose shape is even harder to predict and potentially leads to a higher local recurrence and a lower survival rate [10]. Further, patient-specific factors, such as blood perfusion and location of the tumour, and device-specific parameters, such as delivery of power, strongly influence the lesion zone [4].

In order to address the concerns related to the misprediction of the lesion zone,

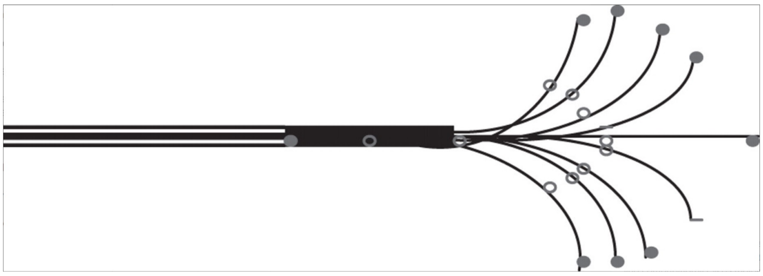

Figure 1.1: RITA probe representation from [11]

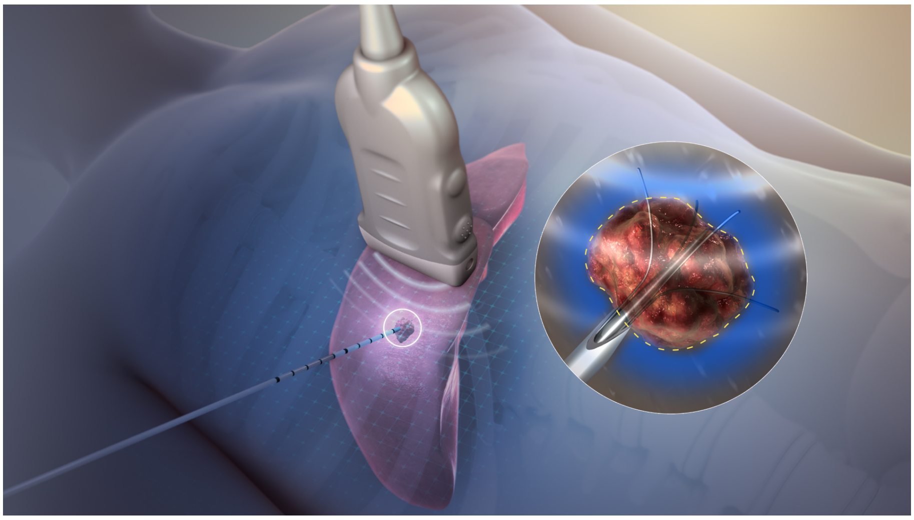

Figure 1.2: Ultrasound-guided RFA using RITA probe from [12]

temperature monitoring during RFA treatment seems inevitable. To this end, both invasive and non-invasive temperature measurement techniques are applicable. Non-invasive methods include ultrasound imaging, computed tomography (CT) and magnetic resonance imaging (MRI). Using these methods, ablation data is reported as image-based thermal maps. Due to the recent advances in data analysis, ultrasound has particularly become popular among non-invasive temperature monitoring methods in RFA. As the temperature increases in the tissues, the changes in speed of sound (SoS) result in echo time shifts. In the past decade, using the echo time shift to visualize temperature variation during RFA has gained the most attention. For this method to be accurate, a coefficient parameter, $k$, which is used to describe the relation between tissue temperature and changes in SoS needs to be estimated [13]; however the estimation of $k$ is challenging and is still an active research topic. In addition, the main issue of non-invasive methods is that the results are not provided in real time which makes it difficult for the physician to assess completion of the ablation procedure.

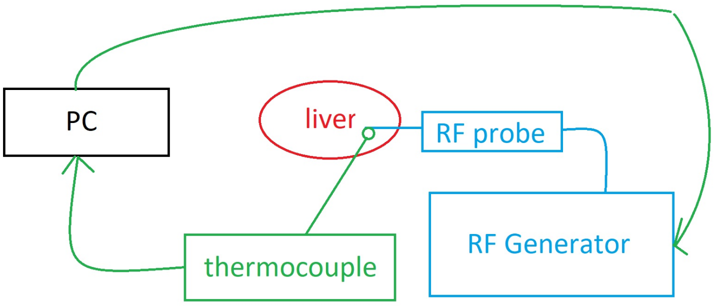

Figure 1.3: Combining invasive temperature monitoring and computer simulation.

The second group of temperature monitoring techniques is invasive methods. Here, invasive sensors are inserted directly into the target zone to measure the temperature during RFA. While being cost-effective compared to non-invasive methods, invasive sensors don't respond as fast to the abrupt temperature changes found in RFA. Additionally, radio frequency (RF) electromagnetic fields could be induced in the circuitry of the thermometer affecting the accuracy of invasive thermometry techniques [8].

Current temperature measurement methods are either incapable of providing real time information or are more or less invasive. Theoretical models and computer simulations are powerful non-invasive tools providing information on the electrical and thermal behaviour of RFA rapidly and at low cost. In recent years, the use of computational models for RFA has gained interest for studying the heat transfer profiles surrounding the ablation probes. Physicians of any experience can use software simulation of RFA and visualize the treatment outcome in advance. In addition, patient-specific factors such as the heat sink effect of hepatic vessels, or the amount of porous tissue perfusion, affect the heat transfer and thereby shape and size of the lesion. Contrast Enhanced Computed Tomography (CECT) allows determination of the vascular anatomy inside the liver and quantification of the liver perfusion. This enables a reliable treatment planning based on simulation models that can be adapted to the physiology of each patient. Finally, computer models can be used alongside other temperature monitoring methods; for instance, Fig. 1.3 shows an illustration of combining an invasive temperature measurement and computer simulation. The measured temperature is sent as an input to the computer program, leading to a more accurate simulation of RFA.

### 1.1.2 Role of high-performance computing

Real-time simulation is getting increasingly important in the prediction of tissue temperature and associated thermal damage in RFA. It has been claimed that real-time prediction of thermal dose could allow precise control of the ablation zone which in turn leads to a better control over the thermal damage to the target tissue without exposing the neighbouring healthy tissue to excessive thermal damage [14]. Also, it helps to avoid recurrence of tumours due to insufficient thermal exposure at target zones [14].

Development of real-time three-dimensional RFA simulation is a challenging task. Many of the current RFA simulation environments are only applicable to pre-treatment predictive analysis since they are focused on numerical accuracy, convergence, and stability, rather than computation time [14]. This is mainly due to the large computational cost of RFA simulation. As will be discussed in Chapter 2, RFA simulation involves solving a nonlinear system of equations in each time step. The nonlinear system of equations is usually solved iteratively based on the Newton–Raphson method. At each iteration, a linear system of equations is solved by either directly computing the inverse of the system matrix or iteratively solving a system of algebraic equations starting from an initial guess. Either way, the numerical simulation of RFA is computationally expensive.

Current approaches for computational simulation of RFA are either too computationally expensive or are based on an over simplified model. Simplification is especially applied to the electrical aspects of the problem. The electrical modelling 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 [15] 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 [11] and used in later RFA simulation frameworks [16], [4], and [17]. In this approach, the power deposition is estimated using a Gaussian distribution around the probe tip. However, these approximations are too much of a simplification as it has been shown by [18] that electrical parameters are critical in models of hepatic RFA.

In order to simulate RFA therapies and predict the extent of the ablation zone, a multi-physics approach that combines the electrical-thermal heating process with a biophysical model of the heat transfer and cellular necrosis is needed. Multi-physics modelling problems generally involve a large number of variables and a complex unstructured mesh which leads to large coupled sets of highly non-linear equations. Therefore, the numerical solution is extremely challenging. As such, the multi-physics software tools need to be implemented on high performance parallel computers. The traditional method to simulate RFA is the finite element method (FEM). 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, 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. For these reasons, a high performance computing (HPC) algorithm for the assembly and solution of strongly coupled multi-physics problems provides great potential for fast and precise RFA simulation.

### 1.1.3 Objectives

The necessity of time-efficient and accurate calculation of the ablation zone on the one hand and the computational cost of the RFA multi-physics modelling on the other hand makes using HPC for RFA simulation an inevitable choice. Within this scope, the main objective of this work is to present a methodology for the multi-physics simulation of RFA procedure and exploit HPC to speed up the simulation time; specifically, as pointed out in the previous section, to address the coupled electrical-thermal problem which has been approximated in the literature because of its high computational cost.

Another important aspect of RFA modelling is dealing with uncertainties. As will be discussed in Chapter 2, the outcome of the RFA simulations depends on various physiological parameters of the liver whose exact values are unknown. Typically, RFA simulations use average values for these properties taken from the literature, resulting in a deterministic simulation outcome. In contrast, one can implement a stochastic simulation model, where the input parameters are modelled as random variables. The stochastic approach computes how the uncertainty in the model parameters propagates to the simulation output, i.e., the ablation zone.

Based on the above discussion, the main questions that we aim to answer in this thesis, are:

1. How to simulate the RFA of hepatic tumours while taking into account the coupled phenomena occurring in the procedure?
2. How to incorporate HPC in our method in order to run the simulation in computational time?
3. How to take into account parameter uncertainty in the simulation?

## 1.2 Main contributions and manuscript organization

### 1.2.1 Main contributions

In this work, three main contributions to the modelling of RFA are proposed as follows:

1. A computational framework based on the Gaussian belief propagation (GaBP) method for weak coupling and strong coupling modelling of the electrical-thermal mechanism involved in RFA.

2. A new parallel implementation of the coupled equations using GaBP method on HPC architectures.

3. A new approach for uncertainty analysis of the ablation zone based on stochastic modelling of input parameters and the GaBP algorithm for continuous variables.

### 1.2.2 Organization of the thesis

This thesis focuses on RFA modelling for hepatic tumours and the associated coupled problems. Chapter 2 introduces the background on the computational modelling of radiofrequency ablation. It also presents the state-of-the-art models used in the literature, highlights their limits and further how those limits are addressed in our choice of RFA modelling.

Chapter 3 provides a review of FEM and its parallel issues for multi-physics problems. The finite element Gaussian belief propagation (FGaBP) algorithm as a parallelizable stochastic variant of FEM is then introduced. The ability of FGaBP to deal with multi-physics problems is also discussed in Chapter 3.

Chapter 4 presents our first contribution based on the FGaBP method introduced in Chapter 3. A new FGaBP algorithm for multi-physics applications is presented in this chapter. The FGaBP method is first modified for solving the heat transfer problem and then verified against an analytical solution. Combined with a Gauss-Seidel algorithm, FGaBP is used to solve the coupled electrical-thermal problem that emerges in RFA of hepatic tumours. The multi-physics FGaBP algorithm is tested on a multi-core personal computer (PC) whit OpenMP used for parallel implementation.

In Chapter 5, the strongest form of coupling algorithms, which is the NR method, is implemented in parallel using the localized computations of FGaBP. 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. In order to verify the correctness of the algorithm, results are compared with COMSOL Multi-physics software [19], showing good fidelity. 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.

Chapter 6 exploits the stochastic nature of the belief propagation algorithm to model the uncertainties involved in RFA simulations. These uncertainties largely result from various physiological parameters of the liver whose exact values are unknown. Taking the randomness of the simulation parameters into consideration, the dependence of the simulation outcome with respect to parameter variations must be evaluated. Such analysis computes how the uncertainty in the model parameters propagates to the simulation output, allowing the physician to estimate the expected range of the treatment outcome. The stochastic finite element method (SFEM) is explicitly developed for solving stochastic partial differential equation (PDE)s that naturally arise from engineering problems with uncertainties. In the last chapter, a variant of belief propagation method called non-parametric belief propagation is utilized in order to develop a novel sample-based solver for stochastic PDEs. The algorithm is different from SFEM in that it doesn't need analytical description of the randomness. Rather, we use samples from the stochastic parameters of the system to locally approximate the uncertainty propagation to the solution.

# Chapter 2

# Computational Models of Hepatic RFA

Research groups have used theoretical models and computer simulations to study different aspects of RFA. Various probe configurations, including single probe [20], RITA probe [21], monopolar probe [22] and bipolar probe [23], probe placement planning [24] and even the effect of the breathing motion on probe positioning [25] are studied in the literature. Special cases such as when the tumour is close to a hepatic vessel [26] or near to the liver boundary [27] are also investigated in the literature. On the other hand, most of the research conducted on computer simulation of RFA focus on its mathematical modelling. These models can be divided in three categories: the electrical models, the tissue bio-heat models, and the cellular necrosis models. This chapter presents a literature review on the computational models of hepatic RFA.

## 2.1 Electrical heating and bio-heat models

At the frequencies employed in RFA (460 kHz – 550 kHz), the wavelength is several orders of magnitude larger than the size of the probes. Thus, the tissue can be considered purely resistive and the electrical power is deposited within a small area around the probe tip through electrical conduction [8]. Consequently, the displacement currents are negligible and a quasi-static electrical conduction model is assumed, which allows us to obtain the electric potential around the probe using Laplace's equation:

$$\nabla \cdot (\sigma(T)\nabla v) = 0, \tag{2.1}$$

where $\sigma(T)$ is the temperature-dependent electrical conductivity (S/m), and $v$ is the electric potential (V).

Even though solving Laplace's equation is theoretically accurate for the electrical problem, it requires a fine volume mesh on the surface of the very thin probe tips and therefore suffers from a high computational cost. Thus, Audigier *et al.* [15] model the electrical heating with a Dirichlet boundary condition on a sphere, whose radius is defined pre-operatively by the protocol followed by the clinician. Another approach to approximate the Joule heating is proposed by [11] and used in later RFA simulation frameworks [16, 4, 17]. In this approach, the power deposition is estimated using a Gaussian distribution around the needle tip. Nevertheless, approximation of the Joule heating instead of solving the electrical problem is too much of simplification as it has been shown by [18] that electrical parameters are critical in models of hepatic RFA.

The most commonly used model for heat transfer in the tissue is the Pennes model:

$$\rho_{ti} c_{ti} \frac{\partial T}{\partial t} = Q + \nabla \cdot (d\nabla T) + H(T_{bl} - T), \tag{2.2}$$

where $T$ is the temperature (K), $c_{ti}$ is the special heat capacity of tissue (J/kg/K), $\rho_{ti}$ is the tissue density (kg/m$^3$), $d$ is the thermal conductivity (W/m/K), $H$ is the convective transfer coefficient (W/m$^3$/K) and $T_{bl}$ is the baseline physiological blood temperature taken to be 310 K. On the right hand side of (2.2), $Q = \sigma|\nabla v|^2$ is the heat source (W/m$^3$) which depends on the electric potential. On the other hand, in the electrical problem, $\sigma$ 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 [15]. Moreover, this model assumes that the blood acts as a spatially homogeneous heat sink inside the tissue; however, studies [28] demonstrate that since liver is a highly vascularized organ, the perfusion does not act as a uniform heat sink, and so the basic assumption behind the Pennes model is not valid when applied to the liver. For these reasons the Wulff-Klinger (WK) model is proposed for a highly perfused organ with small vessels:

$$\rho c_{ti} \frac{\partial T}{\partial t} = Q + \nabla \cdot (d \nabla T) - \epsilon \rho c_{bl} \mathbf{v} \cdot \nabla T, \tag{2.3}$$

where $\epsilon$, $c_{bl}$ and $\mathbf{v}$ stand for blood volume fraction (fraction of blood volume over total volume), the special heat capacity of blood (J/kg/K) and blood velocity (m/s), respectively. Note that the last term in (2.3) accounts for the directional effect of the blood flow and so the WK model is more accurate than the Pennes model where the blood velocity magnitude is low [15].

Combining the Pennes model and the WK model, Payne et al [11] derived a more general model for computing heat diffusion in biological tissues, where each elemental volume contains both tissue and blood with a certain fraction:

$$(1-\epsilon)\rho c_{ti} \frac{\partial T_{ti}}{\partial t} = \underbrace{(1-\epsilon)Q}_{source} + (1-\epsilon)\underbrace{\nabla \cdot (d \nabla T_{ti})}_{diffusion} + H(T_{bl} - T_{ti}), \tag{2.4a}$$

$$\epsilon \rho c_{bl} (\frac{\partial T_{bl}}{\partial t} + \underbrace{\mathbf{v}.\nabla T_{bl}}_{advection}) = \epsilon Q + \epsilon \nabla \cdot (d \nabla T_{bl}) - H(T_{bl} - T_{ti}). \tag{2.4b}$$

In these two coupled equations, subscripts '*ti*' and '*bl*' stand for tissue and blood, respectively. Although large vessels can be clearly identified using state-of-the-art medical imaging, small capillaries are still difficult to image. As a result, modern medical imaging does not provide an accurate measurement of the ratio between blood and liver tissue [15]. Because of this, Audigier et al [15] employ a two-compartment model: Pennes model is used when a point belongs to a large vessel and WK model is used when a point belongs to the parenchyma. Assuming no interaction between large blood vessels and surrounding tissue, the temperature is computed by solving the diffusion equation:

$$\rho c_{ti} \frac{\partial T}{\partial t} = Q + \nabla \cdot (d \nabla T) \tag{2.5}$$

everywhere in the domain, to which they add either the cooling term, $H(T_{bl} - T)/(1-\epsilon)$, for points residing inside a large vessel or the tissue perfusion term, $-\varepsilon \rho c_{bl} \mathbf{v} \cdot \nabla T/((1-\varepsilon)$, when it belongs to the parenchyma.

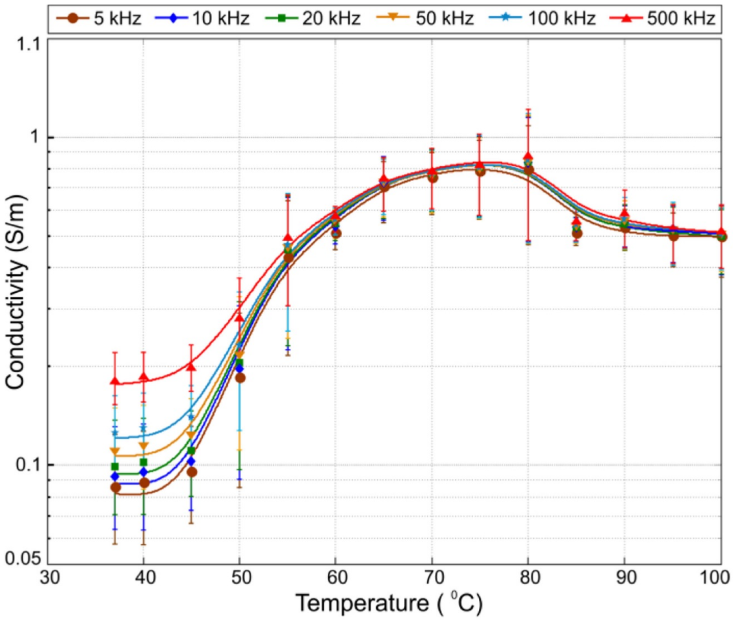

Figure 2.1: Electrical conductivity of ex-vivo porcine liver tissue as a function of temperature at six frequencies from 5 to 500 kHz in slow heating [30].

Besides the different types of equations used to describe the electrical and thermal phenomena in RFA, evaluating the parameters in these equations has been studied extensively in the literature. As for the electrical conductivity, there are two general ways of introducing $\sigma$ in RFA models: (1) using constant values from the measurement data reported in scientific literature, and (2) using mathematical functions which reflect the dependence of $\sigma$ on temperature. The most widely used mathematical function for $\sigma(T)$ in RFA models is a piecewise function which uses different mathematical expressions for different temperature ranges [29]. At temperatures below 100 °C, $\sigma(T)$ increases with temperature, with increase rate from 1.5%/°C to 2%/°C, and increase type of either linear or exponential. At temperatures above 100 °C, vaporization occurs that involves a more or less abrupt drop of $\sigma(T)$. Two to four orders of magnitude drop for temperature between 100 °C and 105 °C are reported by [29]. In addition, [30] studied the effect of different heating rates on $\sigma(T)$. Various heating rates at different frequencies were applied in an ex vivo RFA experiment on native porcine liver tissue. They observed different trends in $\sigma(T)$ for the slow and fast heating rates above 60 °C. The temperature dependency of $\sigma(T)$ at different frequencies and slow heating is shown by Fig. 2.1.

As for the thermal conductivity, $d$, its values is assumed constant in most RFA

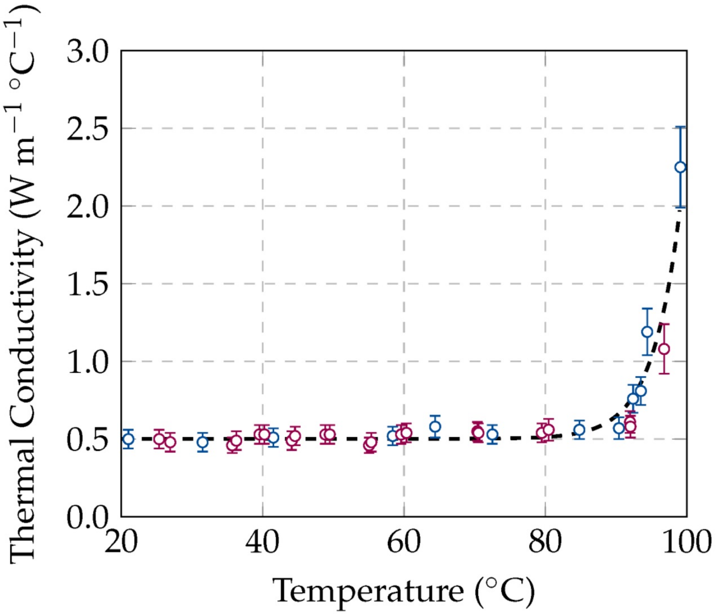

Figure 2.2: Thermal conductivity of ex vivo ovine liver [31].

theoretical models, owing to the fact that changes in $d$ with temperature are not so noticeable as in $\sigma$ [32]. Silva *et al.* [31] conducted ex vivo experiments on ovine liver and observed no changes in thermal conductivity in the range of below 90 °C. As depicted in Fig. 2.2, at higher temperatures approaching the water vaporisation process (at 100 °C), an increase of the thermal conductivity values was recorded. Some models use a piecewise continuous function to consider the temperature dependency of $d$ between 90 °C and 100 °C , e.g., a linear increase of $d$ up to 100 °C, and a constant value above 100 °C [29]. A few studies considered two different values for $d$ before and after water vaporisation temperature [33], modelling the temperature dependence with a non-continuous piecewise function. The typical numerical values of the electrical and thermal model parameters at normal body temperature, i.e. 37 °C, are provided in Table 2.1.

## 2.2 Cellular necrosis models

RFA treatment aims at delivering heat energy into the cancerous tumor while creating a zone of dead cells in its vicinity. The damaged tissue will eventually recover after the ablation procedure. Because of the thermal tolerance of cells, a temperature threshold should be exceeded for a certain time duration for a cell to go to a damaged state. Different cellular necrosis models have been investigated in the literature. The simplest

Table 2.1: Values of electrical and thermal model parameters used in the literature.

| **Parameter** | **value** | **Reference** |
|---|---|---|
| Baseline liver electrical conductivity, $\sigma_{ti}$ (S/m) | 0.33 | [23] |
| Baseline tumor electrical conductivity, $\sigma_{tu}$ (S/m) | 0.640 | [34] |
| Probe electrical conductivity, $\sigma_{probe}$ (S/m) | $10^8$ | [23] |
| Shaft electrical conductivity, $\sigma_{shaft}$ (S/m) | $4 \cdot 10^6$ | [23] |
| Liver and tumor density, $\rho_{ti}$ (kg/m$^3$) | 1060 | [11] |
| Liver and tumor heat capacity, $c_{ti}$ (J/kg· K) | 3600 | [11] |
| Baseline liver thermal conductivity, $d_{ti}$ (W/m· K) | 0.512 | [11] |
| Baseline tumor thermal conductivity, $d_{tu}$ (W/m· K) | 0.640 | [34] |
| Blood density, $\rho_{bl}$ (kg/m$^3$) | 1000 | [23] |
| Blood heat capacity, $c_{bl}$ (J/kg· K) | 4180 | [11] |
| Probe density, $\rho_{probe}$ (kg/m$^3$) | 6450 | [23] |
| Probe heat capacity, $c_{probe}$ (J/kg· K) | 840 | [23] |
| Probe thermal conductivity, $d_{probe}$ (W/m· K) | 18 | [23] |
| Shaft density, $\rho_{shaft}$ (kg/m$^3$) | 21,500 | [23] |
| Shaft heat capacity, $c_{shaft}$ (J/kg· K) | 132 | [23] |
| Shaft thermal conductivity, $d_{shaft}$ (W/m· K) | 71 | [23] |

model is to assume a single critical temperature and a two-state model for cell death; Below the critical temperature, cells are alive and 100% functional, while above it cells are dead with 0% functionality. The most widely used value for this temperature threshold is 50 °C. The ablation zone is then defined by the volume enclosed by the 50 °C isothermal surface [20].

Tissue damage is a function of both temperature and the duration for which cells are at high temperatures [35]; however, the latter is not considered in the isothermal model. For this reason, the Arrhenius equation given by [36] is also used in the literature:

$$\Omega(\mathbf{r}, t) = \int_0^t A \cdot exp(-\frac{\Delta E}{RT(\mathbf{r}, t)})dt. \tag{2.6}$$

In the Arrhenius equation, $\Omega$ is called thermal damage and quantifies the amount of thermal coagulation inside the tissue, $\mathbf{r}$ is the position vector (m) of any point inside the tissue, $t$ is the exposure time (s) , $A$ is the liver frequency factor ($7.39 \times 10^{39}$ s$^{-1}$[23]), $\Delta E$

is the liver activation energy ($2.58 \times 10^5$ J/mol [23]) and $R$ is the universal gas constant (8.31 J/mol $\cdot$ K [8]). The Arrhenius equation represents the percentage probability of cell survival at a specific area due to thermal damage, where $\Omega = 1$ is commonly used to represent 63% of cell death [37] and $\Omega = 4.6$ signifies 99% of cell death [38]. The main drawback of Arrhenius-based models is that they are sensitive to small changes in parameters, and they are not able to predict cellular injury over a wide temperature range [15].

Extending the models that are based on two states (alive and dead), a three-state model that includes an intermediate compartment between the fully alive and dead states is proposed by O'Neill *et al.* [35]. The intermediate compartment is called a vulnerable state from which cells can either go to the dead state or heal and return to the alive state. This model consists of a system of two coupled ordinary differential equations (ODEs) with the constraint $A + V + D = 1$, where the proportions of alive, vulnerable and dead cells are given by $A$, $V$ and $D$ respectively. The system of ODEs is:

$$\frac{dA}{dt} = -k_f A + k_b[1 - A - D], \tag{2.7a}$$

$$\frac{dD}{dt} = k_f[1 - A - D], \tag{2.7b}$$

where $k_f = \overline{k_f} e^{T/T_k}[1 - A]$. The model has three parameters: $\overline{k_f}$, $k_b$ and $T_k$ which are all obtained by fitting to experimental data [35]. In addition, the initial conditions are usually chosen as $A = 0.99$, $V = 0.01$ and $D = 0$ [15]. In this model, thermal-based necrosis is a function of both temperature and time. It also defines an intermediate state that allows to consider the recovery process of the cells. The main drawback of the three-state model is that it requires to estimate several parameters.

Models based on a a parameter known as the thermal dose have also been suggested in the literature. These models are mostly used for RFA treatments where low temperature thermal exposures happens over a longer duration. The thermal dose was introduced to relate the exposure to increased temperatures over different time duration, to an equivalent exposure time held at a fixed temperature of 43 °C. The accumulated thermal dose can be computed using the following mathematical expression [39]:

$$TD_{43} = \int_0^{\tau} C^{(43 - T(t))} dt, \tag{2.8}$$

where $TD_{43}$ is the thermal dose in units of cumulative equivalent minutes at 43 °C , $C$ is approximated as 0.25 for temperatures below 43 °C and 0.5 for temperatures above 43 °C, and $\tau$ represents the final exposure time. A thermal dose of 240 is considered destructive for liver cells. In this model, the cumulative thermal damage causes cell death. However, the possibility of cell recovery is not taken into account in the thermal dose model.

## 2.3 The multi-physics phenomena in RFA modelling

As explained in the previous section, mathematical models of RFA fall into the three categories of electrical models, heat transfer models, and cellular necrosis models. Based on the equations introduced to model each of these categories, the most general system of equations can be written as follows. We have for the electrical model:

$$\nabla \cdot (\sigma(T)\nabla v) = 0, \tag{2.9a}$$

For the heat-transfer model:

$$(1-\epsilon)\rho c_{ti}\frac{\partial T_{ti}}{\partial t} = (1-\epsilon)Q + (1-\epsilon)\nabla \cdot (d\nabla T_{ti}) + H(T_{bl} - T_{ti}) \tag{2.9b}$$

$$\epsilon\rho c_{bl}(\frac{\partial T_{bl}}{\partial t} + \mathbf{v}.\nabla T_{bl}) = \epsilon Q + \epsilon\nabla \cdot (d\nabla T_{bl}) - H(T_{bl} - T_{ti}), \tag{2.9c}$$

and for the cellular necrosis model either:

$$\Omega(\mathbf{r}, t) = \int_0^t A' \cdot exp(-\frac{\Delta E}{RT(\mathbf{r}, t)})dt, \tag{2.9d}$$

or:

$$\frac{dA}{dt} = -k_f A + k_b[1 - A - D], \tag{2.9e}$$

$$\frac{dD}{dt} = k_f[1 - A - D], \tag{2.9f}$$

where $Q = \sigma|\nabla v|^2$ and $k_f = \overline{k_f}e^{T/T_k}[1 - A]$.

The equations in 2.9 are coupled to each other, imposing a multi-physics approach in hepatic RFA modelling: Equation (2.9a) is coupled to (2.9b) and (2.9c) because the electrical conductivity $\sigma$ is temperature dependent. Equations (2.9b) and (2.9c) are coupled together via their last term $H(T_{bl} - T_{ti})$, to (2.9a) because of their source term

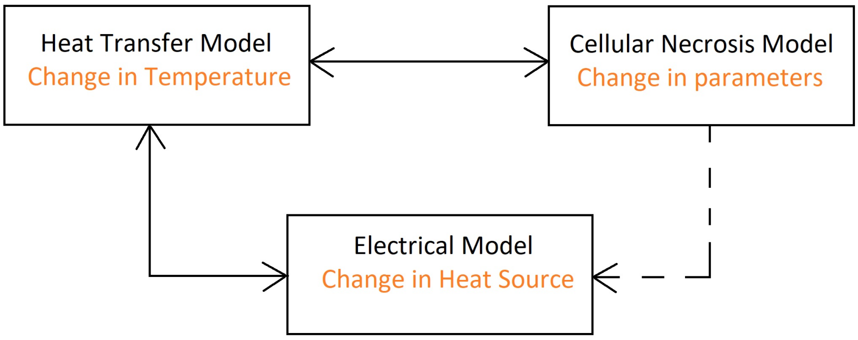

Figure 2.3: Sketch of the coupled *electrical-thermal-cellular necrosis* models.

$Q$, and to the cellular necrosis model because some of the thermal parameters have different values for healthy and damaged cells. Moreover, the advection term in (2.9c), $\mathbf{v}.\nabla T_{bl}$, is locally set to zero when $D > 0.8$ and also heat capacity, $c_{ti}$, is modified when $D < 0.8$ [11]. Finally, (2.9e) and (2.9f) are coupled to each other and coupled to (2.9b) and (2.9c) as $k_f$ is temperature dependent. The coupling between the electrical, thermal, and cellular necrosis models is depicted in Fig. 2.3. The coupling between the electrical model and the cellular necrosis model is shown by a dashed arrow because a change in $\sigma$ due to cellular necrosis is not reported in the literature.

Because of the recent advances in computing power and the growing desire for simulation in various fields of science and engineering, multi-physics simulations have earned more attention in the past decade. A multi-physics system consists of two or more physical models which are controlled by their own physical principles, and yet interact with each other simultaneously. The components involved in a multi-physics system can be simulated either individually or in a coupled manner. The latter is called a multi-physics simulation. Such coupling can introduce new restrictions that don't exist for the individual components, e.g., of stability, accuracy, or robustness limitations [40].

Multi-physics systems are classified from different aspects. They can be categorized based on the level of coupling between the individual components, i.e., either weak or strong. From this perspective, strong (weak) coupling means that physical components of the model are coupled by a strong physical (weak) interaction. Strong coupling could be due to significant overlap of geometric domains or due to a strong impact of one component on another [40]. Moreover, between two components, physical interaction can be strong in one direction and weak in the other, or strong in both directions. The coupling is described as two-way in the latter case while the former is considered as one-way coupling. We need to clarify that throughout this thesis, the term *coupling* is used in the algorithmic level, that is, to indicate the algorithmic components are coupled by either strong or weak interaction.

Besides the level of coupling between physical components, the choice of solution approach for a multi-physics system depends on other factors like the availability of codes for individual components. From a programmer's point of view, if computer codes already exist for individual component solutions, the simplest route to a multi-physics simulation would be to exploit the existing codes based on a reductionist approach such as operator splitting. For example, in the partial differential equation modelling a multi-physics system, if one component is described by diffusion and the other component is described by convection, splitting of diffusion terms and convection terms and exploiting the possibly available individual codes sounds appealing. Such decoupling, however, may overlook strong couplings between physics and give a false sense of completion [40].

Regarding the coupled equations in (2.9), the coupling between (2.9a) and (2.9b) can be considered strong, as the temperature and the electric potential are interacting in the whole domain. On the other hand, the coupling between (2.9b) and (2.9c) can be regarded weak since the tissue temperature and the blood temperature only interact over the tissue-vessel boundaries. The following two sections present a brief review on the weak and strong coupling modelling of multi-physics systems.

### 2.3.1 Weak coupling modelling

The simplest multi-physics system is the coupled equilibrium problem:

$$\begin{cases} f_1(x_1, x_2, ..., x_n) = 0 \\ \\ f_2(x_1, x_2, ..., x_n) = 0 \\ \vdots \\ f_n(x_1, x_2, ..., x_n) = 0. \end{cases} \tag{2.10}$$

Here, $x = (x_1, x_2, \ldots, x_n)$ refers to a multi-physics solution which consists of $n$ components and $f_1$, $f_2$,..., $f_n$ are called the component residuals.

One can follow a reductionist approach to solve (2.10), that is solving the first equation for the first unknown given all the other unknowns, the second equation for the second unknown given all the other unknowns, and so on. This algorithm holds on to the solution of each uni-physics problem, when the coupling is taken into account by iteration over the set of equations. The iteration is typically done in either Gauss-Seidel or Jacobi manner. The Gauss-Seidel method is shown by Algorithm 1.

```
Given initial values $x_1^0, x_2^0, ..., x_n^0$;
for $k = 1, 2, ...,$ (until convergence) do
    for $i = 1, 2, ..., n$ do
        Solve for $x_i^k$ in $f_i(x_1^{k-1}, x_2^{k-1}, ..., x_i^k, ..., x_n^{k-1}) = 0.$;
    end
end
```

**Algorithm 1:** Gauss-Seidel algorithm

When the algorithm converges, the accuracy can be improved by continuing the iterations. In the Gauss-Seidel algorithm, each equation is solved for its own unknown, using the most recent values for the other unknowns. This means the equations in (2.10) need to be solved sequentially. On the other hand, a Jacobi algorithm allows all the equations in the same iteration to be solved simultaneously. In the Jacobi algorithm, each equation in (2.10) uses the values for the other unknowns from the last iteration. Avoiding processor idle time, this further decoupling provides more parallelism, although at the cost of a slower convergence rate. The Jacobi method is particularly advantageous on massively parallel computers.

### 2.3.2 Strong coupling modelling

If the residuals and their derivatives are smooth, Newton's method is the basic algorithm that models the tight coupling between components. The problem in (2.10) can be formulated in terms of a single residual that contains all components:

$$\mathbf{F}(\mathbf{x}) = 0, \tag{2.11}$$

where $\mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), ..., f_n(\mathbf{x}))^T$ is the system of $n$ coupled equations and $\mathbf{x} = (x_1, x_2, ..., x_n)^T$ is the vector of unknowns. The key approach of Newton's method to solve (2.11) is given by Algorithm 2. Here, $\Delta\mathbf{x}$ is called the update vector, $\mathbf{J}(\mathbf{x})$ is an n-by-n Jacobian matrix of $\mathbf{F}(\mathbf{x})$, i.e., $\mathbf{J}_{ij}(\mathbf{x}) = \partial f_i/\partial x_j$, and $k$ is the iteration number. Newton's method is considered as being tightly coupled because of the presence of the off-diagonal blocks in the Jacobian. Here, the off-diagonal blocks of the Jacobian matrix are either dense or sparse with rather large entries. Conversely, a weakly coupled model may contain relatively fewer or smaller off-diagonal entries.

```
Given initial values $\mathbf{x}^0 = (x_1^0, x_2^0, ..., x_n^0)^T$;
for $k = 1, 2, ...,$ (until convergence) do
    Solve $\mathbf{J}(\mathbf{x}^{k-1})\Delta\mathbf{x} = -\mathbf{F}(\mathbf{x}^{k-1})$;
    Update $\mathbf{x}^k = \mathbf{x}^{k-1} + \Delta\mathbf{x}$;
end
```

**Algorithm 2:** Newton's method

Newton's method offers faster (up to quadratic) convergence compared to Gauss-Seidel and Jacobi iterations. However, computation of $\Delta\mathbf{x}$ in Algorithm 2 can be too expensive for large-scale problems. In order to reduce the computational cost of Algorithm 2, inexact Newton methods including Newton-Krylov methods were proposed. In the Newton-Krylov methods, $\Delta\mathbf{x}$ is computed with a Krylov subspace. This approach is beneficial since the Krylov subspace method only needs Jacobian-vector products. Consequently, if these products can be approximated, the Jacobian matrix is never itself required. For instance, by using a finite-difference approximation to the Jacobian-vector products, the Jacobian-free Newton-Krylov (JFNK) method bypasses the need for assembling a Jacobian matrix.

## 2.4 High-performance computing in RFA modelling

Two different approaches are followed by researchers regarding RFA simulation: they either use commercially available software such as COMSOL Multi-physics [27, 23, 22, 41] or develop their own code [25, 34, 4, 16]. COMSOL provides all the necessary tools to build the model, solve the problem, and post-process the results. Also, using its equation-based modelling capabilities, one can create new physics interfaces or modify the underlying equations of a built-in model. This is a key advantage for RFA modelling as it would allow complicated coupled problems (e.g., thermal-electrical-necrosis) to be implemented, based on which the relation between temperature and tissue damage can be simulated. Moreover, COMSOL supports both shared-memory and distributed-memory parallelization, including cluster implementation. Cluster computing is supported using the message passing interface (MPI) or by running the program on cloud computing hardware [42].

The FEM is the main numerical technique used by those researchers who developed their own code. Payne *et al.* [11] present a parallel RFA simulation where the FEM is implemented in parallel based on mesh partition and the MPI. A model based on weighted distance fields to approximate the ablation zone is proposed by Rieder *et al* [43]. The distance fields of varying RF probe types are derived from FEM simulations to allow a fast graphics processing unit (GPU)-based approximation of the ablation zone. Mariappan *et al.* [16] present a GPU-based FEM model of RFA. They use the preconditioned Biconjugate gradient stabilized method (BiCGSTAB) with the assistance of CUDA (dedicated software for NVIDIA's GPUs) libraries such as cuBLAS and cuSPARSE to accelerate their computations on a GPU. RFA Guardian [4] is a comprehensive FEM-based planning and simulation software application for RFA, which supports simulation on the GPU. An MPI-based parallel FEM simulation to analyze the thermal and electrical processes involved in RFA is proposed by [44]. Zhang *et al.* [45] presents a formulation of bio-heat transfer under the effect of soft tissue deformation for a near real-time tissue temperature prediction, based on the fast explicit dynamics finite element method (FED-FEM).

Besides FEM, other numerical methods have also been employed in the parallelization of RFA simulations. Kath *et al.* [25] proposed the finite difference method (FDM) for running RFA simulations on the GPU, and tried different kernel configurations to find an optimum use of the shared memory of CUDA in the calculation of the finite differences. The Lattice-Boltzmann method (LBM) is used by Audigier *et al.* [15] for the simulation of heat transfer in the liver. Looking for a near real-time computation of the heat transfer, they provide a GPU-based version of the model, relying on the CUDA toolkit. In order to show the benefit of their GPU-based approach, it is compared against a central processing unit (CPU) implementation with multithreading. A maximum speed-up of 11 with multithreading and 22 threads (OpenMP) and 45 with GPU implemented in CUDA with respect to a single-core implementation of LBM is reported.

Using neural networks (NN) in RFA modelling has also gained attention. Based on the tissue complex electrical impedance data, a deep neural network and tree-based ensembles were trained for estimating the RFA lesion depth via regression by [46]. A Neural network model of heat transfer in the tissue on three-dimensional regular and irregular grids was developed by [14].

## 2.5 Uncertainty analysis in RFA modelling

The outcome of the RFA simulations depends on various parameters such as electrical and thermal conductivity, heat capacity, density, and water content. These parameters vary from patient to patient, and in fact their exact values are unknown. Typically, RFA simulations use average values for these properties taken from literature, resulting in a deterministic ablation outcome. In contrast, one can implement a stochastic simulation model, where the input parameters are modelled as random variables. The stochastic approach computes how the uncertainty in the input parameters propagates to the simulation output.

The simplest method to handle parameter uncertainty is the Monte Carlo simulation. In this method, a sufficiently large number of samples to completely cover the stochastic space are drawn from the input random variables. A deterministic simulation computes the output for each sample, based on which the output random properties are found. Although the Monte Carlo method is easy to implement, it has a slow convergence rate. Moreover, if the deterministic simulations for each sample is already time-consuming, the total Monte Carlo simulation will be very expensive.

The SFEM is an extension of the FEM that models the uncertainties in a system. Compared to Monte Carlo, more effort is needed in the implementation, but the results are provided faster. In the context of RFA modelling, substituting the uncertain parameters with random variables yields a system of stochastic PDEs, which then can be solved with SFEM. The SFEM propagates the parameter uncertainty via the mathematical model to the output. Therefore, the output temperature is characterized by a random field whose probability density function can be approximated.

SFEM was first introduced in steady-state RFA simulations by Altrogge *et al.* [47]. In their work, the random quantities are described as an expansion of stochastic orthogonal basis functions with deterministic coefficients. Ristovski *et al.* [34] extended this steady-state model to the time-dependent heat equation in RFA. Both methods are based upon the Wiener-Hermite polynomial chaos expansion combined with the spectral stochastic finite element method (SSFEM) of Ghanem and Spanos [48]. The stochastic partial differential system is then solved in the weak form within the random space. The output temperature is also given as a polynomial chaos expansion using Hermite polynomials.

The goal of uncertainty analysis in RFA is to provide the medical doctor with the uncertainties associated with the treatment plan showing its robustness with respect to the variations in the biological parameters of the liver. The medical doctor would adjust the treatment plan if the uncertainty analysis indicates a considerable variation of the optimum control parameters, e.g., the input voltage or the treatment duration, and hence a low confidence that the greatest treatment outcome would be achieved.

In this chapter the computational models of radiofrequency ablation of liver tumours were introduced. It was also outlined that a comprehensive mathematical model of hepatic RFA has multi-physical aspects. The next chapter presents a brief review of FEM and its issues for parallel implementation of multi-physics problems. Also, the FGaBP algorithm and its advantages over FEM on parallel architectures will be explained.

# Chapter 3

# Finite Element Gaussian Belief Propagation

In the previous chapter, the multi-physics aspects of the radiofrequency ablation (RFA) modelling were discussed. The FEM is the most widely used algorithm for computer simulation of RFA. The two computationally expensive stages of FEM, the assembly stage and the solving stage, are even more expensive for multi-physics applications. In a multi-physics simulation, these two stages must be repeated based on the most updated values of all physical components in the system. This is especially so for a non-linear multi-physics problem which is linearized with a NR method. This chapter presents an overview of the FGaBP method as an alternative reformulation of FEM. We start off by explaining the challenges in parallel implementation of multi-physics problems. The main concepts of FGaBP and their advantages over conventional FEM-based multi-physics simulations are then described.

## 3.1 Parallel acceleration of multi-physics problems

A multi-physics system is usually described with a set of coupled PDEs. Assuming the coupled PDEs are spatially discretized with the finite element method (FEM), the multi-physics system is eventually modelled with a linear system $\mathbf{Ax} = \mathbf{b}$, where $\mathbf{A}$ is the sparse stiffness matrix, $\mathbf{x}$ is the multi-physics solution vector, and $\mathbf{b}$ is the right hand side vector. The system matrix $\mathbf{A}$ and right hand side vector $\mathbf{b}$ are dependent on the solution vector $\mathbf{x}$.

As explained in Section 2.3, there are two general approaches for solving a multi-physics system, i.e., weak coupling and strong coupling. The weak coupling model solves each equation for its own unknown, assuming constant values for the other unknowns. As a result, a linear system $\mathbf{A}^{(i)}\mathbf{x}^{(i)} = \mathbf{b}^{(i)}$ for each component needs to be solved, where $\mathbf{A}^{(i)}$ is the sparse matrix corresponding to the $i$th component, $\mathbf{x}^{(i)}$ is the solution vector of the $i$th component, and $\mathbf{b}^{(i)}$ contains the elements in $\mathbf{b}$ associated with the $i$th component. In finite element applications, the coefficient matrices $\mathbf{A}^{(i)}$ are usually very sparse. Consequently, iterative solvers are employed to solve the linear systems associated with each physics. Iterative solvers need to execute different kernels, including the sparse matrix-vector multiplication (SpMV). The SpMV is often considered the performance bottleneck in efficient implementations of iterative solvers.

Strong coupling models are often based on the NR method. In NR, the construction of a Jacobian matrix at each linearizing iteration could be prohibitively expensive for large scale problems. Following the same notation used in Section 2.3, a linear system $\mathbf{J}\Delta\mathbf{x} = -\mathbf{F}(\mathbf{x})$ needs to be solved at each NR iteration, where matrix $\mathbf{J}$ is the Jacobian, vector $\Delta\mathbf{x}$ is the multi-physics solution update, and vector $\mathbf{F}(\mathbf{x})$ is the residual. The JFNK method is the most widely used inexact solver that bypasses the assembly of a Jacobian matrix. 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 gets around the main obstacle associated with the assembly stage of the NR method. For solving $\mathbf{J}\Delta\mathbf{x} = -\mathbf{F}(\mathbf{x})$ using the Krylov method, only matrix–vector products are needed, not the individual elements of $\mathbf{J}$. The action of the Jacobian in the form of matrix–vector products is approximated by:

$$\mathbf{J}\mathbf{v} \approx [\mathbf{F}(\mathbf{x} + \epsilon\mathbf{v}) - \mathbf{F}(\mathbf{x})]/\epsilon, \tag{3.1}$$

where $\epsilon$ is a small perturbation.

The efficiency of JFNK depends critically on preconditioning the inner Krylov solver. The preconditioning reduces the number of Krylov iterations by clustering eigenvalues of the iteration matrix. This operation needs to be done at each Krylov iteration. For instance, if a right preconditioner is applied, (3.1) becomes:

$$\mathbf{J}\mathbf{P}^{-1}\mathbf{v} \approx [\mathbf{F}(\mathbf{x} + \epsilon\mathbf{P}^{-1}\mathbf{v}) - \mathbf{F}(\mathbf{x})]/\epsilon, \tag{3.2}$$

where $\mathbf{P}$ is the preconditioning matrix. Applying the right preconditioner is done in two steps:

1. Solve approximately for $\mathbf{w}$ in $\mathbf{P}\mathbf{w} = \mathbf{v}$
2. Perform matrix-vector approximation: $\mathbf{J}\mathbf{w} \approx [\mathbf{F}(\mathbf{u} + \epsilon\mathbf{w}) - \mathbf{F}(\mathbf{u})]/\epsilon$

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 the SpMV. The SpMV operation can strongly limit the acceleration of the preconditioning stage [49].

Many attempts are made to improve the parallel performance of JFNK preconditioning, such as the Newton-Krylov-Schwarz (NKS) method and the multigrid algorithm as a preconditioner. The NKS method is based on a domain decomposition approach. If the computational domain $\Omega$ is decomposed into a set of $p$ sub-domains $\{\Omega_1, \Omega_2, ..., \Omega_p\}$ such that $\Omega = \bigcup_{i=1}^{p} \Omega_i$, the additive Schwarz preconditioner is computed as:

$$\mathbf{P}^{-1} = \sum_{i=1}^{p} \mathbf{P}_i^{-1}, \tag{3.3}$$

where $\mathbf{P}_i$ is the preconditioner defined on the sub-domain $\Omega_i$. The primary motivation for NKS is to construct $\mathbf{P}^{-1}$ concurrently through divide-and-conquer. Each $\mathbf{P}_i^{-1}$ in the sum can be computed in parallel, resulting in parallelism proportional to the number of sub-domains $p$. Using multigrid preconditioning to compute $\mathbf{P}$ in parallel has also been considered extensively [50]. In the multigrid preconditioned Newton–Krylov method (NKMG), the system $\mathbf{w} = \mathbf{P}^{-1}\mathbf{v}$, is approximately solved for $\mathbf{w}$ using a multigrid algorithm. Nevertheless, the SpMV operation is still the main kernel of these preconditioning methods.

Achieving an efficient SpMV is challenging since its performance is heavily dependent on the density of nonzero entries in the sparse matrix, or its sparsity pattern. The SpMV operation can be executed as a number of concurrent vector dot products between each row of the sparse matrix and the dense vector. The result from each dot product updates the corresponding entry in the output vector. The vector dot products are independent of each other, hence can be executed in parallel. Although this is an embarrassingly parallel approach, it will result in a poor performance in terms of speed-up. The CPU needs to read each row of the sparse matrix and the whole dense vector from the main memory into cache, and then perform the vector dot product. This requires non-coalesced memory access since the data within each row of the matrix is very sparse. Consequently, a very low cache hit rates is achieved which substantially causes the CPU to reach only a small fraction of its peak performance [51].

Optimizing SpMV becomes even harder as the underlying processor architectures are becoming more diverse. The sparse matrix storage format can have a considerable impact on the SpMV performance. While an extensive body of research exists on optimizing SpMV on multi-core architectures [52, 53], it is still unclear how the SpMV performance on many-core platforms depend on different sparse matrix storage formats [54]. To this end, a complicated approach is needed to first identify the performance bottlenecks of SpMV for a specific sparse matrix on a target architecture and then to apply the right optimizations to address those bottlenecks. Reducing the cache-memory traffic, mainly by sparse matrix compression techniques, is the general approach followed by researchers [54]. SpMV is naturally a memory bandwidth bound kernel on multicore machines. Its bandwidth utilization is highly dependent on the sparsity pattern of the matrix and the underlying architecture. On a multi-core machine, the memory requests increase with the number of parallel processes, causing a greater bottleneck for the limited memory bandwidth. This in turn restricts the parallel scalability of the SpMV.

Optimizing SpMV includes a preprocessing stage to analyze the sparse matrix structure followed by sophisticated programming techniques such as code transformations, format conversion, parameter tuning, etc. The cost of this stage can pay off for applications that reuse the same sparse matrix in several instances of SpMV; however, it can outweigh the performance improvements when the convergence is achieved for a small number of iterations, e.g., in preconditioned solvers. The results of SpMV optimization efforts reveal that maintaining good performance for all sparse matrices, even within the same FEM application area, is impossible [51]. There exist standard optimized libraries such as Trilinos [55] and PETSc [56], designed to solve a sparse linear system in parallel. Nevertheless, the varying sparsity pattern of the matrix makes achieving a sustained performance difficult. Furthermore, such libraries do not improve the assembly stage of the matrix, which in should be executed several times for the NR method.

The remainder of this chapter explains the finite element Gaussian belief propagation (FGaBP) algorithm. The FGaBP algorithm is based on probabilistic inference on graphical models. It turns the underlying formulation of the FEM into localized computations involving dense matrices of very small sizes. The algorithm progresses by propagating messages throughout the graph in a flexible manner allowing adaptable memory utilization for various HPC architectures. The FGaBP doesn't construct any large sparse matrices, and eliminates all the global algebraic operations, including the SpMV. This new algorithm provides a highly parallel and adaptable tool for both the assembly and the solving stages of multi-physics problems.

## 3.2 The Gaussian belief propagation algorithm

Belief propagation (BP) [57] is a message-passing algorithm that was originally used in probability problems to find marginal distributions of random variables from their joint distribution, a process called statistical inference. It infers the marginal distribution for each variable conditioned on the other variables. The inference is carried out on probabilistic graphical models, where each node of the graph represents a random variable and the graph edges show the dependency between the variables. It has proved practical success in numerous applications such as turbo codes and low-density parity-check codes [58].

BP performs inference by passing real-valued messages across edges in the graph, iteratively. An undirected graph $\mathcal{G}$ is defined by a set of nodes $\mathcal{V}$ and a corresponding set of undirected edges $\mathcal{E}$. Each node $i \in \mathcal{V}$ is associates with a random variable $x_i$, assuming $x = \{x_i | i \in \mathcal{V}\}$ is the set of all random variables. The joint distribution $p(x)$ is represented in a factor form by [58]:

$$p(x) \propto \prod_{(i,j)\in\mathcal{E}} \psi_{ij}(x_i, x_j) \prod_{i\in\mathcal{V}} \phi_i(x_i), \tag{3.4}$$

where $p(x)$ may only be known up to a normalization coefficient, taken so that it integrates to one. The function $\psi_{ij}(x_i, x_j) > 0$ is called edge potential and indicates the probabilistic dependence between nodes $i$ and $j$. The function $\phi_i(x_i)$ is the self potential of node $i$, containing information about the marginal distribution $p(x_i)$. Note that $\phi_i(x_i)$ does not equal the marginal distribution $p(x_i)$, because of interactions with other nodes.

At each iteration $t$ of the BP algorithm, nodes $i \in \mathcal{V}$ send messages $m^t_{ij}(x_j)$ to neighboring nodes $j \in \mathcal{N}(i)$:

$$m^t_{ij}(x_j) \propto \int_{x_i} \psi_{ij}(x_i, x_j)\phi_i(x_i) \prod_{k \in \mathcal{N}(i) \setminus j} m^{t-1}_{ki}(x_i)\ dx_i. \tag{3.5}$$

The message shows the probability density function (PDF) of node $j$ from the standpoint of node $i$. At message convergence, the marginals are computed as:

$$p(x_j) \propto \phi_j(x_j) \prod_{i \in \mathcal{N}(j)} m_{ij}(x_j). \tag{3.6}$$

This means the marginal distributions depend on the self potentials and also the so called *beliefs* from the neighbouring nodes. If underlying distributions are Gaussian, the algorithm is called Gaussian belief propagation (GaBP). The GaBP update rules can be obtained by substituting Gaussian distributions into the continuous BP equations.

The application of GaBP for solving systems of linear equations was originally proposed by [59]. Let the linear system of interest be expressed as:

$$\mathbf{A}\mathbf{x} = \mathbf{b}. \tag{3.7}$$

One can define the joint Gaussian density function $p(x)$ as:

$$p(x) \sim exp(-\frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} + \mathbf{b}^T\mathbf{x}) \sim \mathcal{N}(\mu = \mathbf{A}^{-1}\mathbf{b}, \mathbf{A}^{-1}), \tag{3.8}$$

where $\mathcal{N}(\mu = \mathbf{A}^{-1}\mathbf{b}, \mathbf{A}^{-1})$ represents a multivariate Gaussian distribution whose mean and covariance are $\mu$ and $\mathbf{A}^{-1}$, respectively. A graphical representation of $p(x)$, $\mathcal{G}$, can be developed whose topology is determined by the non-zero structure of the matrix $\mathbf{A}$. The set of edges $\mathcal{E}$ in $\mathcal{G}$ includes an edge for each non-zero entry $A_{ij}$, for which $j < i$. If matrix $\mathbf{A}$ is symmetric positive definite (SPD), then solving the linear system is equivalent to minimizing the quadratic function $q(x) = \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} - \mathbf{b}^T\mathbf{x}$, which in turn is equivalent to maximizing the exponential $p(x) = e^{-q(x)}$. On the other hand, $p(x)$ is maximized when $\mathbf{x}$ is equal to the marginal mean vector $\mu$. Consequently, by performing GaBP for the joint distribution $p(x)$, one can solve the linear system. In GaBP, the edge potentials and self potentials are defined to be:

$$\psi_{ij}(x_i, x_j) = exp(-\frac{1}{2}x_i A_{ij} x_j), \tag{3.9a}$$

$$\phi_i(x_i) = exp(-\frac{1}{2}A_{ii}x_i^2 + b_i x_i), \tag{3.9b}$$

respectively. Note the self potentials have also a Gaussian form as:

$$\phi_i(x_i) \propto \mathcal{N}(\mu_{ii} = b_i/A_{ii}, P_{ii}^{-1} = A_{ii}^{-1}), \tag{3.10}$$

whose mean and inverse-variance are shown by the scalars $\mu_{ii}$ and $P_{ii}$. The GaBP algorithm is built on the fact that the product of Gaussian densities over a common variable is, up to a constant factor, also a Gaussian density. This means all messages are of Gaussian form also [60]. As a result, the update rules for the messages, (3.5), are reduced to update rules for marginal means and marginal inverse-variances. Assuming the message $m_{ij}$ has a normal distribution $\mathcal{N}(\mu_{ij}, P_{ij}^{-1})$, its mean and inverse-variance are given by:

$$P_{ij} = -A_{ij}^2 P_{i\backslash j}^{-1}, \tag{3.11a}$$

$$\mu_{ij} = -P_{ij}^{-1} A_{ij} \mu_{i\backslash j}, \tag{3.11b}$$

where:

$$P_{i\backslash j} = P_{ii} + \sum_{k \in N(i)\backslash j} P_{ki}, \tag{3.12a}$$

$$\mu_{i\backslash j} = P_{i\backslash j}^{-1}\big(P_{ii}\mu_{ii} + \sum_{k \in N(i)\backslash j} P_{ki}\mu_{ki}\big). \tag{3.12b}$$

Here, $P_{ki}$ and $\mu_{ki}$ are the mean and inverse-variances of the incoming messages $m_{ki}(x_i)$. In the next section, the FEM-based GaBP algorithm proposed by [51] is presented.

## 3.3 Finite element Gaussian belief propagation

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. Factor graphs are used in probability problems to represent factorization of a PDF, enabling efficient computations of marginal distributions through the BP algorithm. A factor graph consists of two different types of nodes, factor node (FN) and variable node (VN). The FNs stand for the factorization of the underlying PDF. The VNs represent the random variables, and can only interact with each other through the FNs. As shown in Fig. 3.1, FGaBP

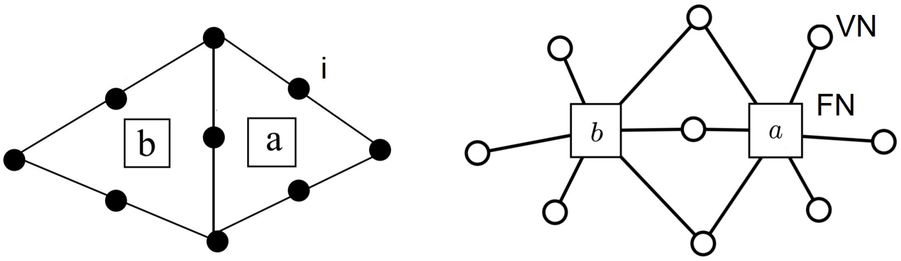

Figure 3.1: (*Left*): Sample FEM mesh of two second order triangles. (*Right*): Two types of nodes in the factor graph representation.

turns the FEM mesh into a factor graph. The variable nodes (nodes of unknowns) are represented by circles, and the factor nodes are represented by squares.

The solution at each VN is considered a random variable with a Gaussian distribution whose shape is defined by two parameters, $\alpha$ and $\beta$, where $\alpha$ is the reciprocal of the variance and $\beta/\alpha$ is the mean. By passing messages between each FN and all its connected VNs, the FGaBP algorithm tries to find the values of $\alpha$ and $\beta$ for each VN. A message, $m_{ai}$, is sent from factor node $a$ ($\mathrm{FN_a}$) to the connected variable node $i$ ($\mathrm{VN_i}$) and represents the most probable solution value at $i$, as observed from $\mathrm{FN_a}$. In return, $\mathrm{VN_i}$ sends a message back to $\mathrm{FN_a}$ representing observations from other connected FNs. The following is the formulation of the FGaBP algorithm update rules [49]. The algorithm propagates the messages over the factor graph edges, iteratively. Before we go over the algorithm, a brief review of the FEM formulation is necessary. The FEM constructs a linear system of equations by looping through all the elements in the computational domain. A local characteristic matrix, $M_e$, and a local source vector, $b_e$, is formed for each element $e$. The global linear system is then assembled by having each element $e$ contribute $M_e$ and $b_e$ to the global stiffness matrix and the global right hand side vector, respectively. As will be explained, FGaBP bypasses the assembly stage. The algorithm can be summarized in the following three steps:

1. $t = 0$: Initialize all messages $\beta^{(0)} = 0$ and $\alpha^{(0)} = 1$.

2. Iterate: $t = 1, 2, ....$

   (a) For each $\mathrm{VN_i}$, compute messages $\alpha_{ia}^{(t)}$ and $\beta_{ia}^{(t)}$ to each connected $\mathrm{FN_a}$ ($a \in$

$\mathcal{N}(i)$) as follows:

$$\alpha_i^{(t)} = \sum_{k \in \mathcal{N}(i)} \alpha_{ki}^{(t-1)}, \quad \alpha_{ia}^{(t)} = \alpha_i^{(t)} - \alpha_{ai}^{(t-1)} \tag{3.13}$$

$$\beta_i^{(t)} = \sum_{k \in \mathcal{N}(i)} \beta_{ki}^{(t-1)}, \quad \beta_{ia}^{(t)} = \beta_i^{(t)} - \beta_{ai}^{(t-1)} \tag{3.14}$$

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

(b) For each $\mathrm{FN_a}$:

i. receive messages $\alpha_{ia}^{(t)}$ and $\beta_{ia}^{(t)}$ from all the VNs $i \in \mathcal{N}(a)$.

ii. Assume $\mathcal{A}^{(\mathrm{t})}$ is a diagonal matrix of incoming $\alpha_{ia}^{(t)}$ messages, and $\mathcal{B}^{(\mathrm{t})}$ is a vector of incoming $\beta_{ia}^{(t)}$ messages, then define matrix $\mathbf{W}$ and vector $\mathbf{K}$ as follows:

$$\mathbf{W}^{(t)} = \mathbf{M} + \mathcal{A}^{(t)} \tag{3.15}$$

$$\mathbf{K}^{(t)} = \mathbf{B} + \mathcal{B}^{(t)} \tag{3.16}$$

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:

$$\mathbf{W}^{(t)} = \begin{bmatrix} W_{\mathcal{L}(i)}^{(t)} & \mathbf{V}^T \\ \mathbf{V} & \bar{\mathbf{W}}^{(t)} \end{bmatrix} \tag{3.17}$$

$$\mathbf{K}^{(t)} = \begin{bmatrix} K_{\mathcal{L}(i)}^{(t)} \\ \bar{\mathbf{K}}^{(t)} \end{bmatrix} \tag{3.18}$$

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:

$$(\mathbf{W}^{(t)})^{-1} = \begin{bmatrix} \tilde{W}_{\mathcal{L}(i)} & \tilde{\mathbf{C}}^T \\ \tilde{\mathbf{C}} & \tilde{\mathbf{W}} \end{bmatrix}. \tag{3.19}$$

v. Compute and send new $\mathrm{FN_a}$ messages $\alpha_{ai}^{(t+1)}$ and $\beta_{ai}^{(t+1)}$ to each $\mathrm{VN_i}$ as follows:

$$\alpha_{ai}^{(t+1)} = \frac{1}{\tilde{W}_{\mathcal{L}(i)}} - \alpha_{ia}^{(t)}. \tag{3.20}$$

$$\beta_{ai}^{(t+1)} = B_{\mathcal{L}(i)} + \frac{1}{\tilde{W}_{\mathcal{L}(i)}} \, (\bar{\mathbf{K}}^{(t)})^T \, \tilde{\mathbf{C}}^T. \tag{3.21}$$

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

   $$u_i = \frac{\beta_i}{\alpha_i} \tag{3.22}$$

   where

   $$\beta_i = \sum_{k \in N(i)} \beta_{ki}, \quad \alpha_i = \sum_{k \in N(i)} \alpha_{ki} \, . \tag{3.23}$$

The messages can be initialized to any arbitrary value given that $0 < \alpha$. Using $1 \leq \alpha$ makes the matrix $\mathbf{W}$ diagonally dominant which improves the numerical properties of its inversion [49].

The FGaBP messages carry information on the most probable states of variable nodes as seen from the standpoint of each connected factor node. As shown by [49], the factor nodes can either represent single elements, or can comprise multiple elements merged together. A vital property of the FGaBP update rules is that they are built on local computations, while the computations of each FN are independent of other FNs at each iteration. As will be explained in the next chapters, we can exploit this property to build local Jacobian matrices associated with each factor node, and implement the NR algorithm at the element level. This provides an efficient tool for parallel implementation of non-linear multi-physics problems on HPC platforms.

Message communication in FGaBP can be implemented based on a specific schedule. Each FGaBP iteration goes over all the FNs exactly once. One of the key empirical properties of the FGaBP algorithm is its flexibility in message scheduling, which allows implementations that provide a trade-off between computation and communication on different parallel architectures [49]. However, message scheduling can significantly affect the number of iterations needed for convergence; an optimum scheduling must exploit the underlying connectivity structure of the graph with minor effect on the iteration count.

There are two primary scheduling schemes for FGaBP message passing, i.e., sequential (asynchronous) and parallel (synchronous). The sequential scheduling computes and communicates the messages one FN at a time, when the FNs are traversed based on a specific order. Therefore, each FN computes its messages based on the most recent messages propagated within the current iteration. This message schedule delivers the lowest number of FGaBP iterations, with the cost of compromising parallelism. In parallel message scheduling on the other hand, all the FNs are processed at the same time while

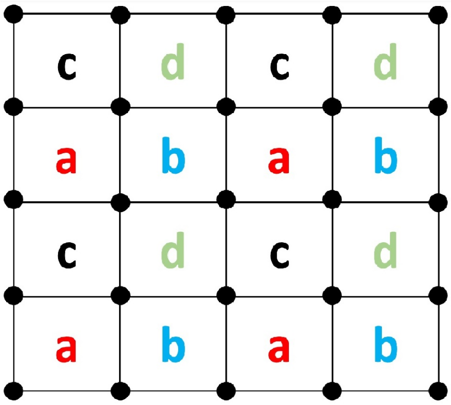

Figure 3.2: Structured quadrilateral mesh containing four colours.

using messages computed at a previous iteration. The parallel scheduling offers a high degree of parallelism; however, due to the slower propagation of information throughout the graph it needs higher number of iterations to converge.

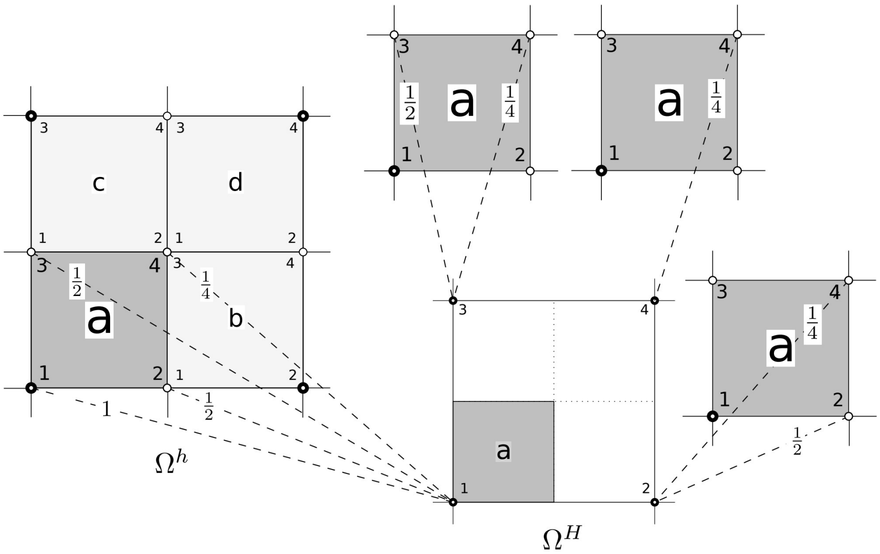

Figure 3.3: Local exchange of residuals between a child element $a$ and its parent element. The numbers stand for the weights that are used to transfer the residual contribution from a node in the child element to a node in the parent element [51].

An element-based colouring schedule is proposed by [49] for shared memory parallelism. This method exploits the parallelism in the factor graph model while not considerably increasing the number of FGaBP iterations. They use an element colouring algorithm when mesh elements are coloured in such a way that every two adjacent element have different colours. Elements are considered adjacent if they share at least a node. A simple mesh colouring scheme is illustrated in Fig. 5.2 using a quadrilateral mesh. FN messages in each colour group are computed and communicated in parallel. This method is particularly effective for multi-threaded implementations on multicore CPUs, as thread-safety is automatically guaranteed here.

A multigrid version of FGaBP, the finite element multigrid Gaussian belief propagation (FMGaBP), to accelerate its convergence was introduced by [61]. The multigrid scheme reduces the iteration count while maintaining all the parallelism features of FGaBP algorithm. Compared with the multigrid preconditioned conjugate gradient (MG-PCG) solvers, FMGaBP algorithm demonstrated considerable iteration reductions and a speedup of up to 2.9 times using eight CPU cores [61]. In addition, the convergence rate of FMGaBP is independent of the problem size on the finest mesh. The FGaBP algorithm benefits from multigrid schemes since communications on coarser levels can serve as bridges to communications between far away nodes on finer levels. This improves the propagation of information throughout the mesh and therefore improves the overall convergence.

The multigrid method in general assumes that the high frequencies of the residual vector are almost removed in a few iterations, but low frequencies are reduced very slowly. The multigrid idea is to turn a fine grid into a coarser grid, on which low frequencies behave like higher frequencies. This is conducted in the following four steps [62]:

- Residual computation: computing residual error on the fine grid.
- Restriction: down sampling the residual error to the coarse grid.
- Prolongation: interpolating the correction from the coarse grid into the fine grid.
- Correction: Adding prolongated coarse grid solution onto the fine grid.

In order to incorporate the above steps into the FGaBP method, El-kurdi *et al.* [61] defined element-wise belief residuals and corrections for each factor node. For each factor on the fine grid, a quantity referred to as the belief residual is defined. The belief residuals of each group of child elements are locally restricted into the parent element. On the other hand, the corrections from the parent elements are mapped onto the child elements using interpolation. This is shown graphically by Fig. 3.3, where $\Omega_h$ and $\Omega_H$ represent the fine and coarse meshes, respectively. The figure shows how each node in the fine mesh (child elements), contributes to the residual in the coarse mesh (parent element).

```
 1 for level $l = 1, 2, \ldots, L$ in the mesh hierarchy do
 2     Perform partition-colour;
 3     Generate elemental dense matrices and source vectors;
 4     Initialize the messages;
 5 end
 6 for cycle $c = 1, 2, \ldots$ in the V-cycle do
 7     for level $l = L, L-1, \ldots, 1$ in the mesh hierarchy do
 8         if $l$ is the coarsest level then
 9             exit;
10         end
11         Execute v1 iterations of FGaBP ;
12         Restrict residuals from child-factors into parent-factors;
13     end
14     Execute FGaBP on coarsest level;
15     for level $l = 1, 2, \ldots, L$ in the mesh hierarchy do
16         Prolongate corrections from parent-factors into child-factors;
17         Execute v2 iterations of FGaBP;
18     end
19     if global tolerance $<$ tolerance then
20         break;
21     end
22 end
```

Figure 3.4: FMGaBP pseudo-code

Figure 3.4 shows the FMGaBP pseudo-code proposed by [61]. The FMGaBP executes the FGaBP algorithm on each level in the multigrid hierarchy. A single iteration of the algorithm is referred to as a V-cycle. The loops on lines 8 and 16 traverse all the levels on the V-cycle except the coarsest level, first going down the cycle and then up the cycle. As the coarsest level includes a very small number of elements, its execution can be rather fast. The application of the multigrid method to multi-physics FGaBP will be described in the next chapters.

In this chapter, we first introduced the main obstacles associated with parallel scalability of multi-physics simulations, i.e., global algebraic operations such as the SpMV. Next, the FGaBP algorithm as an alternative for conventional FEM iterative solvers was marked out. The basic properties of FGaBP that makes it an ideal candidate for parallel multi-physics solvers were also explained. In the following chapters, we will modify the FGaBP to solve the coupled problems that emerge in the simulation of radiofrequency ablation of hepatic tumours.

# Chapter 4

# Parallel Weak Coupling Modelling of RFA

A Multi-physics system consists of more than one simultaneously occurring physics. As discussed in Section 2.3, the physics involved in a multi-physics system interact with each other, where the interactions can be modelled with either strong coupling or weak coupling. Strong coupling finds the monolithic solution of all the physics together, while weak coupling is established on individual solutions of each physics. In weak coupling, the individual solutions converge to the multi-physics solution through iteration, which is done via Jacobi or Gauss–Seidel methods. Both methods deal with individual physics separately. The difference is that the Gauss–Seidel method uses the latest updated values during the iterative process, while the Jacobi method applies the values obtained from the previous iteration. The Gauss-Seidel and Jacobi iterations are shown graphically by figures 4.1 and 4.2, respectively. Here, there are two physical fields in the multi-physics system, depicted by $v$ and $T$.

Besides the level of coupling, another major classification in multi-physics systems is whether the coupling occurs in the whole domain or whether it occurs over an interface. In the former case, the physics share the same computational domain while in the latter case they have separate computational domains interacting through an interface (for example the interaction between a fluid and a structure). The interface coupling is often based on the domain decomposition method (DDM), in which the computational domain is divided into multiple disjoint sub-domains, each belonging to one component. Each component is modelled with a (or a set of) partial differential equation(s) with appropriate initial and boundary conditions. In the multi-physics DDM approach, the equations are solved separately and the coupling happens via boundary conditions. For the sake of clarity, we emphasize that the weak coupling method introduced in this chapter deals with volume coupling. Unlike interface coupling, in volume coupling the individual physics interact everywhere inside the domain, which makes the DDM not applicable.

This chapter is organized as follows: Section 4.1 provides the solution of the heat transfer equation using the FGaBP. The coupled electrical-thermal problem in radiofrequency ablation is revisited in Section 4.2. In Section 4.3, the Gauss-Seidel algorithm is modified and combined with FGaBP method, providing a parallel weak coupling approach. The results are presented and discussed in the last section.

## 4.1 Solving the heat transfer problem

In Section 2.1, the electrical heating and the bio-heat model of heat transfer inside the liver were introduced. The Pennes equation is the most widely used model for heat transfer in the tissue:

$$\rho c \frac{\partial T}{\partial t} = Q + \nabla \cdot (d \nabla T) + H(T_{bl} - T), \tag{4.1}$$

where $T$ is the temperature (K), $c$ is the special heat capacity of tissue (J/kg/K), $\rho$ is the tissue density (kg/m$^3$), $d$ is the thermal conductivity (W/m/K), $H$ is the convective transfer coefficient (W/m$^3$/K) and $T_{bl}$ is the baseline physiological blood temperature taken to be 310 K. On the right hand side of (4.1), $Q = \sigma |\nabla v|^2$ is the heat source (W/m$^3$) where $v$ is the electric potential around the probe tip, and $\sigma$ is the electrical conductivity of the tissue.

The application of FGaBP to solve Laplace's equation has already been studied by [49]. Consequently, this section only provides the solution of the heat transfer equation using FGaBP. Since the last term in (4.1) acts as a heat sink, we assume its effect is included in the source term and drop it for now to have a pure diffusion equation as:

$$\rho c \frac{\partial T}{\partial t} = Q + \nabla \cdot (d \nabla T). \tag{4.2}$$

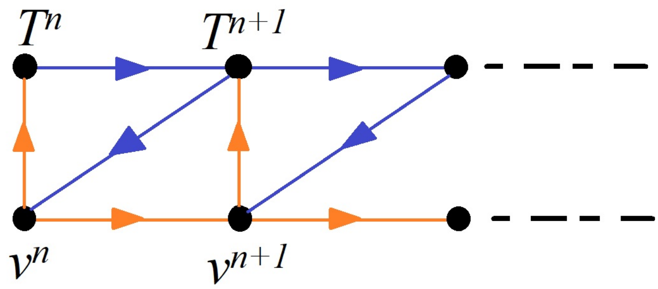

Figure 4.1: Illustration of weak coupling using Gauss-Seidel iteration, with iteration number $n$ advancing to the right.

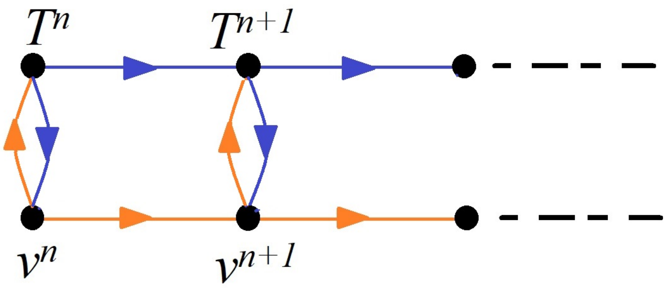

Figure 4.2: Illustration of weak coupling based on Jacobi iteration, with iteration number $n$ advancing to the right.

The spatial discretization of the domain is based on the finite element method (FEM). The theta scheme is chosen for time discretization. The theta scheme generalizes the explicit Euler ($\theta = 0$), implicit Euler ($\theta = 1$), and Crank-Nicolson ($\theta = 0.5$) time discretizations. Since the latter has the highest convergence order, we will choose $\theta = 0.5$, but make it so that changing this parameter remains simple. Employing the theta time discretization, (4.2) becomes:

$$\rho c \, \frac{T^{n+1} - T^n}{\Delta t} = Q \ + \theta \, \nabla \cdot (d \nabla T^{n+1}) + (1 - \theta) \nabla \cdot (d \nabla T^n), \tag{4.3}$$

where the superscript $n$ denotes the n$th$ time step and $\Delta t$ is the time step value. Also, $0 \leq \theta \leq 1$ is the parameter of theta-scheme time discretization. After applying spatial discretization using finite elements, we will have the following discrete equation for each element:

$$\Big[\mathbf{M} - \Delta t \ \theta \ \mathbf{S}\Big] \ \mathbf{T}^{n+1} = \Big[\mathbf{M} + \Delta t \ (1 - \theta) \mathbf{S}\Big] \ \mathbf{T}^n + \Delta t \ \mathbf{f}, \tag{4.4}$$

where $\mathbf{M}_{ij} = \rho c \int N_i N_j \ dV$, $\mathbf{S}_{ij} = d \int \vec{\nabla} N_i \cdot \vec{\nabla} N_j \ dV$ and $\mathbf{f}_j = \int Q N_j \ dV$. Here, $N_i$ and $N_j$ are FEM scalar basis functions.

As described in the previous chapter, FGaBP formulation is based on small dense matrices and vectors instead of global sparse data structures. For the heat transfer equation, the characteristic matrix $\mathbf{M}$ and the source vector $\mathbf{B}$ in the FGaBP update rules, i.e., (3.13) to (3.23) in Chapter 3, should be replaced at each time step with $\mathbf{M} - \Delta t\ \theta\ \mathbf{S}$ and the whole right hand side of (4.4), respectively. Doing so, the $\alpha$ and $\beta$ messages are computed iteratively until message convergence is achieved. At this point, the solution of (4.2) in the current time step and at node $i$ is obtained from the ratio $\beta_i/\alpha_i$, according to (3.22).

In order to evaluate the correctness of the FGaBP heat transfer solver, the results are compared on a regular rectangular domain with an analytical solution. For a source released at $x_0$ at time $t_0$, the 2D analytical solution of the diffusion equation: $(\partial T/\partial t) = \nabla \cdot (d\nabla T) + Q$, inside an infinitely wide domain is [63]:

$$T(x,t) = \frac{K}{[4\pi(t-t_0)d]} \exp\left(\frac{-\|x-x_0\|^2}{4d(t-t_0)}\right). \tag{4.5}$$

We initialized the temperature values at each point of the domain with the analytical solution at time $t = 0$ with this set of parameters: $K = 450$ °C $\cdot$ mm$^2$, $d = 0.15$ mm$^2$/s, $t_0 = -0.1$ s, $x_0 = (0,0)$. The diffusion equation is solved using our FGaBP solver and the temperature at a specific point of the domain is compared with the analytical solution values. For the purpose of being consistent with the analytical solution, the domain was chosen to be large enough with Neumann boundary conditions used at its border, so as to simulate the infinitely wide domain by eliminating the boundary effect at the probed point.

The theta-scheme with $\theta = 0.5$ is also called Crank–Nicolson method. For diffusion equations, it can be shown that the Crank–Nicolson method is unconditionally stable [64]. Consequently, no hard restrictions for the time-step exist; nevertheless, in practice, we still want to make the time-step smaller to decrease the error. In the FGaBP heat transfer simulation a time-step of 0.2 s appeared to be a good compromise between accuracy and computational time. When the time-step is fixed to a constant value, a non-oscillatory criteria imposes an upper bound for the spatial resolution as [65] :

$$\Delta x^2 \leq \frac{20\theta d}{\rho c}\Delta t. \tag{4.6}$$

This means for $\Delta t = 0.2$ s, $\theta = 0.5$, and with other parameters values as in the literature

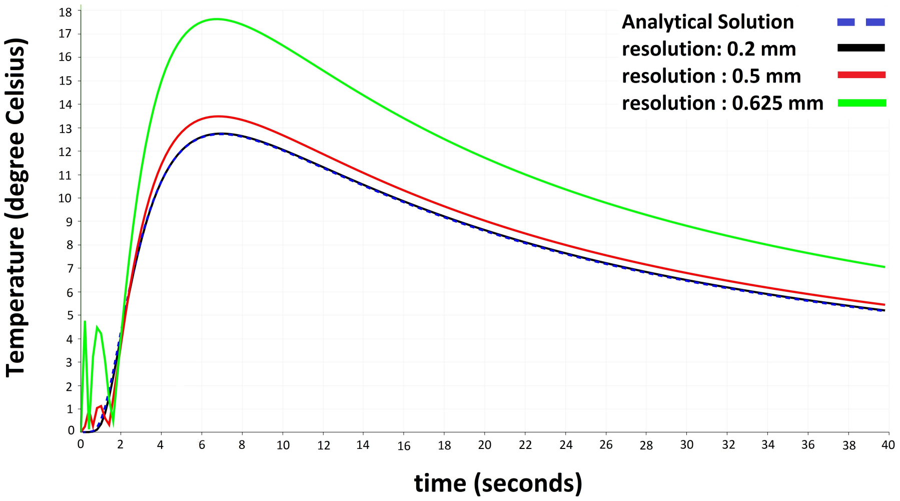

Figure 4.3: Temperature distribution for an instantaneous point source using different space resolutions and time-step = 0.2 s.

[15], $\Delta x$ must be smaller than 0.53 mm. In Fig. 4.3, the analytical solution and FGaBP solution with a time-step of 0.2 s and different spatial resolutions are plotted. Figure 4.3 shows the smaller the spatial resolution, the closer the computed solution is to the analytical one. Quantitatively, the Root-Mean-Square (RMS) errors between the computed solution and the analytical one decreased with the resolution: 3.31 °C, 0.5 °C, 0.14 °C for 0.625 mm, 0.5 mm and 0.2 mm, respectively. This analysis confirms qualitatively and quantitatively the accuracy of the implementation of the heat transfer model with FGaBP.

## 4.2 The coupled electrical-thermal problem

After setting up FGaBP for the thermal problem, it is time to consider the coupled problem. In our multi-physics modelling of RFA, we start with the coupled electrical-thermal problem as described in Section 2.3. The two coupled PDEs are:

$$\nabla \cdot (\sigma(T)\nabla v) = 0, \tag{4.7a}$$

$$\rho c \frac{\partial T}{\partial t} = Q + \nabla \cdot (d\nabla T), \tag{4.7b}$$

where $\sigma(T)$ shows the temperature dependence of electrical conductivity. We assume $\sigma(T)$ increases linearly with the temperature, using a temperature coefficient of +2 %/°C [8]. Since the heat equation is also coupled to the Laplace's equation by its source term $Q = \sigma|\nabla v|^2$, the thermal profile and the electric field are interdependent. After using theta-scheme for time-discretization of the heat transient problem, and similar FEM space discretization for (4.7a) and (4.7b), we will have the following set of coupled discrete equations:

$$\Big[\mathbf{M}_v(T)\Big]\,\{v\} = 0, \tag{4.8a}$$

$$\begin{aligned}\Big[\mathbf{M}_T - \Delta t\ \theta\ \mathbf{S}_T\Big]\,\{T\}^{n+1} = \Big[\mathbf{M}_T + \Delta t\ (1-\theta)\mathbf{S}_T\Big]\,\{T\}^{n} + \\ \Delta t\ ((1-\theta)\{\mathbf{f}(v)\}^{n} +\ \theta\{\mathbf{f}(v)\}^{n+1}),\end{aligned} \tag{4.8b}$$

where $\mathbf{M}_{v,ij}(T) = \int \sigma(T)N_iN_j\ dV$ and $\mathbf{f}_j(v) = \int \sigma(T)N_j|\nabla v|^2\ dV$. The matrices $\mathbf{M}_T$ and $\mathbf{S}_T$ are exactly the same as $\mathbf{M}$ and $\mathbf{S}$ in (4.4).

The choice of solution approach for these coupled equations relies on a number of considerations. Due to the prohibitive cost and numerical complications of multi-physics modelling, the coupled equations are usually divided into separate problems dealing with each physics exclusively. This leads to the simplification of the coupled problem, allowing individual numerical solutions to each PDE based on their own boundary and/or initial conditions. Moreover, from a practical standpoint, existing codes for component solutions often motivate successive substitution as an expeditious route to a first multi-physics simulation capability making use of the separate physics. This approach, however, may ignore strong couplings between physics and give a false sense of completion. In the next section, we illustrate weakly coupled solution strategies within the context of FGaBP and the coupled problem above.

## 4.3 Weak coupling formulation of FGaBP

FGaBP turns the FEM mesh into a factor graph, over which an inference problems is solved to find the FEM solution. There are two distinct nodes in the factor graph, VN and FN. FGaBP assumes the FEM solution at each VN is a random variable with a Gaussian distribution whose shape is defined by two parameters, $\alpha$ and $\beta$, where $\alpha$ is the reciprocal of the variance and $\beta/\alpha$ is the mean. By passing messages in the graph iteratively, FGaBP tries to find the values of $\alpha$ and $\beta$ for each node. A factor node message, $m_{ai}$, is sent from factor node $a$ ($\mathrm{FN_a}$) to the connected variable node $i$ ($\mathrm{VN_i}$) and represents the most probable state of the solution, from the standpoint of $\mathrm{FN_a}$. In return, the variable node message $\eta_{ia}$ represents observations from other connected FNs. The FGaBP can be summarized in the following two steps:

1. The first step updates $\beta$ and $\alpha$ values for all VNs associated with each FN. This operation is done by each FN sending messages to all connected VNs, which then updates the local $\alpha$ and $\beta$ values. This requires solving two small systems of equations [49]:
$$\alpha_{ai}^{(t)} = \frac{1}{\tilde{W}_{\mathcal{L}(i)}} - \alpha_{ia}^{(t-1)}. \tag{4.9}$$
$$\beta_{ai}^{(t)} = B_{\mathcal{L}(i)} + \frac{1}{\tilde{W}_{\mathcal{L}(i)}} \; (\bar{\mathbf{K}}^{(t)})^T \; \tilde{\mathbf{C}}^T - \beta_{ia}^{(t-1)}. \tag{4.10}$$
Noteworthy, the scalars $\tilde{W}_{\mathcal{L}(i)}$ and $B_{\mathcal{L}(i)}$, and the vectors $\bar{\mathbf{K}}$ and $\tilde{\mathbf{C}}$, are local values corresponding to the local factor $a$. Specifically, $\tilde{W}_{\mathcal{L}(i)}$ and $\tilde{\mathbf{C}}$ depend on the characteristic matrix of $a$, while $B_{\mathcal{L}(i)}$ and $\bar{\mathbf{K}}$ depend on the right hand side vector of $a$.

2. In the second step, each VN receives the new beliefs from the connected FNs, computes VN messages accordingly and sends them back to each neighbouring FN [49]:
$$\begin{aligned} \alpha_i^{(t)} &= \sum_{k \in \mathcal{N}(i)} \alpha_{ki}^{(t)}, \quad \alpha_{ia}^{(t)} = \alpha_i^{(t)} - \alpha_{ai}^{(t)}, \\ \beta_i^{(t)} &= \sum_{k \in \mathcal{N}(i)} \beta_{ki}^{(t)}, \quad \beta_{ia}^{(t)} = \beta_i^{(t)} - \beta_{ai}^{(t)}, \end{aligned} \tag{4.11}$$
where $\mathcal{N}(i)$ is the set of all FNs connected to $\mathrm{VN_i}$.

These steps are repeated until the changes in $\alpha$ and $\beta$ values reach a certain threshold. After convergence, the solution is recovered computing the mean as $\mu = \beta/\alpha$. It is important to observe that FNs (where most of the computation is done) are only connected through variable nodes and that messages are only sent to local neighbouring FNs. This is a key feature that gives FGaBP great potential for parallel processing.

Traditional multi-physics algorithms often hold on to already available codes for each of the uni-physics problems, that is, solving each equation for its own unknown, given the other unknowns. This represents a weakly coupled manner and is done iteratively in either Gauss-Seidel or Jacobi approach. After each iteration, a convergence criterion such as L2-norm of the electrical conductivity is evaluated to determine whether or not to repeat the iteration. When the convergence criterion is fulfilled, the accuracy with which the coupled equations are solved can be improved by continuing the iterations. In the Gauss-Seidel algorithm, each equation is solved for its own unknown, using the most recent values for the other unknowns. This means the coupled equations of (4.7) need to be solved sequentially. On the other hand, a Jacobi algorithm allows all the equations in the same iteration to be solved simultaneously. In the Jacobi algorithm, each equation in (4.7) uses the values for the other unknowns from the last iteration.

The Gauss-Seidel and Jacobi methods are depicted by Figures 4.4 and 4.5, respectively. In the Gauss-Seidel method, component 2 must wait for component 1 to finish its computation and send the potential values. Component 2 then starts computation when component 1 is waiting for it to send temperature values. In the Jacobi algorithm, both components can start computation simultaneously. They interact with each other at the end of each iteration. Avoiding processor idle time, the Jacobi method provides more parallelism, although at the cost of a slower convergence rate. It is particularly advantageous on massively parallel computers.

We can exploit the local computations and message passing strategies used in FGaBP to enhance the computational performance of the Gauss–Seidel method. Unlike a classical Gauss-Seidel, in which the uni-physics problems are solved sequentially, the proposed parallel algorithm solves both electrical and thermal problems in parallel. In this method, each component updates its local data structures and messages according to the other component values in the current iteration. Before describing the algorithm in more details, we need to define a local convergence first.

In order to define the local convergence, we start by assigning a multivariate distribution to each individual factor node, referred to as the factor node belief. The factor node belief takes the form [51]:

$$b_a(\mathbf{U}_a) \propto exp[-\frac{1}{2}\mathbf{U}_a^T\mathbf{W}_a\mathbf{U}_a + \mathbf{K}_a^T\mathbf{U}_a] \tag{4.12}$$

where the matrix $\mathbf{W}_a$ and the vector $\mathbf{K}_a$ are defined by (3.15) and (3.16), and represent the inverse covariance and the source vector of the factor node $a$, respectively. Note that the local belief $b_a$ takes an iterative form, where $\mathbf{W}_a$ and $\mathbf{K}_a$ are formed iteratively according to the FGaBP update rules. It has been shown by [51] that for a vector

```
Code for component 1:
for Time steps do
    for coupling iterations do
        Compute potential;
        Send potential;
        Receive temperature;
        Update $\sigma$;
    end
end
```

```
Code for component 2:
for Time steps do
    for coupling iterations do
        Receive potential;
        Update source term;
        Compute temperature;
        Send temperature;
    end
end
```

Figure 4.4: Conventional Gauss-Seidel algorithm

of nodal marginal means, $\boldsymbol{\mu}_a$, as computed by the FGaBP for the set of nodes in the neighbourhood of factor node $a$, at message convergence we have:

$$\boldsymbol{\mu}_a = \mathbf{W}_a^{-1}\mathbf{K}_a. \tag{4.13}$$

Given the above equation, for each factor $a$ we can formulate a vector referred to as the local residual $\mathbf{r}_a$ given by:

$$\mathbf{r}_a = \mathbf{K}_a - \mathbf{W}_a\,\boldsymbol{\mu}_a. \tag{4.14}$$

The factor node $a$ is considered locally converged if $\mathbf{r}_a$ is smaller than a pre-defined threshold.

The proposed parallel weakly coupled FGaBP algorithm is shown by Fig. 4.6. The electrical component sends the updated $\beta_v$ and $\alpha_v$ messages of the locally converged cells to the thermal component. The thermal problem then updates the local right hand side (RHS) matrices of the corresponding cells and starts calculating $\beta_T$ and $\alpha_T$ messages while waiting for more cells in the electrical problem to converge. Whenever a number of cells are converged in the thermal problem, the local matrices of those cells are updated accordingly in the electrical problem and messages are calculated again. This procedure continues until a convergence criterion such as L2-norm of the electrical conductivity is smaller than a threshold. To accelerate FGaBP iterations, at each coupling iteration we use the values of $\alpha$ and $\beta$ messages from the previous coupling iteration. Hence, the messages only need to be initialized before the first iteration and then will continue by

```
Code for component 1:
for Time steps do
    for coupling iterations do
        Compute potential;
        Receive temperature;
        Update $\sigma$;
        Send potential;
    end
end
```

```
Code for component 2:
for Time steps do
    for coupling iterations do
        Compute temperature;
        Send temperature;
        Receive potential;
        Update source term;
    end
end
```

Figure 4.5: Conventional Jacobi algorithm

just updating their values. In fact, as shown by [49], we only need to update $\beta$ messages since $\alpha$ messages converge after a few iterations.

## 4.4 Results and discussion

In order to simulate the FGaBP Gauss-Seidel algorithm, we used the same geometry, mesh size, time-step and parameter values as in the heat transfer model. As shown in Fig. 4.7, the conducting tip of the electrode is embedded into a 8.0 cm by 8.0 cm rectangular region that simulates tissue surrounding the probe tip. A source voltage of 16 V is applied to the conducting tip of the probe. All of the outer boundaries of the rectangular domain serve as a return ground electrode. Regarding the thermal problem, the initial value of the temperature is chosen to be the normal body temperature, i.e. 37 °C and a Dirichlet boundary condition of 37 °C is applied at the outer boundary.

To verify the correctness of the algorithm, the computed temperature at a certain point in the domain is compared to that obtained from COMSOL Multi-physics software [19]. Figure 4.8 shows the transient temperature computed from both methods; the RMS error between them is less than 0.02 °C which confirms the correctness of our algorithm. To test the parallel scalability properties of Algorithm 4.6, a CPU implementation with multi-threading (OpenMP) is provided. For this purpose, the grid was further refined to a medium sized problem with 1,000,000 elements. The code is executed on a Compute

```
Code for component 1:
for coupling iterations do
    for FGaBP iterations do
        for each factor a do
            Compute messages, (4.9), (4.10);
            Compute local residual, (4.14);
            if local convergence then
                Send potential;
            end
            Receive temperature;
            Update $\sigma$;
        end
    end
end
```

```
Code for component 2:
for coupling iterations do
    for FGaBP iterations do
        for each factor a do
            Receive potential;
            Update source term;
            Compute messages, (4.9), (4.10);
            Compute local residual, (4.14);
            if local convergence then
                Send temperature;
            end
        end
    end
end
```

Figure 4.6: Weakly coupled FGaBP algorithm

Canada cluster node. The node contains 2 × 20-core Intel Gold 6148 Skylake 2.4 GHz CPUs with 186 GB Dynamic random-access memory (DRAM). As shown by Fig. 4.9, the performance scales up almost linearly up to sixteen threads, and then experiences fluctuations when the number of threads increases. The results show speedups of sixteen times with respect to one CPU core.

In this chapter, we modified the classical Gauss-Seidel method and designed a new weak coupling algorithm. This approach is not only more parallelizable than the conventional Gauss-Seidel, but is also more efficient than using FGaBP in a classical Gauss-Seidel since it makes FGaBP solver converge faster. This can be confirmed by using FGaBP in a classical Gauss-Seidel to obtain voltage and temperature values at each iteration, and then comparing the total number of FGaBP iterations with that obtained from Algorithm 4.6. The results indicate that with similar convergence criterion, although Algorithm 4.6 needs more Gauss-Seidel iterations to converge, e.g., 11 iterations compared

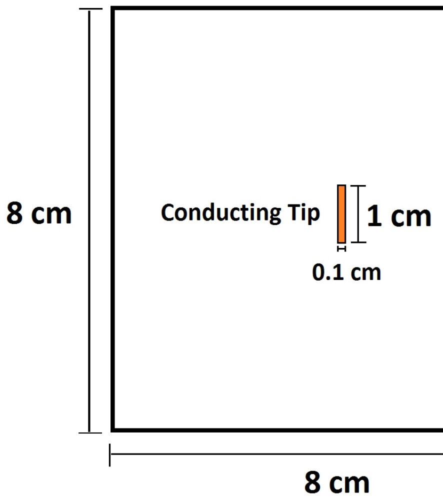

Figure 4.7: The geometry of the simple test case.

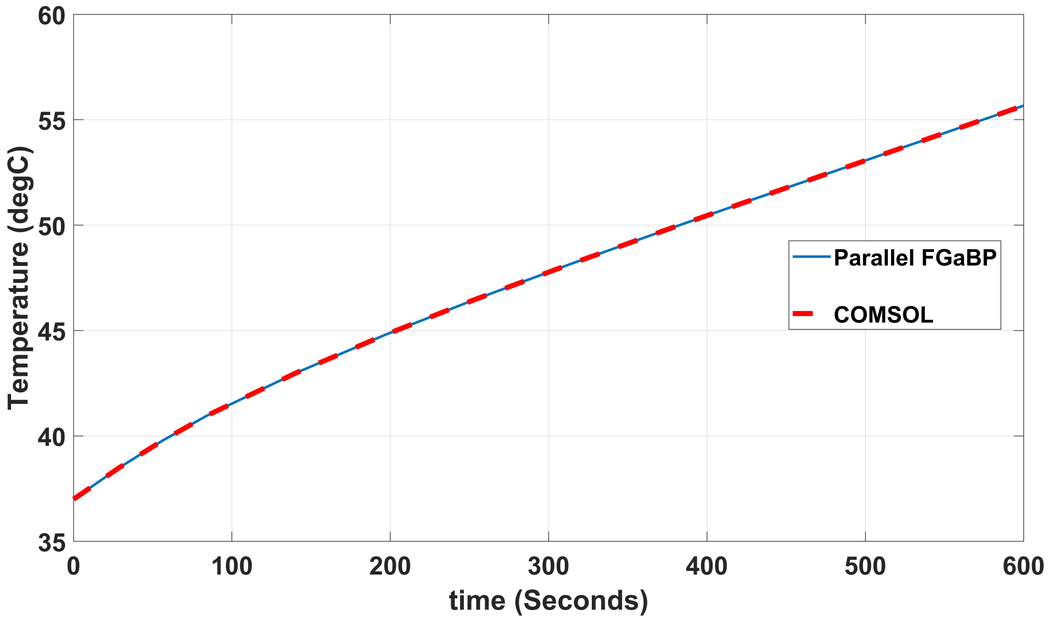

Figure 4.8: Temperature at a specific point over 10 minutes of ablation, obtained from the multi-physics FGaBP algorithm and COMSOL multi-physics software.

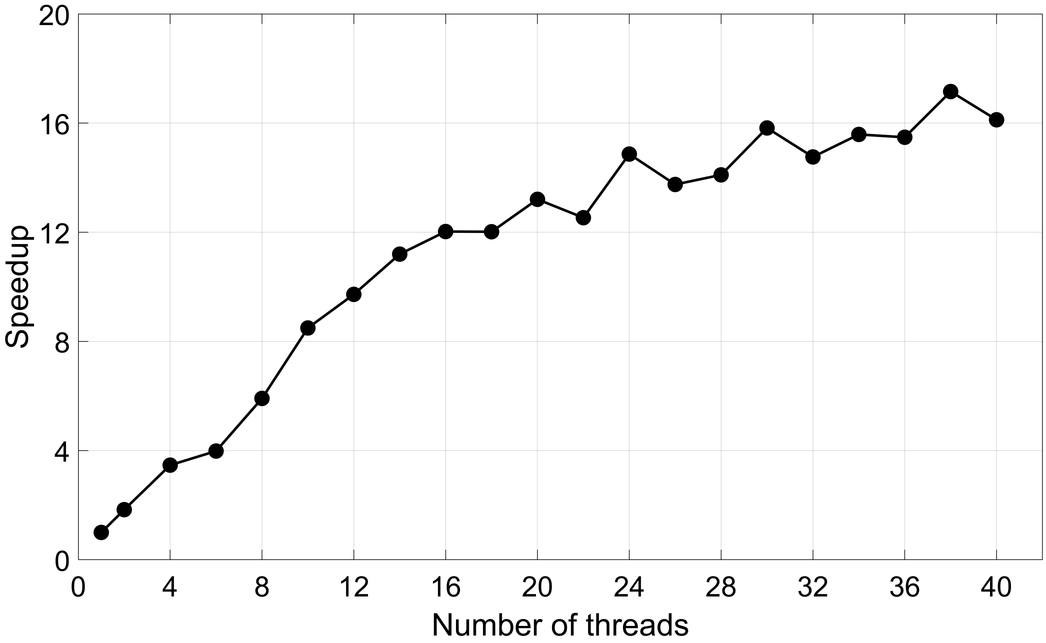

Figure 4.9: Performance scaling of the multi-physics FGaBP method in terms of speedup with respect to 1 core implementation.

to 8 iterations, its total number of FGaBP iterations is about half of that needed in the conventional Gauss-Seidel. In the next chapter, we will present a parallel strong coupling algorithm based on Newton's method and FGaBP.

# Chapter 5

# Parallel Strong Coupling Modelling of RFA

In the previous chapter, the coupled electrical-thermal problem that appears in the multiphysics modelling of RFA was solved following a parallel weak coupling approach, based on the FGaBP algorithm. This chapter provides the strong coupling modelling of the same coupled electrical-thermal problem, and also of the coupled thermal-thermal problem in RFA heat transfer equations.

Strong coupling is usually based on the NR method. As described in Section 2.3.2, the construction of a Jacobian matrix at each linearizing iteration of NR could be prohibitively expensive for large scale problems. The 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 SpMV.

The SpMV operation, as explained by Section 3.1, can strongly limit the acceleration of the solving stage using parallel processing due to its dependency on the underlying sparse data-structure [49]. 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.

This chapter presents a NR reformulation of the FGaBP in order to exploit its localized computations and message passing scheme for solving multi-physics problems in parallel. Similar to the JFNK, FGaBP 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.

## 5.1 Parallel strong coupling formulation of the coupled electrical-thermal problem

The focus of this section is the coupled electrical-thermal problem. For this reason, the coupling between the tissue temperature and the blood temperature is neglected here and the thermal problem is modelled by the the Pennes equation [66]:

$$\rho_{ti} c_{ti} \frac{\partial T}{\partial t} = Q + \nabla \cdot (d \nabla T) + H(T_{bl} - T), \tag{5.1}$$

where $T$ is the temperature (K), $c_{ti}$ is the special heat capacity of tissue (J/kg/K), $\rho_{ti}$ is the tissue density (kg/m$^3$), $d$ is the thermal conductivity (W/m/K), $H$ is the convective transfer coefficient (W/m$^3$/K) and $T_{bl}$ is the baseline physiological blood temperature taken to be 310 K. The electric potential around the probe is solved using Laplace's equation:

$$\nabla \cdot (\sigma(T) \nabla v) = 0, \tag{5.2}$$

where $\sigma(T)$ is the temperature-dependent electrical conductivity (S/m), and $v$ is the electric potential (V).

We use the theta-scheme [67] for time discretization of (5.1), and similar FEM mesh for (5.1) and (5.2), to obtain the following set of coupled discrete equations:

$$\Big[\mathbf{M}_v(T)\Big]\left\{\mathbf{v}\right\} = 0, \tag{5.3a}$$

$$\begin{aligned}\Big[\mathbf{M}_T + \delta t\ \theta\ \mathbf{S}_T\Big]\left\{\mathbf{T}\right\}^{n+1} &= \Big[\mathbf{M}_T - \delta t\ (1-\theta)\mathbf{S}_T\Big]\left\{\mathbf{T}\right\}^{n} \\ &+ \delta t\ \Big\{(1-\theta)\left\{\mathbf{f}(v)\right\}^{n} +\ \theta\left\{\mathbf{f}(v)\right\}^{n+1}\Big\},\end{aligned} \tag{5.3b}$$

where $M_{v,ij}(T) = \int \sigma(T)\overrightarrow{\nabla} N_i \cdot \overrightarrow{\nabla} N_j\ dV$ 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 $S_{T,ij} = d\int \overrightarrow{\nabla} N_i \cdot \overrightarrow{\nabla} N_j\ dV$. Note that $N_i$ and $N_j$ are FEM scalar basis functions, $n$ is the time step number, $\delta t$ is the time step value and $0 \leq \theta \leq 1$ is the parameter of the theta-scheme time discretization.

### 5.1.1 Formulation

Owing to its 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 [40]. If a solver code that goes beyond the weak-coupling is needed, the NR method is the simplest algorithm. This chapter provides an efficient FEM-based NR algorithm for solving the coupled electrical-thermal problem of (5.3) in a strong coupling manner. For this purpose, a parallel FEM formulation based on FGaBP algorithm is modified for solving multi-physics problems.

The FGaBP algorithm is already explained in Chapter 3. The message passing scheme of FGaBP is shown graphically by Fig. 5.1. In a uni-physics problem, each message contains information about the value of the unknown parameter at each node. For example, in Fig. 5.1, 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

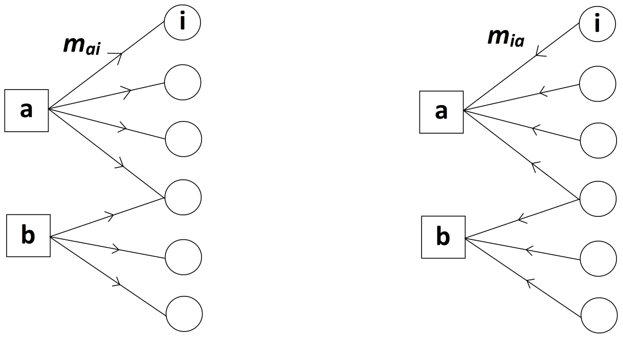

Figure 5.1: (*Left*): A message $m_{ai}$ is sent from $\mathrm{FN_a}$ to $\mathrm{VN_i}$. (*Right*): A message $m_{ia}$ is sent back from $\mathrm{VN_i}$ to $\mathrm{FN_a}$.

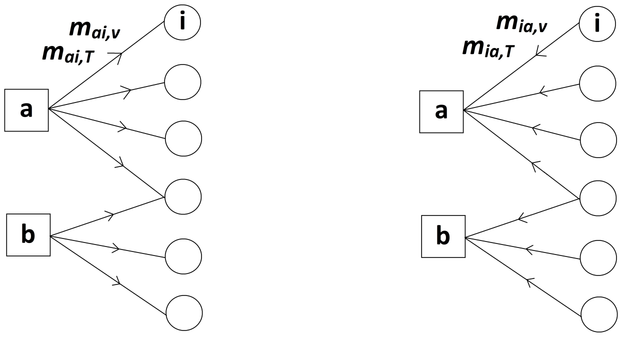

Figure 5.2: (*Left*): A message $m_{ai,v}$ ($m_{ai,T}$) is sent from $\mathrm{FN_a}$ to $\mathrm{VN_i}$ in the electrical (thermal) problem . (*Right*): A message $m_{ia,v}$ ($m_{ia,T}$) is sent back from $\mathrm{VN_i}$ to $\mathrm{FN_a}$ in the electrical (thermal) problem.

their values. Figure 5.2 shows the message passing scheme in a multi-physics scenario. Here, a massage $m_{ai,v}$ ($m_{ai,T}$) sent in the electrical (thermal) problem carries information about $v_i$ ($T_i$) and also $dv_i/dT_j$ ($dT_i/dv_j$).

When FGaBP update rules introduced by (3.13) to (3.23) are applied to the coupled electrical-thermal problem, it should be noted that, in the electrical problem, element $a$ characteristic matrix, $\mathbf{M}_v$, depends on the temperature values at nodes $i \in \mathcal{N}(a)$, where $\mathcal{N}(a)$ shows the neighbourhood of factor node $a$. On the other hand, in the thermal problem, element $a$ source vector, $\mathbf{B}_T$, depends on the electric potential values at nodes $i \in \mathcal{N}(a)$. Consequently, the $\alpha$ and $\beta$ messages update rules can be rewritten as:

$$\alpha_{ai,v}^{(t)} = \frac{1}{\tilde{W}_{\mathcal{L}(i),v}(T_j)} - \alpha_{ia,v}^{(t_*)} , \tag{5.4a}$$

$$\beta_{ai,v}^{(t)} = B_{\mathcal{L}(i),v} + \frac{1}{\tilde{W}_{\mathcal{L}(i),v}(T_j)} \; (\bar{\mathbf{K}}_v{}^{(t_*)}(v_j))^T \; \tilde{\mathbf{C}}_v(T_j) , \tag{5.4b}$$

$$\alpha_{ai,T}^{(t)} = \frac{1}{\tilde{W}_{\mathcal{L}(i),T}} - \alpha_{ia,T}^{(t_*)} \,, \tag{5.4c}$$

$$\beta_{ai,T}^{(t)} = B_{\mathcal{L}(i),T}(v_j) + \frac{1}{\tilde{W}_{\mathcal{L}(i),T}} \; (\bar{\mathbf{K}_T}^{(t_*)}(v_j, T_j))^T \; \tilde{\mathbf{C}_T}, \tag{5.4d}$$

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 (5.4a) to (5.4d) present a coupled system of non-linear equations at each FGaBP iteration inside factor node $a$. The rest of this section provides a parallel strong coupling algorithm to solve these coupled equations. According to (5.4c), $\alpha$ messages in the thermal problem are independent of the potential values; therefore, we only consider the coupling between (5.4a), (5.4b), and (5.4d):

$$\begin{cases} \alpha_{ai,v}^{(t)} = \frac{1}{\tilde{W}_{\mathcal{L}(i),v}(T_j)} - \alpha_{ia,v}^{(t_*)} \,, \\[2ex] \beta_{ai,v}^{(t)} = B_{\mathcal{L}(i),v} + \frac{1}{\tilde{W}_{\mathcal{L}(i),v}(T_j)} \; (\bar{\mathbf{K}_v}^{(t_*)}(v_j))^T \; \tilde{\mathbf{C}}_v(T_j), \\[2ex] \beta_{ai,T}^{(t)} = B_{\mathcal{L}(i),T}(v_j) + \frac{1}{\tilde{W}_{\mathcal{L}(i),T}} \; (\bar{\mathbf{K}_T}^{(t_*)}(v_j, T_j))^T \; \tilde{\mathbf{C}_T}. \end{cases} \tag{5.5}$$

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

$$\mathbf{F}(\mathbf{x}) \; = 0, \tag{5.6}$$

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

$$\mathbf{J}(\mathbf{x}^{(m)})\Delta\mathbf{x}^{(m)} = -\mathbf{F}(\mathbf{x}^{(m)}), \tag{5.7}$$

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

$$\mathbf{x}^{(m+1)} = \mathbf{x}^{(m)} + \Delta\mathbf{x}^{(m)}. \tag{5.8}$$

Here, we apply the NR method to (5.5), i.e., at the element-level. 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 factor node $a$:

$$\begin{cases} \alpha_{ak,v}^{(m,t)} - f_k(v_l^{(m,t_*)}, T_l^{(m-1)}) = 0 \\ \beta_{ak,v}^{(m,t)} - g_k(v_l^{(m,t_*)}, T_l^{(m-1)}) = 0 \\ \beta_{ak,T}^{(m,t)} - h_k(v_l^{(m-1)}, T_l^{(m,t_*)}) = 0\,, \end{cases} \tag{5.9}$$

where $f_k$, $g_k$ and $h_k$ are the right-hand-sides of the equations in (5.5). 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 factor $a$. After FGaBP iterations have converged, the non-linear system of (5.5) can be solved with a pure NR method, using local data for each FN. A local Jacobian matrix is constructed for factor node $a$ based on (5.5) and the sensitivities calculated during FGaBP as follows:

$$\mathbf{J}_a = \begin{bmatrix} \mathbf{I}_{2nc\times 2nc} & \mathbf{J}_{a,v} \\ \mathbf{J}_{a,T} & \mathbf{I}_{nc\times nc} \end{bmatrix}_{3nc\times 3nc}, \tag{5.10}$$

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 Appendix A. 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 $a$; Thus, the Jacobian matrix associated with factor $a$, $\mathbf{J}_a$, is only dependent on the local data structure of factor $a$, i.e., dense matrix $\mathbf{M}_a$, vector $\mathbf{B}_a$, and messages $(\alpha_{ai}, \beta_{ai})$. After FGaBP iterations have converged and the local Jacobian is formed for factor node $a$, the message updates are computed as:

$$\begin{Bmatrix} \left\{\Delta\alpha_{ak,v}\right\}_{nc\times 1} \\ \left\{\Delta\beta_{ak,v}\right\}_{nc\times 1} \\ \left\{\Delta\beta_{ak,T}\right\}_{nc\times 1} \end{Bmatrix}^{(m)} = \left[\mathbf{J}_a^{-1}\right]^{(m)} \cdot \begin{Bmatrix} \left\{f_k - \alpha_{ak,v}\right\}_{nc\times 1} \\ \left\{g_k - \beta_{ak,v}\right\}_{nc\times 1} \\ \left\{h_k - \beta_{ak,T}\right\}_{nc\times 1} \end{Bmatrix}^{(m)}. \tag{5.11}$$

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 (5.11):

$$\begin{Bmatrix} \left\{\alpha_{ak,v}\right\} \\ \left\{\beta_{ak,v}\right\} \\ \left\{\beta_{ak,T}\right\} \end{Bmatrix}^{(m+1)} = \begin{Bmatrix} \left\{\alpha_{ak,v}\right\} \\ \left\{\beta_{ak,v}\right\} \\ \left\{\beta_{ak,T}\right\} \end{Bmatrix}^{(m)} + \begin{Bmatrix} \left\{\Delta\alpha_{ak,v}\right\} \\ \left\{\Delta\beta_{ak,v}\right\} \\ \left\{\Delta\beta_{ak,T}\right\} \end{Bmatrix}^{(m)}. \tag{5.12}$$

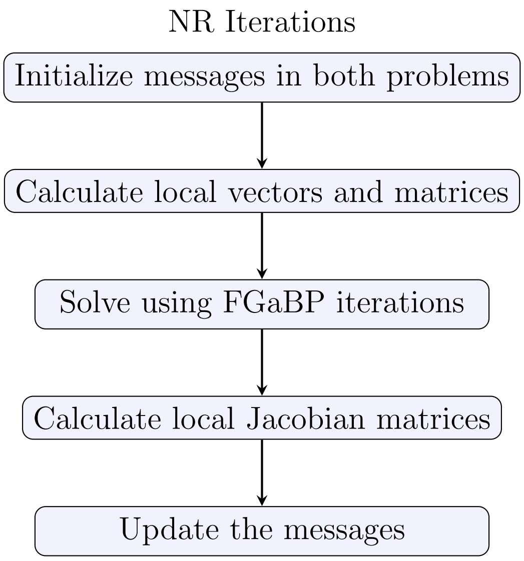
NR Iterations
Initialize messages in both problems
Calculate local vectors and matrices
Solve using FGaBP iterations
Calculate local Jacobian matrices
Update the messages

Figure 5.3: The parallel NR algorithm.

The updated messages are then used in the next NR iteration. After the NR iterations have converged, we proceed to the next time step. Figure 5.3 shows the element-wise NR iterations graphically.

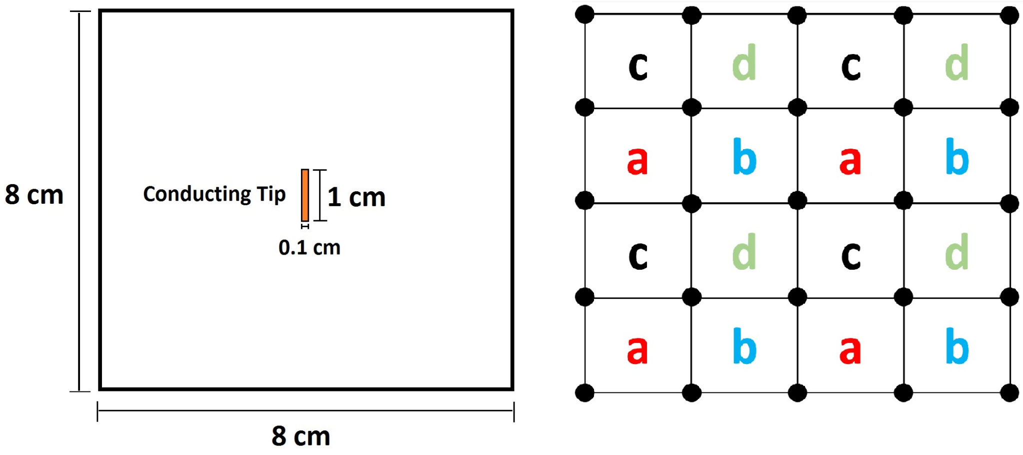

Figure 5.4: (*Left*): The geometry of the test case in two dimensions. (*Right*): Structured quadrilateral mesh containing four colours.

```
Import problem geometry and mesh;
for NR iteration $m = 1, 2, \ldots$ do
    for each level in the mesh hierarchy do
        for cell $a = 1, 2, \ldots$ do
            for node $i \in \mathcal{N}(a)$ do
                Initialize messages $m_{ai,(v,T)}$;
            end
            Calculate local vector $\mathbf{f}$ and local matrices $\mathbf{M}_{\mathrm{v,T}}$ inside cell $a$;
        end
    end
    for cycles $= 1, 2, \ldots$ in the V-cycle do
        for Mesh levels from fine to coarse do
            Execute $v_1$ iterations of Algorithm 2;
            Restrict;
        end
        Execute Algorithm 2 on the coarsest level;
        for Mesh levels from coarse to fine do
            Execute $v_2$ iterations of Algorithm 2;
            Prolongate;
        end
        if global tolerance $<$ tolerance then
            break;
        end
    end
    Execute Algorithm 3 on the finest level;
    if global residual $<$ NR tolerance then
        break;
    end
end
```

Figure 5.5: Parallel NR pseudo-code in each time step.

### 5.1.2 Implementation

In this section, the parallel NR algorithm is applied to the coupled equations of (5.3) in two dimensional (2D) and three dimensional (3D) geometries. As shown in Fig. 5.4, left, tissue surrounding the probe tip is modelled by an 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 modelled 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 (5.1) and (5.2) are chosen according to Table 2.1. 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 [68].

```
for FGaBP iteration $t = 1, 2, \ldots$ do
    for colour $c = 1, 2, \ldots$ do
        for cell $a$ in colour $c$ do
            for node $i \in \mathcal{N}(a)$ do
                Calculate $m_{ai(v,T)}$, i.e., (3.13) to (3.23);
                Calculate $J_a$, i.e., (5.10) ;
                Update $v_i$ and $T_i$, i.e., (5.12) ;
                Calculate local message tolerance;
            end
        end
        Update global message tolerance;
    end
    if global tolerance ¡ FGaBP tolerance then
        return(global tolerance);
        break;
    end
end
```

Figure 5.6: FGaBP with local Jacobian calculation

```
for colour $c = 1, 2, \ldots$ do
    for cell $a$ in colour $c$ do
        Update $m_{ai,(v,T)}$;
        Update $v_i$ and $T_i$;
        Calculate local NR residual;
        Calculate local vector $\mathbf{f}$ and local matrices $\mathbf{M}_{\mathrm{v,T}}$ inside cell $a$.
    end
    Update global NR residual;
end
```

Figure 5.7: NR parallel update

An element-based colouring 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. 5.2, 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 colouring. The mesh elements are coloured so that no two adjacent elements have the same colour symbol. In this way, the messages in each colour 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 colouring diagram in 2D is illustrated in Fig. 5.4, right, using a quadrilateral mesh.

Mesh refinement is conduced 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 is 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 sensitivity of these values with respect to their neighbouring nodes. The transfer of information between different refinement levels is conducted in a multi-grid scheme [61].

Figure 5.5 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 [61]. The algorithms depicted by Fig. 5.6 and Fig. 5.7 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 steps in Fig. 5.5; Hence, these two steps are implemented in parallel exactly as proposed by [61] and described in Section 3.3. Because of the distributed nature of our method, no global Jacobian matrices are assembled which means the application of a conventional preconditioner is not needed; however, the multi-grid approach acts as a preconditioner in reducing the number of iterations on the finest level.

### 5.1.3 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 algorithm at a specific location during one minute of ablation are shown in Fig. 5.8. Quantitatively, the 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 Fig. 5.5, 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 Fig. 5.5, 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 [61]. Note that the number of iterations in the inner FGaBP execution, i.e., line 19 in Fig. 5.5, 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 and 3D implementations. Figure 5.9

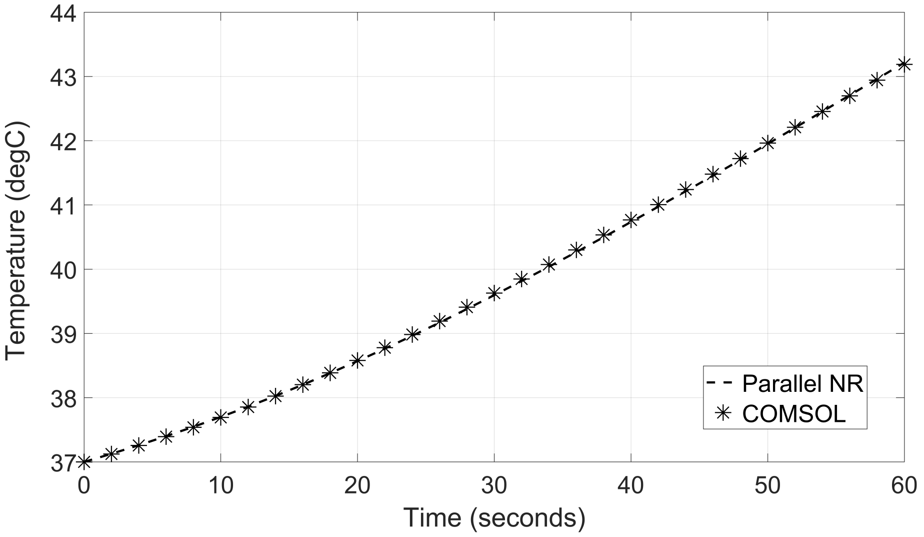

Figure 5.8: Temperature values over time at a specific point, i.e., $x = 1$ cm and $y = 1$ cm in 2D domain, when the ablation probe tip is located at the center of the domain, i.e. $x = 0$ cm and $y = 0$, cm . The time step size $\delta t$ is equal to 0.5 s.

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 [69], 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 varied from 410,881 to 13,148,192 unknowns. The scaled speedup data and a linear curve fitted to it based on Gustafson's law [70] are depicted in Fig. 5.10. 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. 5.11, a comparison between static and dynamic scheduling with different chunk sizes is provided. The number of threads is chosen to be 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. 5.11, the best performance observed is similar to that delivered by the default scheduling in Fig. 5.9. Also, for

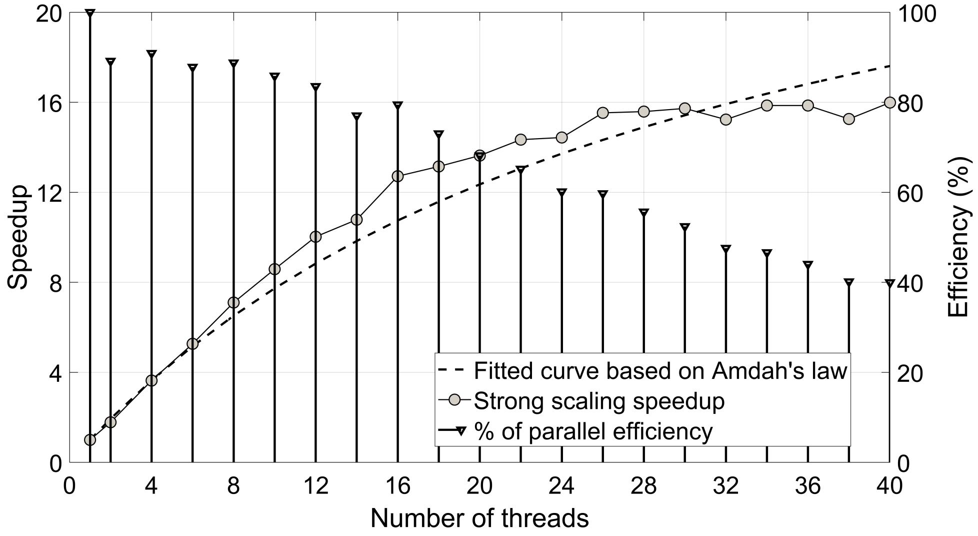

Figure 5.9: Strong scaling of the multi-physics NR method in terms of speedup with respect to 1 core implementation. The dashed line is the fitted curve based on Amdah's law.

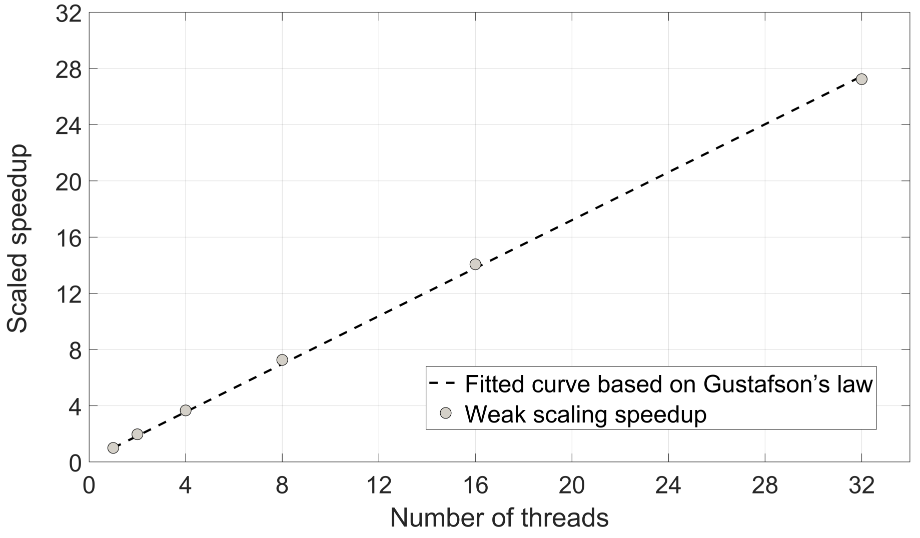

Figure 5.10: Weak scaling of the multi-physics NR method in terms of scaled speedup with respect to 1 core implementation. The dashed line is the fitted curve based on Gustafson's law

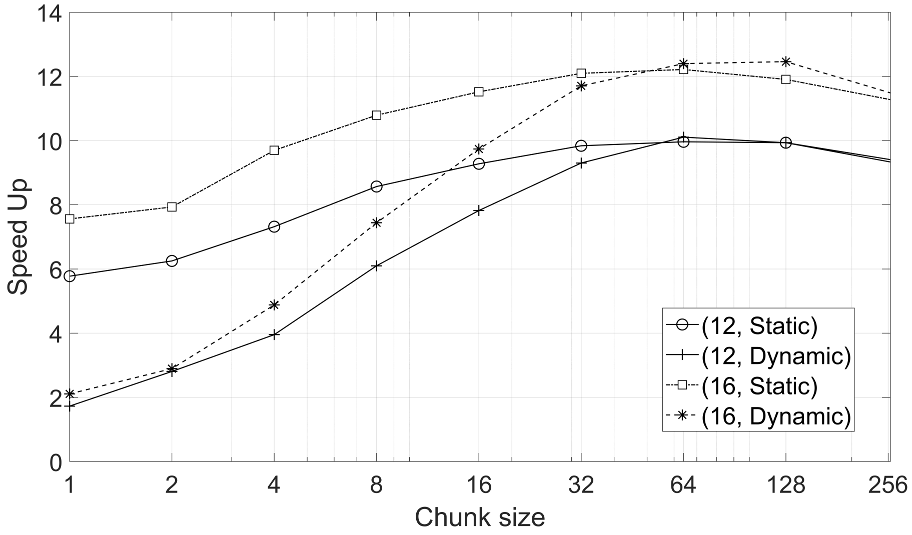

Figure 5.11: Speedup of the parallel NR method in terms of OpenMP *for* loop's *scheduling type* and *chunk size* when the number of threads = 12 and 16.

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 Portable, Extensible Toolkit for Scientific Computation (PETSc)[71, 72, 73]. PETSc employs the MPI standard for communication 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) [74, 75] is selected here. Figure 5.12 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.

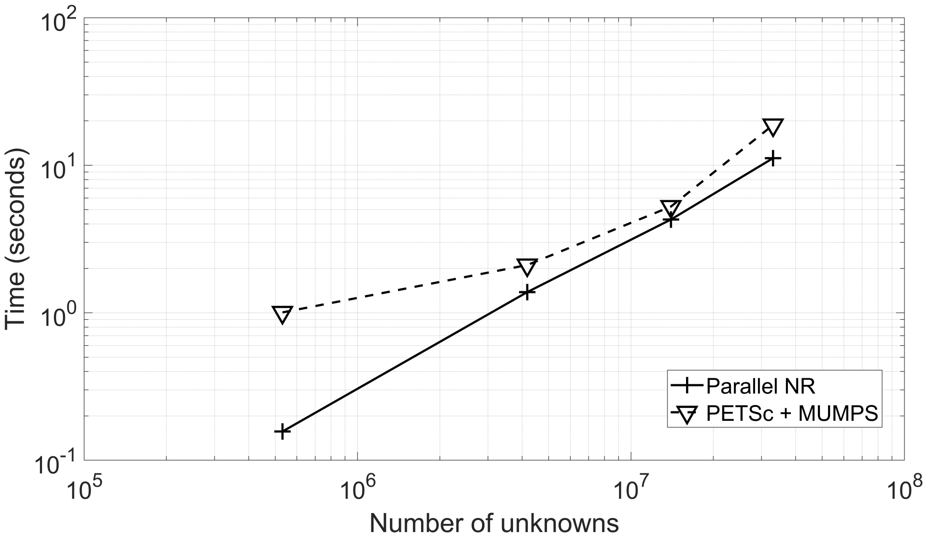

Figure 5.12: Execution times using 16 cores.

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. 5.13. 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.

## 5.2 Parallel strong coupling formulation of the thermal tissue-blood interaction

This section presents an MPI-based extension of the multi-physics algorithm in Section 5.1. Instead of the electrical-thermal coupling, the multi-physics heat transfer problem in radiofrequency ablation is considered in this section. The Pennes model which was used in the previous section assumes the blood temperature is constant, i.e., 310 K. However, this is only true within and close to large vessels [15]. For this reason, Payne *et al.* [11] model the heat transfer problem with two coupled equations:

$$(1-\epsilon)\rho_{ti}c_{ti}\frac{\partial T_{ti}}{\partial t} = (1-\epsilon)Q + (1-\epsilon)\nabla \cdot (d\nabla T_{ti}) + H(T_{bl} - T_{ti}), \tag{5.13a}$$

$$\epsilon\rho_{bl}c_{bl}(\frac{\partial T_{bl}}{\partial t} + \mathbf{v} \cdot \nabla T_{bl}) = \epsilon Q + \epsilon\nabla \cdot (d\nabla T_{bl}) - H(T_{bl} - T_{ti}). \tag{5.13b}$$

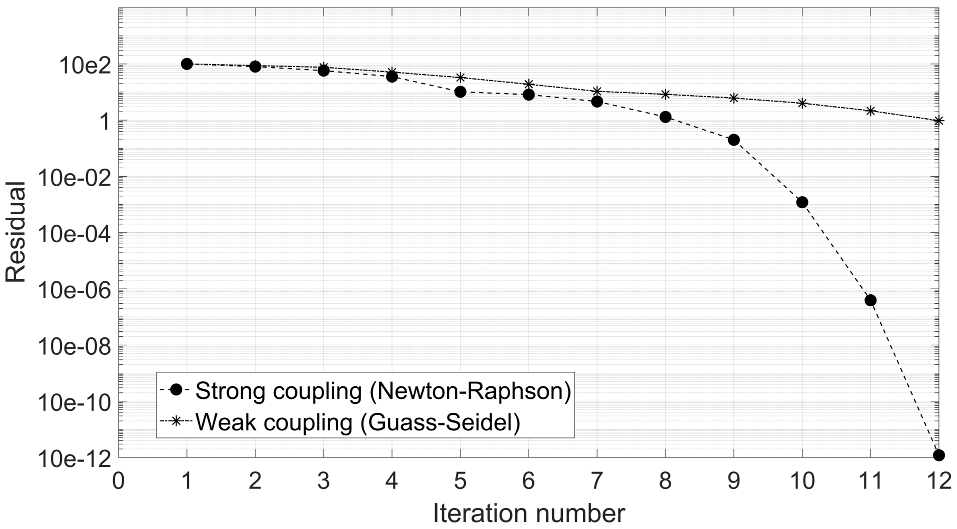

Figure 5.13: The convergence plot of the parallel NR algorithm compared to that of the Gauss-Seidel algorithm.

In these two equations, subscripts '*ti*' and '*bl*' stand for tissue and blood, respectively. The two additional parameters $\epsilon$ and $\mathbf{v}$ are the fraction of blood volume over total volume and blood velocity (m/s), respectively. These equations are nonlinear due to the linear dependency of thermal conductivity $d$ on temperature. Payne *et al.* [11] solve the coupled blood and tissue equations based on the Picard linearization which is only valid if the non-linearity is mild. In this section, the Newton Raphson method is used to solve the non-linear coupled equations of (5.13). The spatial discretization is based on FEM and the backward differentiation formula of order two is employed for time discretization:

$$(1-\epsilon)\rho_{ti}c_{ti}\frac{3T_{ti}^{n+1}-4T_{ti}^{n}+T_{ti}^{n-1}}{2\Delta t} = (1-\epsilon)Q + \tag{5.14a}$$

$$(1-\epsilon)\nabla\cdot(d\nabla T_{ti}^{n+1}) + H(T_{bl}^{n+1}-T_{ti}^{n+1}),$$

$$\epsilon\rho_{bl}c_{bl}\frac{3T_{bl}^{n+1}-4T_{bl}^{n}+T_{bl}^{n-1}}{2\Delta t} = \epsilon Q \ + \epsilon\nabla\cdot(d\nabla T_{bl}^{n+1}) \tag{5.14b}$$

$$+H(T_{ti}^{n+1}-T_{bl}^{n+1}) - (\epsilon\rho_{bl}c_{bl})\mathbf{v}\cdot\nabla T_{bl}^{n+1},$$

where the superscript $n$ denotes the n$th$ time step and $\Delta t$ is the time step value.

### 5.2.1 Algorithm

Similar to the coupled electrical-thermal problem in the previous section, when applied to the coupled thermal equations in (5.14), the FGaBP messages should carry information

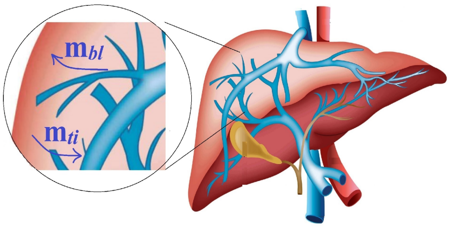

Figure 5.14: The tissue-blood interaction with message passing, where $m_{bl}$ ($m_{ti}$) contains blood (tissue) temperature information [76].

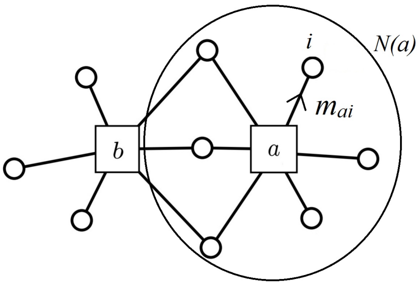

Figure 5.15: The neighbourhood of factor node $a$.

on how the tissue (blood) temperature depends on the blood (tissue) temperature. By doing so, we can deal with the tissue-blood interaction in the element level, avoiding the construction of a global Jacobian which is needed by the conventional NR. The key property of FGaBP which allows this is that the messages sent by a FN at each FGaBP iteration only depend on the local data, i.e., the values in the neighbourhood of the FN. Mathematically speaking, we can write:

$$\begin{cases} m_{ai}^{ti} = f_i(\mathbf{T}_a^{ti}, \mathbf{T}_a^{bl}) \\ \\ m_{ai}^{bl} = g_i(\mathbf{T}_a^{ti}, \mathbf{T}_a^{bl}). \end{cases} \tag{5.15}$$

Here, $m_{ai}^{ti}$ and $m_{ai}^{bl}$ are messages sent from factor node $a$ to variable node $i \in \mathcal{N}(a)$ for tissue and blood temperature, respectively. The vectors $\mathbf{T}_a^{ti}$ and $\mathbf{T}_a^{bl}$ contain tissue temperature and blood temperature in the neighborhood of factor node $a$, $\mathcal{N}(a)$. The functions $f_i$ and $g_i$ describe the dependency of the messages on the temperature values. It's important to note that $f_i$ ($g_i$) is a non-linear function of $\mathbf{T}_a^{ti}$ ($\mathbf{T}_a^{bl}$) because of the temperature dependency of $d$ in (5.13).

At each FGaBP iteration, the non-linear system of (5.15) can be solved with Newton's method. A local Jacobian matrix, $J_a$ is constructed based on the functions $f_i$ and

$g_i$ as follows:

$$\mathbf{J}_a = \begin{bmatrix} \mathbf{J}_{ti,ti} & \mathbf{J}_{ti,bl} \\ \mathbf{J}_{bl,ti} & \mathbf{J}_{bl,bl} \end{bmatrix}_{2n\times 2n}, \tag{5.16}$$

where $n$ is the number of nodes per element. The entries of $J_a$ are found based on the derivatives of $f_i$ and $g_i$ with respect to $m_{ai}^{ti}$ and $m_{ai}^{bl}$. For instance, the sub-matrix $J_{ti,ti}$ is formed as:

$$\mathbf{J}_{ti,ti}(\mathcal{L}(i), \mathcal{L}(j)) = \frac{\partial(m_{ai}^{ti} - f_i)}{\partial m_{aj}^{ti}}, \tag{5.17}$$

where $1 \leq \mathcal{L}(i), \mathcal{L}(j) \leq n$ are the local indices corresponding to the global variable nodes $i, j \in \mathcal{N}(a)$. These partial derivatives are calculated analytically based on the update rules of FGaBP introduced by [61]. After the local Jacobian is formed, message updates are computed as:

$$\left\{\begin{matrix} \left\{\Delta m_{ai}^{ti}\right\}_{n\times 1} \\ \left\{\Delta m_{ai}^{bl}\right\}_{n\times 1} \end{matrix}\right\}_{2n\times 1} = \left[\mathbf{J}_a^{-1}\right] \cdot \left\{\begin{matrix} \left\{f_i - m_{ai}^{ti}\right\}_{n\times 1} \\ \left\{g_i - m_{ai}^{bl}\right\}_{n\times 1} \end{matrix}\right\}_{2n\times 1}. \tag{5.18}$$

The above update rule is carried out for each factor node $a$ and then the FGaBP algorithm propagates the updated messages throughout the mesh. After the FGaBP iterations have converged, we proceed to the next Newton iteration and compute the local Jacobians and update messages again. This continues until the local update messages are smaller than a threshold, meaning the cell-wise Newton's method has converged for the current time step.

### 5.2.2 Implementation and Results

The computational domain is discretized using a hexahedral mesh. Starting by a coarse mesh (and its corresponding graph), mesh refinement is conduced by splitting each hexahedral element into eight smaller elements successively. As proposed by [61], a multi-grid scheme to transfer information between different refinement levels is used to accelerate FGaBP iterations. The multi-grid approach resembles a preconditioner in reducing the number of iterations on the finest level.

An MPI (message passing interface) version of the multi-physics FGaBP method is developed in this section. The MPI code is built upon the parallel distributed computing

```
while NR threshold < global NR residual do
    for cell each a in sub-domain s, and for node each $i \in \mathcal{N}(a)$ do
        Calculate local Jacobian $J_a$;
        Calculate local residual;
        Update the messages $m_{ai}^{ti}$ and $m_{ai}^{bl}$;
        if i is at the interface then
            Send out $m_{ai}^{ti}$ and $m_{ai}^{bl}$ to the adjacent sub-domains;
        end
    end
    Update global residual;
    for FGaBP iteration $t = 1, 2, \ldots$ do
        for each sub-domain s in the domain do
            Load input messages from adjacent sub-domains;
            for cell a in sub-domain s do
                for node $i \in \mathcal{N}(a)$ do
                    Propagate the messages $m_{ai}^{ti}$ and $m_{ai}^{bl}$ inside $s$;
                    if i is at the interface then
                        Send out $m_{ai}^{ti}$ and $m_{ai}^{bl}$ to the adjacent sub-domain;
                    end
                    Update $T_i^{ti}$ and $T_i^{bl}$;
                    Calculate and send out local message residual;
                end
            end
            Update global message residual;
        end
        if global message residual < FGaBP threshold then
            break;
        end
    end
end
```

**Algorithm 3:** Parallel NR with local Jacobian calculation.

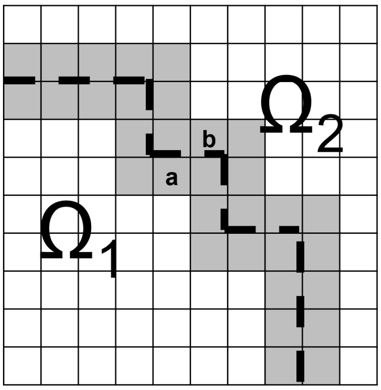

Figure 5.16: Mesh partioning using p4est. The dashed line shows the interface between two sub-domains $\Omega_1$ and $\Omega_2$.

in deal.II [77]. The parallel pseudo-code is shown by Algorithm 3. For massively parallel computations, deal.II builds on the p4est [78] library and so does our code for mesh partitioning. Deal.II assigns each part of the partitioned mesh to one MPI process. Each processor stores the cells it owns and also one layer of adjacent cells—called ghost cells—that are owned by other processors. The locally owned cells and ghost cells can be identified by specific indices that deal.II assigns to them. We exploit this feature to define adjacent factor nodes and interface nodes in each sub-domain, and communicate messages between MPI processes. Such communication occurs at the end of each FGaBP iteration. This is shown graphically by Fig. 5.16. The dashed line is the interface between the two partitions, while the layers shown by grey elements are the ghost layers and are shared between the two partitions. Consequently, the messages sent inside elements *a* and *b* are communicated between the two processors.

The numerical results of the parallel algorithm are verified against the built-in RFA model in COMSOL Multiphysics software. Figure 5.17 shows the geometry of the model in COMSOL. The tissue temperature from COMSOL and our algorithm at a specific location inside the domain are depicted in Fig. 5.18. The RMS error for 10 minutes of ablation is 0.021 °C. To assess the parallel efficiency of our method, we perform a strong scaling analysis. The global unstructured mesh is partitioned first. We map each partition to one MPI process (node), and use one thread per process. The algorithm is tested on Compute Canada cluster nodes. Each node contains an Intel Gold 6148 Skylake 2.4 GHz CPU with 186 GB DRAM. Figure 5.19 shows the speedup for solving the multi-physics system of 200,000 double-precision unknowns. The speed-up degrades from 1 to

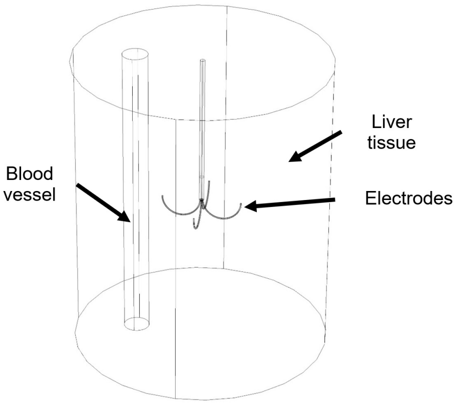

Figure 5.17: The built-in COMSOL model of RFA.

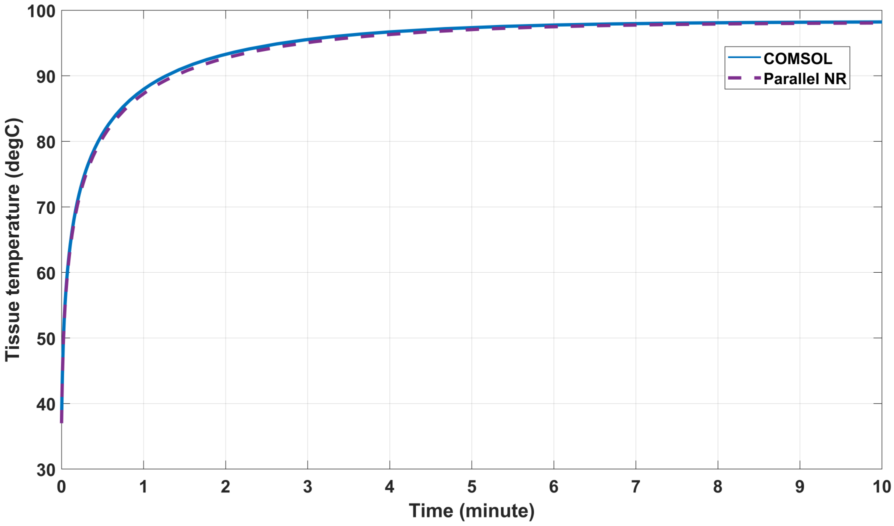

Figure 5.18: Comparison between COMSOL and the parallel NR algorithm. Plot shows tissue temperature at a specific point for 10 minutes of ablation.

2 processes because of the introduction of mesh partitioning and MPI communication. The parallel performance then improves by increasing the number of processes up to 128. After this point, the overhead of communication between the processes and also the assembly of the solution vector dominates the computation time, resulting in the degradation of parallel scalability.

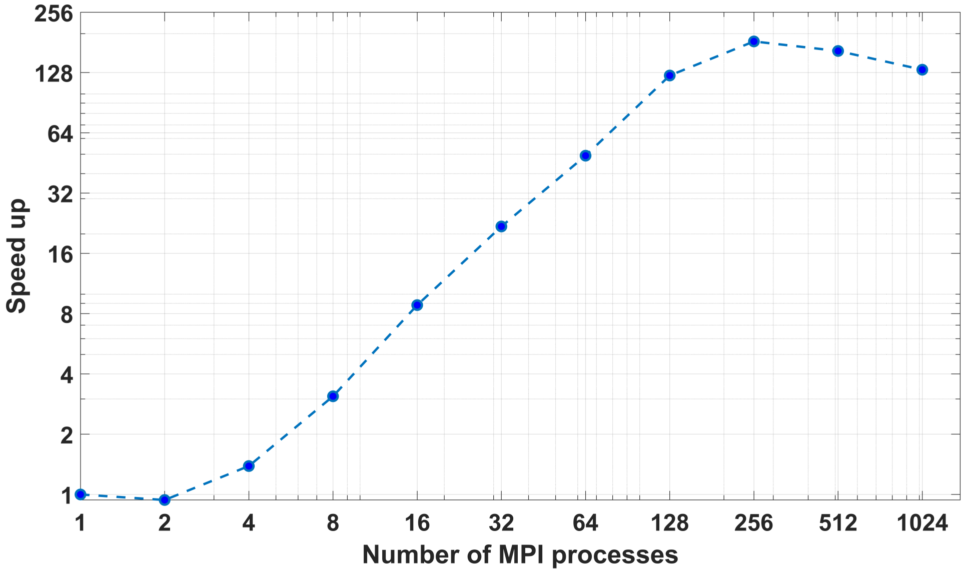

Figure 5.19: Strong scaling analysis.

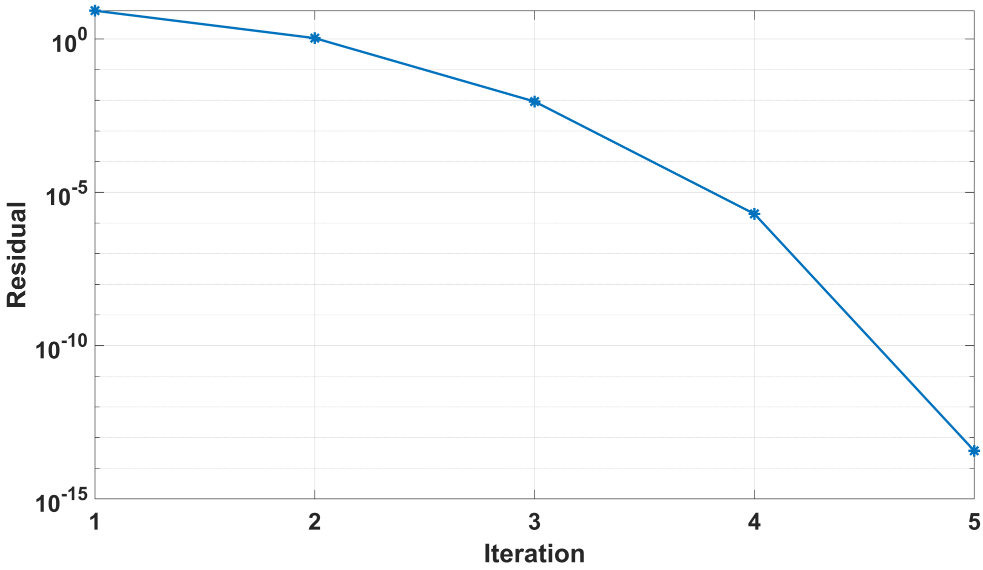

Figure 5.20: Convergence plot of Algorithm 3.

The convergence plot of the parallel NR method is depicted in Fig. 5.20. If the initial messages are chosen sufficiently close to the solution, the quadratic convergence can be observed in NR iterations until the convergence criterion is achieved. To ensure the initial guess is close enough to the NR solution, we had to run the algorithm in a Gauss-Seidel manner for a few iterations first.

The FGaBP algorithm was modified and applied to the coupled electrical-thermal and thermal-thermal problems that emerge in the mathematical modelling of RFA. By forming local Jacobian matrices instead of a global sparse Jacobian, our method provides a highly scalable strong coupling algorithm for solving the non-linear coupled equations on multiple cores (shared-memory) and multiple cluster nodes (distributed-memory). This is important in the context of fast and accurate simulation of RFA treatment for hepatic tumours, considering that the coupled problems are often solved in a decoupled manner because of their high computational cost. The correctness of the algorithms provided in this chapter are verified by comparing against COMSOL multiphysics software. The parallel implementation of the algorithms shows reasonable speedups on both shared-memory and distributed-memory architectures. The next chapter exploits the probabilistic properties of belief propagation in order to deal with the uncertainties in RFA modelling.

# Chapter 6

# Uncertainty analysis of RFA

Uncertainties are inevitable when simulating RFA treatment. These uncertainties largely result from various physiological parameters of the liver whose exact values are unknown. Even small uncertainties in measuring the physiological parameters can have a substantial impact on the outcome [4]. Taking the uncertainty of the simulation parameters into consideration, the dependence of the simulation outcome with respect to parameter variations must be evaluated. Such analysis computes how the uncertainty in the model parameters propagates to the simulation output, allowing the physician to estimate the expected range of the treatment outcome. In this chapter, we start off by a brief overview of SFEM and its basic concepts. Then, we will put together a novel sample-based solver for uncertainty analysis of RFA, using the SFEM concepts and a variant of the belief propagation method called non-parametric belief propagation.

## 6.1 The stochastic finite element method

Engineering problems that contain uncertainties are often described by stochastic differential equations. The stochastic finite element method (SFEM) is a numerical approach specifically developed for solving stochastic PDEs. It takes advantage of numerical techniques for deterministic problems in order to characterize stochastic properties of the solution. If one or more components of a PDE exhibit random behaviour, the response field will also be random. The SFEM studies the propagation of randomness from the

PDE components, such as material properties, geometry, and external forces, to the solution. The SFEM consists of three basic steps [79]: 1) discretization of the stochastic fields representing the random properties of the system, 2) construction of the local stochastic matrices and then assembly of the global stochastic matrix and finally, 3) calculation of response statistics. These three steps will be explained in this section.

### 6.1.1 Discretization of stochastic fields

The first step in the analysis of stochastic PDEs is the representation of the uncertain quantities of the system. These uncertain quantities are described with the definition of random fields. A random field is a random function $\tilde{f}(\mathbf{x})$ taking on a random value at each point $\mathbf{x}$ inside a multi-dimensional space $\Omega$. The stochastic characteristics of $\tilde{f}(\mathbf{x})$ are often obtained through experimental measurements. The continuous stochastic field $\tilde{f}(\mathbf{x})$ is then discretized, that is, replaced by a finite number of indexed random variables forming a random vector:

$$\tilde{f}(\mathbf{x}) \mapsto \{f_i\}, \tag{6.1}$$

where index $i$ depicts the position at which $\tilde{f}(\mathbf{x})$ is evaluated.

The discretization methods comprise two basic categories: point discretization methods and average-type discretization methods. In the point discretization, the set of random variables $\{f_i\}$, is simply calculated based on the values of the continuous stochastic field $\tilde{f}(\mathbf{x})$ at specific points inside the domain $\Omega$. Assuming the domain is already discretized using a finite element mesh, the evaluation points could be element centroid, element nodes, or integration points inside each element. In the average-type discretization method, a random variable is assigned to each finite element, whose value is found from the integral of $\tilde{f}(\mathbf{x})$ over the element. Both point discretization and average-type discretization methods have been widely utilized in the SFEM literature resulting in different levels of accuracy.

The stochastic mesh used for the discretization of the stochastic field doesn't have to be the finite element mesh. The stochastic mesh is controlled by the spatial variability of the random field, while the finite element mesh is determined by the geometry and the gradients of the response field. As these two meshes are selected based on different criteria, the use of different meshes may be more efficient in certain problems. For instance, if the stochastic field has strong spatial correlation, the problem size can be reduced by choosing a stochastic mesh that is coarser than the finite element [79].

### 6.1.2 Formulation of the stochastic linear system

In the context of SFEM, the stochastic behaviour of the solution can be due to the random material properties, problem geometry, and the forcing field. In this chapter, we assume the material properties—represented by a stochastic field $\tilde{f}(\mathbf{x})$—are the sources of randomness, considering the geometry and the force to be deterministic.

To follow the standard notation of FEM, stochastic quantities are often denoted by adding a tilde to their deterministic counterpart in the SFEM literature. Thus, the stochastic element matrix is indicated by $\widetilde{\mathbf{K}}_e$, and $\mathbf{K}_e$ stands for a realization of this matrix. The global properties are formed by the standard assembling procedure of FEM, that is, the individual element contributions are added to the global stiffness matrix and forcing vector:

$$\widetilde{\mathbf{K}} = \sum \widetilde{\mathbf{K}}_e, \qquad \widetilde{\mathbf{q}} = \sum \widetilde{\mathbf{q}}_e. \tag{6.2}$$

The global system of equations is then obtained as:

$$\widetilde{\mathbf{K}}\widetilde{\mathbf{u}} = \widetilde{\mathbf{q}} \ , \tag{6.3}$$

where $\widetilde{\mathbf{u}}$ is the vector containing the solution values at the nodes. The key task of SFEM is to estimate the stochastic properties of $\widetilde{\mathbf{u}}$ from those of the global stiffness matrix $\widetilde{\mathbf{K}}$. For this reason, the right-hand-side vector is often considered deterministic [80].

### 6.1.3 Calculation of response statistics

The Monte Carlo simulation (MCS) is the simplest method for calculating response variability in the context of SFEM. Assuming the right-hand-side vector is deterministic, $n$ samples of the stochastic stiffness matrix $\widetilde{\mathbf{K}}$ are generated and the linear system of (6.3) is solved $n$ times, producing $n$ samples of the solution vector $\widetilde{\mathbf{u}}$. Basic statistical properties of the solution vector are then calculated from these samples. For example, the mean and variance of the solution at node $i$ are:

$$E(u_i) = \frac{1}{n}\sum_{j=1}^{n} u_i(j), \tag{6.4}$$

$$\kappa^2(u_i) = \frac{1}{n}\sum_{j=1}^{n}\big[u_i(j) - E(u_i)\big]^2. \tag{6.5}$$

The accuracy of above estimations depends on the number of samples. In the case of large-scale systems, the solution of $n$ deterministic problems has a prohibitively high computational cost. Therefore, the MCS is often used as a robust reference for validating the results of other methods in the SFEM literature.

The MCS treats the stochastic system in a black-box approach, i.e., the random properties are never characterized directly. The SFEM on the other hand characterizes randomness analytically by deriving approximate series expansions of the stochastic stiffness matrix and of the solution. Two popular cases are Taylor and Karhunen–Loève series expansions, known as perturbation method and spectral method, respectively. Assume the stochastic field $\tilde{f}(\mathbf{x})$ representing the random behaviour of the material is discretised into $N$ random variables $\{f_i\}_{i=1}^{N}$. In the perturbation method, Taylor series expansion of the stochastic stiffness matrix is computed as [79]:

$$\mathbf{K} = \mathbf{K}_0 + \sum_{i=1}^{N}\mathbf{K}_\mathbf{i}^\mathbf{I} f_i + \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\mathbf{K}_\mathbf{ij}^\mathbf{II} f_i f_j + ..., \tag{6.6}$$

where

$$\mathbf{K}_\mathbf{i}^\mathbf{I} = \left.\frac{\partial \mathbf{K}}{\partial f_i}\right|_{f_i=0} \tag{6.7}$$

and

$$\mathbf{K}_\mathbf{ij}^\mathbf{II} = \left.\frac{\partial^2 \mathbf{K}}{\partial f_i \partial f_j}\right|_{f_i,f_j=0}. \tag{6.8}$$

Taylor series expansion of the right-had-side and solution vectors are also required:

$$\mathbf{q} = \mathbf{q}_0 + \sum_{i=1}^{N}\mathbf{q}_\mathbf{i}^\mathbf{I} f_i + \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\mathbf{q}_\mathbf{ij}^\mathbf{II} f_i f_j + \dots, \tag{6.9}$$

$$\mathbf{u} = \mathbf{u}_0 + \sum_{i=1}^{N}\mathbf{u}_\mathbf{i}^\mathbf{I} f_i + \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\mathbf{u}_\mathbf{ij}^\mathbf{II} f_i f_j + \dots. \tag{6.10}$$

The solution vector is then calculated in an iterative approach as:

$$\begin{aligned}
\mathbf{u_0} &= \mathbf{K_0^{-1}q_0},\\
\mathbf{u_i^I} &= \mathbf{K_0^{-1}}(\mathbf{q_i^I} - \mathbf{K_i^I u_0}),\\
\mathbf{u_{ij}^{II}} &= \mathbf{K_0^{-1}}(\mathbf{q_{ij}^{II}} - \mathbf{K_i^I u_j^I} - \mathbf{K_j^I u_i^I} - \mathbf{K_{ij}^{II} u_0}).
\end{aligned} \tag{6.11}$$

If the right-hand-side vector is deterministic, $\mathbf{q} = \mathbf{q}_0$ and $\mathbf{q_i^I} = \mathbf{q_{ij}^I} = 0$. The main drawback of the perturbation method is its need for calculation of the partial derivatives which increases the computational cost of the approach in large-scale problems.

The SSFEM was introduced by Ghanem and Spanos [48] for solving PDEs with uncertain material properties. If $\bar{f}(\mathbf{x})$ and $C(\mathbf{x}_1, \mathbf{x}_2)$ stand for the mean function and covariance kernel of the stochastic field $\tilde{f}(\mathbf{x})$, the Karhunen-Loeve (K-L) expansion of $\tilde{f}(\mathbf{x})$ is:

$$\tilde{f}(\mathbf{x}) = \bar{f}(\mathbf{x}) + \sum_{i=1}^{\infty} \sqrt{\lambda_i}\phi_i(\mathbf{x})\tilde{\zeta}_i, \tag{6.12}$$

where $\{\tilde{\zeta}_i\}$ is an infinite set of random variables representing the randomness of $\tilde{f}(\mathbf{x})$. The deterministic parameters $\lambda_i$ and $\phi_i$ are eigenvalues and eigenfunctions of $C(\mathbf{x}_1, \mathbf{x}_2)$, obtained from the integral equation:

$$\int_{\Omega} C(\mathbf{x}_1, \mathbf{x}_2)\phi_i(\mathbf{x}_1)d\mathbf{x}_1 = \lambda_i\phi_i(\mathbf{x}_2), \tag{6.13}$$

in which $\Omega$ is the domain of interest and the normalized orthogonal eigenfunctions satisfy:

$$\int_{\Omega} \phi_i(x)\phi_j(x)dx = \begin{cases} 1 & i = j \\ 0 & i \neq j. \end{cases} \tag{6.14}$$

In practice, a finite number of terms with the largest eigenvalues are retrained in the infinite K-L series of (6.12). It can be proved [48] that the terms of the random series $\{\tilde{\zeta}_i\}$ are obtained from the following integration:

$$\tilde{\zeta}_i = \frac{1}{\lambda_i}\int_{\Omega} \tilde{\alpha}(\mathbf{x})\phi_i(\mathbf{x})d\mathbf{x}, \tag{6.15}$$

where $\tilde{\alpha}(\mathbf{x}) \triangleq \tilde{f}(\mathbf{x}) - \bar{f}(\mathbf{x})$ is a zero-mean random field.

The stochastic field $\tilde{f}(\mathbf{x})$ is discretized as explained in Section 6.1.1. This turns the function $\bar{f}(\mathbf{x})$ and kernel $C(\mathbf{x}_1, \mathbf{x}_2)$ into a mean vector and covariance matrix, respectively. The K-L expansion of $\tilde{f}(\mathbf{x})$ is then evaluated at each discretization point $\mathbf{x}_d$ of the domain:

$$\tilde{f}(\mathbf{x}_d) = \bar{f}(\mathbf{x}_d) + \sum_{i=1}^{\infty} \sqrt{\lambda_i}\phi_i(\mathbf{x}_d)\tilde{\zeta}_i. \tag{6.16}$$

Substituting (6.16) in the element matrices and assembling the global stiffness matrix leads to series expansion of the stiffness matrix. Assuming $M$ terms in the K-L expansion are retained, the stochastic stiffness matrix will have the following form:

$$\tilde{\mathbf{K}}_{N\times N} = \bar{\mathbf{K}} + \sum_{i=1}^{M} \mathbf{K}_i \, \tilde{\zeta}_i. \tag{6.17}$$

Here, $N$ is the number of degrees of freedom, $\bar{\mathbf{K}}$ is the mean matrix of $\tilde{\mathbf{K}}$, and $\mathbf{K}_i$ are deterministic matrices. The random solution vector $\tilde{\mathbf{u}}_{N\times 1}$ is represented using polynomial chaos (PC) [81] expansion:

$$\tilde{\mathbf{u}} = \sum_{j=1}^{\infty} \mathbf{u}_j \tilde{\Psi}_j, \tag{6.18}$$

where $\mathbf{u}_j$ are the unknown vector coefficients to be found, and $\tilde{\Psi}_j$ are known random polynomials. Truncating the PC series at the $P$th term, and assuming the source vector is deterministic, the linear system becomes:

$$\big(\bar{\mathbf{K}} + \sum_{i=1}^{M} \mathbf{K}_i \tilde{\zeta}_i\big)\big(\sum_{j=1}^{P} \mathbf{u}_j \tilde{\Psi}_j\big) = \mathbf{q}. \tag{6.19}$$

The optimal set of coefficients $\{\mathbf{u}_j\}_{j=1}^{P}$ is found by minimizing the residual $\mathcal{R}_{M,P}$ defined as:

$$\mathcal{R}_{M,P} \triangleq \big(\bar{\mathbf{K}} + \sum_{i=1}^{M} \mathbf{K}_i \tilde{\zeta}_i\big)\big(\sum_{j=1}^{P} \mathbf{u}_j \tilde{\Psi}_j\big) - \mathbf{q}, \tag{6.20}$$

with respect to $\{\mathbf{u}_j\}_{j=1}^{P}$. This finally leads to solving an $NP \times NP$ linear system of equations [48]:

$$\begin{bmatrix} K_{11} & K_{12} & \cdots & K_{1,NP} \\ K_{21} & K_{22} & \cdots & K_{2,NP} \\ \vdots & \vdots & \ddots & \vdots \\ K_{NP,1} & K_{NP,2} & \cdots & K_{NP,NP} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{NP} \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_{NP} \end{bmatrix}. \tag{6.21}$$

The advantage of SSFEM over the perturbation method is that it doesn't require calculation of partial derivatives. However, as the number of dofs $N$ is multiplied by $P$ in the dimension of the linear system, the computational cost required for the solution of (6.21) is much larger than that of its corresponding deterministic system. In the next section, we devise a novel technique based on belief propagation algorithm for the solution of stochastic PDEs.

## 6.2 Non-parametric belief propagation

The BP algorithm was introduced in Chapter 3. Given the joint distribution $p(x_1, x_2, ..., x_n)$ of $n$ random variables $\{x_1, x_2, ..., x_n\}$, BP aims to find their marginal distributions $p(x_i)$

for $1 \leq i \leq n$. The probabilistic dependence between the random variables can be depicted by a graph. If these dependencies are symmetrical, the graph is defined by a set of nodes $\mathcal{V}$ and a corresponding set of undirected edges $\mathcal{E}$. At each iteration $t$ of the BP algorithm, nodes $i \in \mathcal{V}$ send messages $m_{ij}^t(x_j)$ to the neighbouring nodes $j \in \mathcal{N}(i)$:

$$m_{ij}^t(x_j) \propto \int_{x_i} \psi_{ij}(x_i, x_j)\phi_i(x_i) \prod_{k\in\mathcal{N}(i)\backslash j} m_{ki}^{t-1}(x_i)\; dx_i, \tag{6.22}$$

where the function $\psi_{ij}(x_i, x_j) > 0$ is called the edge potential and indicates the probabilistic dependence between nodes $i$ and $j$. The function $\phi_i(x_i)$ is the self potential of node $i$, containing information about the marginal distribution $p(x_i)$. At message convergence the marginal probability at node $j$ is computed as:

$$p(x_j) \propto \phi_j(x_j) \prod_{i\in\mathcal{N}(j)} m_{ij}(x_j). \tag{6.23}$$

Most BP applications assume each random variable $x_i$ takes one of $k$ possible discrete values, meaning that messages and marginal probabilities can be represented by $k$-dimensional vectors. As a result, the message update integral of (6.22) becomes a matrix–vector multiplication. For graphical models with continuous variables on the other hand, closed-form evaluation of the integral in (6.22) is only feasible when the variables are jointly Gaussian. The resulting algorithm is called GaBP, in which the messages $m_{ij}(x_i)$ can be described by their mean and variance. These parameters are iteratively updated via (6.22). With the exception of multivariate Gaussian problems, finding the marginals is challenging for continuous random variables; the messages are continuous functions and so, are expensive to compute and transmit, in which case the BP approach is not computationally feasible.

Various techniques have been proposed in the literature to moderate the complexity of continuous BP in different applications [82, 83, 84, 85]. These techniques often approximate the messages as mixtures of normal distributions [85] or as weighted particles [83, 84]. A low-complexity alternative to continuous BP based on expanding the messages in stochastic orthogonal series was proposed by [86]. All these techniques are categorized as non-parametric belief propagation (NBP), a variant of BP for graphical models containing continuous non-Gaussian random variables. The NBP represents the message $m_{ij}(x_j)$ using kernel–based density estimation. Assume $\mathcal{N}(x; \mu, \kappa)$ indicates a Gaussian density of mean $\mu$ and variance $\kappa$, we may then approximate $m_{ij}(x_j)$ by a mixture of $L$ Gaussian kernels as:

$$m_{ij}(x_j) = \sum_{l=1}^{L} w_l \, \mathcal{N}(x_j; \mu_l, \kappa_l), \tag{6.24}$$

where $w_l$ is the $l$th kernel's weight. Although other options for kernel functions are also suggested, in order to exploit the simplicity of the GaBP algorithm, we only consider Gaussian kernels in this chapter. The NBP algorithm uses sampling methods to estimate the parameters of (6.24).

One can break down the message update rule in (6.22) into two stages. First, the product of all incoming messages, $\prod_k m_{ki}(x_i)$, is computed. Second, the product is combined with the self-potential $\phi_i$ and edge potential $\psi_{ij}$ and integrated to produce the outgoing message. The NBP algorithm approximates these two stages using a sampling-based approach. To approximate the incoming message product, the NBP notes that the multiplication of $d$ Gaussian distributions of a random variable $x_i$ is itself Gaussian:

$$\prod_{k=1}^{d} \mathcal{N}(x_i; \mu_k, \kappa_k) \propto \mathcal{N}(x_i; \mu, \kappa), \tag{6.25}$$

where:

$$\kappa^{-1} = \sum_{k=1}^{d} \kappa_k^{-1}, \qquad \kappa^{-1}\mu = \sum_{k=1}^{d} \kappa_k^{-1}\mu_k. \tag{6.26}$$

Therefore, one can conclude that if the incoming messages $m_{ki}^{t-1}(x_i)$ in (6.22) are Gaussian mixtures each containing $L$ components, the output of the message product will be a Gaussian mixture with $L^d$ components.

Although the message updates could be computed using (6.22) to (6.26), the number of mixture components grows exponentially with each iteration. In other words, if we start with $L$–component messages in the first iteration, the second iteration messages will have $L^d$ components and the messages in the third iteration will have $(L^d)^d$ components and so forth. As $d$, the number of nodes in the neighbourhood of each node, is usually greater then one, the exponential growth of the number of components makes direct application of (6.25) and (6.26) intractable. Thus, the NBP algorithm approximates the $L^d$–component outgoing messages by first drawing independent samples from them, and then fitting an $L$-component Gaussian mixture to the samples. This technique is called mixture reduction and is an extensive research topic in non-parametric statistics [87]. Mixture reduction can be as simple as eliminating Gaussian components with lower probabilities (smaller $w_l$ in (6.24)), while more advanced techniques such as merging components according to certain similarity criteria [88] also exist.

The theoretical background of NBP and its convergence criteria is explored in statistics literature [86] and is not within the scope of this chapter. In the remainder of this chapter, we exploit the main ideas of NBP and combine them with those of SFEM in order to design a new algorithm for solving stochastic PDEs. The Laplace's equation with random conductivity plays a key role in the uncertainty analysis of RFA, and so is chosen as an illustrative example of solving stochastic PDEs with NBP in the next section.

## 6.3 Solving stochastic Laplace's equation with NBP

In this section we combine the ideas from the SFEM and NBP in order to develop a novel method for solving stochastic PDEs. We use SFEM to construct a stochastic linear system from a random field. Then, instead of following conventional SFEM approaches, we will give our attention to NBP for solving the stochastic linear system. The electrical conductivity of the liver is considered to be a major source for uncertainty in RFA modelling. For this reason, the Laplace's equation with random conductivity is considered in this section.

### 6.3.1 Assembly of the stochastic linear system

RFA is a type of hyperthermia treatment that uses high temperatures to *cook* the cancerous cells. The source of this heat is the pass of electric current through the conductive medium of the liver, a phenomenon known as Joule heating. As such, Laplace's equation is used to model the electric potential and subsequently the heat source in RFA simulations:

$$\nabla \cdot (\sigma(x)\nabla v) = 0, \qquad x \in \Omega, \tag{6.27}$$

where $\Omega$ is the spatial domain of interest, $\sigma$ (S/m) is the electrical conductivity of the liver, and $v$ (V) is the electric potential. As explained in Section 2.1, the electrical conductivity of the liver depends on the temperature. Due to the spatial variations of the temperature,

$\sigma(x)$ depends on space as well. Liver conductivity also depends on the physical state of the tissue, and varies from day to day depending on the patient's physical condition. The range of values outlined in the literature marks out this uncertainty. In the ambient body temperature (37 °C), the conductivity of liver tissue is reported as [89]:

$$\sigma = 0.17 - 0.6 \quad \text{S/m}. \tag{6.28}$$

The uncertainties associated with $\sigma$ mean it can be modelled as a random field. The random conductivity induces uncertainty in the electric potential. Following the notation introduced in Section 6.1, the stochastic Laplace's equation inside the domain $\Omega$ with boundary $\Gamma$ becomes:

$$\begin{aligned} \nabla \cdot (\tilde{\sigma}(x)\nabla\tilde{v}(x)) = 0, \qquad & x \in \Omega, \\ \tilde{v}(x) = v_0, \qquad & x \in \Gamma, \end{aligned} \tag{6.29}$$

where $\tilde{\sigma}(x)$ and thus $\tilde{v}(x)$ are random fields. The term *field* is used in statistics for a function that takes on random values at each point inside a domain. For simplicity, we assume $\tilde{\sigma}(x)$ is a Gaussian random field, i.e., any spatial discretization of it produces a Gaussian random vector $\big(\tilde{\sigma}(x_1), \ldots, \tilde{\sigma}(x_n)\big)$, where $\{x_1, \ldots, x_n\}$ is the set of discretization points. Assume the FEM mesh divides the computational domain $\Omega$ into $N_e$ elements. If the midpoint method is used for random field discretization, $\tilde{\sigma}(x)$ is approximated inside each finite element $\Omega_e$ by its value at the centroid of the element:

$$\tilde{\sigma}(\boldsymbol{x}) \approx \tilde{\sigma}(\boldsymbol{x}_c), \qquad\qquad \boldsymbol{x} \in \Omega_e, \tag{6.30}$$

where $\boldsymbol{x}_c$ is the center of element $\Omega_e$. The approximated field is then completely defined by the random vector $\boldsymbol{\mathcal{X}} = (\tilde{\sigma}(\boldsymbol{x}_c^1), \ldots \tilde{\sigma}(\boldsymbol{x}_c^{N_e}))$, whose mean vector and covariance matrix are obtained from the mean function and covariance kernel of the Gaussian field $\tilde{\sigma}(x)$. Note that the Gaussian random field $\tilde{\sigma}(x)$ is entirely defined by its mean function, $\mu(x)$, and covariance kernel, $C(x, x')$.

The stochastic element matrix $\tilde{\mathrm{K}}_e$ associated with $\Omega_e$ is then approximated as:

$$\tilde{\mathrm{K}}_e(i,j) = \int_{\Omega_e} \tilde{\sigma}(\boldsymbol{x})\vec{\nabla}N_i \cdot \vec{\nabla}N_j \; dV \approx \tilde{\sigma}(\boldsymbol{x}_c)\int_{\Omega_e} \vec{\nabla}N_i \cdot \vec{\nabla}N_j \; dV, \tag{6.31}$$

where $N_i$ and $N_j$ are FEM scalar basis functions. Finally, the global stochastic linear system is assembled:

$$\tilde{\mathrm{K}}\tilde{\mathrm{v}} = \mathrm{b}, \tag{6.32}$$

where the right-hand-side vector $b$ is deterministic due to fixed boundary conditions.

### 6.3.2 NBP algorithm for solving $\widetilde{\mathrm{K}}\tilde{\mathrm{v}} = \mathrm{b}$

In Chapter 3, solving a deterministic linear system $\mathbf{Kv} = \mathbf{b}$ using Gaussian belief propagation (GaBP) was explained. It was shown that assuming the solution vector $\mathbf{v}$ is a jointly Gaussian random vector, solving the linear system is equivalent to finding the marginal mean of $\mathbf{v}$. GaBP defines the edge potentials and self potentials respectively as:

$$\psi_{ij}(v_i, v_j) = exp(-\frac{1}{2}v_i K_{ij} v_j), \tag{6.33a}$$

$$\phi_i(v_i) = exp(-\frac{1}{2}K_{ii}v_i^2 + b_i v_i). \tag{6.33b}$$

It was shown that substituting these potentials in (6.22), the message $m_{ij}^t(v_j)$ will have the form of a Gaussian distribution, whose mean and precision are determined by (3.11). If the system matrix $\mathbf{K}$ is stochastic and represented by $\widetilde{\mathbf{K}}$, the potentials in (6.33) and the message update rules are valid for each realization of $\widetilde{\mathbf{K}}$. In other words, the self and edge potentials are also random variables that are computed as:

$$\widetilde{\psi}_{ij}(v_i, v_j) = exp(-\frac{1}{2}v_i \widetilde{K}_{ij} v_j), \tag{6.34a}$$

$$\widetilde{\phi}_i(v_i) = exp(-\frac{1}{2}\widetilde{K}_{ii}v_i^2 + b_i v_i). \tag{6.34b}$$

Hence, the deterministic update rules of (3.11) and (3.12) will take a stochastic form as well:

$$\widetilde{P}_{ij} = -\widetilde{K}_{ij}^2\, \widetilde{P}_{i\backslash j}^{-1}, \tag{6.35a}$$

$$\widetilde{\mu}_{ij} = -\widetilde{P}_{ij}^{-1}\widetilde{K}_{ij}\, \widetilde{\mu}_{i\backslash j}, \tag{6.35b}$$

where:

$$\widetilde{P}_{i\backslash j} = \widetilde{K}_{ii} + \sum_{k \in N(i)\backslash j} \widetilde{P}_{ki}, \tag{6.36a}$$

$$\widetilde{\mu}_{i\backslash j} = \widetilde{P}_{i\backslash j}^{-1}\big(b_i + \sum_{k \in N(i)\backslash j} \widetilde{P}_{ki}\, \widetilde{\mu}_{ki}\big). \tag{6.36b}$$

Even if the PDF of $\widetilde{K}_{ii}$ and $\widetilde{K}_{ij}$ are known, computing those of $\widetilde{P}_{ij}$ and $\widetilde{\mu}_{ij}$ is not analytically tractable. An alternative approach is to represent the messages non-parametrically using kernel–based density estimation [85]. For notation simplicity, the random mean and precision messages sent from node $i$ to node $j$ will be shown by a single random variable $\widetilde{m}_{ij}$, where $m \in \{\mu, P\}$. Following the NBP method introduced in Section 6.2, $\widetilde{m}_{ij}$ may be approximated by a mixtures of $L$ Gaussian kernels:

$$\widetilde{m}_{ij}(x_j) = \sum_{l=1}^{L} w_l \, \mathcal{N}(x_j; \mu_l, \kappa_l). \tag{6.37}$$

Here, $w_l$, $\mu_l$, and $\kappa_l$ are the weight, mean, and precision associated with component $l$ in the mixture. These parameters can be approximated by drawing independent samples from $\widetilde{m}_{ij}$.

Sampling from $\widetilde{m}_{ij}$ is straightforward. In equation (6.36), the two summations $\widetilde{S}_1 = \sum_{k \in N(i)\backslash j} \widetilde{P}_{ki}$ and $\widetilde{S}_2 = \sum_{k \in N(i)\backslash j} \widetilde{P}_{ki} \, \widetilde{\mu}_{ki}$ contain the incoming messages $\widetilde{P}_{ki}$ and $\widetilde{\mu}_{ki}$. If the incoming messages have the form of Gaussian mixtures like (6.37), then $\widetilde{S}_1$ and $\widetilde{S}_2$ are also Gaussian mixtures models (GMM). In particular, $\widetilde{S}_1$ is a GMM containing $Ld$ components, where $d$ is the number of nodes in the neighbourhood of node $i$ excluding node $j$. The second summation $S_2$ contains the multiplication of two GMMs, $\widetilde{P}_{ki}$ and $\widetilde{\mu}_{ki}$, each containing $L$ components. Since the multiplication of two Gaussian distributions is itself a Gaussian distribution, $\widetilde{S}_2$ is a GMM with $L^2 \, d$ components.

Assuming the PDF of the stochastic matrix $\widetilde{\mathbf{K}}$ is known, $\widetilde{K}_{ii}$ and $\widetilde{S}_1$ are sampled to generate independent samples for $\widetilde{P}_{i\backslash j}$. Independent samples from $\widetilde{\mu}_{i\backslash j}$ are drawn in a similar way. In the next step, the sampled data from $\widetilde{P}_{i\backslash j}$ and $\widetilde{\mu}_{i\backslash j}$ are made use of to sample from $\widetilde{P}_{ij}$ and $\widetilde{\mu}_{ij}$ according to (6.35). Finally, by fitting two GMMs to the driven data for $\widetilde{P}_{ij}$ and $\widetilde{\mu}_{ij}$, the outgoing messages are expressed in the mixture form of (6.37). These steps can be summarized in Algorithm 4.

### 6.3.3 Implementation and results

Laplace's equation with random conductivity is solved in two dimensions:

$$\nabla \cdot (\tilde{\sigma}(x) \nabla \tilde{v}(x)) = 0, \qquad x \in \Omega, \tag{6.38}$$

where the computational domain $\Omega$ is shown in Fig. 6.1. Dirichlet boundary conditions are applied at the conducting tip (16 V) and the outer boundary (ground). A triangular FEM mesh of first order is used to discretize the domain (Fig. 6.2).

The conductivity $\widetilde{\sigma}(\mathbf{x})$ is a Gaussian random field with pre-defined mean function $\bar{\sigma}(\mathbf{x}) = 0.4$ S/m and covariance kernel $C(\mathbf{x}_1, \mathbf{x}_2) = \kappa \, e^{-(\mathbf{x}_1 - \mathbf{x}_2)/d}$, where $C(\mathbf{x}_1, \mathbf{x}_2)$ is an

**Algorithm 4:** The NBP algorithm for solving $\tilde{\mathrm{K}}\tilde{\mathrm{v}} = \mathrm{b}$

```
1 Receive input messages $\widetilde{P}_{ki}$ and $\widetilde{\mu}_{ki}$ for $k \in \mathcal{N}(i) \setminus j$ in the form of GMMs and
   construct output messages in the form of GMMs following 2 to 8;
2 Construct a new GMM from the summation over the input messages
  $\widetilde{S}_1 = \sum_{k \in \mathcal{N}(i) \setminus j} \widetilde{P}_{ki}$ ;
3 Draw $M$ independent samples from $\widetilde{P}_{i\setminus j}$ by sampling $\widetilde{S}_1$ and $\widetilde{K}_{ii}$;
4 Construct a new GMM from the summation over the input messages product
  $\widetilde{S}_2 = \sum_{k \in \mathcal{N}(i)} \widetilde{P}_{ki}\widetilde{\mu}_{ki}$ ;
5 Draw $M$ independent samples from $\widetilde{\mu}_{i\setminus j}$ by sampling $\widetilde{S}_2$ and using the sampled
   data from $\widetilde{P}_{i\setminus j}$ in step 3;
6 Draw $M$ independent samples from $\widetilde{P}_{ij}$ by sampling $\widetilde{K}_{ij}$ and $\widetilde{P}_{i\setminus j}$ samples;
7 Draw $M$ independent samples from $\widetilde{\mu}_{ij}$, using $\widetilde{K}_{ij}$, $\widetilde{\mu}_{i\setminus j}$, and $\widetilde{P}_{ij}$ samples;
8 Using the samples drawn in steps 4 and 5, express $\widetilde{P}_{ij}$ and $\widetilde{\mu}_{ij}$ in terms of two
   GMMs;
```

exponential covariance with $\kappa$ being the variance at each point and $d$ being the correlation length (m) between two points $\mathbf{x}_1$ and $\mathbf{x}_2$. The correlation length $d$ is a measure of the statistical correlation between two points of the field. The midpoint method is used for random field discretization and thus $\tilde{\sigma}(x)$ is approximated inside each finite element $\Omega_e$ by its value at the centroid of the element:

$$\tilde{\sigma}(\boldsymbol{x}) \approx \tilde{\sigma}(\boldsymbol{x}_c), \qquad \boldsymbol{x} \in \Omega_e. \tag{6.39}$$

The random field is then represented by the random vector $\mathcal{X} = (\tilde{\sigma}(\boldsymbol{x}_c^1), \ldots \tilde{\kappa}(\boldsymbol{x}_c^{N_e}))$, where $N_e$ is the total number of elements. Next, the stochastic element matrices $\tilde{\mathrm{K}}_e$ and the stochastic linear system $\tilde{\mathbf{K}}\tilde{\mathbf{v}} = \mathbf{b}$ are constructed according to (6.31) and (6.32), respectively.

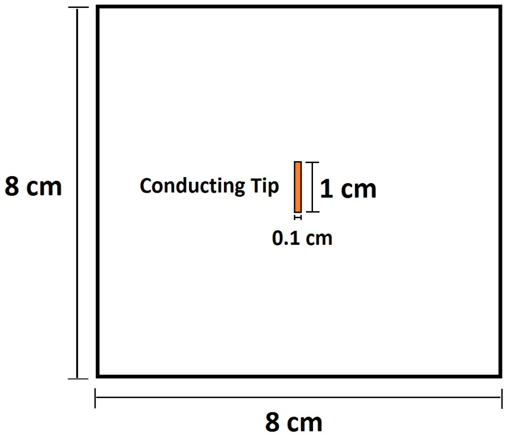

Figure 6.1: The geometry of the stochastic Laplace's problem.

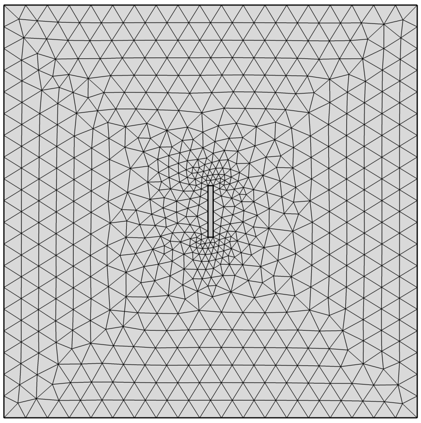

Figure 6.2: Sample triangular FEM mesh used to discretize the random field.

The stochastic system matrix $\tilde{\mathbf{K}}$ has to be sampled, as depicted by Algorithm 4, steps 3 and 6. On the other hand, the FEM connectivity matrix dictates a linear relation between $\tilde{\mathbf{K}}$ and the random vector $\mathcal{X}$. Such linear relation means the non-zero entries of $\tilde{\mathbf{K}}$ have jointly Gaussian distribution. This is because if $\mathcal{X} \sim \mathcal{N}(\mu, \mathbf{\Sigma})$ is a multivariate Gaussian distribution and $\mathcal{Y} = \mathbf{B}\mathcal{X}$, then $\mathcal{Y}$ is also a multivariate Gaussian distribution:

$$\mathcal{Y} \sim \mathcal{N}(\mathbf{B}\,\mu, \mathbf{B}\mathbf{\Sigma}\mathbf{B}^T). \tag{6.40}$$

Consequently, assuming the non-zero entries of $\tilde{\mathbf{K}}$ are contained in the random vector $\mathcal{Y}$, samples from $\tilde{\mathbf{K}}$ are generated from the distribution of (6.40). Nevertheless, one can also generate samples from $\tilde{\sigma}(\mathbf{x})$ and then construct large number of $\tilde{\mathbf{K}}$ samples prior to Algorithm 4. This approach is similar to the MCS in that a large number of global

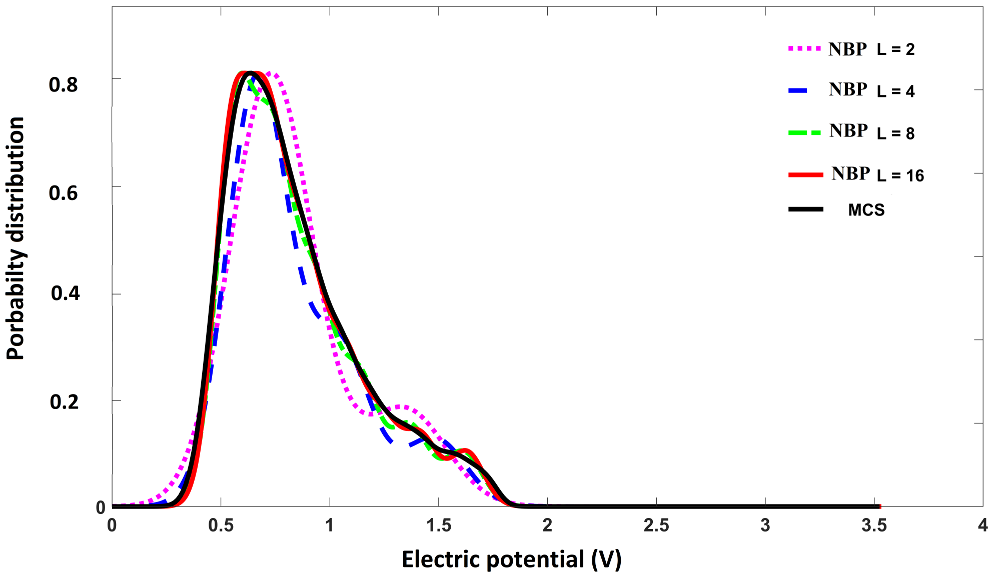

Figure 6.3: The PDF obtained from the MCS and NBP methods

random matrices $\widetilde{\mathbf{K}}$ are constructed. However, unlike the MCS, these matrices are not inverted to generate the solution vectors directly, and rather are used in Algorithm 4 to sample the NBP messages.

In order to investigate the NBP as a probabilistic solver for the stochastic linear system $\widetilde{\mathbf{K}}\widetilde{\mathbf{v}} = \mathbf{b}$, Algorithm 4 is implemented in the same iterative manner as of belief propagation. This means the non-parametric messages are sent along the edges of the graph associated with the sparse matrix $\widetilde{\mathbf{K}}$ (See Section 6.2). The iterations continue until the mean of the messages converge. Note that since the messages are represented as Gaussian mixture series, the mean of each message is the weighted sum of the means of its mixture components.

Figure 6.3 depicts the PDF of a specific element of the solution vector, $\tilde{v}_i$, obtained from the MCS and the NBP with different number of mixture components. The number of unknowns (number of degrees of freedom in the FEM mesh) is 2500. The Number of samples for both MCS and NBP is 10,000. According to the figure, by increasing the number of mixture components $L$, the NBP algorithm shows better agreement with MCS by better capturing of the random properties of the solution vector. Figure 6.4 shows the variance of the solution vector at a specific point versus that of the random field $\tilde{\sigma}$, i.e., $\kappa$ in the exponential covariance kernel $C(\mathbf{x}_1, \mathbf{x}_2) = \kappa\, e^{-(\mathbf{x}_1 - \mathbf{x}_2)/d}$. The correlation length $d$ is 0.5 cm for all simulations.

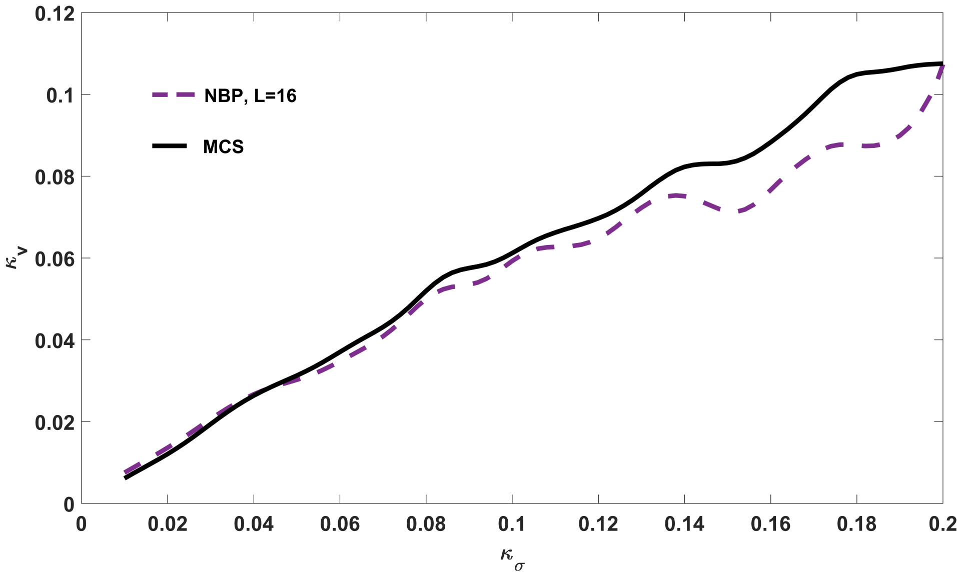

Figure 6.4: Variance of the solution vector versus random conductivity variance.

The idea of NBP is to involve the random field samples to locally approximate the BP messages. By doing so, the statistical information recorded in the random field samples at each discretization point of the domain are propagated to the whole system. Comparison to the MCS is often performed in the literature to validate new stochastic approaches. In this regard, the results of figures 6.3 and 6.4 prove the capability of the algorithm to model the uncertainty propagation from the input random field (the conductivity in this example) to the system response (the electric potential).

The computational cost of Algorithm 4 increases with the number of Gaussian components $L$ and the number of samples $M$. Efficient implementation of NBP on parallel computers can reduce the simulation time. For instance, similar to the finite element Gaussian belief propagation algorithm which was introduced for deterministic systems in Chapter 3, Algorithm 4 can be implemented on a factor graph to gain more parallelism. Also, a multigrid version of the algorithm can be devised for accelerating its convergence. Nevertheless, this chapter aimed to build a proving ground for using the NBP as a sample-based solver for stochastic linear systems and more research is needed to improve the computational efficiency of this novel technique.

# Chapter 7

# Summary and future work

Computer simulations of RFA have gained attention over the last decade and many research groups have already developed software for RFA simulation over the last few years. The main challenge involved in computer modelling of RFA is its high computational cost. Therefore, how to speed up the simulation time is an active area of research in computer-aided RFA. Throughout this thesis study, we aimed to outline the main obstacles associated with fast RFA simulations and proposed solutions to bypass these obstacles.

Being the principal technique in RFA simulations, the FEM and its challenges for parallel implementation of multi-physics problems were described. Then, the finite element Gaussian belief propagation (FGaBP) algorithm as a parallel FEM-based solver was introduced and revised for solving the coupled electrical-thermal and thermal-thermal problems that emerge in RFA models. The local computations and message passing strategies of FGaBP were made use of to enhance the computational performance of the two general approaches for solving multi-physics systems, i.e., weak coupling and strong coupling methods.

The weak coupling approach solves each equation for its own unknown, assuming constant values for the other unknowns. This is usually conducted in either Gauss-Seidel or Jacobi iterations. Gauss-Seidel solves each uni-physics problem for its own unknown, taking the most recent values for the other unknowns. This means the individual uni-physics problems need to be solved sequentially. On the other hand, the Jacobi algorithm allows all the equations in the same iteration to be solved simultaneously, where each equation uses the values for the other unknowns from the last iteration. Avoiding processor idle time, Jacobi iterations provide more parallelism, although at the cost of a slower convergence rate.

In this thesis, we combined the Gauss-Seidel and Jacobi approaches to devise a parallel weak coupling algorithm. Unlike the conventional Gauss-Seidel where the uniphysics problems are solved sequentially, the proposed algorithm solves the equations concurrently. A local convergence was defined and used to pass the updated values of each component to the other components, using the message passing scheme of FGaBP. This approach provides more parallelism than the classical Gauss-Seidel and is particularly advantageous on large number of computer processors. According to the results this method is more efficient than using FGaBP in a classical Gauss-Seidel since it makes FGaBP solver converge faster. In addition, in contrast to the Jacobi algorithm where the component values are taken from the previous iteration, the new algorithm uses updated values from the current iteration.

The strongest form of coupling algorithms, which is the NR method, is also implemented in parallel using the localized computations of FGaBP. In the NR method, the construction of a Jacobian matrix at each linearizing iteration could be prohibitively expensive for large scale problems. The JFNK method is typically used as an inexact variant of the NR, whose advantage over the traditional NR method is that it does not require to compute and store the Jacobian matrix. On the other hand, parallel acceleration of JFNK is restricted due to its dependence on the execution of global algebraic operations in each JFNK iteration, such as the sparse matrix-vector multiplication. 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.

In this thesis, a NR reformulation of FGaBP was presented to exploit its localized computations and message passing scheme for solving multi-physics problems in parallel. Similar to the JFNK, FGaBP 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 amenable 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. The algorithm was executed on both shared-memory and distributed memory architectures. According to the results, the parallel scalability of the FGaBP is retained in the new multi-physics algorithm, while the quadratic convergence of NR is also preserved.

The last chapter exploits the probabilistic message passing attribute of belief propagation for solving stochastic linear systems, establishing an explicit connection between the two fields for the first time. The non-parametric belief propagation is combined with SFEM in order to develop a novel method that makes use of the appealing sample-based approach in MCS and yet is computationally efficient. The algorithm is also different from SFEM in that it doesn't need analytical description of the randomness. Instead, we use samples from the stochastic parameters of the system to locally approximate the uncertainty propagation to the solution. The algorithm is applied to stochastic Laplace's equation and validated by comparison to the Mont Carlo method.

Future work can be pursued in two distinct directions. First, the parallel weak-coupling and strong coupling algorithms provided in this thesis are tested on simplified models of RFA. Regarding the model geometry, state of the art medical imaging techniques can produce three-dimensional anatomical models of the liver and its vascular system. As such, treatment planning based on individual patient images is the current trend in computer-aided RFA. Implementing the algorithms on complex liver models obtained from medical images can better reflect their capabilities for fast simulation of RFA. In addition, using realistic medical images, simulation outcomes can be validated by comparison to clinical data.

Besides the model geometry, simplifications are also assumed in the multi-physics couplings. While a fully coupled RFA model includes the interactions between electrical, thermal, and cellular necrosis processes simultaneously, these couplings are studied separately in this thesis. For instance, the electrical-thermal coupling is modelled while the thermal-thermal and thermal-cellular necrosis couplings are ignored. While such a divide and conquer method provides the building blocks for a fully coupled model, incorporating the individual couplings is essential for the development of a comprehensive simulation environment.

A second research direction could be to investigate further the non-parametric belief propagation algorithm introduced in the last chapter. Chapter 6 explores the potential of belief propagation as a probabilistic solver for stochastic linear systems that arise from the finite element formulation of stochastic partial differential equations. The results provided in Chapter 6 serve as a proof of concept in this regard. However, more research is required to investigate the computational efficiency and parallel scalability of non-parametric belief propagation for solving stochastic linear systems.

# Appendices

# Appendix A

A local Jacobian matrix is constructed for $\mathrm{FN_a}$, based on the sensitivities calculated during FGaBP:

$$\mathbf{J}_a^{(m)} = \begin{bmatrix} \mathbf{I}_{2nc\times 2nc} & \mathbf{J}_{a,v}^{(m)} \\ \mathbf{J}_{a,T}^{(m)} & \mathbf{I}_{nc\times nc}. \end{bmatrix} \tag{A.1}$$

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:

$$\mathbf{J}_{a,v}^{(m,t)}(k,l) = \frac{\partial \alpha_{ak,v}^{(m,t)}}{\partial \beta_{al,T}^{(m-1)}} = \frac{\partial \alpha_{ak,v}^{(m,t)}}{\partial T_l^{(m-1)}} \frac{\partial T_l^{(m-1)}}{\partial \beta_{al,T}^{(m-1)}} = \frac{1}{\alpha_{l,T}^{(m-1)}} \frac{\partial \alpha_{ak,v}^{(m,t)}}{\partial T_l^{(m-1)}}, \tag{A.2}$$

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

$$\mathbf{J}_{a,v}^{(m,t)}(k,l) = \frac{\partial \beta_{ak,v}^{(m,t)}}{\partial \beta_{al,T}^{(m-1)}} = \frac{\partial \beta_{ak,v}^{(m,t)}}{\partial T_l^{(m-1)}} \frac{\partial T_l^{(m-1)}}{\partial \beta_{al,T}^{(m-1)}} = \frac{1}{\alpha_{l,T}^{(m-1)}} \frac{\partial \beta_{ak,v}^{(m,t)}}{\partial T_l^{(m-1)}}, \tag{A.3}$$

for $nc \leq k < 2nc$. Here the chain rule and also the fact that based on (3.21) we can write $\partial T_l / \partial \beta_{al,T} = 1/\alpha_{l,T}$ are used. Now, the elements of the sub-matrix $\mathbf{J}_{a,v}^{(m,t)}$ are updated as:

$$\mathbf{J}_{a,v}^{(m,t)}(k,l) = \frac{1}{\alpha_{l,T}^{(m-1)}} \frac{\partial f_k}{\partial T_l^{(m-1)}}, \tag{A.4}$$

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

$$\mathbf{J}_{a,v}^{(m,t)}(k,l) = \frac{1}{\alpha_{l,T}^{(m-1)}} \Big( \frac{\partial g_k}{\partial T_l^{(m-1)}} + \sum_{l' \in \mathcal{N}(a)} \frac{\partial g_k}{\partial v_{l'}^{(m,t_*)}} \frac{\partial v_{l'}^{(m,t_*)}}{\partial T_l^{(m-1)}} \Big), \tag{A.5}$$

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

$$\frac{\partial f_k}{\partial T_l^{(m-1)}} = \frac{-1}{\tilde{W}^2{}_{k,v}} \frac{\partial \tilde{W}_{k,v}}{\partial T_l^{(m-1)}} \ , \tag{A.6}$$

$$\begin{aligned}\frac{\partial g_k}{\partial T_l^{(m-1)}} = & \frac{-1}{\tilde{W}^2{}_{k,v}} (\bar{\mathbf{K}}_v{}^{(m,t_*)})^T \cdot \tilde{\mathbf{C}}_v \frac{\partial \tilde{W}_{k,v}}{\partial T_l^{(m-1)}} + \\ & \frac{(\bar{\mathbf{K}}_v{}^{(m,t_*)})^T}{\tilde{W}_{k,v}} \cdot \frac{\partial \tilde{\mathbf{C}}_v}{\partial T_l^{(m-1)}} \ ,\end{aligned} \tag{A.7}$$

and

$$\frac{\partial g_k}{\partial v_{l'}^{(m,t_*)}} = \begin{cases} 0 & k = l' \\ \frac{1}{\tilde{W}_{k,v}(T_l)} \frac{\partial \left(\bar{\mathbf{K}}_v{}^{(m,t_*)}\right)^T}{\partial v_{l'}^{(m,t_*)}} \cdot \tilde{\mathbf{C}}_v(T_l) & k \neq l'. \end{cases} \tag{A.8}$$

Finally, the partial derivatives in A.6 to A.8 are calculated analytically based on (3.14) to (3.18).

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

$$\begin{aligned}\mathbf{J}_{a,T}^{(m,t)}(k,l) &= \frac{\partial \beta_{ak,T}^{(m,t)}}{\partial \beta_{al,v}^{(m-1)}} = \frac{\partial \beta_{ak,T}^{(m,t)}}{\partial v_l^{(m-1)}} \frac{\partial v_l^{(m-1)}}{\partial \beta_{al,v}^{(m-1)}} = \frac{1}{\alpha_{l,v}^{(m-1)}} \frac{\partial \beta_{ak,T}^{(m,t)}}{\partial v_l^{(m-1)}} \\ &= \frac{1}{\alpha_{l,v}^{(m-1)}} \left( \frac{\partial h_k}{\partial v_l^{(m-1)}} + \sum_{l' \in \mathcal{N}(a)} \frac{\partial h_k}{\partial T_{l'}^{(m)}} \frac{\partial T_{l'}^{(m)}}{\partial v_l^{(m-1)}} \right),\end{aligned} \tag{A.9}$$

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

$$\begin{aligned}\mathbf{J}_{a,T}^{(m,t)}(k,l) &= \frac{\partial \beta_{ak,T}^{(m,t)}}{\partial \alpha_{al,v}^{(m-1)}} = \frac{\partial \beta_{ak,T}^{(m,t)}}{\partial v_l^{(m-1)}} \frac{\partial v_l^{(m-1)}}{\partial \alpha_{al,v}^{(m-1)}} = \frac{-\beta_{l,v}^{(m-1)}}{[\alpha_{l,v}^{(m-1)}]^2} \frac{\partial \beta_{ak,T}^{(m,t)}}{\partial v_l^{(m-1)}} \\ &= \frac{-\beta_{l,v}^{(m-1)}}{[\alpha_{l,v}^{(m-1)}]^2} \left( \frac{\partial h_k}{\partial v_l^{(m-1)}} + \sum_{l' \in \mathcal{N}(a)} \frac{\partial h_k}{\partial T_{l'}^{(m)}} \frac{\partial T_{l'}^{(m)}}{\partial v_l^{(m-1)}} \right),\end{aligned} \tag{A.10}$$

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

$$v_l = \frac{\beta_{l,v}}{\alpha_{l,v}} = \frac{1}{\alpha_{l,v}} \sum_{a' \in \mathcal{N}(l)} \beta_{a'l,v} \ , \tag{A.11}$$

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 A.6 to A.8, the sensitivities $\partial h_k / \partial T_{l'}$ and $\partial h_k / \partial v_l$ are calculated as follows:

$$\frac{\partial h_k}{\partial T_{l'}^{(m)}} = \begin{cases} 0 & k = l' \\ \frac{1}{\tilde{W}_{k,v}} \frac{\partial \left(\bar{\mathbf{K}_T}{}^{(m,t_*)}\right)^T}{\partial T_{l'}^{(m)}} \cdot \tilde{\mathbf{C}}_T & k \neq l \end{cases} \ , \tag{A.12}$$

$$\frac{\partial h_k}{\partial v_l^{(m-1)}} = \frac{\partial B_{k,T}}{\partial v_l^{(m-1)}} + \frac{1}{\tilde{W}_{k,v}} \frac{\partial \left(\bar{\mathbf{K}}_T{}^{(t_*)}\right)^T}{\partial v_l^{(m-1)}} \, \tilde{\mathbf{C}}_T \,, \tag{A.13}$$

in which the partial derivatives are calculated analytically based on (3.14) to (3.18).

# Bibliography

[1] J. D. Yang, P. Hainaut, G. J. Gores, A. Amadou, A. Plymoth, and L. R. Roberts, “A global view of hepatocellular carcinoma: trends, risk, prevention and management,” *Nature reviews Gastroenterology & hepatology*, pp. 1–16, 2019.

[2] S. F. Altekruse, K. A. McGlynn, and M. E. Reichman, “Hepatocellular carcinoma incidence, mortality, and survival trends in the united states from 1975 to 2005,” *Journal of clinical oncology*, vol. 27, no. 9, p. 1485, 2009.

[3] S. G. Delis and C. Dervenis, “Selection criteria for liver resection in patients with hepatocellular carcinoma and chronic liver disease,” *World Journal of Gastroenterology: WJG*, vol. 14, no. 22, p. 3452, 2008.

[4] P. Voglreiter, P. Mariappan, M. Pollari, R. Flanagan, R. B. Sequeiros, R. H. Portugaller, J. Fütterer, D. Schmalstieg, M. Kolesnik, and M. Moche, “Rfa guardian: Comprehensive simulation of radiofrequency ablation treatment of liver tumors,” *Scientific reports*, vol. 8, no. 1, p. 787, 2018.

[5] J. H. Oh, D. H. Sinn, G.-S. Choi, J. M. Kim, J.-W. Joh, T. W. Kang, D. Hyun, W. Kang, G.-Y. Gwak, Y.-H. Paik *et al.*, “Comparison of outcome between liver resection, radiofrequency ablation, and transarterial therapy for multiple small hepatocellular carcinoma within the milan criteria,” *Annals of Surgical Treatment and Research*, vol. 99, no. 4, p. 238, 2020.

[6] M. B. Glassberg, S. Ghosh, J. W. Clymer, R. A. Qadeer, N. C. Ferko, B. Sadeghirad, G. W. Wright, and J. F. Amaral, “Microwave ablation compared with radiofrequency ablation for treatment of hepatocellular carcinoma and liver metastases: a

systematic review and meta-analysis," *OncoTargets and therapy*, vol. 12, p. 6407, 2019.

[7] F. Izzo, V. Granata, R. Grassi, R. Fusco, R. Palaia, P. Delrio, G. Carrafiello, D. Azoulay, A. Petrillo, and S. A. Curley, "Radiofrequency ablation and microwave ablation in liver tumors: an update," *The oncologist*, vol. 24, no. 10, p. e990, 2019.

[8] E. J. Berjano, "Theoretical modelling for radiofrequency ablation: state-of-the-art and challenges for the future," *BioMed. Eng. Online*, vol. 5, no. 1:24, 2006.

[9] S. L. Wong, P. B. Mangu, M. A. Choti, T. S. Crocenzi, G. D. Dodd III, G. S. Dorfman, C. Eng, Y. Fong, A. F. Giusti, D. Lu *et al.*, "American society of clinical oncology 2009 clinical evidence review on radiofrequency ablation of hepatic metastases from colorectal cancer," *Journal of Clinical Oncology*, vol. 28, no. 3, pp. 493–508, 2009.

[10] J. H. Kim, H. J. Won, Y. M. Shin, S. H. Kim, H.-K. Yoon, K.-B. Sung, and P. N. Kim, "Medium-sized (3.1–5.0 cm) hepatocellular carcinoma: transarterial chemoembolization plus radiofrequency ablation versus radiofrequency ablation alone," *Annals of surgical oncology*, vol. 18, no. 6, pp. 1624–1629, 2011.

[11] S. Payne, R. Flanagan, M. Pollari, T. Alhonnoro, C. Bost, D. O'Neill, T. Peng, and P. Stiegler, "Image-based multi-scale modelling and validation of radio-frequency ablation in liver tumours," *Phil. Trans. R. Soc. A*, vol. 369, no. 1954, pp. 4233–4254, 2011.

[12] Scientific animations. [Online]. Available: https://www.scientificanimations.com/

[13] Y.-D. Liu, Q. Li, Z. Zhou, Y.-W. Yeah, C.-C. Chang, C.-Y. Lee, and P.-H. Tsui, "Adaptive ultrasound temperature imaging for monitoring radiofrequency ablation," *Plos one*, vol. 12, no. 8, p. e0182457, 2017.

[14] J. Zhang and S. Chauhan, "Neural network methodology for real-time modelling of bio-heat transfer during thermo-therapeutic applications," *Artificial Intelligence in Medicine*, vol. 101, p. 101728, 2019.

[15] C. Audigier, T. Mansi, H. Delingette, S. Rapaka, V. Mihalef, D. Carnegie, E. Boctor, M. Choti, A. Kamen, N. Ayache *et al.*, "Efficient lattice boltzmann solver for

patient-specific radiofrequency ablation of hepatic tumors," *IEEE Transactions on Medical Imaging*, vol. 34, no. 7, pp. 1576–1589, 2015.

[16] P. Mariappan, P. Weir, R. Flanagan, P. Voglreiter, T. Alhonnoro, M. Pollari, M. Moche, H. Busse, J. Futterer, H. R. Portugaller *et al.*, "Gpu-based rfa simulation for minimally invasive cancer treatment of liver tumours," *International journal of computer assisted radiology and surgery*, vol. 12, no. 1, pp. 59–68, 2017.

[17] M. Reinhardt, P. Brandmaier, D. Seider, M. Kolesnik, S. Jenniskens, R. B. Sequeiros, M. Eibisberger, P. Voglreiter, R. Flanagan, P. Mariappan *et al.*, "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*, vol. 8, pp. 25–32, 2017.

[18] S. K. Hall, E. H. Ooi, and S. J. Payne, "Cell death, perfusion and electrical parameters are critical in models of hepatic radiofrequency ablation," *International Journal of Hyperthermia*, vol. 31, no. 5, pp. 538–550, 2015.

[19] Comsol multiphysics® v. 5.4., comsol ab, stockholm, sweden. [Online]. Available: www.comsol.com

[20] D. Panescu, J. G. Whayne, S. D. Fleischman, M. S. Mirotznik, D. K. Swanson, and J. G. Webster, "Three-dimensional finite element analysis of current density and temperature distributions during radio-frequency ablation," *IEEE Transactions on Biomedical Engineering*, vol. 42, no. 9, pp. 879–890, 1995.

[21] S. Tungjitkusolmun, S. T. Staelin, D. Haemmerich, J.-Z. Tsai, H. Cao, J. G. Webster, F. T. Lee, D. M. Mahvi, and V. R. Vorperian, "Three-dimensional finite-element analyses for radio-frequency hepatic tumor ablation," *IEEE transactions on biomedical engineering*, vol. 49, no. 1, pp. 3–9, 2002.

[22] F. Soetaert, G. Crevecoeur, and L. Dupré, "Coupled electrical-thermal model for monopolar and bipolar radiofrequency liver tumor ablation," in *2016 International Symposium on Fundamentals of Electrical Engineering (ISFEE)*. IEEE, 2016, pp. 1–5.

[23] J. K. Cheong, S. Yap, E. T. Ooi, and E. H. Ooi, "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*, vol. 176, pp. 17–32, 2019.

[24] R. Chen, F. Lu, K. Wang, D. Kong *et al.*, "Semiautomatic radiofrequency ablation planning based on constrained clustering process for hepatic tumors," *IEEE Transactions on Biomedical Engineering*, vol. 65, no. 3, pp. 645–657, 2017.

[25] N. Kath, H. Handels, and A. Mastmeyer, "Robust gpu-based virtual reality simulation of radio-frequency ablations for various needle geometries and locations," *International journal of computer assisted radiology and surgery*, pp. 1–11, 2019.

[26] X. Lu, H. Kikuchi, K. Hirooka, Y. Isobe, H. Watanabe, Y. Kobayashi, T. Miyashita, and M. G. Fujie, "Method for estimating the temperature distribution associated with the vessel cooling effect in radio frequency ablation," in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC).* IEEE, 2015, pp. 4836–4839.

[27] E. H. Ooi, K. W. Lee, S. Yap, M. A. Khattab, I. Y. Liao, E. T. Ooi, J. J. Foo, S. R. Nair, and A. F. M. Ali, "The effects of electrical and thermal boundary condition on the simulation of radiofrequency ablation of liver cancer for tumours located near to the liver boundary," *Computers in biology and medicine*, vol. 106, pp. 12–23, 2019.

[28] T. Peng, D. O'Neill, and S. Payne, "A two-equation coupled system for determination of liver tissue temperature during thermal ablation," *International Journal of Heat and Mass Transfer*, vol. 54, no. 9-10, pp. 2100–2109, 2011.

[29] M. Trujillo and E. Berjano, "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*, vol. 29, no. 6, pp. 590–597, 2013.

[30] D. D. Yero, F. G. González, D. Van Troyen, and G. A. Vandenbosch, "Dielectric properties ofex vivoporcine liver tissue characterized at frequencies between 5 and

500 khz when heated at different rates," *IEEE Transactions on Biomedical Engineering*, vol. 65, no. 11, pp. 2560–2568, 2018.

[31] N. P. Silva, A. Bottiglieri, R. C. Conceição, M. O'Halloran, and L. Farina, "Characterisation of ex vivo liver thermal properties for electromagnetic-based hyperthermic therapies," *Sensors*, vol. 20, no. 10, p. 3004, 2020.

[32] S. A. Solazzo, Z. Liu, S. M. Lobo, M. Ahmed, A. U. Hines-Peralta, R. E. Lenkinski, and S. N. Goldberg, "Radiofrequency ablation: importance of background tissue electrical conductivity—an agar phantom and computer modeling study," *Radiology*, vol. 236, no. 2, pp. 495–502, 2005.

[33] J. Abraham and E. M. Sparrow, "A thermal-ablation bioheat model including liquid-to-vapor phase change, pressure-and necrosis-dependent perfusion, and moisture-dependent properties," *International Journal of Heat and Mass Transfer*, vol. 50, no. 13-14, pp. 2537–2544, 2007.

[34] G. Ristovski, N. Garbers, H. K. Hahn, T. Preusser, and L. Linsen, "Uncertainty-aware visual analysis of radiofrequency ablation simulations," *Computers & Graphics*, vol. 79, pp. 24–35, 2019.

[35] D. P. O'Neill, T. Peng, P. Stiegler, U. Mayrhauser, S. Koestenbauer, K. Tscheliessnigg, and S. J. Payne, "A three-state mathematical model of hyperthermic cell death," *Annals of biomedical engineering*, vol. 39, no. 1, pp. 570–579, 2011.

[36] J. A. Pearce, "Models for thermal damage in tissues: processes and applications," *Critical Reviews™ in Biomedical Engineering*, vol. 38, no. 1, 2010.

[37] I. A. Chang, "Considerations for thermal injury analysis for rf ablation devices," *The open biomedical engineering journal*, vol. 4, p. 3, 2010.

[38] M. Trujillo, J. Bon, M. José Rivera, F. Burdío, and E. Berjano, "Computer modelling of an impedance-controlled pulsing protocol for rf tumour ablation with a cooled electrode," *International Journal of Hyperthermia*, vol. 32, no. 8, pp. 931–939, 2016.

[39] P. X. Mouratidis, I. Rivens, J. Civale, R. Symonds-Tayler, and G. Ter Haar, "Relationship between thermal dose and cell death for "rapid" ablative and "slow"

hyperthermic heating," *International Journal of Hyperthermia*, vol. 36, no. 1, pp. 228–242, 2019.

[40] D. E. Keyes, L. C. McInnes, C. Woodward, W. Gropp, E. Myra, M. Pernice *et al.*, "Multiphysics simulations: Challenges and opportunities," *The International Journal of High Performance Computing Applications*, vol. 27, no. 1, pp. 4–83, 2013.

[41] B. Duan and C.-K. Chui, "Multiscale modeling of liver bio-impedance and frequency control for radiofrequency ablation," in *2016 IEEE Region 10 Conference (TENCON)*. IEEE, 2016, pp. 1532–1535.

[42] Running comsol® in parallel on clusters. [Online]. Available: https://www.comsol.com/support/knowledgebase/1001

[43] C. Rieder, T. Kroeger, C. Schumann, and H. K. Hahn, "Gpu-based real-time approximation of the ablation zone for radiofrequency ablation," *IEEE transactions on visualization and computer graphics*, vol. 17, no. 12, pp. 1812–1821, 2011.

[44] Y. Vutov, D. Nikolov, I. Lirkov, and K. Georgiev, "Computer simulation of a saline enhanced radio-frequency hepatic ablation process," in *International Conference on Large-Scale Scientific Computing*. Springer, 2019, pp. 382–390.

[45] J. Zhang and S. Chauhan, "Fast computation of soft tissue thermal response under deformation based on fast explicit dynamics finite element algorithm for surgical simulation," *Computer Methods and Programs in Biomedicine*, vol. 187, p. 105244, 2020.

[46] E. Besler, Y. C. Wang, and A. V. Sahakian, "Real-time radiofrequency ablation lesion depth estimation using multi-frequency impedance with a deep neural network and tree-based ensembles," *IEEE Transactions on Biomedical Engineering*, 2019.

[47] I. Altrogge, T. Pätz, T. Kröger, H.-O. Peitgen, and T. Preusser, "Optimization and fast estimation of vessel cooling for rf ablation," in *World Congress on Medical Physics and Biomedical Engineering, September 7-12, 2009, Munich, Germany.* Springer, 2009, pp. 1202–1205.

[48] R. G. Ghanem and P. D. Spanos, *Stochastic finite elements: a spectral approach.* Courier Corporation, 2003.

[49] Y. El-Kurdi, M. M. Dehnavi, W. J. Gross, and D. Giannacopoulos, "Parallel finite element technique using gaussian belief propagation," *Comput. Phys. Commun.*, vol. 193, pp. 38–48, 2015.

[50] D. A. Knoll and D. E. Keyes, "Jacobian-free newton–krylov methods: a survey of approaches and applications," *Journal of Computational Physics*, vol. 193, no. 2, pp. 357–397, 2004.

[51] Y. El-Kurdi, "Parallel finite element processing using gaussian belief propagation inference on probabilistic graphical models," Ph.D. dissertation, McGill University Libraries, 2015.

[52] W. Liu and B. Vinter, "Csr5: An efficient storage format for cross-platform sparse matrix-vector multiplication," in *Proceedings of the 29th ACM on International Conference on Supercomputing*, 2015, pp. 339–350.

[53] C. A. Linte, J. J. Camp, D. R. Holmes, M. E. Rettmann, and R. A. Robb, "Toward online modeling for lesion visualization and monitoring in cardiac ablation therapy," in *International Conference on Medical Image Computing and Computer-Assisted Intervention.* Springer, 2013, pp. 9–17.

[54] S. Chen, J. Fang, D. Chen, C. Xu, and Z. Wang, "Optimizing sparse matrix-vector multiplication on emerging many-core architectures," *arXiv preprint arXiv:1805.11938*, 2018.

[55] M. A. Heroux, R. A. Bartlett, V. E. Howle, R. J. Hoekstra, J. J. Hu, T. G. Kolda, R. B. Lehoucq, K. R. Long, R. P. Pawlowski, E. T. Phipps *et al.*, "An overview of the trilinos project," *ACM Transactions on Mathematical Software (TOMS)*, vol. 31, no. 3, pp. 397–423, 2005.

[56] Petsc - portable, extensible toolkit for scientific computation. [Online]. Available: https://www.mcs.anl.gov/petsc/

[57] J. Pearl, *Probabilistic reasoning in intelligent systems: networks of plausible inference.* Elsevier, 2014.

[58] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Understanding belief propagation and its generalizations," *Exploring artificial intelligence in the new millennium*, vol. 8, pp. 236–239, 2003.

[59] O. Shental, P. H. Siegel, J. K. Wolf, D. Bickson, and D. Dolev, "Gaussian belief propagation solver for systems of linear equations," in *2008 IEEE International Symposium on Information Theory.* IEEE, 2008, pp. 1863–1867.

[60] D. Bickson, "Gaussian belief propagation: Theory and aplication," *arXiv preprint arXiv:0811.2518*, 2008.

[61] Y. El-Kurdi, W. J. Gross, and D. Giannacopoulos, "Parallel multigrid acceleration for the finite-element gaussian belief propagation algorithm," *IEEE transactions on magnetics*, vol. 50, no. 2, pp. 581–584, 2014.

[62] M. T. Heath, *Scientific Computing: An Introductory Survey, Revised Second Edition.* SIAM, 2018.

[63] E. Kersal, "Analytic solutions of partial differential equations," *University of Leeds, Leeds*, 2004.

[64] J. W. Thomas, *Numerical partial differential equations: finite difference methods.* Springer Science & Business Media, 2013, vol. 22.

[65] W. Cui, K. Gawecka, D. Taborda, D. Potts, and L. Zdravković, "Time-step constraints in transient coupled finite element analysis," *International Journal for Numerical Methods in Engineering*, vol. 106, no. 12, pp. 953–971, 2016.

[66] H. H. Pennes, "Analysis of tissue and arterial blood temperatures in the resting human forearm," *Journal of applied physiology*, vol. 85, no. 1, pp. 5–34, 1998.

[67] T. Szabó, "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, 2008, pp. 564–571.

[68] W. Bangerth, R. Hartmann, and G. Kanschat, "deal. ii—a general-purpose object-oriented finite element library," *ACM Transactions on Mathematical Software (TOMS)*, vol. 33, no. 4, pp. 24–es, 2007.

[69] D. P. Rodgers, "Improvements in multiprocessor system design," *ACM SIGARCH Computer Architecture News*, vol. 13, no. 3, pp. 225–231, 1985.

[70] J. L. Gustafson, "Reevaluating amdahl's law," *Communications of the ACM*, vol. 31, no. 5, pp. 532–533, 1988.

[71] S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, A. Dener, V. Eijkhout, W. D. Gropp, D. Karpeyev, D. Kaushik, M. G. Knepley, D. A. May, L. C. McInnes, R. T. Mills, T. Munson, K. Rupp, P. Sanan, B. F. Smith, S. Zampini, H. Zhang, and H. Zhang, "PETSc users manual," Argonne National Laboratory, Tech. Rep. ANL-95/11 - Revision 3.13, 2020. [Online]. Available: https://www.mcs.anl.gov/petsc

[72] ——, "PETSc Web page," https://www.mcs.anl.gov/petsc, 2019. [Online]. Available: https://www.mcs.anl.gov/petsc

[73] S. Balay, W. D. Gropp, L. C. McInnes, and B. F. Smith, "Efficient management of parallelism in object oriented numerical software libraries," in *Modern Software Tools in Scientific Computing*, E. Arge, A. M. Bruaset, and H. P. Langtangen, Eds. Birkhäuser Press, 1997, pp. 163–202.

[74] P. Amestoy, A. Buttari, J.-Y. L'Excellent, and T. Mary, "Performance and Scalability of the Block Low-Rank Multifrontal Factorization on Multicore Architectures," *ACM Transactions on Mathematical Software*, vol. 45, pp. 2:1–2:26, 2019.

[75] P. Amestoy, I. S. Duff, J. Koster, and J.-Y. L'Excellent, "A fully asynchronous multifrontal solver using distributed dynamic scheduling," *SIAM Journal on Matrix Analysis and Applications*, vol. 23, no. 1, pp. 15–41, 2001.

[76] Know your body. [Online]. Available: https://knowyourbody.net

[77] W. Bangerth, C. Burstedde, T. Heister, and M. Kronbichler, "Algorithms and data structures for massively parallel generic adaptive finite element codes," *ACM Transactions on Mathematical Software (TOMS)*, vol. 38, no. 2, pp. 1–28, 2012.

[78] C. Burstedde, L. C. Wilcox, and O. Ghattas, "p4est: Scalable algorithms for parallel adaptive mesh refinement on forests of octrees," *SIAM Journal on Scientific Computing*, vol. 33, no. 3, pp. 1103–1133, 2011.

[79] G. Stefanou, "The stochastic finite element method: past, present and future," *Computer methods in applied mechanics and engineering*, vol. 198, no. 9-12, pp. 1031–1051, 2009.

[80] M. A. Gutiérrez and S. Krenk, "Stochastic finite element methods," *Encyclopedia of Computational Mechanics Second Edition*, pp. 1–25, 2018.

[81] N. Wiener, "The homogeneous chaos," *American Journal of Mathematics*, vol. 60, no. 4, pp. 897–936, 1938.

[82] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Transactions on signal processing*, vol. 50, no. 2, pp. 174–188, 2002.

[83] A. Smith, *Sequential Monte Carlo methods in practice.* Springer Science & Business Media, 2013.

[84] A. Ihler and D. McAllester, "Particle belief propagation," in *Artificial intelligence and statistics.* PMLR, 2009, pp. 256–263.

[85] E. B. Sudderth, A. T. Ihler, M. Isard, W. T. Freeman, and A. S. Willsky, "Nonparametric belief propagation," *Communications of the ACM*, vol. 53, no. 10, pp. 95–103, 2010.

[86] N. Noorshams and M. J. Wainwright, "Belief propagation for continuous state spaces: Stochastic message-passing with quantitative guarantees," *The Journal of Machine Learning Research*, vol. 14, no. 1, pp. 2799–2835, 2013.

[87] D. F. Crouse, P. Willett, K. Pattipati, and L. Svensson, "A look at gaussian mixture reduction algorithms," in *14th international conference on information fusion.* IEEE, 2011, pp. 1–8.

[88] J. E. Harmse, "Reduction of gaussian mixture models by maximum similarity," *Journal of Nonparametric Statistics*, vol. 22, no. 6, pp. 703–709, 2010.

[89] E. J. Berjano and F. Hornero, "Thermal-electrical modeling for epicardial atrial radiofrequency ablation," *IEEE transactions on biomedical engineering*, vol. 51, no. 8, pp. 1348–1357, 2004.

[90] H. B. El-Serag, J. A. Davila, N. J. Petersen, and K. A. McGlynn, "The continuing increase in the incidence of hepatocellular carcinoma in the united states: an update," *Annals of internal medicine*, vol. 139, no. 10, pp. 817–823, 2003.

[91] S. Garrean, J. Hering, A. Saied, W. S. Helton, and N. J. Espat, "Radiofrequency ablation of primary and metastatic liver tumors: a critical review of the literature," *The American Journal of Surgery*, vol. 195, no. 4, pp. 508–520, 2008.

[92] E. Schena, D. Tosi, P. Saccomandi, E. Lewis, and T. Kim, "Fiber optic sensors for temperature monitoring during thermal treatments: an overview," *Sensors*, vol. 16, no. 7, p. 1144, 2016.

[93] I. Chang, "Finite element analysis of hepatic radiofrequency ablation probes using temperature-dependent electrical conductivity," *Biomedical engineering online*, vol. 2, no. 1, p. 12, 2003.

[94] A. Nakayama and F. Kuwahara, "A general bioheat transfer model based on the theory of porous media," *International Journal of Heat and Mass Transfer*, vol. 51, no. 11-12, pp. 3190–3199, 2008.

[95] W. Roetzel and Y. Xuan, "Transient response of the human limb to an external stimulust," *International journal of heat and mass transfer*, vol. 41, no. 1, pp. 229–239, 1998.

[96] A.-R. Khaled and K. Vafai, "The role of porous media in modeling flow and heat transfer in biological tissues," *International Journal of Heat and Mass Transfer*, vol. 46, no. 26, pp. 4989–5003, 2003.

[97] D. Shrivastava and J. T. Vaughan, "A generic bioheat transfer thermal model for a perfused tissue," *Journal of biomechanical engineering*, vol. 131, no. 7, p. 074506, 2009.

[98] S. Mahjoob and K. Vafai, "Analytical characterization and production of an isothermal surface for biological and electronic applications," *Journal of Heat Transfer*, vol. 131, no. 5, p. 052604, 2009.

[99] S. Labonte, "Numerical model for radio-frequency ablation of the endocardium and its experimental validation," *IEEE Transactions on Biomedical Engineering*, vol. 41, no. 2, pp. 108–115, 1994.

[100] J. Chato, "Heat transfer to blood vessels," *Journal of biomechanical engineering*, vol. 102, no. 2, pp. 110–118, 1980.

[101] H. Klinger, "Heat transfer in perfused biological tissue—i: General theory," *Bulletin of mathematical biology*, vol. 36, no. 4, pp. 403–415, 1974.

[102] W. Wulff, "The energy conservation equation for living tissue," *IEEE transactions on biomedical engineering*, no. 6, pp. 494–495, 1974.

[103] F. Henriques Jr and A. Moritz, "Studies of thermal injury: I. the conduction of heat to and through skin and the temperatures attained therein. a theoretical and an experimental investigation," *The American journal of pathology*, vol. 23, no. 4, p. 530, 1947.

[104] A. R. Moritz and F. Henriques Jr, "Studies of thermal injury: Ii. the relative importance of time and surface temperature in the causation of cutaneous burns," *The American journal of pathology*, vol. 23, no. 5, p. 695, 1947.

[105] D. Haemmerich, J. Webster, and D. Mahvi, "Thermal dose versus isotherm as lesion boundary estimator for cardiac and hepatic radio-frequency ablation," in *Engineering in Medicine and Biology Society, 2003. Proceedings of the 25th Annual International Conference of the IEEE*, vol. 1. IEEE, 2003, pp. 134–137.

[106] M. A. Wood, K. M. Shaffer, A. L. Ellenbogen, and E. D. Ownby, "Microbubbles during radiofrequency catheter ablation: composition and formation," *Heart Rhythm*, vol. 2, no. 4, pp. 397–403, 2005.

[107] E. J. Berjano, J. L. Al, and J. Saiz, "Modeling for radio-frequency conductive keratoplasty: implications for the maximum temperature reached in the cornea," *Physiological measurement*, vol. 26, no. 3, p. 157, 2005.

[108] J. Gopalakrishnan, "A mathematical model for irrigated epicardial radiofrequency ablation," *Annals of biomedical engineering*, vol. 30, no. 7, pp. 884–893, 2002.

[109] X. Liu, M. Smelyanskiy, E. Chow, and P. Dubey, "Efficient sparse matrix-vector multiplication on x86-based many-core processors," in *Proceedings of the 27th international ACM conference on International conference on supercomputing*, 2013, pp. 273–282.

[110] Q. Zhang and S. Cen, *Multiphysics Modeling: Numerical Methods and Engineering Applications: Tsinghua University Press Computational Mechanics Series.* Elsevier, 2015.

[111] X.-C. Cai and D. E. Keyes, "Nonlinearly preconditioned inexact newton algorithms," *SIAM Journal on Scientific Computing*, vol. 24, no. 1, pp. 183–200, 2002.

[112] G. Otto, C. Dber, M. Hoppe-Lotichius, J. Knig, M. Heise, and M. B. Pitton, "Radiofrequency ablation as first-line treatment in patients with early colorectal liver metastases amenable to surgery," *Annals of surgery*, vol. 251, no. 5, pp. 796–803, 2010.

[113] D. Haemmerich, L. Chachati, A. S. Wright, D. M. Mahvi, F. T. Lee, and J. G. Webster, "Hepatic radiofrequency ablation with internally cooled probes: effect of coolant temperature on lesion size," *IEEE Transactions on Biomedical Engineering*, vol. 50, no. 4, pp. 493–500, 2003.

[114] Y. El-Kurdi, D. Giannacopoulos, and W. J. Gross, "Relaxed gaussian belief propagation," in *Information Theory Proceedings (ISIT), 2012 IEEE International Symposium on.* IEEE, 2012, pp. 2002–2006.

[115] Y. Tang, T. Jin, and R. C. Flesch, "Numerical temperature analysis of magnetic hyperthermia considering nanoparticle clustering and blood vessels," *IEEE Trans. Magn.*, vol. 53, no. 10, pp. 1–6, 2017.

[116] S. Ho, S. Niu, and W. Fu, "Design and analysis of novel focused hyperthermia devices," *IEEE Trans. Magn.*, vol. 48, no. 11, pp. 3254–3257, 2012.

[117] M. Bullo, V. D'Ambrosio, F. Dughiero, and M. Guarnieri, "A 3-d cell method formulation for coupled electric and thermal problems," *IEEE Tras. Magn.*, vol. 43, no. 4, pp. 1197–1200, 2007.

[118] Y. El-Kurdi, W. J. Gross, and D. Giannacopoulos, “Efficient implementation of gaussian belief propagation solver for large sparse diagonally dominant linear systems,” *IEEE Trans. Magn.*, vol. 48, no. 2, pp. 471–474, 2012.

[119] H. Jiang, P. Luo, D. Zhi, F. Qi, X. Huang, Z. Lu, M. Ge, L. Wang, and B. Qiu, “In vitro and in vivo experimental studies of a novel mr-guided method for bipolar radiofrequency liver ablation,” *IEEE Access*, vol. 6, pp. 21 859–21 866, 2018.

[120] B. Duan, R. Wen, C.-B. Chng, W. Wang, P. Liu, J. Qin, J. L. Peneyra, S. K.-Y. Chang, P.-A. Heng, and C.-K. Chui, “Image-guided robotic system for radiofrequency ablation of large liver tumor with single incision,” in *2015 12th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI).* IEEE, 2015, pp. 284–289.

[121] J. J. Pérez, J. J. Pérez-Cajaraville, V. Muñoz, and E. Berjano, “Computer modeling of electrical and thermal performance during bipolar pulsed radiofrequency for pain relief,” *Medical physics*, vol. 41, no. 7, p. 071708, 2014.

[122] R. S. Dembo, S. C. Eisenstat, and T. Steihaug, “Inexact newton methods,” *SIAM Journal on Numerical analysis*, vol. 19, no. 2, pp. 400–408, 1982.

[123] C. Audigier, T. Mansi, H. Delingette, S. Rapaka, T. Passerini, V. Mihalef, M.-P. Jolly, R. Pop, M. Diana, L. Soler *et al.*, “Comprehensive preclinical evaluation of a multi-physics model of liver tumor radiofrequency ablation,” *International journal of computer assisted radiology and surgery*, vol. 12, no. 9, pp. 1543–1559, 2017.

[124] X. Wang, H. Gao, S. Wu, T. Jiang, Z. Zhou, and Y. Bai, “Numerical evaluation of ablation zone under different tip temperatures during radiofrequency ablation,” *Math. Biosci. Eng*, vol. 16, pp. 2514–2531, 2019.

[125] C. Rossmann, F. Rattay, and D. Haemmerich, “Platform for patient-specific finite-element modeling and application for radiofrequency ablation,” *Visualization, Image Processing and Computation in Biomedicine*, vol. 1, no. 1, 2012.

[126] A. Akbari and D. Giannacopoulos, “An efficient multi-threaded newton–raphson algorithm for strong coupling modeling of multi-physics problems,” *Computer Physics Communications*, vol. 258, p. 107563.