IT 09 060

Examensarbete 30 hp
November 2009

Finite Element Methods for Microelectromechanical Systems

Yang Zeng

Institutionen för informationsteknologi
Department of Information Technology

UPPSALA
UNIVERSITET

**Teknisk- naturvetenskaplig fakultet
UTH-enheten**

Besöksadress:
Ångströmlaboratoriet
Lägerhyddsvägen 1
Hus 4, Plan 0

Postadress:
Box 536
751 21 Uppsala

Telefon:
018 – 471 30 03

Telefax:
018 – 471 30 00

Hemsida:
http://www.teknat.uu.se/student

Abstract

Finite Element Methods for Microelectromechanical Systems

Yang Zeng

The stationary Joule heating problem is a crucial multiphysical problem for many microelectromechanical (MEMS) applications. In our paper, we derive a finite element method for this problem and introduce iterative solution-techniques to compute the numerical simulation. Further we construct an adaptive algorithm for mesh refinement based on a posteriori error estimation.
Finally, we present two numerical tests: convergences analysis of different iterative methods for distinct materials which are classified by electrical conductivities, and a test of the new adaptive refinement algorithm. All the numerical implementations have been done in MATLAB.

Handledare: Axel Målqvist
Ämnesgranskare: Maya Neytcheva
Examinator: Anders Jansson
IT 09 060
Tryckt av: Reprocentralen ITC

Contents

List of Figures

Acknowledgements

First of all, I would like to express my warmest gratitude to my supervisor, Axel Målqvist, for his patience and instructive suggestions on the writing of this thesis. Without his invaluable help and generous encouragement, I would not accomplish my thesis.

Besides, I wish to thank the department of information technology of Uppsala University for giving me the opportunity to study here. I am also greatly indebted to the professors and teachers in Uppsala University, who have instructed and helped me a lot in the past two years.

The last but not the least, my thanks would go to my beloved family and friends for their loving considerations and great confidence in me all through these years.

Chapter 1

Introduction

Partial differential equations are used to model physical phenomena. In many important applications, several different physical processes are active at the same time. One such example is the design of microelectromechanical systems (MEMS).

MEMS are e.g. used to build sensors on micrometer scale. Here electric potential, heat transfer and mechanical stresses are coupled in a system of non-linear elliptic partial differential equations. The stationary Joule heating problem is a model problem for this application. A voltage is applied at the boundary of a device. A current that flows through the device is produced, which leads to heating of the material. The equations describing electrostatical potential and the temperature are coupled. The heat equation is driven by the electrical current and the electric conductivity depends on the temperature, which means we get coupling in both directions.

Many industrial codes are optimized for solving single physics problems, such as the two individuals equations in the Joule heating application. In order to take advantage of this infrastructure, engineers typically couple together such optimized single physics solvers when solving coupled problems. One approach is to iterate between two problems in a Jacobi or Gauss-Seidel fashion with a given initial guess. However, it is hard to predict whether the iteration will converge or not.

In this paper, we will study numerical simulation of the Joule heating problem using the Finite Element Method, then construct an adaptive algorithm for mesh refinement based on a posteriori error estimates as well as analyze convergence of different iterative methods for distinct materials in MEMS devices.

Chapter 2

Physical Background

2.1 Single Physics Problems

2.1.1 Heat equation

The heat equation is an important partial differential equation which describes the distribution of heat (or variation in temperature) in a given region over time. The stationary heat equation reads

$$-\nabla \cdot k\nabla u = f \tag{2.1}$$

where k is the thermal conductivity, u is the temperature and f is a given heat source.

2.1.2 Potential equation

Electrostatic phenomena arises from the forces that electric charges exert on each other. One of the cornerstones of electrostatics is the posing and solving of problems described by the Poisson equation

$$-\nabla \cdot \sigma(u)\nabla\phi = g \tag{2.2}$$

where $\sigma(u)$ is the electrical conductivity, which is strongly dependent on temperature, ϕ is the electric potential and g is a given function.

2.2 Multiphysics Problems

Joule heating is generated by the resistance of materials to electric current and presents in any electric conductor. Therefore it is crucial in many MEMS applications. Let u be the absolute temperature and ϕ be the electric potential in a solid electric conductor represented by a bounded domain Ω. Under steady conditions, the stationary Joule heating problem consists of the following nonlinear elliptic system

$$-\nabla \cdot k\nabla u = \sigma(u) \cdot |\nabla\phi|^2, \qquad \text{in } \Omega \tag{2.3}$$

$$-\nabla \cdot \sigma(u)\nabla\phi = f, \qquad \text{in } \Omega \tag{2.4}$$

with some suitable boundary conditions, where k and $\sigma(u)$ are the thermal and electrical conductivities respectively which we suppose to be given positive functions.

A number of different materials can be used in MEMS technology. Silicon is the material used to create most integrated circuits used in consumer electronic in the modern world. Because of availabilities of cheap high-quanlity and incorporating electronics functionality, silicon has been exploited for a wide variety of MEMS applications. Metals can also be used to create MEMS elements because of their high reliability. The devices based on so called high-temperature superconductivity have been paid more and more attention in past two decades. These different materials exhibit distinct physical properties, but we are more interested in their electrical conductivities in our paper.

2.3 Electrical Conductivities

2.3.1 Common definition

When an electrical potential difference is placed across a conductor, its movable charges flow, giving rise to an electrical current. The conductivity σ is defined as the ratio of the current density J to the electrical field strength

E

$$J = \sigma E. \tag{2.5}$$

Conductivity is the reciprocal of electrical resistivity ρ, and has SI unit of siemens per meter $S \cdot m^{-1}$

$$\sigma = \frac{1}{\rho}. \tag{2.6}$$

2.3.2 Classifications

According to the electrical conductivity, materials can be classified into the following categories:

- A conductor such as metal has high conductivity and a low resistivity;
- An insulator like glass or a vacuum has low conductivity and a high resistivity;
- The conductivity of a semiconductor is generally intermediate, but varies widely under different conditions, such as exposure of the material to electric fields or specific frequencies of light and, most important, with temperature and composition of the semiconductor material;
- Superconductivity occurs at extremely low temperature (not far from absolute zero), and materials have been found to exhibit very high electrical conductivity in this phenomenon.

Based on this classifications, we will test convergences of iterative solution-techniques when we are solving the Joule heating problem for three sorts of materials: metals, semiconductors and superconductors.

2.3.3 Temperature dependency

Electrical conductivity is strongly dependent on temperature. This dependence is often expressed using a conductivity-vs-temperature graph. Let us define σ' to be the electrical conductivity at a standard temperature u' and

α to be the temperature compensation slope. Based on the physical property of metals such that electrical conductivity decreases with the increasing temperature, we can describe this kind of conductivity-vs-temperature graph as

$$\sigma(u) = \begin{cases} \sigma' & u \le u' \\ \frac{\sigma'}{1+\alpha(u-u')} & u > u' \end{cases} \tag{2.7}$$

where $\sigma(u)$ is electrical conductivity at temperature u.

In semiconductors, electrical conductivity increases as temperature is increasing, hence its dependence can be written as

$$\sigma(u) = \begin{cases} \sigma' & u \le u' \\ \sigma'\big(1+\alpha(u-u')\big) & u > u' \end{cases} \tag{2.8}$$

where $\sigma(u)$, σ', α and u' have the same explanations as equation (2.7).

Finally let us consider superconductors. As we know, one of the most important physical properties is that when temperature is decreased to a very low value (not far from absolute zero), the electrical resistance of a superconductor lowers to exact zero. Electrical conductivity is the inverse of electrical resistivity, i.e. it will be extremely large in such situation. Based on this property, we can define the conductivity-vs-temperature graph for superconductors as

$$\sigma(u) = \begin{cases} \infty & u < u' \\ \frac{\sigma'}{\alpha(u-u')} & u \ge u' \end{cases} \tag{2.9}$$

where $\sigma(u)$, σ', α and u' are the same as equations (2.7) and (2.8).

Chapter 3

Mathematical Model and FEM

3.1 Preliminaries

First of all, let us settle some definitions and notations that will be frequently used in the paper. The scalar product $(\cdot, \cdot)$ is the ordinary $L^2 = L^2(\Omega)$ product and $\| \cdot \|$ is the corresponding norm. A triangulation, or mesh, $\mathcal{K}$ of Ω is a set $\{K\}$ of triangles K such that $\Omega = \bigcup_{K \in \mathcal{K}} K$ and intersection of two triangles is either a triangle edge, a triangle corner, or empty. A closed polyline $\partial\mathcal{K}$ of $\partial\Omega$ is a set $\{E\}$ of edges E such that $\partial\Omega = \bigcup_{E \in \partial\mathcal{K}} E$. In other words, $\{E\}$ is a set of sides of triangles which are on the boundary. The boundary of a triangle K is denoted by ∂K. If an edge is shared by two triangles, we call it an interior edge, otherwise it will be on the boundary of Ω, i.e. $\partial K \cup \partial\Omega = E \in \partial\mathcal{K}$. We denote the longest edge of triangle K by h_K and the length of edge E by h_E.

Let K be a triangle with nodes at the corners $N_1 = (x_1^1, x_2^1)$, $N_2 = (x_1^2, x_2^2)$, and $N_3 = (x_1^3, x_2^3)$ and let $\mathcal{P}_1(K)$ denote the vector space of linear polynomials defined on K

$$\mathcal{P}_1(K) = \{v : v(x_1, x_2) = c_0 + c_1 x_1 + c_2 x_2, c_0, c_1, c_2 \in R\}. \tag{3.1}$$

Let V_h be the vector space of all continuous piecewise linear polynomials

$$V_h = \{v : v \in \mathcal{C}(\Omega), v|_K \in \mathcal{P}_1(K) \ \ \forall K \in \mathcal{K}\} \tag{3.2}$$

where $\mathcal{C}(\Omega)$ denotes the space of all continuous functions on Ω and $\mathcal{P}_1(K)$ is the space of linear polynomials on K as defined by (3.1). We denote the basis functions $\varphi_j(N_i) \subset V_h$, such that

$$\varphi_j(N_i) = \begin{cases} 0 & i = j \\ 1 & i \neq j \end{cases}$$

for $i, j = 1, 2, \dots, N$. Using these basis functions, all $v \in V_h$ can be written as

$$v(x_1, x_2) = \sum_{i=1}^{N} \alpha_i \varphi_i(x_1, x_2) \tag{3.3}$$

where the coefficients α_i are the nodal values of the function v, that is

$$\alpha_i = v(N_i), \quad i = 1, 2, \dots, N. \tag{3.4}$$

Given a continuous function $f \in \mathcal{C}(K)$ on a triangle K with nodes at $N_i = (x_1^i, x_2^i)$, $i = 1, 2, 3$, we define the interpolant $\pi f \in \mathcal{P}_1(K)$ of f as

$$\pi f = \sum_{i=1}^{3} f(N_i)\varphi_i.$$

We let Df and $D^2 f$ be defined by

$$\begin{aligned} Df &= (|\frac{\partial f}{\partial x}|^2 + |\frac{\partial f}{\partial y}|^2)^{1/2}, \\ D^2 f &= (|\frac{\partial^2 f}{\partial x^2}|^2 + 2|\frac{\partial^2 f}{\partial x \partial y}|^2 + |\frac{\partial^2 f}{\partial y^2}|^2)^{1/2}. \end{aligned}$$

Using these notations, the following proposition for the interpolation error estimates has been shown, i.e. in [2].

Proposition 3.1. *The following estimates hold*

$$\begin{aligned} \sum_K \|f - \pi f\|^2_{L^2(K)} &\leq C \sum_K h_K^2 \|Df\|^2_{L^2(K)} \\ \sum_K \|D(f - \pi f)\|^2_{L^2(K)} &\leq C \sum_K \|Df\|^2_{L^2(K)} \end{aligned}$$

where C is a constant.

3.2 The Model Problem

Let us define a model problem for the stationary Joule heating problem with mixed boundary conditions. It can be written as

$$-\nabla \cdot k\nabla u = \sigma(u) \cdot |\nabla \phi|^2 \qquad \text{in } \Omega \tag{3.5}$$

$$\mathbf{n} \cdot k\nabla u = \kappa_1(g_1 - u) \qquad \text{on } \partial\Omega \tag{3.6}$$

$$-\nabla \cdot \sigma(u)\nabla \phi = f \qquad \text{in } \Omega \tag{3.7}$$

$$\phi = g_2 \qquad \text{on } \Gamma_D \tag{3.8}$$

$$\mathbf{n} \cdot \sigma(u)\nabla \phi = \kappa_2(g_3 - \phi) \qquad \text{on } \Gamma_N \tag{3.9}$$

where g_1, g_2, g_3 and f are given functions, Ω is a two-dimensional domain with the boundary $\partial\Omega$, $\partial\Omega = \Gamma_D \cup \Gamma_N$, and $\mathbf{n}$ is the outward normal of $\partial\Omega$, as shown in Figure 3.1. Next we will derive a finite element method for this model.

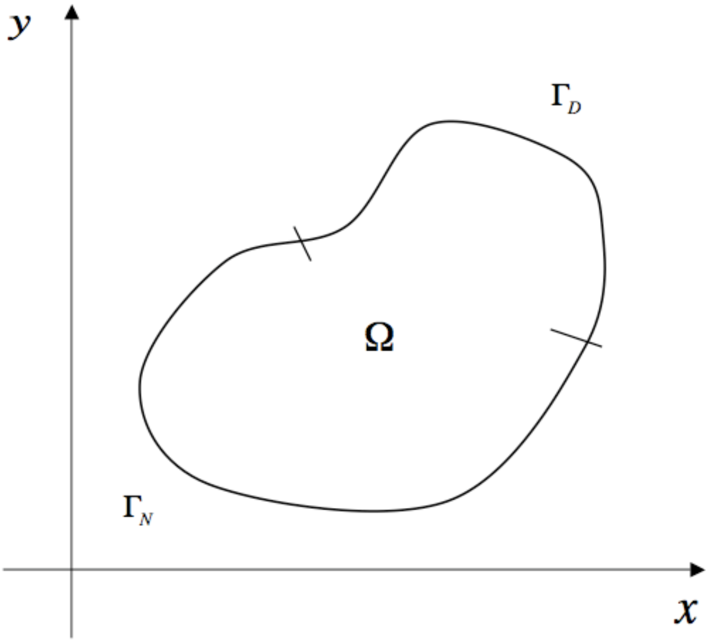

Figure 3.1: A domain Ω and boundary $\partial\Omega$, $\partial\Omega = \Gamma_D \cup \Gamma_N$.

3.3 Finite Element Method

3.3.1 Heat equation

In this section, we consider the stationary heat equation (3.5) – (3.6). We assume for now that the right hand side of (3.5) is a given function. To distinguish between u and ϕ in the left hand sides of (3.5) and (3.7), we denote it by $\sigma(u^*) \cdot |\nabla \phi^*|^2$.

Let us introduce a space

$$V = \{v : \|v\| + \|\nabla v\| < \infty\}. \tag{3.10}$$

Multiplying (3.5) by a test function $v \in V$, and integrating on both sides, then using *Green's formula*, we get

$$\begin{aligned} -\int_\Omega (\nabla \cdot k\nabla u) \cdot v dx &= \int_\Omega k\nabla u \cdot \nabla v dx - \int_{\partial\Omega} \mathbf{n} \cdot k\nabla u \cdot v ds \\ &= \int_\Omega k\nabla u \cdot \nabla v dx + \int_{\partial\Omega} \kappa_1 (u - g_1) \cdot v ds \\ &= \int_\Omega \sigma(u^*) \cdot \ |\nabla \phi^*|^2 \cdot v dx \end{aligned}$$

since $\mathbf{n} \cdot k\nabla u = \kappa_1(g_1 - u)$ on $\partial\Omega$. Thus we obtain the variational formulation for the problem (3.5) – (3.6): *Find* $u \in V$ *such that*

$$\begin{aligned} \int_\Omega k\nabla u \cdot \nabla v dx \quad &+ \quad \int_{\partial\Omega} \kappa_1 u \cdot v ds \\ &= \int_\Omega \sigma(u^*) \cdot \ |\nabla \phi^*|^2 \cdot v dx + \int_{\partial\Omega} \kappa_1 g_1 \cdot v ds, \ \ \forall v \in V. \end{aligned} \tag{3.11}$$

Based on this form, we define a finite element method: *Find* $U \in V_h \subset V$, *such that*

$$\begin{aligned} \int_\Omega k\nabla U \cdot \nabla v dx \quad &+ \quad \int_{\partial\Omega} \kappa_1 U \cdot v ds \\ &= \int_\Omega \sigma(u^*) \cdot \ |\nabla \phi^*|^2 \cdot v dx + \int_{\partial\Omega} \kappa_1 g_1 \cdot v ds, \ \ \forall v \in V_h \end{aligned} \tag{3.12}$$

From the variational formulation and the finite element method, we get the following theorem:

Theorem 3.2. *For the problem* (3.5) – (3.6) *with a given exact right hand side of* (3.5) *which has been denoted as* $\sigma(u^*)\cdot|\nabla\phi^*|^2$*, the following Galerkin orthogonality property holds*

$$\int_\Omega k\nabla e\cdot\nabla v dx + \int_{\partial\Omega}\kappa_1 evds = 0, \quad \forall v\in V_h \tag{3.13}$$

where $e = u - U$ *is the error between the exact and the finite element solution.*

Proof. Subtracting two equations (3.11) and (3.12), then using the fact that $V_h\subset V$ immediately proves this claim. □

3.3.2 Potential equation

Let us seek a solution to the potential problem which consisting of (3.7) – (3.9). Assume $\sigma(u)$ is a given function which is the same as in the right hand side of heat equation (3.5), hence we can rewrite it as $\sigma(u^*)$.

Let us introduce a space

$$V_{g_2,D} = \{v : \|v\| + \|\nabla v\| < \infty, v|_{\Gamma_D} = g_2\}. \tag{3.14}$$

Multiplying (3.7) by a test function $v\in V_{0,D}$, and integrating on both sides, then using *Green's formula*, we get

$$\begin{aligned}
-\int_\Omega(\nabla\cdot\sigma(u^*)\nabla\phi)\cdot vdx &= \int_\Omega\sigma(u^*)\nabla\phi\cdot\nabla vdx - \int_{\partial\Omega}\mathbf{n}\cdot\sigma(u^*)\nabla\phi\cdot vds\\
&= \int_\Omega\sigma(u^*)\nabla\phi\cdot\nabla vdx + \int_{\Gamma_N}\kappa_2(\phi-g_3)\cdot vds\\
&= \int_\Omega f\cdot vdx
\end{aligned}$$

since $\mathbf{n}\cdot\sigma(u^*)\nabla\phi = \kappa_2(g_3-\phi)$ on Γ_N and $v = 0$ on Γ_D. Thus the variational problem reads as follows: *Find* $\phi\in V_{g_2,D}$ *such that*

$$\int_\Omega\sigma(u^*)\nabla\phi\cdot\nabla vdx + \int_{\Gamma_N}\kappa_2\phi\cdot vds = \int_\Omega f\cdot vdx + \int_{\Gamma_N}\kappa_2 g_3\cdot vds, \quad \forall v\in V_{0,D}. \tag{3.15}$$

We assume g_2 to be piecewise polynomial and continuous on the boundary Γ_D. That means, there is a function $\Phi_{g_2}\in V_{h,D}$ such that $\Phi_{g_2} = g_2$ on Γ_D.

We introduce the affine subspace $V_{h,g_2,D} = \{v \in V_h : v|_{\Gamma_D} = g_2\}$, then the finite element method reads: *Find* $\Phi \in V_{h,g_2,D}$*, such that*

$$\int_\Omega \sigma(u^*)\nabla\Phi \cdot \nabla v dx + \int_{\Gamma_N} \kappa_2 \Phi \cdot v ds = \int_\Omega f \cdot v dx + \int_{\Gamma_N} \kappa_2 g_3 \cdot v ds, \quad \forall v \in V_{h,0,D}. \tag{3.16}$$

From the variational formulation and the finite element method, we obtain the following theorem:

Theorem 3.3. *For the problem defined by* (3.7) – (3.9) *with a given function* $\sigma(u^*)$*, the following Galerkin orthogonality property holds*

$$\int_\Omega \sigma(u^*)\nabla e \cdot \nabla v dx + \int_{\Gamma_N} \kappa_2 e v ds = 0, \quad \forall v \in V_{h,0,D} \tag{3.17}$$

where $e = \phi - \Phi$ *is the error between the exact and the finite element solution.*

Proof. Subtracting two equations (3.15) and (3.16), then using the fact that $V_{h,0,D} \subset V_{0,D}$ immediately proves this claim. □

To derive an equation for Φ, we will use a technique presented in [1]. We write Φ in the form

$$\Phi = \Phi_0 + \Phi_{g_2} \tag{3.18}$$

where Φ_{g_2} is any fixed function in $V_{h,g_2,D}$ and $\Phi_0 = 0$ on Γ_D and thus $\Phi_0 \in V_{h,0,D}$. This construction of Φ will satisfy the boundary conditions because of $\Phi_{g_2} = g_2$ on Γ_D. Since Φ_{g_2} is known it remains to determine Φ_0, we get the equation: *Find* $\Phi_0 \in V_{h,0,D}$*, such that*

$$\begin{aligned}&\int_\Omega \sigma(u^*)\nabla\Phi_0 \cdot \nabla v dx + \int_{\Gamma_N} \kappa_2 \Phi_0 \cdot v ds \\ &= \int_\Omega f \cdot v dx + \int_{\Gamma_N} \kappa_2 g_3 \cdot v ds - \int_\Omega \sigma(u^*)\nabla\Phi_{g_2} \cdot \nabla v dx - \int_{\Gamma_N} \kappa_2 \Phi_{g_2} \cdot v ds\end{aligned} \tag{3.19}$$

for $\forall v \in V_{h,0,D}$. This is a problem of the same kind as above but with a modified right hand side. One can prove that the solution $\Phi = \Phi_0 + \Phi_{g_2}$ is independent of the particular choice of the function Φ_{g_2}. In practice Φ_{g_2} is often chosen to be zero at all interior nodes plus all the nodes on Γ_N. And more details about this technique will be illustrated in the later section.

Next, let us derive linear systems resulting from these two finite element problems. However, we need to introduce quadrature rules which will be utilized in this paper.

3.4 Gauss Quadrature Rule

In numerical analysis, a quadrature rule is an approximation of the definite integral of a function, usually has the form of a weighted sum of function values at specified points within the domain of integration. Assume f is a given function, a general quadrature rule on a triangle K or a edge E takes the form

$$\int_K f dx \approx \sum_j w_j f(q_j)$$

or

$$\int_E f ds \approx \sum_j w_j f(q_j)$$

where q_j is the set of quadrature points within K or on E.

3.4.1 Area coordinates

To explain the interpolation functions with higher degree, we will introduce the definition so-call *area coordinates* in [5].

Definition 3.4 (**Area coordinates**). *For triangular elements, it is possible to construct three non-dimensional coordinates $L_i (i = 1, 2, 3)$, which vary in a direction normal to the sides directly opposite each node. The coordinates are defined such that*

$$L_i = \frac{A_i}{A} \quad i = 1, 2, 3 \tag{3.20}$$

$$A = \sum_{i=1}^{3} A_i \tag{3.21}$$

where A_i is the area of the triangle formed by nodes j and k $(j, k = 1, 2, 3)$ and arbitrary point P in the element, and A is the total area of the element.

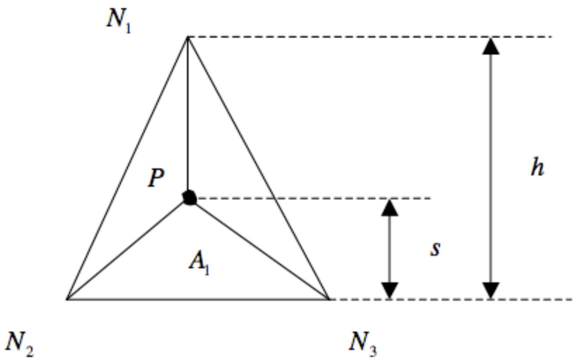

Figure 3.2: The example of area coordinates

For example, assume A_1 is the area of the triangle which is formed by nodes N_2 and N_3 and point P, as showed in Figure 3.2. The point P is at a distance of s from the side connecting nodes N_2 and N_3. We have

$$\begin{aligned} A_1 &= \frac{b \cdot s}{2} \\ A &= \frac{b \cdot h}{2} \end{aligned}$$

where h is the distance from the node 1 to the side connecting nodes N_2 and N_3 and b is the length of this side. Hence,

$$L_1 = \frac{A_1}{A} = \frac{s}{h}.$$

Clearly, L_1 is zero on side $N_2 - N_3$ (hence, zero at nodes N_2 and N_3) and has a value of unity at node N_1. In other words, L_1 is the finite element basis function associated with node N_1. Similarly, L_2 and L_3 are the basis functions associated with nodes N_2 and N_3.

3.4.2 Gauss quadrature rule on triangles

On an arbitrary triangle, using the *Gauss quadrature rule*, we can obtain the following approximation

$$\int_K f(x)dx \approx |K| \cdot \sum_{i=1}^{N} w_i f(G_i)$$

where N is the number of the Gaussian points; G_i is the i^{th} Gaussian point; w_i is the weight of i^{th} Gaussian point; $|K|$ is the area of the triangle. Assume this triangle is defined by three nodes (x_1^1, x_2^1), (x_1^2, x_2^2) and (x_1^3, x_2^3), then G_i can be calculated by the formula

$$G_i = \begin{bmatrix} x_1^1 & x_2^1 & x_3^1 \\ x_1^2 & x_2^2 & x_3^2 \end{bmatrix} \cdot \begin{bmatrix} L_1^i \\ L_2^i \\ L_3^i \end{bmatrix}$$

where L_1^i, L_2^i and L_3^i are the area coordinates of point G_i in the reference triangle. They can be can be found in Table A.1 in Appendix.

3.4.3 Gauss quadrature rule on edges

On the boundary, it is possible to construct two non-dimensional coordinates L_1 and L_2. If the edge E lies between two boundary nodes (x_1^1, x_2^1) and (x_1^2, x_2^2), then we use the *Gauss quadrature rule* to obtain the following approximation

$$\int_E f(x)dx \approx |h_E| \cdot \sum_{i=1}^{N} w_i f(G_i)$$

where N is the number of the Gaussian points; G_i is the i^{th} Gaussian point; w_i is the weight of i^{th} Gaussian point; $|h_E|$ is the length of the edge. Further, G_i is defined by

$$G_i = \begin{bmatrix} x_1^1 & x_2^1 \\ x_1^2 & x_2^2 \end{bmatrix} \cdot \begin{bmatrix} L_1^i \\ L_2^i \end{bmatrix}$$

where L_1^i and L_2^i are the area coordinates of point G_i on the reference edge. They can be given in Table A.2 in Appendix.

3.5 Implementation Details

3.5.1 Derivation of the linear system for the heat equation

Let $\{\varphi_i\}_{i=1}^N$ be basis functions of V_h which are defined on the mesh $\mathcal{K}$. Then we write U as a linear combination of the basis functions

$$U = \sum_{j=1}^{N} \zeta_j \varphi_j \tag{3.22}$$

with unknown coefficients ζ_j where $j = 1, 2, \ldots, N$. Inserting (3.22) into (3.12), we get

$$\begin{aligned} \int_\Omega k(\nabla \sum_{j=1}^{N} \zeta_j \varphi_j) \cdot \nabla \varphi_i dx \quad &+ \quad \int_{\partial\Omega} \kappa_1 (\sum_{j=1}^{N} \zeta_j \varphi_j) \varphi_i ds \\ &= \quad \int_\Omega \sigma(u^*) \cdot |\nabla \phi^*|^2 \varphi_i dx + \int_{\partial\Omega} \kappa_1 g_1 \varphi_i ds \end{aligned} \tag{3.23}$$

for $i = 1, 2, \ldots, N$. Introducing the notations

$$a_{i,j}^1 = \int_\Omega k \nabla \varphi_i \cdot \nabla \varphi_j dx, \tag{3.24}$$

$$m_{i,j}^1 = \int_{\partial\Omega} \kappa_1 \varphi_i \varphi_j ds, \tag{3.25}$$

$$b_i^1 = \int_\Omega \sigma(u^*) \cdot |\nabla \phi^*|^2 \varphi_i dx, \tag{3.26}$$

$$r_i^1 = \int_{\partial\Omega} \kappa_1 g_1 \varphi_i ds, \tag{3.27}$$

for $i, j = 1, 2, \ldots, N$. Then we get

$$\sum_{j=1}^{N} (a_{i,j}^1 + m_{i,j}^1) \zeta_j = b_i^1 + r_i^1 \tag{3.28}$$

for $i = 1, 2, \ldots, N$, which is a linear system for the coefficients ζ_j. In the matrix form we write as

$$(A^1 + M^1)\zeta = b^1 + r^1 \tag{3.29}$$

with $N \times N$ matrices A^1 and M^1, $N \times 1$ vectors b^1, r^1 are defined above respectively.

3.5.2 Derivation of the linear system for the potential equation

Let $\{\varphi_i\}_{i=1}^N$ be basis functions of V_h which are defined on the mesh $\mathcal{K}$ as well. Then we write Φ as a linear combination of the basis functions

$$\Phi = \sum_{j=1}^{N} \xi_j \varphi_j \tag{3.30}$$

with unknown coefficients ξ_j where $j = 1, 2, \ldots, N$. Inserting (3.30) into equation (3.16), we get

$$\begin{aligned} \int_\Omega \sigma(u^*)(\nabla \sum_{j=1}^{N} \xi_j \varphi_j) \cdot \nabla \varphi_i dx \quad &+ \quad \int_{\Gamma_N} \kappa_2 (\sum_{j=1}^{N} \xi_j \varphi_j) \varphi_i ds \\ &= \quad \int_\Omega f \varphi_i dx + \int_{\Gamma_N} \kappa g_3 \varphi_i ds \end{aligned} \tag{3.31}$$

for $i = 1, 2, \ldots, N$. Introducing the notations

$$a_{i,j}^2 = \int_\Omega \sigma(u^*) \nabla \varphi_i \cdot \nabla \varphi_j dx, \tag{3.32}$$

$$m_{i,j}^2 = \int_{\Gamma_N} \kappa_2 \varphi_i \varphi_j ds, \tag{3.33}$$

$$b_i^2 = \int_\Omega f \varphi_i dx, \tag{3.34}$$

$$r_i^2 = \int_{\Gamma_N} \kappa_2 g_3 \varphi_i ds, \tag{3.35}$$

for $i, j = 1, 2, \ldots, N$. Then we get

$$\sum_{j=1}^{N} (a_{i,j}^2 + m_{i,j}^2) \xi_j = b_i^2 + r_i^2 \tag{3.36}$$

for $i = 1, 2, \ldots, N$, which is a linear systems for the coefficients ξ_j. In the matrix form we write as

$$(A^2 + M^2)\xi = b^2 + r^2 \tag{3.37}$$

with $N \times N$ matrixes A^2 and M^2, $N \times 1$ vectors b^2 and r^2 are defined above respectively.

Based on (3.18) and (3.19), let us assume that the first S nodes are interior nodes including the nodes on Γ_N, while the last $N - S$ nodes are boundary nodes on Γ_D for the linear system of the potential equation (3.37). These boundary nodes are fixed since the nodal values of Φ should be g_2. Using these nodes numbering, we can partition the linear system $(A^2 + M^2)\xi = b^2 + r^2$ into the following form

$$\begin{bmatrix} A^2_{0,0} + M^2_{0,0} & A^2_{0,g_2} + M^2_{0,g_2} \\ A^2_{g_2,0} + M^2_{g_2,0} & A^2_{g_2,g_2} + M^2_{g_2,g_2} \end{bmatrix} \cdot \begin{bmatrix} \xi_0 \\ \xi_{g_2} \end{bmatrix} = \begin{bmatrix} b^2_0 + r^2_0 \\ b^2_{g_2} + r^2_{g_2} \end{bmatrix}$$

where $A^2_{0,0}$ and $M^2_{0,0}$ are the upper left $S \times S$ block of A^2 and M^2, while $A^2_{g_2,g_2}$ and $M^2_{g_2,g_2}$ are the lower right $(N-S)\times(N-S)$ block of A^2 and M^2. Rearranging the first S equations of this linear system, we have the $S \times S$ linear system

$$(A^2_{0,0} + M^2_{0,0})\xi_0 = (b^2_0 + r^2_0) - (A^2_{0,g_2} + M^2_{0,g_2})\xi_{g_2} \tag{3.38}$$

from which the unknown interior nodal values as well as the nodal values on Γ_N of Φ can be determined.

3.5.3 Assembling of the stiffness matrices

Recall the notations (3.24) and (3.32), the local 3×3 stiffness matrices are given by

$$\begin{aligned} A^{1,K}_{i,j} &= \int_K k\nabla\varphi_i\nabla\varphi_j dx, \\ A^{2,K}_{i,j} &= \int_K \sigma(u^*)\nabla\varphi_i\nabla\varphi_j dx, \end{aligned}$$

for $i, j = 1, 2, 3$.

Consider a triangle K with the nodes (x^1_1, x^1_2), (x^2_1, x^2_2) and (x^3_1, x^3_2). To each node N_i $(i = 1, 2, 3)$, there is a hat function φ_i associated, which takes on the value 1 at node N_i and 0 at other nodes. Each hat function is a linear function on K so it takes the form

$$\varphi_i = a_i + b_i x_i + c_i x_2, \quad i = 1, 2, 3$$

where the coefficients a_i, b_i and c_i are determined by the following linear systems

$$\begin{bmatrix} 1 & x_1^1 & x_2^1 \\ 1 & x_1^2 & x_2^2 \\ 1 & x_1^3 & x_2^3 \end{bmatrix} \cdot \begin{bmatrix} a_1 \\ b_1 \\ c_1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = e_1,$$

$$\begin{bmatrix} 1 & x_1^1 & x_2^1 \\ 1 & x_1^2 & x_2^2 \\ 1 & x_1^3 & x_2^3 \end{bmatrix} \cdot \begin{bmatrix} a_2 \\ b_2 \\ c_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = e_2,$$

$$\begin{bmatrix} 1 & x_1^1 & x_2^1 \\ 1 & x_1^2 & x_2^2 \\ 1 & x_1^3 & x_2^3 \end{bmatrix} \cdot \begin{bmatrix} a_3 \\ b_3 \\ c_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = e_3.$$

The gradient of φ_i is just the constant vector $\nabla\varphi_i = [b_i \ c_i]$. Then using the *Gauss quadrature rule*, we get

$$\begin{aligned} A_{i,j}^{1,K} &= \int_K k\nabla\varphi_i\nabla\varphi_j dx \\ &= (b_ib_j + c_ic_j)\int_K kdx \\ &\approx (b_ib_j + c_ic_j)\cdot|K|\cdot\sum_{l=1}^{Q}(w_l\cdot k(G_l)), \\ A_{i,j}^{2,K} &= \int_K \sigma(u^*)\nabla\varphi_i\nabla\varphi_j dx \\ &\approx (b_ib_j + c_ic_j)\cdot|K|\cdot\sum_{l=1}^{Q}(w_l\cdot\sigma(u_l^*)) \end{aligned}$$

where i, j $= 1, 2, 3$; G_l is the l^{th} Gaussian point; w_l is the weight of l^{th} Gaussian point; L_i^l is the i^{th} area coordinate of the l^{th} Gaussian point defined by (3.20); $\sigma(u_l^*)$ is the value of $\sigma(u^*)$ on the l^{th} Gauss point; Q is the number of Gaussian points.

3.5.4 Assembling of the load vectors

Recall the notations (3.26) and (3.34), on each element K, we get local 3×1 element vectors b_K^1 and b_K^2 with entries

$$\begin{aligned} b_i^{1,K} &= \int_K \sigma(u^*) \cdot |\nabla \phi^*|^2 \varphi_i dx, \\ b_i^{2,K} &= \int_K f \varphi_i dx \end{aligned}$$

for $i = 1, 2, 3$. Using the *Gauss quadrature rule* to compute these integrals, we obtain

$$\begin{aligned} b_i^{1,K} &\approx |K| \cdot \sum_{l=1}^{Q} (w_l \cdot \sigma(u_l^*) \cdot |\nabla \phi_l^*|^2 \cdot L_i^l) \\ b_i^{2,K} &\approx |K| \cdot \sum_{l=1}^{Q} (w_l \cdot f(G_l) \cdot L_i^l), \end{aligned}$$

where i, j $= 1, 2, 3$; G_l is the l^{th} Gaussian point; w_l is the weight of l^{th} Gaussian point; L_i^l is the i^{th} *area coordinate* of the l^{th} Gaussian point defined by (3.20); $\sigma(u_l^*)$ is the value of $\sigma(u^*)$ on the l^{th} Gauss point; $|\nabla \phi_l^*|^2$ is the value of $|\nabla \phi^*|^2$ on the l^{th} Gauss point; Q is the number of the Gaussian points.

3.5.5 Assembling of the boundary contributions

Two nodes of a triangle K lie along the domain $\partial\Omega$, then the edge between them will contribute to matrices entries $M_{i,j}^1$, $M_{i,j}^2$ and vectors entries r_i^1, r_i^2

$$\begin{aligned} M_{i,j}^{1,E} &= \int_E \kappa_1 \varphi_i \varphi_j ds \approx \kappa_1 |h_E| \cdot \sum_{l=1}^{Q} (w_l \cdot L_i^l \cdot L_j^l) \\ r_i^{1,E} &= \int_E \kappa_1 g_1 \varphi_i ds \approx \kappa_1 |h_E| \cdot \sum_{l=1}^{Q} (w_l \cdot g_1(G_l) \cdot L_i^l) \\ M_{i,j}^{2,E} &= \int_E \kappa_2 \varphi_i \varphi_j ds \approx \kappa_2 |h_E| \cdot \sum_{l=1}^{Q} (w_l \cdot L_i^l \cdot L_j^l) \\ r_i^{2,E} &= \int_E \kappa_2 g_3 \varphi_i ds \approx \kappa_2 |h_E| \cdot \sum_{l=1}^{Q} (w_l \cdot g_3(G_l) \cdot L_i^l) \end{aligned}$$

where i, j $= 1, 2, 3$; G_l is the l^{th} Gaussian point; w_l is the weight of l^{th} Gaussian point; L_i^l is the i^{th} *area coordinate* of the l^{th} Gaussian point defined by (3.20); Q is the number of the Gaussian points.

3.6 Iterative Methods for non-linear problems

So far we have derived two linear systems of two single physical problems which are based on an assumption that $\sigma(u^*) \cdot |\nabla\phi^*|^2$ and $\sigma(u^*)$ are two given functions. Let us write the relationship between u^*, ϕ^* and U , Φ in the following way

$$\begin{aligned} U &= f_1(u^*, \phi^*), & (3.39) \\ \Phi &= f_2(u^*). & (3.40) \end{aligned}$$

However, our model problem is a multiphysical problem involving both heat and potential equations. Hence u^* and ϕ^* must be the exact solutions u and ϕ defined by (3.5) – (3.9). Then as the numerical solutions of u and ϕ in the model problem, (3.39) and (3.40) can be represented as

$$\begin{aligned} U &= f_1(U, \Phi), & (3.41) \\ \Phi &= f_2(U). & (3.42) \end{aligned}$$

To solve these linear systems, we need to introduce iterative methods in our model.

An iterative method attempts to solve a problem by finding successive approximations to the solution starting from an initial guess. If an equation can be put into the form $F(X) = X$ (here, X could be a vector and contains several elements), a solution X is an attractive fixed point of the function F, then one may begin with a point $X^{(1)}$ in the basin of attraction of X. Let $X^{(i+1)} = F(X^i)$ for $i \geq 1$, and the sequence $\{X^{(i)}\}_{i\geq1}$ will converge to the solution X.

Examples of iterative methods are *Jacobi* method, *Gauss-Seidel* (*GS*) method and *Successive Over-relaxation* (*SOR*) method.

- *Jacobi* method: is an iterative technique that solves present values X by using previous values X in the right hand side. This method can be written as $X^{(i+1)} = F(X^{(i)})$;

- *GS* method: the computation of $x_j^{(i+1)}$ uses the elements of $X^{(i+1)}$ that have already been computed (denoted as $X_*^{(i+1)}$) and the elements of $X^{(i)}$ that have to be advanced to iteration $i+1$ (denoted as $X_{**}^{(i)}$). This method can be written as $X^{(i+1)} = F(X_*^{(i+1)}, X_{**}^{(i)})$;

- *SOR* method: is a variant of the *GS method*, resulting in faster convergence. It introduces a relaxation factor ω, which is a constant and greater than 0. This method can be written as $X^{(i+1)} = (1-\omega)X^{(i)} + \omega F(X_*^{(i+1)}, X_{**}^{(i)})$. When $\omega = 1$, it is the *GS* method. Since the similar method can be used for any slowly converging iterative process, we can use it to improve *Jacobi* method as well.

In this paper, assume we start from two guessing values U^0 and Φ^0, then these iterative methods would be implemented between heat and potential equations as:

- *Jacobi* method : $\{U^{(i)}, \Phi^{(i)}\} = \{f_1(U^{(i-1)}, \Phi^{(i-1)}), f_2(U^{(i-1)})\}$;

- *GS* method : $\{U^{(i)}, \Phi^{(i)}\} = \{f_1(U^{(i-1)}, \Phi^{(i-1)}), f_2(U^{(i)})\}$;

- *SOR* method : $\{U^{(i)}, \Phi^{(i)}\} = \{(1-\omega)U^{(i-1)} + \omega f_1(U^{(i-1)}, \Phi^{(i-1)}), (1-\omega)\Phi^{(i-1)} + \omega f_2(U^{(i)})\}$

where $i \geq 1$ and all procedures stop when a certain tolerance is reaching.

Chapter 4

Adaptive Finite Element Method

4.1 A Posteriori Error Estimates

Let us revisit the model problem

$$
\begin{aligned}
-\nabla \cdot k\nabla u &= \sigma(u) \cdot |\nabla\phi|^2 && \text{in } \Omega, && (4.1)\\
\mathbf{n} \cdot k\nabla u &= \kappa_1(g_1 - u) && \text{on } \partial\Omega, && (4.2)\\
-\nabla \cdot \sigma(u)\nabla\phi &= f && \text{in } \Omega, && (4.3)\\
\phi &= g_2 && \text{on } \Gamma_D, && (4.4)\\
\mathbf{n} \cdot \sigma(u)\nabla\phi &= \kappa_2(g_3 - \phi) && \text{on } \Gamma_N. && (4.5)
\end{aligned}
$$

For the heat equation defined by (4.1) – (4.2), we have a posteriori estimate:

Theorem 4.1. *For the finite element approximation U of the exact solution u to* (4.1) *and* (4.2) *with a given right hand side of* (4.1)*, $\sigma(u^*) \cdot |\nabla\phi^*|^2$, the following a posteriori error estimate holds*

$$\|\nabla(u-U)\|^2_{L^2(\Omega)} + \|u-U\|^2_{L^2(\partial\Omega)} \leq C\big(\sum_{K\in\mathcal{K}} \rho^2_K(U,u^*,\phi^*) + \sum_{E\in\partial\mathcal{K}} \rho^2_E(U)\big) \tag{4.6}$$

where C is a constant, the element residual in the interior domain $\rho_K(U,u^*,\phi^*)$

is defined by

$$\begin{aligned}\rho_K(U,u^*,\phi^*) &= h_K\|\sigma(u^*)\cdot|\nabla\phi^*|^2+\nabla\cdot k\nabla U\|_{L^2(K)}\\ &\quad+\frac{1}{2}h_K^{1/2}\|[\mathbf{n}\cdot k\nabla U]\|_{L^2(\partial K\setminus\partial\Omega)}\end{aligned}\tag{4.7}$$

and the element residual on the boundary $\rho_E(U)$ is defined by

$$\rho_E(U)=h_K^{1/2}\|\kappa_1(g_1-U)-\mathbf{n}\cdot k\nabla U\|_{L^2(E\cap\partial\Omega)}.\tag{4.8}$$

Here $[\mathbf{n}\cdot k\nabla U]$ denotes the jump in the k times normal derivative of U at an interior edge $\partial K_1\cap\partial K_2$, i.e.

$$[\mathbf{n}\cdot k\nabla U]|_{\partial K_1\cap\partial K_2}=\mathbf{n_1}\cdot k\nabla U_1+\mathbf{n_2}\cdot k\nabla U_2\tag{4.9}$$

with $U_i=U|_{K_i}$ and $\mathbf{n}_i$ is the exterior unit normal of K_i.

Proof. Let $e=u-U$ be the error. If $k,\kappa_1\geq\alpha>0$, we have

$$\begin{aligned}&\alpha\cdot\left(\|\nabla e\|^2_{L^2(\Omega)}+\|e\|^2_{L^2(\partial\Omega)}\right)\\ \leq\ &\int_\Omega k\nabla e\cdot\nabla e dx+\int_{\partial\Omega}\kappa_1 e\cdot e ds\\ =\ &\int_\Omega k\nabla e\cdot\nabla(e-\pi e)dx+\int_{\partial\Omega}\kappa_1 e\cdot(e-\pi e)ds\end{aligned}$$

where we have used the Galerkin orthogonality (see Theorem 3.2) to subtract the interpolant πe. Splitting this into a sum over the elements and using the *Green's formula*, or integration by parts, we further have

$$\begin{aligned}&\alpha\cdot\left(\|\nabla e\|^2_{L^2(\Omega)}+\|e\|^2_{L^2(\partial\Omega)}\right)\\ \leq\ &\sum_{K\in\mathcal{K}}\int_K k\nabla e\cdot\nabla(e-\pi e)dx+\sum_{E\in\partial\mathcal{K}}\int_E\kappa_1 e\cdot(e-\pi e)ds\\ =\ &\sum_{K\in\mathcal{K}}(-\int_K\nabla\cdot k\nabla e\cdot(e-\pi e)dx+\int_{\partial K}\mathbf{n}\cdot k\nabla e\cdot(e-\pi e)ds)\\ &+\sum_{E\in\partial\mathcal{K}}\int_E\kappa_1 e\cdot(e-\pi e)ds.\end{aligned}$$

First of all, let us consider the element residuals in the interior domain.

According to the equation (4.1), we obtain

$$\begin{aligned} & \sum_{K\in\mathcal{K}} -\int_K \nabla\cdot k\nabla e\cdot(e-\pi e)dx \\ = & \sum_{K\in\mathcal{K}} -\int_K \nabla\cdot k\nabla(u-U)\cdot(e-\pi e)dx \\ = & \sum_{K\in\mathcal{K}} \int_K (\sigma(u^*)\cdot|\nabla\phi^*|^2+\nabla\cdot k\nabla U)(e-\pi e)dx. \end{aligned}$$

Then using the *Cauchy-Schwartz inequality* and an interpolation error estimate (see Proposition 3.1), we can get

$$\begin{aligned} & \sum_{K\in\mathcal{K}} -\int_K \nabla\cdot k\nabla e\cdot(e-\pi e)dx \\ \leq & \sum_{K\in\mathcal{K}} \|\sigma(u^*)\cdot|\nabla\phi^*|^2+\nabla\cdot k\nabla U\|_K\|e-\pi e\|_K \\ \leq & \sum_{K\in\mathcal{K}} \|\sigma(u^*)\cdot|\nabla\phi^*|^2+\nabla\cdot k\nabla U\|_K Ch_K\|De\|_K \qquad (4.10) \end{aligned}$$

where C is a constant.

For each interior edge $\partial K_1\cap\partial K_2$, there are two contributions, one from triangle K_1 and the other one from triangle K_2. Summing these contributions, we get

$$\int_{\partial K_1\cap\partial K_2} (\mathbf{n_1}\cdot k\nabla e_1(e_1-\pi e_1)+\mathbf{n_2}\cdot k\nabla e_2(e_2-\pi e_2))ds,$$

where $e_i=e|_{K_i}$ and $\mathbf{n_i}$ is the exterior unit normal of K_i for $i=1,2$. Using the fact that the exact solution has a continuous normal derivative and that the error and its interpolant are continuous, we get

$$\sum_{K\in\mathcal{K}}\int_{\partial K\backslash\partial\Omega}\mathbf{n}\cdot k\nabla e(e-\pi e)ds=\sum_{K\in\mathcal{K}}\int_{\partial K\backslash\partial\Omega}([\mathbf{n}\cdot k\nabla U]/2)(e-\pi e)ds.$$

since all interior edges are considered twice. Then using the *Cauchy-Schwartz inequality* again and the so-called *trace inequality* in [1] followed an interpolation error estimate in Proposition 3.1, we can get

$$\begin{aligned}
&\sum_{K\in\mathcal{K}}\int_{\partial K\backslash\partial\Omega}([\mathbf{n}\cdot k\nabla U]/2)(e-\pi e)ds \\
\leq\ &\sum_{K\in\mathcal{K}}\|[\mathbf{n}\cdot k\nabla U]/2\|_{\partial K}\|e-\pi e\|_{\partial K} \\
\leq\ &\sum_{K\in\mathcal{K}}\|[\mathbf{n}\cdot k\nabla U]/2\|_{\partial K}C(h_K^{-1}\|e-\pi e\|_K^2+h_K\|D(e-\pi e)\|_K^2)^{1/2} \\
\leq\ &\sum_{K\in\mathcal{K}}\|[\mathbf{n}\cdot k\nabla U]/2\|_{\partial K}Ch_K^{1/2}\|De\|_K \qquad (4.11)
\end{aligned}$$

where C is a constant.

Now let us consider edges of a triangle which are not shared by others. They also have contributions to the element residuals but only on the boundary. Hence for each $\partial K\in\partial\mathcal{K}$, using the *Cauchy-Schwartz inequality*, the boundary condition (4.2) and *trace inequality* in [1] followed an interpolation error estimate in Proposition 3.1, we can get

$$\begin{aligned}
&\sum_{E\in\partial K}\int_E(\mathbf{n}\cdot k\nabla e+\kappa_1 e)(e-\pi e)ds \\
\leq\ &\sum_{E\in\partial K}\|\kappa_1(g_1-U)-\mathbf{n}\cdot k\nabla U\|_E\|e-\pi e\|_E \\
\leq\ &\sum_{E\in\partial K}\|\kappa_1(g_1-U)-\mathbf{n}\cdot k\nabla U\|_E\|e-\pi e\|_{\partial K} \\
\leq\ &\sum_{E\in\partial K}\|\kappa_1(g_1-U)-\mathbf{n}\cdot k\nabla U\|_E C\big(h_K^{-1}\|e-\pi e\|_K^2+h_K\|D(e-\pi e)\|_K^2\big)^{1/2} \\
\leq\ &\sum_{E\in\partial K}\|\kappa_1(g_1-U)-\mathbf{n}\cdot k\nabla U\|_E Ch_E^{1/2}\|De\|_K \qquad (4.12)
\end{aligned}$$

Based on inequalities (4.10), (4.11) and (4.12), we can get

$$\begin{aligned}
& \alpha \cdot \big(\|\nabla e\|^2_{L^2(\Omega)} + \|e\|^2_{L^2(\partial\Omega)}\big) \\
\leq\ & \sum_{K\in\mathcal{K}} \big(\|\sigma(u^*)\cdot|\nabla\phi^*|^2 + \nabla\cdot k\nabla U\|_K C h_K \|De\|_K \\
& +\|[\mathbf{n}\cdot k\nabla U]/2\|_{\partial K} C h_K^{1/2}\|De\|_K\big) \\
& + \sum_{E\in\partial\mathcal{K}} \|\kappa_1(g_1-U) - \mathbf{n}\cdot k\nabla U\|_E C h_E^{1/2}\|De\|_K \\
\leq\ & C\Big(\big(\sum_{K\in\mathcal{K}} h_K^2\|\sigma(u^*)\cdot|\nabla\phi^*|^2 + \nabla\cdot k\nabla U\|^2_K \\
& + h_K\|[\mathbf{n}\cdot k\nabla U]/2\|^2_{\partial K}\big)^{1/2}\big(\sum_{K\in\mathcal{K}}\|De\|^2_K\big)^{1/2} \\
& +\big(\sum_{E\in\partial\mathcal{K}} h_E\|\kappa_1(g_1-U)-\mathbf{n}\cdot k\nabla U\|^2_E\big)^{1/2}\big(\sum_{K\in\mathcal{K}}\|De\|^2_K\big)^{1/2}\Big) \\
\leq\ & C\Big(\big(\sum_{K\in\mathcal{K}} h_K^2\|\sigma(u^*)\cdot|\nabla\phi^*|^2 + \nabla\cdot k\nabla U\|^2_K \\
& + h_K\|[\mathbf{n}\cdot k\nabla U]/2\|^2_{\partial K}\big)^{1/2}\cdot\|De\|_{L^2(\Omega)} \\
& +\big(\sum_{E\in\partial\mathcal{K}} h_E\|\kappa_1(g_1-U)-\mathbf{n}\cdot k\nabla U\|^2_E\big)^{1/2}\cdot\|De\|_{L^2(\Omega)}\Big) \\
\leq\ & C\Big(\big(\sum_{K\in\mathcal{K}} h_K^2\|\sigma(u^*)\cdot|\nabla\phi^*|^2 + \nabla\cdot k\nabla U\|^2_K + h_K\|[\mathbf{n}\cdot k\nabla U/2]\|^2_{\partial K}\big)^{1/2} \\
& +\big(\sum_{E\in\partial\mathcal{K}} h_E\|\kappa_1(g_1-U)-\mathbf{n}\cdot k\nabla U\|^2_E\big)^{1/2}\Big)\cdot\big(\|\nabla e\|_{L^2(\Omega)} + \|e\|_{L^2(\partial\Omega)}\big)
\end{aligned}$$

where C is a constant.

By dividing $\|\nabla e\|_{L^2(\Omega)} + \|e\|_{L^2(\partial\Omega)}$ and taking squares on both sides, the inequality (4.6) can be proved directly. □

Next, we consider the potential problem defined by (4.3) – (4.5). We have a posteriori estimate:

Theorem 4.2. *For the finite element approximation Φ of the exact solution ϕ to (4.3), (4.4) and (4.5) with a given function $\sigma(u^*)$, the following a posteriori error estimate holds*

$$\|\nabla(\phi-\Phi)\|^2_{L^2(\Omega)} + \|\phi-\Phi\|^2_{L^2(\partial\Omega)} \leq C\big(\sum_{K\in\mathcal{K}}\eta^2_K(\Phi,u^*) + \sum_{E\in\partial\mathcal{K}}\eta^2_E(\Phi,u^*)\big) \tag{4.13}$$

where C is a constant, the element residual in the interior domain $\eta_K(\Phi, u^*)$ is defined by

$$\eta_K(\Phi, u^*) = h_K \|f + \nabla \cdot \sigma(u^*)\nabla\Phi\|_{L^2(K)} + \frac{1}{2} h_K^{1/2} \|[\mathbf{n} \cdot \sigma(u^*)\nabla\Phi]\|_{L^2(\partial K \setminus \partial\Omega)} \tag{4.14}$$

and the element residual on the boundary $\rho_E(U)$ is defined by

$$\eta_E(\Phi, u^*) = h_E^{1/2} \|\kappa_2(g_3 - \Phi) - \mathbf{n} \cdot \sigma(u^*)\nabla\Phi\|_{L^2(E\cap\partial\Omega)}. \tag{4.15}$$

Proof. Let $e = \phi - \Phi$ be the error. Assume $\sigma(u^*), \kappa_2 \geq \beta > 0$, then we have

$$\begin{aligned} & \beta \cdot \left(\|\nabla e\|^2_{L^2(\Omega)} + \|e\|^2_{L^2(\partial\Omega)}\right) \\ \leq & \int_\Omega \sigma(u^*)\nabla e \cdot \nabla e dx + \int_{\Gamma_N} \kappa_2 e \cdot e ds \\ = & \int_\Omega \sigma(u^*)\nabla e \cdot \nabla(e - \pi e) dx + \int_{\Gamma_N} \kappa_2 e \cdot (e - \pi e) ds \end{aligned}$$

where we have used the Gelerkin orthogonality (see Theorem 3.2) to subtract the interpolant πe. The rest procedure of this proof is similar to Theorem 4.1. □

In both Theorem 4.1 and 4.2, we have assumed that u^* and ϕ^* are known. In this paper, the model problem is a multiphysical problem which is constructed by both heat and potential equations. That means u^* and ϕ^* are the exact solution of the model problem. Hence, in the numerical method of this multiphysical problem, u^* and ϕ^* are replaced by their numerical solution U and Φ. It makes us to derive a posteriori estimation more difficultly. However, we can state a conjecture encouraged by Proposition 2.1 in [3].

Conjecture 4.3. *For the finite element approximation U and Φ of the exact solution u and ϕ to the multiphysical problem defined by* (4.1) – (4.5)*, the following a posteriori error estimate holds*

$$\begin{aligned} \|\nabla(u - U)\|^2_{L^2(\Omega)} \quad + & \quad \|\nabla(\phi - \Phi)\|^2_{L^2(\Omega)} + \|u - U\|^2_{L^2(\partial\Omega)} + \|\phi - \Phi\|^2_{L^2(\partial\Omega)} \\ & \leq C\big(\sum_{K\in\mathcal{K}} \mu_K^2(U, \Phi) + \sum_{E\in\partial\mathcal{K}} \mu_E^2(U, \Phi)\big) \end{aligned} \tag{4.16}$$

where C is a constant and the element residuals in the interior domain $\mu_K(U,\Phi)$ is defined by

$$\begin{aligned}\mu_K(U,\Phi) &= h_K\|\sigma(U)\cdot|\nabla\Phi|^2+\nabla\cdot k\nabla U\|_{L^2(K)}+\frac{1}{2}h_K^{1/2}\|[\mathbf{n}\cdot k\nabla U]\|_{L^2(\partial K\setminus\partial\Omega)}\\ &\quad +h_K\|f+\nabla\cdot\sigma(U)\nabla\Phi\|_{L^2(K)}+\frac{1}{2}h_K^{1/2}\|[\mathbf{n}\cdot\sigma(U)\nabla\Phi]\|_{L^2(\partial K\setminus\partial\Omega)}\end{aligned}\tag{4.17}$$

and the element residuals on the boundary $\mu_E(U,\Phi)$ is defined by

$$\begin{aligned}\mu_E(U,\Phi) &= h_E^{1/2}\|\kappa_1(g_1-U)-\mathbf{n}\cdot k\nabla U\|_{L^2(E\cap\partial\Omega)}\\ &\quad +h_E^{1/2}\|\kappa_2(g_3-\Phi)-\mathbf{n}\cdot\sigma(U)\nabla\Phi\|_{L^2(E\cap\partial\Omega)}\end{aligned}\tag{4.18}$$

4.2 Adaptive Mesh Refinement

When constructing a refinement algorithm, two important issues need to be considered:

- Invalid triangles (e.g with hanging nodes): are not allowed and we wish to refine as few elements as possible which are not in the list of elements to be refined;
- Minimal angle: is kept as large as possible.

Here we will introduce two methods: *Regular* refinement and *Rivara* refinement. See Figures 4.1. The first method consists of splitting each triangle into four smaller ones. In the second method a triangle is always refined by inserting a new edge from the midpoint of the longest edge to the opposite corner.

In MATLAB, a geometry of a given domain can be created by a mesh generator included in PDE toolbox. We call it **geom**. A mesh or triangulation on the domain **geom** is stored by three matrices: the point matrix **p**, the connectivity matrix **t** and the edge matrix **e** which contains the node numbers of the triangle edges making up the boundary of the mesh. The command **refinemesh** refines the mesh with a list of triangles to be refined. The refinement method is either *Rivara* or *Regular* refinement. The

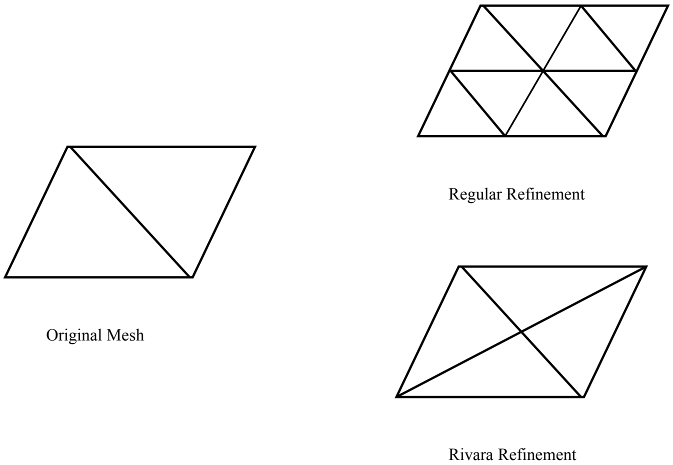

Figure 4.1: Refinement algorithms: Rivara and Regular

command **pdejmps** calculates the error indicators $R(K)$ of the differential equation $-\nabla\cdot(c\nabla u)+au=f$ for each triangle on a given mesh by equation

$$R(K)=\alpha\|h^m(f-aU)\|_K+\beta(\frac{1}{2}\sum_{\tau\in\partial K}h_\tau^{2m}[\mathbf{n}_\tau\cdot c\nabla U]^2)^{1/2}$$

where $\mathbf{n}_\tau$ is the unit normal of the edge τ and the braced term is the jump on an interior edge, α and β are weight indices and m is an order parameter.

In our code, we calculate indicators $\big($as defined by (4.17) $\big)$ on the mesh by **pdejmps** with $a=0$. And error residuals on the boundary defined by equations (4.18) is computed by a function called **ResidualOnBoundary**. Finally a pseudocode of this new refinement algorithm based on a posteriori error estimate defined in the previous section can be written as:

```
Algorithm 4.2.1: ADAPTIVEMESHREFINEMENT($geom, p, e, t$)

while $size(t, 2) < maximum\_degrees\_of\_freedom$
  do
    Compute $U$ and $\Phi$ with iterative method;
    comment: Calculate the element residual on mesh
    $RH \leftarrow$ PDEJMPS($p, e, t, k(x), 0, \sigma(U) \cdot |\nabla\Phi|^2, U, 1, 1, 1$);
    $RP \leftarrow$ PDEJMPS($p, e, t, \sigma(U), 0, f, \Phi, 1, 1, 1$);
    comment: Calculate the element residual on boundary
    $RB \leftarrow$ RESIDUALONBOUNDARY($p, e, t, U, \Phi$);
    $R \leftarrow RH + RP + RB$;
    Find elements that have to be refined;
    $[p, e, t] \leftarrow$ REFINEMESH($geom, p, e, t, elements$);
return $(p, e, t)$
```

Chapter 5

Numerical Examples

5.1 The Model Description

An example for the Joule heating problem reads as

$$\begin{aligned}
-\nabla \cdot k\nabla u &= \sigma(u) \cdot |\nabla\phi|^2 && \text{in } \Omega, \\
\mathbf{n} \cdot k\nabla u &= \kappa_1(g_1 - u) && \text{on } \partial\Omega, \\
-\nabla \cdot \sigma(u)\nabla\phi &= f && \text{in } \Omega, \\
\phi &= g_2 && \text{on } \Gamma_1, \\
\phi &= g_3 && \text{on } \Gamma_2, \\
\mathbf{n} \cdot \sigma(u)\nabla\phi &= \kappa_2(g_4 - \phi) && \text{on } \Gamma_3.
\end{aligned}$$

The domain Ω is a U-shape. See in Figure 5.1. Its boundary $\partial\Omega$ contains three parts (i.e. $\partial\Omega = \Gamma_1 \cup \Gamma_2 \cup \Gamma_3$) : Γ_1 and Γ_2 are showed in the picture with bold lines, and the rest of boundary is called Γ_3 . Then we let $f = 0, k = 1, \kappa_1 = 1, \kappa_2 = 0, g_1 = g_3 = g_4 = 0, g_2 = 1$, and choose $\alpha = 1, \beta = 1, m = 1$ in **pdejmps**.

In the following, there are two kinds of numerical implementations: one is about the convergences of different iterative methods, and the other one is about adaptive finite elements.

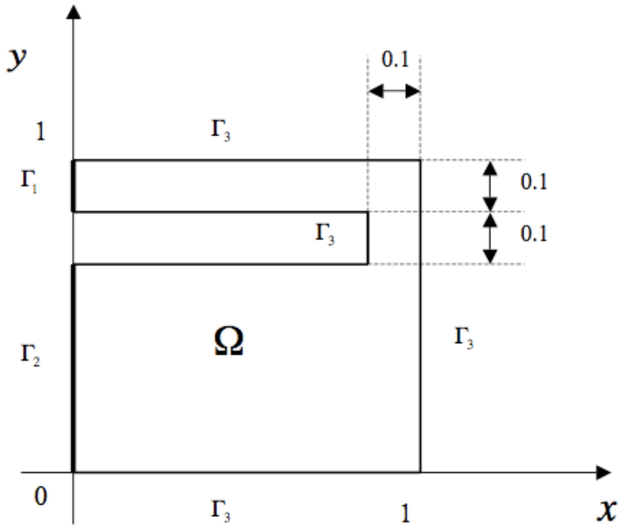

Figure 5.1: U-shape

5.2 Convergences of Iterative Solution-techniques for Different Materials

In the section 2.3, we have introduced three types of materials used in MEMS applications based on the different physical properties of electrical conductivities. Hence, there are three different choices of $\sigma(u)$ in the example problem. On the other hand, when we analyzed the numerical method of the multiphysical problem, we described four iterative techniques: Jacobi, GS, SOR and Jacobi with a relaxation factor. In this numerical implementation, we are interested in which iterative method will converge fast for these three materials.

According to (2.7), (2.8) and (2.9), let the electrical conductivity of a metal be defined by

$$\sigma_1(u) = \begin{cases} 100 & u \leq 0 \\ \frac{100}{1+u} & u > 0 \end{cases}$$

and $\sigma(u)$ of a semiconductor is given by

$$\sigma_2(u) = \begin{cases} 10 & u \leq 0 \\ 10\,(1+u) & u > 0. \end{cases}$$

For a superconductor, we can not implement $\sigma = \infty$ as $u < u'$ directly in our code. Hence we need to use a large value instead. In order to keep the function continuous, we introduce a factor ε. Then we get

$$\sigma_3(u) = \begin{cases} \frac{\sigma'}{\varepsilon} & u \leq u' + \varepsilon \\ \frac{\sigma'}{u-u'} & u > u' + \varepsilon \end{cases}$$

where $0 < \varepsilon < 1$. In our code, $\varepsilon = 0.1$, $u' = 1$ and $\sigma' = 10^5$ for example. Figures 5.2, 5.3 and 5.4 present the variations of σ_1, σ_2 and σ_3 as the temperature changes from 0 to 50.

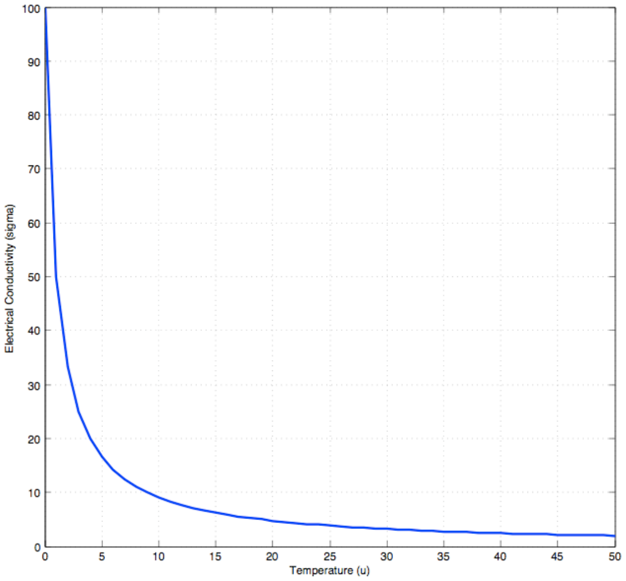

Figure 5.2: Electrical conductivity of a metal

Since we use a relaxation factor ω for both Jacobi and GS methods, we present results in two groups: one is Jacobi methods with variable choices of ω and the other group is GS methods with different ω. Figure 5.5 and 5.6 illuminate numbers of iterations for different iterative procedures for metal

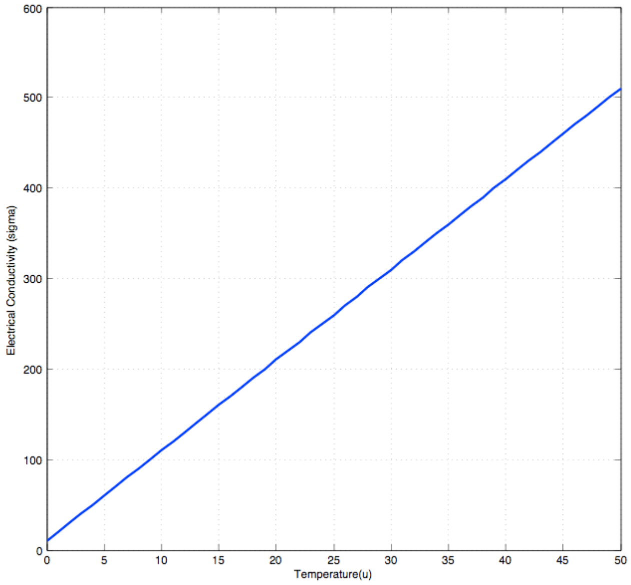

Figure 5.3: Electrical conductivity of a semiconductor

and semiconductor. For the metal, the most fast process uses 18 iterations as $\omega = 0.9$ with GS method. And it is 9 iterations for the semiconductor at best with GS method (here, $\omega = 1$). Figure 5.7 shows the result of the super conductor with the factor $\varepsilon = 0.1$. It takes at least 69 iterations to converge when $\omega = 0.8$ with GS method. However, we are also interested in how the different ε influences the convergence of the iterative procedures. Table 5.1 presents numbers of iterations of GS methods with different ω on the semiconductor with variable ε. The numbers of iterations depend very weakly on ε. However when ε becomes smaller, we see a slight increase in the numbers of iterations. And they always converge most quickly when $\omega = 0.8$. When we use Jacobi methods with different ε, it shows the same result. Comparing three materials, we can conclude that the electrical conductivity of a superconductor has a higher cost of convergence and its behavior makes our numerical methods more sensitive and we need to choose iterative methods more carefully, since there is only a small range of ω to be valid.

The following table 5.2 summaries the ranges of ω when Jacobi and GS methods converge for each material within 500 iterations. Otherwise, for

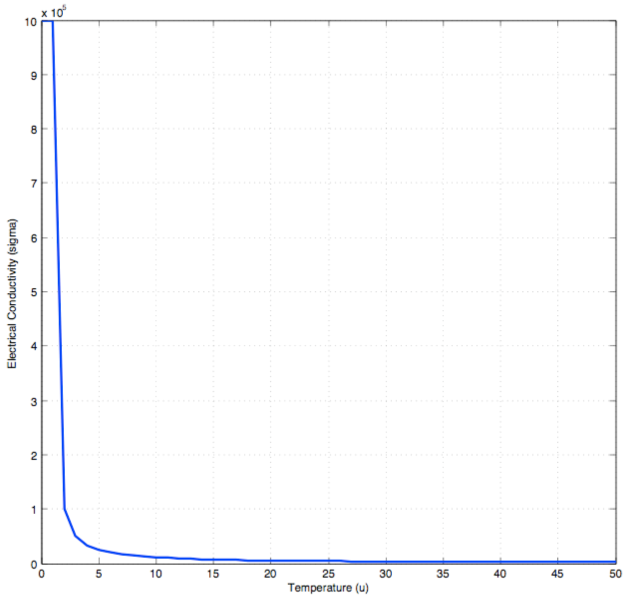

Figure 5.4: Electrical conductivity of a superconductor

larger value of ω these methods diverge and for smaller value of ω these methods do not converge within 500 iterations. From the table, we can conclude that GS method with the relaxation factor is a more reliable iterative method to get convergence for all materials.

5.3 Adaptive Finite Elements

In this section, the adaptive mesh refinement Algorithm 4.2.1 is implemented on the example problem but only on the metal and the semiconductor with σ_1 and σ_2 defined in section 5.2. Because it is more difficult to test our new algorithm on the superconductor based on the conclusion in the previous section, we do not consider it in this implementation.

We choose the fastest iterative solution-techniques for two refinement procedures since we have already measured in last implementation and assume 35% of triangles are worse in each refinement process. For the metal, Figure 5.8 shows the final refined mesh. Figures 5.9 and 5.10 are the finite element approximations U and Φ on the final refined mesh.

Table 5.1: Numbers of Iterations of GS methods with different ω on the semiconductor with variable ε

ε	$\omega = 0.5$	$\omega = 0.7$	$\omega = 0.8$	$\omega = 0.9$
0.1	146	88	69	88
0.01	149	90	70	89
0.001	153	92	72	90
0.0001	156	94	73	91

Table 5.2: Ranges of ω as Jacobi and GS methods converge for each material

	Jacobi	GS
Metal	$0.4 \leq \omega \leq 0.9$	$0.2 \leq \omega \leq 1.2$
Semiconductor	$0.2 \leq \omega \leq 1.5$	$0.2 \leq \omega \leq 1.3$
Superconductor	$\omega = 0.7$	$0.5 \leq \omega \leq 0.9$

Since the exact solutions are not known we instead compute a reference solution on a fine mesh in order to evaluate the solutions on coarser meshes. Computing the energy norm $En(i)$ of the solution U_i and Φ_i by

$$En(i) = U_i^T \cdot A_i^1 \cdot U_i + \Phi_i^T \cdot A_i^2 \cdot \Phi_i \tag{5.1}$$

where U_i and Φ_i are the finite element approximations on the mesh which is adapted by i^{th} refinement process; A_i^1 and A_i^2 are the stiffness matrices corresponding to U_i and Φ_i respectively. Let us call the total times of refinement processes is I. Hence $En(I)$ is the reference solution. Then the relative error energy norm can be calculated by $\frac{|(En(i)-En(I))|}{En(i)}$ for $i = 1, \ldots, I-1$. Figure 5.11 shows these results and values on *x-coordiate* are numbers of nodes on the mesh. The results for the semiconductor are illuminated by Figures 5.12 – 5.15.

From Figures 5.11 and 5.15, we can conclude that the new adaptive algorithm based on a posteriori error estimate we stated in chapter 4 seems to give good results for the example problem. The error decreases in each

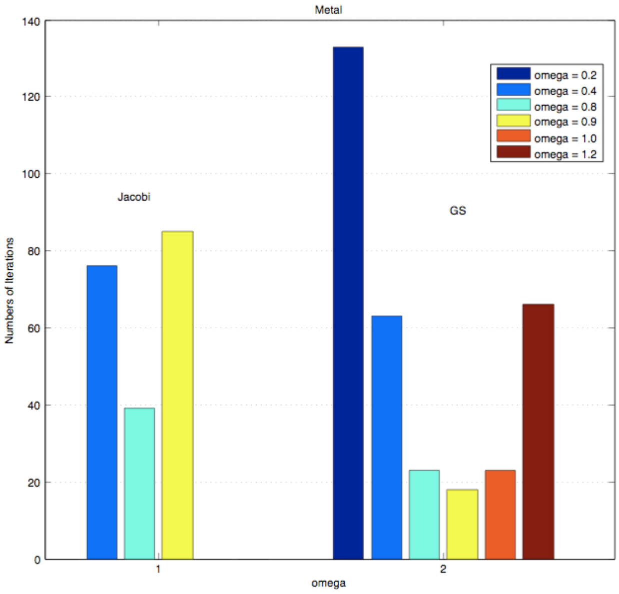

Figure 5.5: Numbers of iterations for the metal with different ω in Jacobi and GS methods

refinement on both examples.

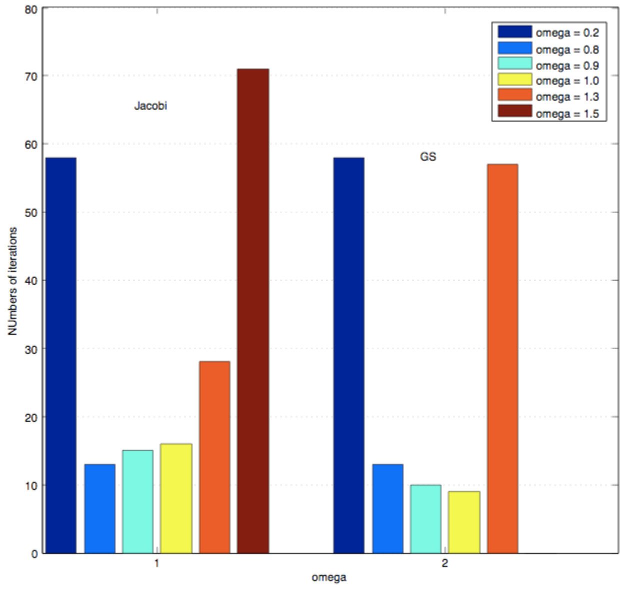

Figure 5.6: Numbers of iterations for the semiconductor with different ω in Jacobi and GS methods

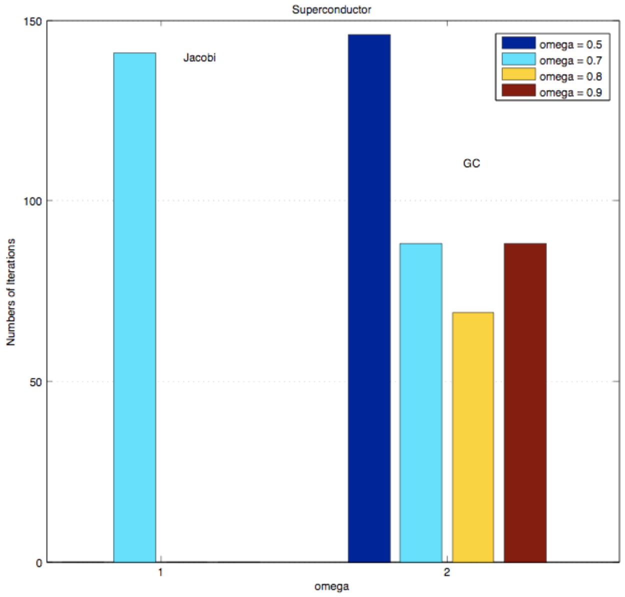

Figure 5.7: Numbers of iterations for the semiconductor with different ω in Jacobi and GS methods

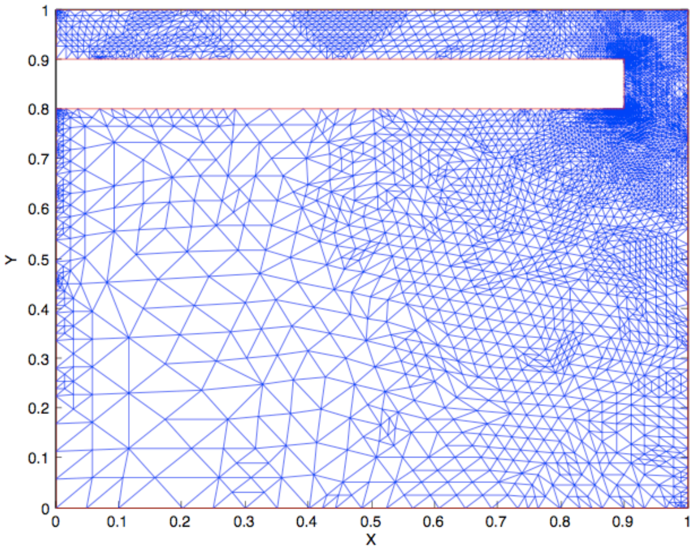

Figure 5.8: Final refined mesh on the metal

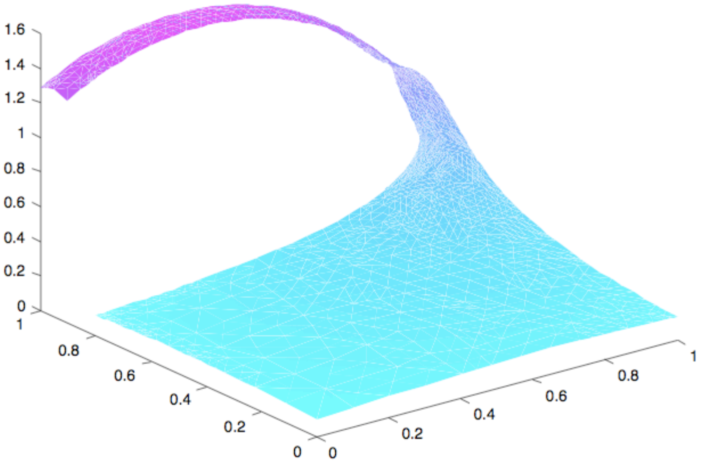

Figure 5.9: Finite element approximation U on the metal

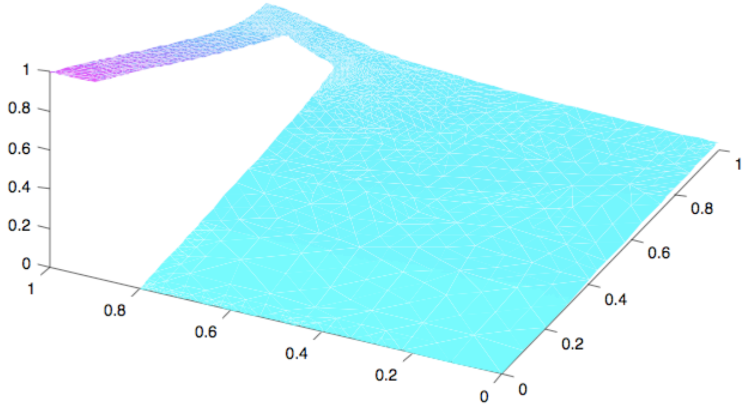

Figure 5.10: Finite element approximation Φ on the metal

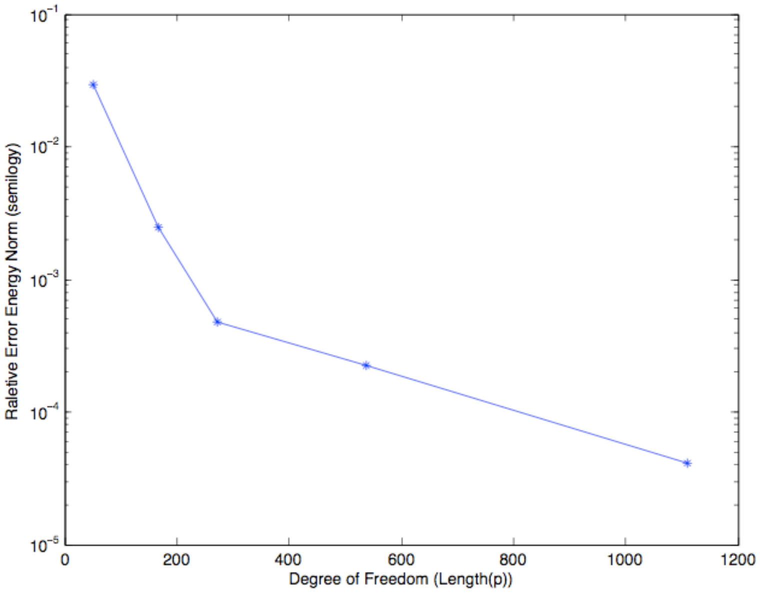

Figure 5.11: Relative error energy norm for the metal

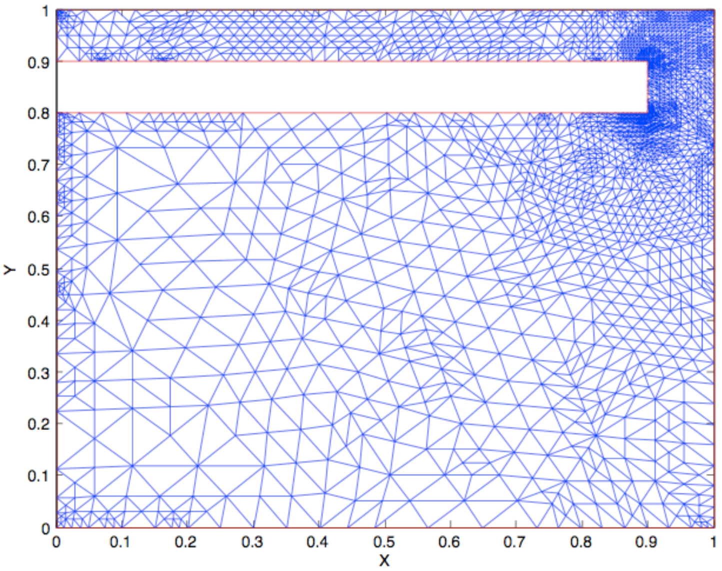

Figure 5.12: Final refined mesh on the semiconductor

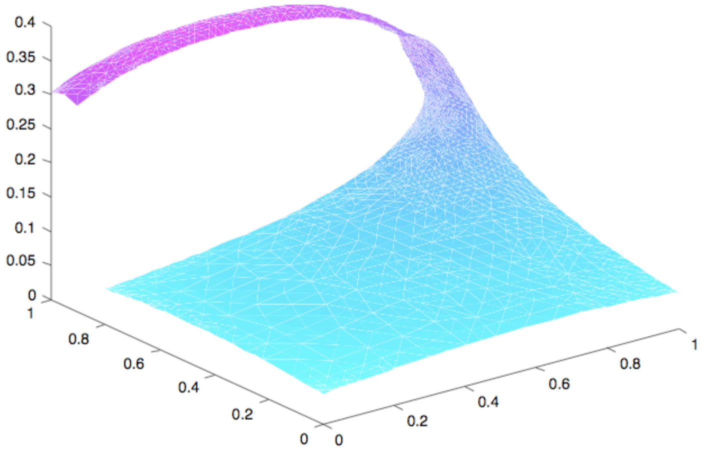

Figure 5.13: Finite element approximation U on the semiconductor

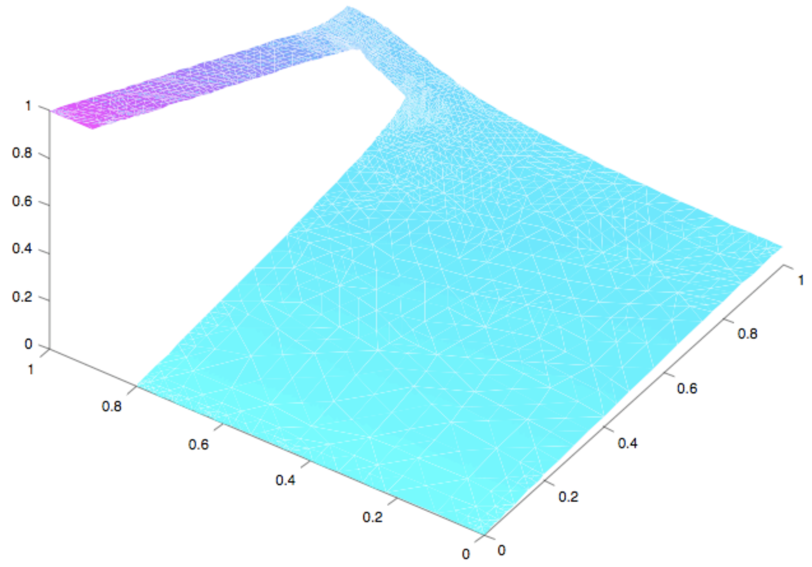

Figure 5.14: Finite element approximation Φ on the semicondctor

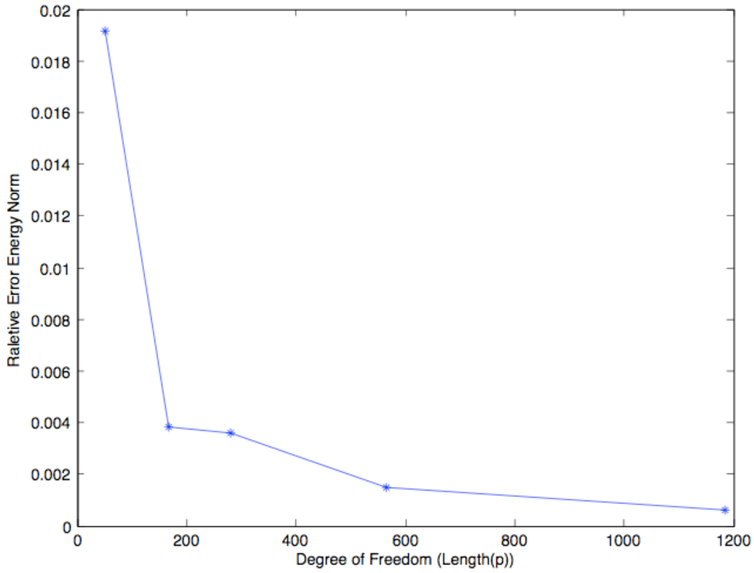

Figure 5.15: Relative error energy norm for the semiconductor

Chapter 6

Conclusion

We consider the Joule heating problem. In the first two chapters we introduce the problem and the physical background. In the third chapter, we derive a finite element method for the model problem and describe implementation details. Since the problem is non-linear we use iterative methods to solve the problem. In the following chapter, we analyze a posteriori estimate for each of single physical problems, then we derive a conjecture for the model problem. Based on this result we derive a new adaptive mesh refinement algorithm. Finally, we present two numerical examples. In the first one we focus on convergences of different iterative methods for different materials. These methods are Jacobi, GS and SOR. We also use a relaxation factor in Jacobi to improve its performance. Three materials have been tested: metals, semiconductors and superconductors. In the second example we implement the adaptive algorithm based on the description in Chapter 4.

Finally we get three conclusions: (1) By adjusting the relaxation factor ω, iterations procedures can converge for all three materials. Gauss-Seidel method with the relaxation factor shows a higher reliability. (2) The convergence of iterations for superconductors seems more difficult and sensitive. (3) The adaptive algorithm produces good results and the error decays at each refinement.

Appendix A

Table A.1: The area coordinates and weights for one-point, three-point, four-point and six-point Gauss quadrature rules over triangular elements

Degree	Points	Area Coordinate	Weights
		$(L_1^i,\, L_2^i,\, L_3^i)$	
One	a	$\frac{1}{3},\, \frac{1}{3},\, \frac{1}{3}$	1
Two	a	$\frac{1}{2},\, \frac{1}{2},\, 0$	$\frac{1}{3}$
	b	$0,\, \frac{1}{2},\, \frac{1}{2}$	$\frac{1}{3}$
	c	$\frac{1}{2},\, 0,\, \frac{1}{3}$	$\frac{1}{3}$
Three	a	$\frac{1}{3},\, \frac{1}{3},\, \frac{1}{3}$	$-\frac{27}{48}$
	b	0.6, 0.2, 0.2	$\frac{25}{48}$
	c	0.2, 0.6, 0.2	$\frac{25}{48}$
	d	0.2, 0.2, 0.6	$\frac{25}{48}$
Four	a	$p_1,\, p_2,\, p_2$	w_1
	b	$p_2,\, p_2,\, p_1$	w_1
	c	$p_2,\, p_1,\, p_2$	w_1
	d	$p_3,\, p_4,\, p_4$	w_2
	e	$p_4,\, p_4,\, p_3$	w_2
	f	$p_4,\, p_3,\, p_4$	w_2
	where	$p_1 = 0.108103018168070$	
		$p_2 = 0.445948490915965$	$w_1 = 0.223381589678011$
		$p_3 = 0.816847572980459$	$w_2 = 0.109951743655322$
		$p_4 = 0.091576213509771$	

Table A.2: The area coordinates and weights for one-point, two-point, three-point and four-point Gauss quadrature rules on edges

Degree	Points	Area Coordinate $(L_1^i,\, L_2^i)$	Weights
One	a	$\frac{1}{2},\, \frac{1}{2}$	1
Three	a	$\frac{1}{2} + \frac{1}{2\sqrt{3}},\, \frac{1}{2} - \frac{1}{2\sqrt{3}}$	$\frac{1}{2}$
	b	$\frac{1}{2} - \frac{1}{2\sqrt{3}},\, \frac{1}{2} + \frac{1}{2\sqrt{3}}$	$\frac{1}{2}$
Five	a	$\frac{1+\sqrt{3/5}}{2},\, \frac{1-\sqrt{3/5}}{2}$	$\frac{5}{18}$
	b	$\frac{1}{2},\, \frac{1}{2}$	$\frac{8}{18}$
	c	$\frac{1-\sqrt{3/5}}{2},\, \frac{1+\sqrt{3/5}}{2}$	$\frac{5}{18}$
Seven	a	$\frac{1+p_1}{2},\, \frac{1-p_1}{2}$	$\frac{w_1}{2}$
	b	$\frac{1+p_2}{2},\, \frac{1-p_2}{2}$	$\frac{w_2}{2}$
	c	$\frac{1-p_2}{2},\, \frac{1+p_2}{2}$	$\frac{w_2}{2}$
	d	$\frac{1-p_1}{2},\, \frac{1+p_1}{2}$	$\frac{w_1}{2}$
	where	$p_1 = 0.861136311594053$	$w_1 = 0.347854845137454$
		$p_2 = 0.339981043584856$	$w_2 = 1 - w_1$

Bibliography

[1] M. G. Larson and F. Bengzon *A First Course in Finite Elements Lecture Notes*, The Computational Mathematics Laboratory, Department of Mathematics and Mathematical Statistics, Umeå University, 2007.

[2] S. C. Brenner and L. R. Scott *The Mathematical Theory of Finite Element Methods*, Second Edition, Springer, 2002.

[3] R. Verfürth *A Review of A Posteriori Error Estimation and Adaptive Mesh-refinement Techniques*, Wiley-Teubner, 1996.

[4] M. G. Larson and F. Bengzon *Adaptive Finite Element Approximation of Multiphysics Problems*, 2006.

[5] K. E. Atkinson *The Numerical Solution of Integral Equations of the Second Kind*, 1997.

[6] D. K. Cheng *Fundamentals of Engineering Electromagnetics*, 1993.

[7] J. N. Reddy and D. K. Gartling *The Finite Element Method in Heat Transfer and Fluid Dynamics*, Second Edition, 2000.

[8] O. C. Zienkiewicz and R. L. Taylor *The Finite Element Method*, Volume 1, The Basis, Fifth Edition, 2000.

[9] A. C. Rose-Innes and E. H. Rhoderick *Introduction to Superconductivity*, Second Edition, 1978.