UC Riverside
UC Riverside Electronic Theses and Dissertations

Title

FEM Based Multiphysics Analysis of Electromigration Voiding Process in Nanometer Integrated Circuits

Permalink

Author

Zhao, Hengyang

Publication Date

2018

UNIVERSITY OF CALIFORNIA
RIVERSIDE

FEM Based Multiphysics Analysis of Electromigration Voiding Process in
Nanometer Integrated Circuits

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Electrical Engineering

by

Hengyang Zhao

December 2018

Dissertation Committee:

    Dr. Sheldon Tan, Chairperson
    Dr. Daniel Wong
    Dr. Jianlin Liu

The Dissertation of Hengyang Zhao is approved:

_____

_____

_____
Committee Chairperson

University of California, Riverside

Acknowledgments

This thesis could not have been completed without the great support that I have received from so many people over the years. I wish to offer my most heartful thanks to the following people.

I would like to thank my advisor, Dr. Sheldon Tan for guiding and supporting me over the years. Dr. Tan is a great example of excellence as a researcher, a mentor, an instructor and most importantly a scholar persuing rigours theorems. His kindness, insight and suggestions always guide me to the right direction.

I would like to thank my committee members, Dr. Daniel Wong and Dr. Jianlin Liu for their direction, dedication and invaluable advice.

I would like to thank all the members in our VSCLAB, especially Chase Cook, Haibao Chen, Han Zhou, Jinwei Zhang, Kai He, Lebo Wang, Shaoyi Peng, Sheriff Sadiqbatcha, Shuyuan Yu, Taeyoung Kim, Wentian Jin, Yan Zhu, Yue Zhao, and Zeyu Sun, for the collaborative research works, discussion and help, which lead to the presented work in this thesis. I appreciate the friendship of my fellow students in UCR.

I would like to thank my friend Daniel Quach, my first good friend in the United States, who took great efforts to help me to settle down and fit in the new Ph.D. life.

I would like to thank my wife Yunduan and my parents for the love, support, and constant encouragement during the years of my study. I undoubtedly could not have done this without them.

To my wife Duānduān for all the love and support.

ABSTRACT OF THE DISSERTATION

FEM Based Multiphysics Analysis of Electromigration Voiding Process in Nanometer
Integrated Circuits

by

Hengyang Zhao

Doctor of Philosophy, Graduate Program in Electrical Engineering
University of California, Riverside, December 2018
Dr. Sheldon Tan, Chairperson

Failures introduced by the electromigration (EM) effect in copper interconnect is
one of the top reliability issues in modern integrated circuits (ICs) in 10 nm technology
and below. International Technology Roadmap for Semiconductors (ITRS) predicts that
the required current density for driving a normal gate will exceed the EM current density
limit in 2024 if the industry continues current technology scaling with existing interconnect
materials and EM design rules based on current density.

Because of EM effect, under extreme current densities, copper interconnect is
highly probable to fail over time, most of which is caused by void formation. The behavior
of void in copper interconnect, which includes void generation, growth, migration, merging,
and vanishing, interacting very tightly with mechanical hydrostatic stress, is critical to the
EM reliability. Among the variety void behaviors, void growth and migration dominates
the impact on the consequence of open-circuit failures. The analysis of void growth and
migration and circuit failures thereby is considerably difficult because of the coupling of
a variety physical systems (temperature, electrical current density, void shape, and hydro-

static stress).

In this work, a multi-physics finite element method (FEM) based analysis method for void growth simulation of confined copper interconnects is proposed. The purpose is to facilitate the analysis of void behavior and EM-induced copper interconnect failures. The proposed method for the first time considers four important physics simultaneously in the EM failure process and their time-varying interactions: the hydrostatic stress in the confined interconnect wire, the current density, copper-void boundary evolution, and Joule heating induced temperature. A solver based on finite element method is proposed to solve the coupled systems in time-dependent manner. The proposed work ranges from deriving the partial differential equations governing the interaction of the hydrostatic stress and copper-void boundary evolution, to their variational form derivation, and finally to a software implementation.

The experiment data acquired from the software matches well with the behavior observed in real silicon experiments, which are in the aspects of hydrostatic stress and void shape evolution, final hydrostatic stress distribution, and Joule heating effect.

Contents

List of Figures

List of Tables

Listings

Chapter 1

Introduction

Electromigration (EM) is the top reliability concern for copper based back-end-of-line (BEOL) interconnects of current and future ICs in 10 nm technology and below. International Technology Roadmap for Semiconductors (ITRS) predicts that EM lifetime of interconnects in VLSI chips will be reduced by half for each generation of nodes [80] due to the increasing current density and shrinking wire line cross sectional area, which determine the critical size of EM effect.

1.1 Major Interconnect Reliability Issues

In modern integrated circuits design scaling down to nanometer scale, various effects have become more important in terms of their impacts on IC lifetime and failure patterns. Four major effects, including electromigration that will be focused on in the rest of this thesis, are introduced and discussed in the following.

1.1.1 Electromigration

Electromigration is the transport of material caused by the gradual movement of the ions in a conductor due to the momentum transfer between conducting electrons and diffusing metal atoms. The effect is important in applications where high direct current densities are used, such as in microelectronics and related structures. As the IC feature size decreases, the practical significance of this effect increases. [21]

Mechanisms of Electromigration Induced Failure

In a homogeneous crystalline structure, because of the uniform lattice structure of the metal ions, there is hardly any momentum transfer between the conduction electrons and the metal ions. However, this symmetry does not exist at the grain boundaries and material interfaces, and so here momentum is transferred much more vigorously. Since the metal ions in these regions are bonded more weakly than in a regular crystal lattice, once the electron wind has reached a certain strength, atoms become separated from the grain boundaries and are transported in the direction of the current. This direction is also influenced by the grain boundary itself, because atoms tend to move along grain boundaries.

Diffusion processes caused by electromigration can be divided into grain boundary diffusion, bulk diffusion and surface diffusion. In general, grain boundary diffusion is the major electromigration process in aluminum wires, whereas surface diffusion is dominant in copper interconnects.

In an ideal conductor, where atoms are arranged in a perfect lattice structure, the electrons moving through it would experience no collisions and electromigration would not

occur. In real conductors, defects in the lattice structure and the random thermal vibration of the atoms about their positions causes electrons to collide with the atoms and scatter, which is the source of electrical resistance (at least in metals; see electrical conduction). Normally, the amount of momentum imparted by the relatively low-mass electrons is not enough to permanently displace the atoms. However, in high-power situations (such as with the increasing current draw and decreasing wire sizes in modern VLSI microprocessors), if many electrons bombard the atoms with enough force to become significant, this will accelerate the process of electromigration by causing the atoms of the conductor to vibrate further from their ideal lattice positions, increasing the amount of electron scattering. High current density increases the number of electrons scattering against the atoms of the conductor, and hence the speed at which those atoms are displaced.

In integrated circuits, electromigration does not occur in semiconductors directly, but in the metal interconnects deposited onto them (see semiconductor device fabrication).

Electromigration is exacerbated by high current densities and the Joule heating of the conductor (see electrical resistance), and can lead to eventual failure of electrical components. Localized increase of current density is known as current crowding.

Electromigration Related Reliability Research

In modern dual damascene [40] process of copper interconnect manufacturing, failures induced by electromigration are observed by many works. Two major categories of such failures are open circuit failure caused by copper depletion [51, 33, 64, 58] as shown in figure 1.1, and short circuit failure caused by accumulated copper hillocks [26, 34, 76, 30] as shown in figure 1.2. Comparing to the short circuit failures caused by hillocks, work [35]

3
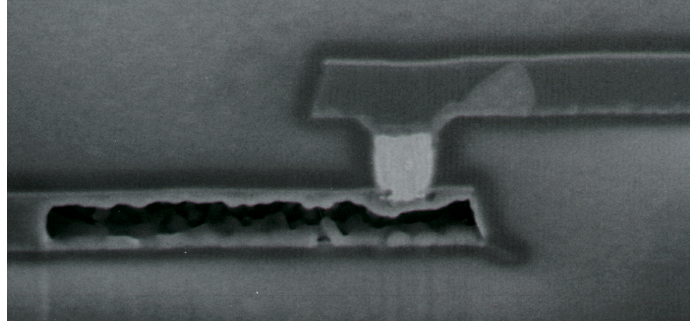
Figure 1.1: EM induced open circuit failure observed in [51].
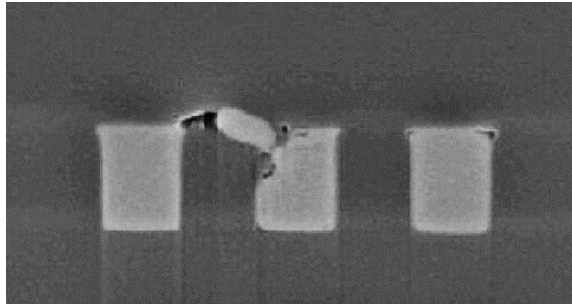
Figure 1.2: EM induced short circuit failure observed in [30].

pointed out that open circuit failures dominates the EM induced failures.

Because electromigration has severe impact on the life time of modern ICs, researchers have done a variety of modeling works on electromigration effects. In addition, to incorporate electromigration induced life time shortening at the design phase of chips, especially on the power/ground network design, different levels of EM life time coupling at design phase has been investigated. Work [75] modeled the interconnect as a many-resister network and applied linear programming to achieve fast optimization of EM lifetime. Work [67] used voltage drop as failure criterion and proposed a technique for calculating the stress distribution inside a multi-branch interconnect tree to avoid over optimistic failure prediction. Work [38] used a physics model to model EM lifetime for a given copper interconnect.

Work [14] proposed a fast stress calculation method for given time-varying temperature and current densities for the nucleation phase. Work [71] used a novel method to estimate the EM-induced stress in general multi-branch interconnects based on the terminal voltages. Work [36] applied a statistic model to perform full-chip EM lifetime evaluation. Work [37] proposed a dynamic EM model based on the direct analytical solution of one-dimensional Korhonen's equation [45] with load driven by any unipolar or bipolar current waveforms. Work [78] used a full-chip assessment algorithm to precisely determine the location and time of EM-induced void nucleation by directly checking the time-changing hydrostatic stresses of all the wires.

Many numerical methods are proposed in the past years to calculate the EM-induced hydrostatic stress and circuit life time change. Work [50] used a finite element approach to analysis the electromigration effects in confined interconnect under different microstructure. Work [47] used level set functions to model and simulate electromigration voiding process in 2D. Work [27] used a combined analytical and numerical solution scheme to calculate the atomic fluxes and the evolution of mechanical stress and avoided the difficulties associated with finite element based approaches. Work [7] used a phase field model to simulate electromigration and stress-induced void evolution in interconnect lines, in which a modified Cahn-Hilliard equation [23] is used. Work [61] presented a finite difference method to simulate of the electromigration-induced shape evolution of quasi-two-dimensional cylindrical voids in metallic thin films. Work [48] used fully coupled mechanical diffusion equations to model electromigration and presented methods to acquire numerical solutions. Work [12] proposed a simulated annealing based numerical method to extract the

diffusion and electromigration parameters for multi-phase intermetallic compounds formed as a result of material reactions between under bump metallization and solder joints. Work [49] studied a numerical simulation method for electromigration in IC device and solder joint in a package with different physics variables of high current density, thermal load, and mechanical load considered altogether. While this was not a direct study about electromigration of confined IC interconnect metal, it proposed a multiphysics simulator which is innovative to the work of this thesis. Work [22] applied finite element method to model electronmigration on copper domain and used a simulated solidification algorithm to generate random copper grains. Work [39] used ANSYS software to numerically simulate the void incubation and propagation, and gave the insights of the underlying electron wind force, stress gradients, temperature gradients, and atomic concentration gradient. Work [19] proposed an effective method based on finite difference method for the EM effects in multi-branch interconnects based on the kinetics of the first principle of EM physics.

1.1.2 Hot Carrier Injection

In IC reliability area, hot carrier injection (HCI) is a phenomenon in solid-state electronic devices, mostly metal-oxide-semiconductor field-effect transistors (MOSFETs), where an electrons at a site gain sufficient kinetic energy to overcome their potential barrier to be conducted through the dielectric [10]. *Hot* here denotes the effective temperature used to model carrier density. Since the charge carriers can become trapped in the gate dielectric of a MOS transistor, the switching characteristics of the transistor can be permanently changed. Hot-carrier injection is one of the mechanisms that adversely affects the reliability of MOSFETs in modern ICs.

1.1.3 Negative Bias Temperature Instability

Negative-bias temperature instability (NBTI) is a key reliability issue in MOS-FETs. NBTI manifests as an increase in the threshold voltage and consequent decrease in drain current and transconductance of a MOSFET [63, 62] . The degradation exhibits a power-law dependence on time. It is of immediate concern in p-channel MOS devices, since they almost always operate with negative gate-to-source voltage; however, the very same mechanism also affects nMOS transistors when biased in the accumulation regime, i.e. with a negative bias applied to the gate.

1.1.4 Time Dependent Dielectric Breakdown

Time-dependent dielectric breakdown (TDDB) is a failure mechanism in MOS-FETs, when the gate oxide breaks down as a result of long-time application of relatively low electric field (as opposed to immediate breakdown, which is caused by strong electric field). The breakdown is caused by formation of a conducting path through the gate oxide to substrate due to electron tunneling current, when MOSFETs are operated close to or beyond their specified operating voltages.

The defect generation in the dielectric is a stochastic process. There are two modes of breakdown, intrinsic and extrinsic. Intrinsic breakdown is caused by electrical stress induced defect generation. Extrinsic breakdown is caused by defects induced by the manufacturing process. For Integrated circuits, the time to breakdown is dependent on the thickness of the dielectric (gate oxide) and also on the material type, which is dependent on the manufacturing process node. Older generation products with gate oxide thickness ¿

4nm are based on silicon dioxide and the advanced process nodes with gate oxide ¡ 4nm are based on high-k dielectric materials [29].

The failure types for Integrated Circuit (IC) components follow the classic bath tub curve. There is infant mortality, which is decreasing failure rate typically due to manufacturing defects. A low constant failure rate which is random in nature. Wearout failures which are increasing failures due to aging semiconductor degradation mechanisms. TDDB is one of the intrinsic wear out failure mechanisms. Performance of the IC components can be evaluated for semiconductor wearout mechanisms including TDDB for any given operating conditions. The breakdown models mentioned above could be used to predict the time to fail for the component due to TDDB.

There are different breakdown models and thickness of the gate oxide determines the validity of the model. E model [53], 1/E model [52], and exponential model [72]are common models which depict the breakdown behavior.

1.2 Existing Works on Electromigration Modeling

As a result, EM induced reliability needs to be addressed at various design stages, even at system level to ensure reliability targets are met. However, conservative design rules based on worst cases (highest possible temperature and power consumption) and simple EM models such as Black's equation can lead to significant overdesign and 2X-3X enlarged guard bands [3]. Such conservative and overdesign rules, however, will hardly remain as an option in current and future technologies because a 3X guard band increase will significantly increase the buffer size and many other aspects of chips, which can lead

to increased currents, and hence, increased cost and chip power consumption. As a result, more accurate physics-based EM modeling and assessment techniques are critical for the EM-aware physical design and system level EM lifetime optimization [74].

The currently employed method of predicting time-to-failure (TTF) is based on the approximation and statistical methods such as Black's equation [8] and Blech's limit [9]. They are subject to growing criticism due to their over conservativeness and lack of considerations of multi-segment interconnect wires [36]. To mitigate this problem, a number of new physics-based EM modeling approaches have been proposed recently [69, 36, 68, 13, 15, 70, 18] based on solving the Korhonen's hydrostatic stress diffusion equation [45]:

$$\frac{\partial \sigma}{\partial t} = \nabla \cdot \left(\frac{D_a B}{k_B T} (\Omega \nabla \sigma - eZ\rho\vec{j}) \right) \quad \text{in } \Omega_L, \tag{1.1}$$

$$\nabla \sigma = \frac{eZ\rho\vec{j}_{N_i}}{\Omega} \quad \text{on } \partial\Omega_L \cap \Gamma_{N_i}, i = 1, ..., k, \tag{1.2}$$

$$\nabla \sigma = \frac{\sigma}{\delta} \quad \text{on } \partial\Omega_L \cap \Gamma_{\text{void}}, \tag{1.3}$$

where σ is hydrostatic stress, product eZ is effective charge of the migrating atoms, ρ is electrical resistivity, $\vec{j}$ is current density. k_B is the Boltzmann's constant, δ is the effective thickness of the copper-void boundary, T is the operating temperature, and E_a is the EM activation energy. Ω_L is the domain of simulated copper interconnect and $\partial\Omega_L$ denotes its boundary. Γ_{N_i} is the ith flux termination boundary, where normal current density j_{N_i} are prescribed, among k termination boundaries. Γ_{void} denotes the void boundary, which can be a union of several discrete voids. D_a is the atomic diffusion coefficient. D_a is given by $D_a = D_0 \text{exp}(-\frac{E_a}{k_B T})$ where D_0 is the diffusion coefficient. [1] Copper grain size is a key factor

---

[1]In fact, equation 1.1 can be written as

$$\frac{\partial \sigma}{\partial t} = \nabla \cdot \left(\frac{D_a B \Omega}{k_B T} \Omega \nabla \sigma \right) \quad \text{in } \Omega_L,$$

that affects EM lifetime [56]. Effective diffusion coefficient can be much higher in copper grain boundaries than grain bodies, which leads to different D_0 over the copper interconnect. In equation (1.1), a uniform D_0, synthesized from different body and boundary diffusion coefficients, is used. Note that Korhonen's hydrostatic diffusion equations can be applied to 3D multi-segment interconnect wires with multiple flux-termination boundary nodes, allowing any number of evolving voids to be simulated.

Equations (1.1) (1.2) and (1.3) can be solved by using finite element method (FEM) or finite difference method (FDM) in three dimensional cases, as well as one- or two-dimensional with certain level of abstraction. In void nucleation phase, under high current density, copper atoms are forced to slowly migrate along the electron wind direction towards the anode. Wire boundaries or terminals block such atom migration and consequently tensile stress will start to be built up at the cathode and compressive stress will develop at the anode. When the stress at the cathode reaches the critical stress, voids will form and start to grow and sometimes merge or migrate. We remark that when the compressive stress at the anode continues to be built up, hillocks or extrusion will form, which can potentially cause short-circuit failures. But the void nucleation is still the dominant EM failure effects [35].

However it is a notoriously difficult task to simulate the post-voiding phase due to several reasons. First, void shapes or their volumes and boundaries keep changing, which makes it more difficult to model this microstructural evolution process as it requires very fine

---

since in realistic copper interconnect, the electrical resistivity is constant and the current divergence is always zero:
$$\nabla \cdot (eZ\rho\vec{j}) = 0,$$
except for the interconnect terminals which are in contact with other materials such as liner metal.

meshed elements around the boundary. This needs dedicated adaptive re-meshing method in order to keep the system matrix relatively small. Second, the void volume (shapes) and the stress distributions of the remaining wire are correlated by the following atom conservation equation [44]

$$V_{\mathrm{S}}(t) = \frac{1}{B} \int_{\Omega_L} \sigma(V, t) dV, \quad \forall t, \tag{1.4}$$

where Ω_L is the volume of remaining interconnect, V_S is void volume, and B is the effective bulk modulus. However, most of the existing post-void simulation methods fail to consider this void volume conservation constraint.

Method in [68] simplifies the problem into one-dimensional case and only considers the stress distribution without considering the void volume change and its impact on the stress distribution. Bhate *et al* [6] proposed a 2D finite element method (FEM) based void growth analysis method by considering stress and void growth using coupled diffusion equations. In their method, Cahn-Hilliard equations are coupled to model the voids with shapes changing over time. In this method, the stress condition around the void (which should be zero) is not properly modeled. Also, it is not clear whether the conservation in (1.4) is satisfied or not.

1.3 Contributions of This Thesis

In this work, we try to mitigate the aforementioned problems. We propose a novel finite element method (FEM) based post-voiding stress simulation technique for confined copper interconnect wires. The new method can be applied for 1D, 2D, and 3D stress analysis of the wire for different trade-offs between accuracy and speed. It provides many

useful insights of the EM wear-out and failure process for multi-segment interconnects, which can be used to guide more compact system EM modeling and assessment methods.

The new method can perform stress analysis for both void nucleation and void growth process. But in this article, we focus on the post-voiding process, which is difficult to simulate due to the interaction between the void shape and stress development. Different from most of the existing EM analysis methods, which only focuses on stress evolution in the wires during the void growth phase, the new method explicitly considers the interplay between void growth and evolving stress distribution in the remaining interconnect, with physics-based conservation constraint taken into account. To this line, we introduce a separate phase field to model the void boundary evolution process, which leads to two coupled partial differential equations (in addition to the dynamic stress field partial differential equation) to describe the post-voiding stress evolution process. To consider the inherent physical interaction between the void volume and stress distributions in a confined interconnect wire, a novel regulated void growth velocity is introduced, which ensures that the void volume is always consistent with the stress distribution on the wire all the time. The proposed FEM-based method models all the three regions varying over time: the void, the void boundary and rest of the wires based on different stress governing equations and boundary conditions. The numerical results show that the calculated stress distribution and extracted current exponent fits the published measured results better than a recent proposed post-voiding analysis method [68]. Furthermore, the Joule heating effect is also considered in a simulation of upstream-configured copper interconnect where temperature and resistance jump happens when void grows to the critical volume. The proposed method can predict

12

the unique transient wire resistance change pattern for copper interconnect wires, which were well observed by the published experimental data.

In addition, we present multi-physics finite element method based solution for void growth simulation of confined copper interconnects for EM failure analysis. Our contributions lies in the following aspects: First, for the first time, three important physics and their transient interactions are considered simultaneously in the new method: the hydrostatic stress in the confined interconnect wire, the current density and Joule heating induced temperature. Second, we end up solving a coupling partial differential equations which consists of stress differential equation (Korhonen's equation), the phase field equation (for modeling void boundary move), the Laplace equation for current density and the heat diffusion equation for Joule heating and wire temperature. We show that each of the physics will have different physical domains and corresponding boundary conditions, and how such coupled multi-physic FEM transient analysis was carried out, and how different time scales are properly handled. Last but not least, numerical results show that by considering all three coupled physics — the stress, current density, and temperature — and their transient behaviors, the proposed FEM post-void EM solver can predict the unique transient wire resistance change pattern in which resistance jump happens when void grows to the critical volume for copper interconnect, which were well observed by published experiment data. We also show that the simulated void growth speed is less conservative than a recently proposed compact EM model.

1.4　Thesis Organization

This thesis is organized as the following to propose the FEM based multiphysics electromigration voiding analyzer as a whole, complete work. Chapter 2 formalize the electromigration problem as a coupled system of four physics systems, for both time dependent and stationary analysis. Chapter 3 derives the weak form introduced in chapter 2 providing a bridge between the math forms and the software implementation. Chapter 4 describes software implementation of the proposed analyzer, as well as applied simplifications. Chapter 5 shows results acquired from the proposed electromigration voiding analyzer, which are compared against existing modeling works and observations from real-silicon tests, as a validation of the effectiveness of proposed method.

Chapter 2

Problem Formulation

Electromigration analysis includes not only the hydrostatic stress, but also the consequent distribution change of temperature, current density, and most importantly how the void behave and impact on the potential failures. This chapter will introduce the coupled physics systems that will be solved later in order to acquire comprehensive analysis of EM lifetime, heating effect, resistance change, etc. In this chapter, all the physics systems of temperature, current density, hydrostatic stress, and void boundary will be described in their corresponding mathematical forms for both time dependent and stationary uses.

2.1   Coupled Physics Systems

In this section, we present the multi-physics simulation problem that we are facing to model and characterize the post-voiding process in a confined copper interconnect. Figure 2.1 shows an *up-stream* interconnect wire structure in which the electron flux is flowing from the bottom metal wire (metal 1 or M1) through the via and into the upper metal (metal 2
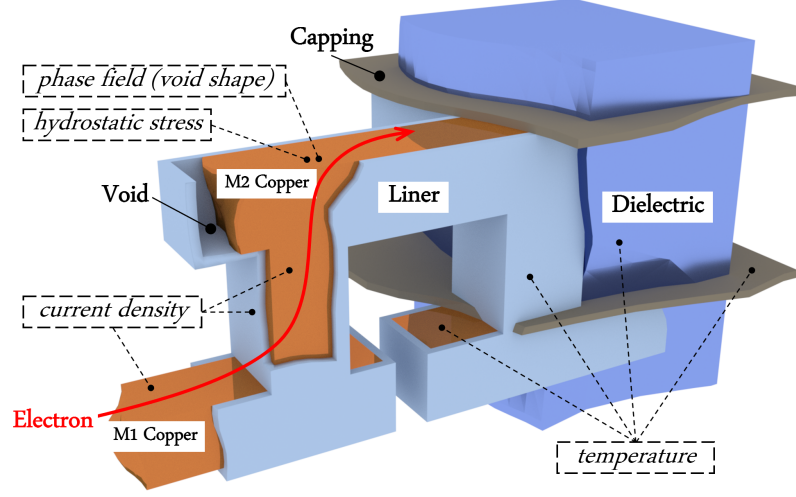
Figure 2.1: 3D illustration of up-stream interconnect structure and simulated physical systems.

or M2).

As we can see, basically we have three physics involved and three overlapped simulation domains accordingly. First is the hydrostatic stress evolution in the copper metal, which is marked as the brown in the figure. Stress will be developed once the current is applied in M2 (for up-stream, we only focus on M2 as void will be developed in the cathode, which is to the left of M2). This is called copper wire domain Ω_C and stress on the metal 2 will be governed by the stress differential equation shown below. The second domain is related to the electrical current density, which consists of both copper and barrier or liner surrounding the copper as current can flow through both the copper and the liner. We refer it as the wire domain or Ω_W and it will be described by the Laplace equation of electrical potential. The third domain is the thermal related domain, which consists of everything: copper, liner, capping layers, surrounding dielectric material. We

16

call this thermal domain or Ω_T.

As we can see, all three physics are coupled in this EM post-voiding analysis process: current density and temperature will have impacts on the stress development; void growth will change the current density distribution over time, which will change both current density and temperature profile; and eventually the resistance of the interconnect will change, which is the key determination of the EM failure criteria.

2.2   Electrical Current Density

The second physics to be modeled is the current density distribution in a wire as the current density $\vec{j}$ is an important variable in the Korhonen's equation 1.1 and varies significantly around intersections and corners, where voids are often form or evolve at.

The steady-state (electrostatic) electric field in an interconnect, can be modeled by the Laplace equation with Dirichlet and Neumann boundary conditions. Specifically, for electrical field in the steady state, we have

$$\vec{E} = -\nabla U \tag{2.1}$$

and

$$\vec{j} = \frac{1}{\rho}\vec{E}, \tag{2.2}$$

where U is the voltage potential, $\vec{E}$ is the electric field, and $\vec{j}$ is the current density, and ρ is the electrical resistivity. Given the conservation of charge carriers, current density $\vec{j}$ must have zero divergence over the conductor:

$$0 = \nabla \cdot \vec{j} = -\nabla \cdot \left(\frac{1}{\rho}\nabla U\right). \tag{2.3}$$

In general, the steady state electrical potential U can be prescribed by the Laplace equation with both Dirichlet (voltage) and Neumann (current density) boundary conditions given as follows:

$$\nabla \cdot \left(\frac{1}{\rho} \nabla U \right) = 0, \text{ in } \Omega_W, \tag{2.4}$$

$$U = u_V, \text{ on } \partial \Omega_W \cap \Gamma_V, \tag{2.5}$$

$$\nabla U \cdot \vec{n} = g_V, \text{ on } \partial \Omega_W \cap \Gamma_I, \tag{2.6}$$

where Ω_W is the domain of conductor including copper and liner, Γ_V is the boundaries where voltage (Dirichlet) boundary conditions are given, Γ_I is boundaries where current density (Neumann) boundary conditions are given, U is the unknown electrical potential to be found, u_V, and g_V are given voltage sources and current sources at the boundaries.

2.3 Temperature

In the circuit and layout level, the heat transfer phenomena are governed by the following heat differential equation [17, 5]:

$$\rho_M C_p \frac{\partial T}{\partial t} - \nabla \cdot (\kappa \nabla T) = g_T, \tag{2.7}$$

subject to the following convectional thermal boundary condition (Robin boundary condition)

$$\kappa \nabla T \cdot \vec{n} = h(T_{\text{ext}} - T). \tag{2.8}$$

Here, T is the temperature (K), ρ_M is the mass density of the material $(\text{kg} \cdot \text{m}^{-3})$, C_p is the mass heat capacity $(\text{J} \cdot \text{kg}^{-1} \cdot \text{K}^{-1})$, κ is the thermal conductivity $(\text{W} \cdot \text{m}^{-1} \cdot \text{K}^{-1})$,

and g_T is the heat energy generation rate $(\mathrm{W} \cdot \mathrm{m}^{-3})$, $\vec{n}$ is the outward direction normal to the boundary, h is the heat-transfer coefficient $(\mathrm{W} \cdot \mathrm{m}^{-2} \cdot \mathrm{K}^{-1})$, and T_{ext} is the ambient temperature surrounding the thermal system. If $h = 0$, the boundary condition is adiabatic (convectionally isolated), otherwise it is convective. Note that the thermal conductivity κ differs for different materials.

For a copper wire with surrounding liners, capping layers, and dielectrics, which all together are defined as the domain Ω_T, the heat source comes from Joule heating effect. As a result, we have

$$g_T = j^2 \rho \tag{2.9}$$

In this work, it is assumed that κ is constant for each material. Then, (2.7) can be written as

$$\rho_M C_p \frac{\partial T}{\partial t} - \kappa \nabla^2 T = j^2 \rho_M, \quad \text{in } \Omega_T, \tag{2.10}$$

$$\kappa \nabla T \cdot \vec{n} = h(T_{\mathrm{ext}} - T), \quad \text{on } \partial\Omega_T \cap \Gamma_T, \tag{2.11}$$

subject to the initial condition $T = T_0$ at $t = 0$, where Γ_T is the boundary surface for Robin boundary conditions.

2.4 Copper-Void Boundary

2.4.1 Boundary Modeling Methods

To model void boundary evolution over time, an effective modeling method is required. In an IC interconnect using dual-damascene process, copper strictly confined by liner material and capping layer. Since this work only focuses on void formulation

19

and evolution without considering hillocks, the domain of copper and void is also strictly confined in a fixed volume. Therefore the union of copper and void domains does not require expanding to a larger extent. Within the confined domain, a boundary is necessary to be modeled in order to separate the void domain and the copper domain, while always distributes within the domain.

Given the above observation, several techniques are possible to be applied on modeling the copper-void boundary, or explicitly representing the boundary by respectively indicating the two materials: copper and void (nothing/vacuum). These techniques are discussed below and then the advantages and disadvantages are listed.

Mesh Deformation and Remeshing

In the method of mesh deformation and remeshing, only copper is meshed in the space discretization stage, while anywhere in void is not part of finite element calculation. This method explicitly models the copper-void boundary by its domain boundary. To model boundary evolution, the coordinates of mesh nodes need to be modified. While only modifying the coordinates will distort the elements severely at some point[1], which will lead to low-quality elements, causing computability issues. When this point is reached, the copper domain, described by the modified domain boundary, should be meshed again to have well-shaped elements. For simplicity, this deformation and remeshing process is demonstrated in figure 2.2. Assume the copper-void domain is in 1D, where the blue part on the left is void, and the red part on the left is copper[2]. After meshing, the copper-void

---

[1]This is generally true for most of complexed geometries. Exceptions are, for example meshed using quad elements, cubes with its faces moving normal to themselves.

[2]Here in the demonstration, the void domain is also meshed into elements, showing the capability of modeling the boundary of two different materials. In this void evolution study, the void just depleted copper

boundary, denoted by the purple solid line, is explicitly represented by the copper domain boundary, shown in figure 2.2a. When the void starts growing, the boundary is supposed to advance to the right. This will cause modification of node coordinates, as shown in figures 2.2b and 2.2c. In this process, the elements near the boundary are distorted and their qualities are declined in general. For this reason, when the worst element quality reaches a critical point (figure 2.2d), a remeshing action will be triggered and a new set of elements, with good qualities are generated, shown in figure 2.2e. This process will be repeated until the void evolution simulation is complete.

(a) Initial 1D mesh.

(b) Boundary moved.

(c) Boundary moved.

(d) Boundary moved, element shape/size reaching critical level.

(e) Remesh.

(f) Boundary moved.
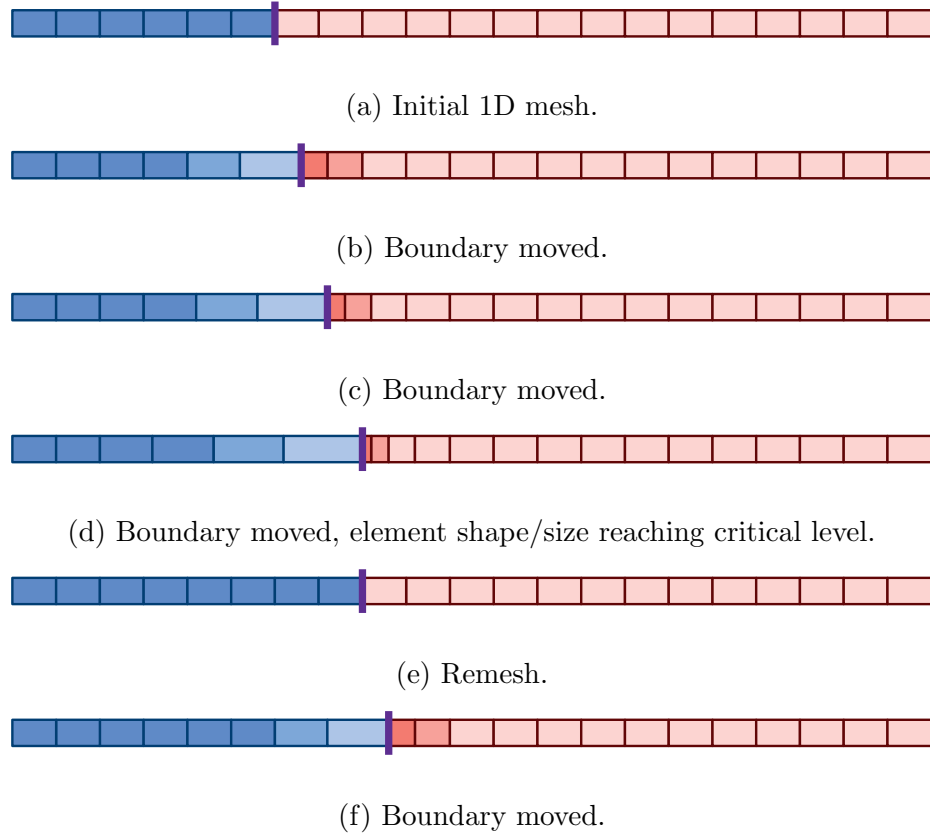
Figure 2.2: Mesh deformation and remeshing demonstration for a growing void in 1D.

so nothing needs to be modeled or meshed.

Mesh deformation and remeshing performs well in 1D modelings, but it lacks a major capability to be applied in simulation of void evolution in 2D and 3D dimensions. Specifically, the modeled boundary can not move over labeled domain boundaries[3]. For example, figure 2.3, shows a case where the void grows toward the right. During this process, the left-most boundary must be excluded from the finite element analysis. This brings significant difficulty in finite element analysis if the removed boundary was prescribed with a boundary condition.

(a) Initial void. (b) Void boundary moved. (c) Void boundary moved.
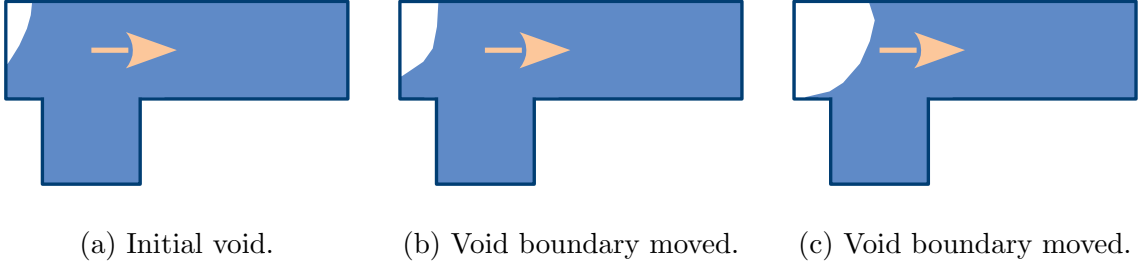
Figure 2.3: 2D growing void boundary moves over labeled domain boundaries.

While the above incapability seems possible to be overcome by leverage some sophisticated implementation related techniques, another disadvantage is critical as well. In void evolution analysis, it is allowed for a void to shrink or even vanish or heal if the applied electrical current is reversed or reduced. Under such circumstance, if the void is already grown and obtained significant volume, it then will shrink and let copper occupy the depleted space again. As demonstrated in figure 2.4, where the ideal final void shape shown in figure 2.4d, it is difficult for the growing copper to recover its shape. This is because

---

[3]In finite element method, in order to correctly apply boundary conditions, the domain boundary is usually managed by labeling separately. Take figure 2.3 as an example, every single line segment can be treated as a uniquely labeled domain boundary, facilitating the prescription of different kinds of boundary conditions. "Moving over labeled domain boundaries" here means boundary intersects to another line segment boundary from a previous one.

the geometry information is lost during the previous void growth. While storing the lost geometry is theoretically possible, it is very difficult to implement or configure practically.

(a) Initial void. (b) Void shrank.

(c) Void shrank. (d) Final void shape.
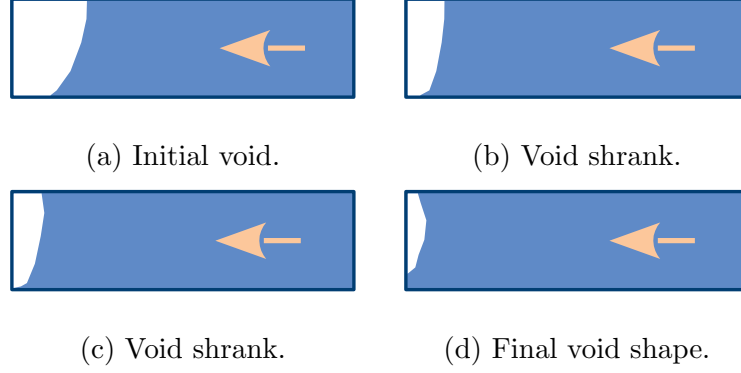
Figure 2.4: 2D shrinking void boundary moves over labeled domain boundaries.

Mesh Element Removal

Mesh element removal method is widely applied in material depletion modeling and simulations [79, 73, 66]. Once an element is determined as occupying an depleted material, the element is excluded from finite element analysis. Comparing to the mesh deformation and remeshing method, the element nodes are not moved, i.e., all the node coordinates are fixed. Sophisticated finite element software uses this method to model material removal process [66]. Similar to the mesh deformation method, only copper domain is meshed and the copper-void boundary is explicitly modeled by the boundary of the meshed copper domain. When considering void shrinking, it requires this method to be applied in the opposite way, which is adding elements to the existing domain. This exposes the similar disadvantage, since the copper interconnect is confined by liner material and capping layers, which does not allow it to grow freely. Moreover, for irregular geometries, where aligned

mesh is not always possible or practical to acquire, adding element can become difficult. On the boundary labeling aspect, it requires more user configuration effort to specify and anticipate the labels and boundary conditions for those boundaries that are going to built up later.

Phase Field

Phase field method [55] has many varieties and has been applied in a wide range of different areas, such as phase transition [57], microstructure evolution [16], solidification process [42], multiphase systems [65], etc. It is also applied recently in an electromigration research [45] for void growth analysis. A phase field is a scalar variable depending on spatial coordinates, that indicates or implies the type of underlying material of the corresponding coordinate. The phase field variable is governed by equations in the form of

$$\frac{\partial \phi}{\partial t} = \mathcal{T}(\phi) + \mathcal{D}(\phi, \vec{v}). \tag{2.12}$$

Here, ϕ is the phase field variable. Function $\mathcal{T}$ is for self-stabilization and function $\mathcal{D}$ is for modifying the phase field to reflect the migration of the implied material[4]. Without the contribution of $\mathcal{D}$, the sole term of $\mathcal{T}(\phi)$ will enforce the phase field to stay still[5]. Under such self-balance, the phase field variable will in general value in fixed limit, within which range different materials are implied by corresponding values[6]. Thus, the copper-void boundary

---

[4]For instance, in the phase field later applied, these two functions can be
$$\mathcal{T}(\phi) = \nabla^2 \phi + \phi - \phi^3,$$
and
$$\mathcal{D}(\phi) = \vec{v} \cdot \nabla \phi.$$

[5]In general, we do not enforce a point-wise equilibrium. Instead, only the integral over the interested domain is required to be stationary without $\mathcal{D}$ driving source.

[6]Take the ϕ used in later chapters as example, its range will be $[-1, 1]$, within which value 1 implies copper, value -1 implies void, any values in between imply the void boundary.

is represented by the transition region implicitly. This implicit boundary will be driven to move by term $\mathcal{D}(\phi, \vec{v})$.

The phase field method requires space discretization both on the void and the copper, comparing to the previously introduced methods, which only discretize the copper domain. This method does not require dynamically change node coordinates or add/remove element entities. Therefore, all the predefined boundary labels and boundary conditions are not subject to change during the void shape evolving, even if the boundary moves over different boundaries.

2.4.2   Comparison

According to the description of the three copper-void boundary modeling methods above, a detailed comparison list is given in table 2.1.

Despite the phase method neither explicitly models the copper-void boundary nor drops the elements on void in computation, it does facilitate 2D and 3D void growth and shrinking modeling. Moreover, remeshing is fully optional in phase field, which means adaptive local mesh refinement [4] is allowed when finer solution is needed. In the detailed finite element method description and software implementation presented later, we will choose phase field as the boundary modeling method unless specified otherwise.

2.4.3   Phase Field Method

In this section, we present the new FEM-based void growth simulation method, which is governed by the coupled stress diffusion and phase field equations, to model the void boundary and dynamic stress distribution evolving over time in the copper interconnects.

	(a)	(b)	(c)
Explicit boundary	Yes	Hard	No
Boundary moving prescription	Explicit	Implicit	Explicit
1D void growth	Yes	Yes	Yes
2D void growth	Yes	Yes	Yes
3D void growth	Yes	Yes	Yes
1D void shrinking	Yes	Yes	Yes
2D void shrinking	Hard	Hard	Yes
3D void shrinking	Hard	Hard	Yes
Extra computation	No	No	Yes
BC† Friendliness	Low	Low	High
Remesh	Required	Difficult	Allowed
Node translation	Yes	No	No
Element operation	No	Yes	No

(a): mesh deformation and remeshing; (b): element removal; (c): phase field.

†BC: boundary condition.

Table 2.1: Comparison between copper-void boundary modeling methods.

Specifically, to model the interaction of hydrostatic stress σ and void shape (implied by phase field ϕ), we introduce a new phase field variable ϕ, which is defined as a unit-less function of space that represents the material by its value. As time evolves: (1) hydrostatic stress derivative leads to atomic flux which drives void-copper boundary to move; (2) movement of a void-copper boundary relaxes the hydrostatic stress from being built up; (3) void volume and stress distribution conservation should be maintained. These three aspects will be discussed in the following sections in which partial differential equations are derived to reflect the mechanisms of the interaction between σ and ϕ, as well as their evolution over time.

2.4.4 Phase field and weight functions

Phase field ϕ is defined as a function of space by which the underlying material is represented. Over time, its evolution is governed by

$$\frac{\partial \phi}{\partial t} = \frac{1}{\tau_\phi}\left(\frac{\delta^2}{2}\nabla^2\phi + \phi(1-\phi^2)\right) - k_R\vec{v}\cdot\nabla\phi, \tag{2.13}$$

where δ is the thickness of the void-copper diffusive boundary, τ_ϕ is the stabilizing time constant, k_R is regulation factor and will be discussed in section 2.5.2, and $\vec{v}$ is the velocity of phase field boundary, to be described in section 2.4.5. In general, domain of ϕ is locked approximately to $[-1, 1]$ by the diffusion and source terms in (2.13). Comparing to the phase field equation used in work [68], which does not have consistent units, term τ_ϕ is introduced to control the field convergence speed and make the units match on both sides of the equation. Materials are implied as copper if $\phi \approx 1$, void if $\phi \approx -1$, copper-void boundary if $-1 < \phi < 1$. To recognize different materials in the PDEs in the following sections, three weight functions, w_V, w_B, w_C, are defined as the softmax of three likelihood functions, respectively z_V, z_B, z_C, indicting the regions of void, boundary, and copper:

$$w_V = \mathcal{S}(z_V|z_V, z_B, z_C), \tag{2.14}$$

$$w_B = \mathcal{S}(z_B|z_V, z_B, z_C), \tag{2.15}$$

$$w_C = \mathcal{S}(z_C|z_V, z_B, z_C). \tag{2.16}$$

where the softmax function $\mathcal{S}(z_k|z_1, z_2, \cdots, z_N)$ is defined as

$$\mathcal{S}(z_k|z_1, z_2, \cdots, z_N) = \frac{e^{z_k}}{\sum_{i=1}^{N}e^{z_i}}. \tag{2.17}$$
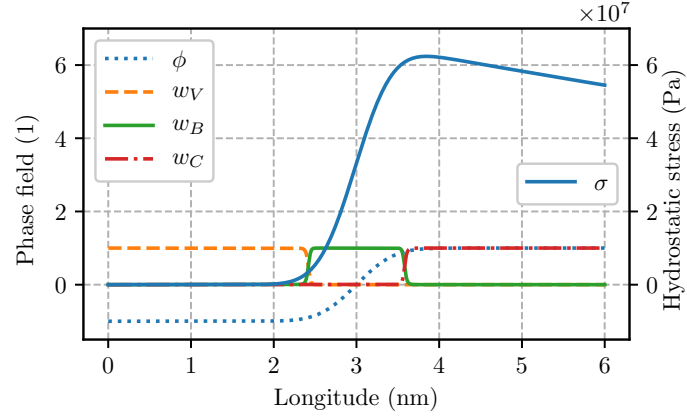
Figure 2.5: Near-boundary plot of hydrostatic stress, phase field, and weight functions.

The likelihood functions depend on phase field ϕ:

$$z_V = k_s \frac{-\phi - \phi_{\text{th}}}{1 - \phi_{\text{th}}}, \tag{2.18}$$

$$z_B = k_s \frac{\phi_{\text{th}}^2 - \phi^2}{\phi_{\text{th}}^2}, \tag{2.19}$$

$$z_C = k_s \frac{\phi - \phi_{\text{th}}}{1 - \phi_{\text{th}}}, \tag{2.20}$$

where k_s is the sharpness coefficient and ϕ_{th} is the phase field threshold. An illustration of σ, ϕ, w_V, w_B, and w_C near void-copper boundary is shown in Figure 2.5.

Figure 2.6 demonstrates the relationship of a 2D phase field and the corresponding weight functions[7].

2.4.5   Void-copper boundary evolution

Void-copper boundary is implied by the phase field variable ϕ. In equation (2.13), one key unknown variable is the boundary movement velocity $\vec{v}$. The relationship between

---

[7]Here $k_s = 3$ and $\phi_{\text{th}} = 0.8$ for sake of demonstration
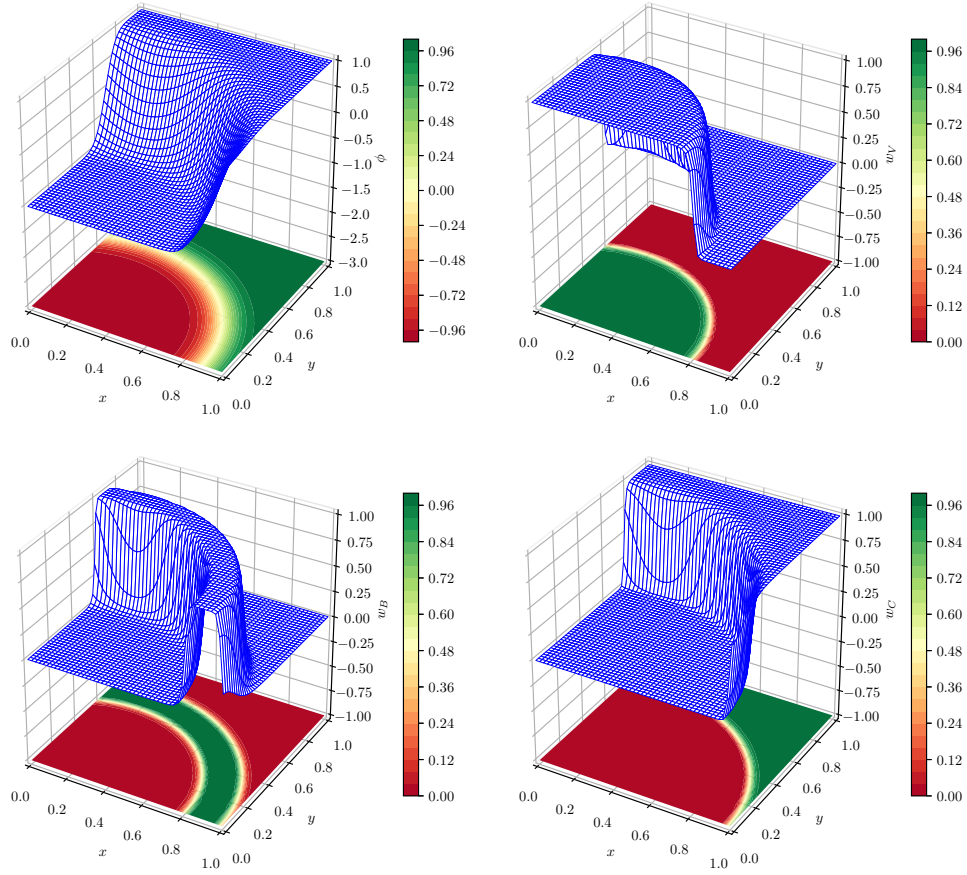
Figure 2.6: 2D weight functions indicating underlying materials.

$\vec{v}$ and the atomic flux J_a is

$$\vec{v} = \vec{J_a}\Omega, \tag{2.21}$$

according to the observation of number of copper atoms moved through infinitesimal area dA:

$$J_a \cdot dA \cdot dt = \frac{dV}{\Omega} = \frac{dA \cdot vdt}{\Omega}. \tag{2.22}$$

The flux of atoms $\vec{J_a}$ is a function of electric field $\vec{E}$ and the gradient of chemical

potential μ:

$$\vec{J_a} = -\frac{D_a C_a}{k_B T} \left(\nabla \mu + \vec{E} q^* \right), \tag{2.23}$$

where $\mu = \mu_0 - \Omega \sigma$ and C_a denotes the copper atom concentration. On void-copper boundary where the normal vector is denoted by $\vec{n}$, we have $\vec{E} \cdot \vec{n} = 0$, which leads the following simplification:

$$\vec{J_a} \cdot \vec{n} = -\frac{D_a C_a}{k_B T} \left(\nabla \mu \right) \cdot \vec{n} = \frac{D_a C_a}{k_B T} \Omega \nabla \sigma \cdot \vec{n}. \tag{2.24}$$

Apply the approximation $C_a \Omega \approx 1$ we get the boundary movement velocity

$$\vec{v} \cdot \vec{n} = \frac{D_a \Omega}{k_B T} \nabla \sigma \cdot \vec{n}. \tag{2.25}$$

By substituting $\vec{v}$ in equation (2.13) using (2.25), the evolution of phase field ϕ regulated by $\nabla \sigma$ is described as:

$$\frac{\partial \phi}{\partial t} = \frac{1}{\tau_\phi} \left(\frac{\delta^2}{2} \nabla^2 \phi + \phi(1 - \phi^2) \right) - k_R \frac{D_a \Omega}{k_B T} \nabla \sigma \cdot \nabla \phi. \tag{2.26}$$

Note that $\nabla \sigma$ is the stress gradient on void-copper interface as depicted in Figure 2.5.

2.4.6 Discussion on an Existing Phase Field Application

A phase field method is applied in work [68] to model 1D copper void boundary behavior. The phase field is governed by equation

$$\frac{\partial \phi}{\partial t} - \xi \nabla^2 \phi + (\phi^2 - 1)\phi + V \nabla \phi = 0 \tag{2.27}$$

where ϕ is the unit-less phase field variable; ξ depends on the prescribed interface thickness δ and have the unit of m^2; V is the boundary velocity. It should be noted that the units in

this equation is misconfigured. Specifically, terms $\frac{\partial \phi}{\partial t}$ and $V\nabla\phi$ have unit s^{-1} while terms $\xi\nabla^2\phi$ and $(\phi^2 - 1)\phi$ are unit-less.

This configuration will be equivalent to

$$\frac{\partial \phi}{\partial t} - \frac{1}{\tau}\xi\nabla^2\phi + \frac{1}{\tau}(\phi^2 - 1)\phi + V\nabla\phi = 0 \tag{2.28}$$

where $\tau = 1$s under the international system of units [54]. τ, as referred as τ_ϕ in the rest of this thesis, is the *phase convergence* time constant, prescribes how fast a phase field, especially its transition band, reaches the steady state where the diffusion term $\xi\nabla^2\phi$ and pseudo free energy term $(\phi^2 - 1)\phi$ cancel each other. Since 1 second is significantly smaller than the simulated time scale[8], the default time constant of 1 second by the unit-mismatched equation used in [68] has trivial negative effect in the simulation accuracy. While from the numerical computation perspective, 1 second is too small comparing to he expected simulation time scale, which is usually from months to years. This is because the solver has to adjust the time step down to the same time scale as τ to acquire precise phase field solution, i.e., copper-void boundary evolution, where very small time resolution is not necessary when an interconnect is expected to fail over a long time, but the computation resource might be wasted for achieving such high time resolution and overall analysis time would be unnecessarily prolonged.

Therefore a phase field convergence time constant τ_ϕ is introduced in equation (2.13) to allow adjustment of time resolution of simulated copper-void boundary evolution.

---

[8]Simulated copper-void boundary usually has moving speed at the level of $1 \text{ nm} \cdot \text{s}^{-1}$. Simulated MTTF (mean time to failure) of an interconnect can be up to years.

2.5 Hydrostatic Stress

2.5.1 Hydrostatic stress modeling in confined metal wires

Hydrostatic stress σ in confined metal wire has different governing equations which need to be separately derived for void, void-copper boundary, and copper.

Governing equation of σ in copper

In confined copper, hydrostatic stress σ builds up over time [45]:

$$\frac{\partial \sigma}{\partial t} = \nabla \cdot \left(\frac{D_a B}{k_B T} (\Omega \nabla \sigma - e Z \rho \vec{j}) \right). \tag{2.29}$$

Within the copper domain where the electrical resistance ρ and effective charge number Z is constants, the term $\nabla \cdot e Z \rho \vec{j}$ has value zero, given the conservation of charge carrier $\nabla \cdot \vec{j} = 0$. The divergence of current density $\vec{j}$ is non-zero only at the copper domain boundaries where electrical current is prescribed to flow through.

Governing equation of σ on void-copper boundary interface

On copper-void boundary interface, source term $e Z \rho \vec{j}$ in equation (2.29) is dropped since there is no current flowing in the normal direction. As boundary moving with a certain speed, σ decreases as a relaxation effect of the void growth and the consequent copper atom migration. At a certain point in copper, it takes $\Delta t = \frac{\delta}{v}$ for the boundary to leave since its inclusion. During Δt, stress σ is reduced to zero from the original value when it was included in the void-copper boundary. A term $-\frac{\sigma v k_\tau}{\delta}$ is therefore added to the right hand

side to model this behavior:

$$\frac{\partial \sigma}{\partial t} = \nabla \cdot \left(\frac{D_a B \Omega}{k_B T} \nabla \sigma \right) - \frac{\sigma v k_\tau}{\delta}, \tag{2.30}$$

where k_τ is the speed-up factor which ensures that the stress is relaxed to a near-zero level.

Governing equation of σ inside a void

Once a void has nucleated, which is typically at or near a terminal node, the stress at the void location will immediately vanish. However, the stress around the void will be close to the same stress level as immediately prior to the nucleation [45, 68]. As a result, a very large stress gradient will be formed around the void at nucleation time, which can be described by [45]. The hydrostatic stress of copper inside a void is governed by

$$\frac{\partial \sigma}{\partial t} = -\frac{\sigma}{\tau_{\text{void}}}, \tag{2.31}$$

where τ_{void} is the vanishing time constant to ensure σ valued at 0 in the void.

The combined final governing equation

To consider the three equations together in domain Ω_L, weight functions introduced in section 2.4.4 are used to combine them into one PDE

$$\frac{\partial \sigma}{\partial t} = w_C \nabla \cdot \left(\frac{D_a B \Omega}{k_B T} \nabla \sigma \right) - w_V \frac{\sigma}{\tau_{\text{void}}} + w_B \left(\nabla \cdot \left(\frac{D_a B \Omega}{k_B T} \nabla \sigma \right) - \frac{\sigma v k_\tau}{\delta} \right). \tag{2.32}$$

2.5.2 Conservation enforcement by relating void volume and hydrostatic stress distribution

To enforce the conservation equation (1.4), one approach is to introduce a closed-loop feedback between the two coupled PDE systems so that if the volumes computed by

33

the stress distribution and the actual void volume are not equal, the void boundary growth speed will be adjusted. To achieve this, we need to compute the void volume in two different ways which are equivalent in terms of continuum physics.

Specifically, according to the definition of weight functions, the modeled void volume can be calculated by integrating appropriate combination of w_B and w_V:

$$V_W = \int_{\Omega_L} \left(w_V + \frac{1}{2} w_B \right) dV. \tag{2.33}$$

In addition, the stress and the void saturation volume need to satisfy the relationship defined in (1.4).

Due to the large atom flux at the void-copper boundary, atoms can diffuse into the whole copper interconnect very fast if the void volume and stress distribution interaction is not considered. This will manifest the growing discrepancy between V_W and V_S over time. To mitigate this problem, we introduce a new regulation factor k_R in phase field governing equation (2.13) and (2.26):

$$k_R = \frac{1}{2} \left(1 + \operatorname{erf} \left(\frac{1}{c_R} (V_W - V_S) \right) \right), \tag{2.34}$$

where $\operatorname{erf}(\cdot)$ is the Gauss error function to force a bounded output range $(-1, 1)$, c_R is a coefficient controlling the resolution of matching two derived void volumes. As a result, the regulated velocity of the void shape boundary will be expressed as $\vec{v} k_R$. Consequently, if V_W is larger than V_S, $k_B \vec{v}$ will increase so that the void volume, V_S, will increase as well, which will make the difference between the two volumes smaller. This is also the case when V_W is smaller than V_S. Figure 2.7 shows the modeled void volume V_W being consistent with the void saturation volume V_S from the proposed simulation method.
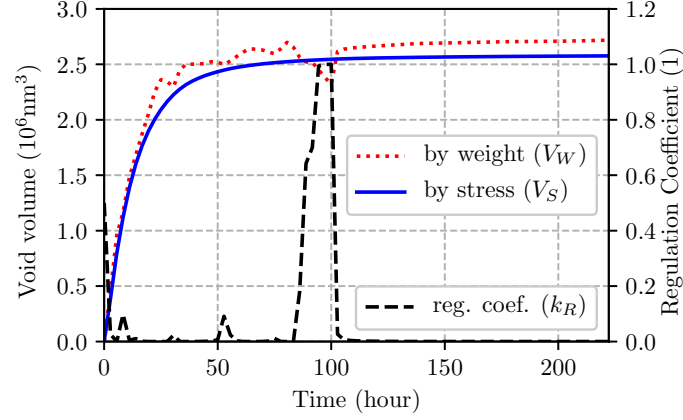
Figure 2.7: Void volume calculated by weight functions and by stress over time.

2.6 Joule Heating Modeling
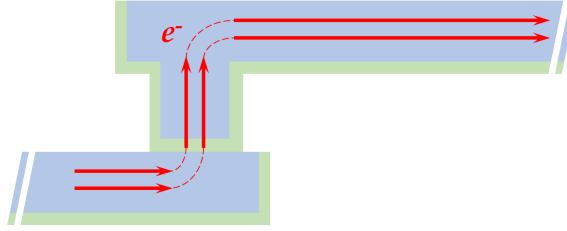
Figure 2.8: Illustration of electron flow in an upstream configuration.

Void growth simulation enables observing temperature and resistance jump due to Joule heating effect. Under an upstream configuration, where electron flows from via to upper level of copper interconnect as illustrated in Figure 2.8, a temperature and resistance jump is observed in work [24]. When a growing void, expanding from the reservoir side toward the anode direction, covers all the via copper, the current originally conducting by copper will be shunted to the surrounding barrier metal. As shown in Figure 2.9, current flows through a narrow portion of barrier metal in such case, where the Joule heating is highly concentrated and heat dissipation is limited. We refer this portion of barrier, and

the copper (if any) close by, as *the hotspot*. A hotspot is suddenly heated up at the time the via copper is cut off from the main interconnect by the growing void. Consequently the temperature will noticeably increase near the hotspot. It should be noted that, under a prescribed current density, hotspot temperature and resistance will converge at a higher level due to a weak positive feedback in-between. To be specific, higher temperature leads to higher hotspot resistance (given their positive resistivity temperature coefficient), followed by higher Joule heating power, which feeds back to higher temperature. The convergence of the hotspot temperature and resistance is guaranteed since a) a realistic metal or alloy has low resistivity temperature coefficient (usually less than 0.01); b) temperature itself is in a negative feedback loop because heat dissipation rate is proportional to media temperature difference, according to Fourier's law of thermal conduction.
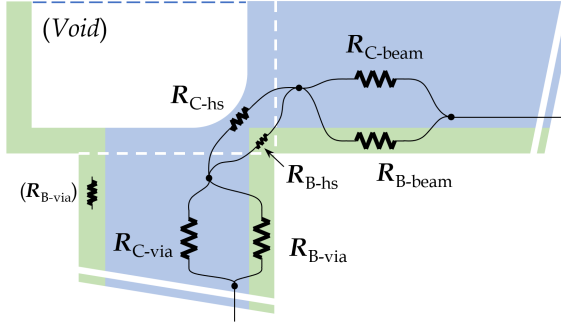
Figure 2.9: Discretized approximation of metal connectivity before the temperature jump.

Figure 2.9 and 2.10 illustrate how the electrical connectivity is modeled. The total resistance is calculated using

$$R = (R_{\text{C-via}} \| R_{\text{B-via}}) + (R_{\text{C-hs}} \| (R_{\text{B-hs}} + R_{\text{B-neck}})) + (R_{\text{C-beam}} \| R_{\text{B-beam}}), \qquad (2.35)$$

where "$\|$" is the impedance parallel operator: $z_1 \| z_2 = \frac{z_1 z_2}{z_1 + z_2}$, subscript prefix "C-" and
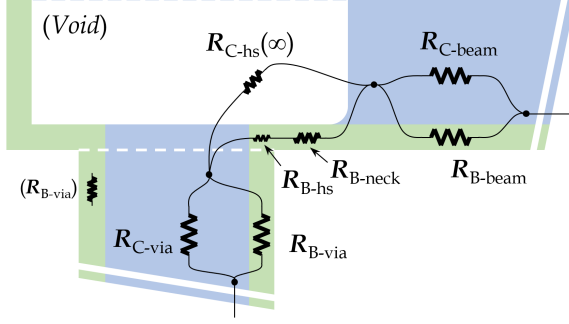
Figure 2.10: Discretized approximation of metal connectivity after the temperature jump.

"B-" respectively denotes *copper* and *barrier*, subscript postfix "-hs" stands for *hotspot*.

On the hotspot, the temperature is modeled by balancing the heat dissipation and generation:

$$\frac{\lambda A(T - T_0)}{\delta} = I^2 \left[R_{\text{B-hs}}(1 + \alpha_B(T - T_{\text{ref}})) \| R_{\text{C-hs}}(1 + \alpha_C(T - T_{\text{ref}})) \right], \tag{2.36}$$

where λ is the heat conduction coefficient; T is the temperature to solve; T_0 is the temperature of the surrounding metal; T_{ref} is the reference temperature for barrier resistivity calculation; α_B and α_C are respectively the resistivity of the barrier material and copper; A and δ are the effective heat dissipation area and thickness; I is the prescribed electrical current. We assume that the heating hotspot is small when compared to the interconnect scale such that the temperature surrounding metal does not change much because of the hotspot. $R_{\text{B-hs}}$ can be calculated as a constant since the barrier does not have structure change while void is evolving. Copper resistance at the hotspot $R_{\text{C-hs}}$ can be measured by applying probes during simulation. Note that in Figure 2.9 and 2.10, due to its 2-dimensional nature, it cannot fully demonstrate the barrier connectivity in 3D cases. The green-shaded barrier material on the left side and right side are actually mechanically and

electrically connected in 3D cases, as they are surrounding the copper interconnect except near the copper-capping layer boundary. Here capping layer refers to the isolation layer in dual damascene process[28]. $R_{\text{B-hs}}$, $R_{\text{B-via}}$, and $R_{\text{B-neck}}$ are therefore not as separable in 3D as in the illustrated 2D case.

By collecting all the above-mentioned values, temperature change caused by Joule-heating can be calculated. According to equations (2.35) and (2.36), resistance change and temperature change can be divided into three phases.

In the first phase, when the hydrostatic stress is not built up to the critical level to drive the void to grow, or the growing void is still mostly contained in the reservoir, thus the resistance of copper at the hotspot remains approximately unchanged. In this phase, the total resistance temperature does not change noticeably over time, since the structure of conducting material is not changed. Consequently, heating characteristic and temperature stay approximately unchanged.

In the second phase, the void size is large enough to partially cover the via cross-section area. As depicted in figure 2.9, a large portion of the copper that conducts current between the via and the rest of the interconnect (marked as *beam*) has been depleted. This will cause the effective resistance $R_{\text{C-hs}}$ to noticeably increase, compared to the first phase. It keeps increasing until there is no copper connecting the via and the *beam*, where effectively, $R_{\text{C-hs}} = +\infty$. During the end of this phase, $R_{\text{C-hs}}$ accelerates approaching infinity, as the effective hotspot copper conductivity (reciprocal to resistivity) decreases to zero, with the void growing at a steady speed. In this phase, $R_{\text{C-hs}}$ and $R_{\text{B-hs}}$ dominants the whole resistance growth.

In the third phase, it is merely the barrier material that conducts current between the via and the rest of the copper. We use $R_{\text{B-neck}}$ to denote this part of barrier, next to which the copper is already depleted in the previous phases. As the void continues to grow, $R_{\text{B-neck}}$ increases due to Ohm's law. Comparing to the second phase, $R_{\text{B-neck}}$ dominates the whole resistance growth in this phase and increases linearly as the void growing at a steady speed.

2.7   Stationary Analysis

Because of the effect of stress relaxing near moving void boundary, the steady state of hydrostatic stress is zero at void boundary, providing no driving force to advance the boundary further. Therefore we have velocity term $\vec{v}$ in (3.56) vanishes:

$$\vec{v} \cdot \vec{n} = \frac{D_a \Omega}{k_B T} \nabla \sigma \cdot \vec{n} = 0. \tag{2.37}$$

which implies zero-valued Neumann boundary condition prescribed on the copper-void boundary

$$\nabla \sigma \cdot \vec{n} = 0. \tag{2.38}$$

Along with current density induced boundary conditions, the steady state hydrostatic stress govern equation is

$$\nabla \cdot \left(\frac{D_a B \Omega}{k_B T} \nabla \sigma \right) = 0 \quad \text{in } \Omega_C, \tag{2.39}$$

$$\nabla \sigma \cdot \vec{n} = 0 \quad \text{on } \partial \Omega_C \cap \partial \Omega_V, \tag{2.40}$$

$$\nabla \sigma \cdot \vec{n} = \frac{e Z \rho}{\Omega} j \quad \text{on } \partial \Omega_C \cap \Gamma_I, \tag{2.41}$$

39

where Ω_C and Ω_B respectively denote the copper domain and the void domain; Γ_I denotes the boundaries where non-zero current densities are prescribed.

For the temperature system described in equations (2.10) and (2.11), the corresponding stationary case is where

$$\frac{\partial T}{\partial t} = 0 \tag{2.42}$$

is achieved. Under such condition, the stationary temperature is governed by

$$-\kappa \nabla^2 T = j^2 \rho_M \quad \text{in } \Omega_T, \tag{2.43}$$

$$\kappa \nabla T \cdot \vec{n} = h(T_{\text{ext}} - T) \quad \text{on } \partial \Omega_T \cap \Gamma_T, \tag{2.44}$$

where Γ_T is the boundary surfaces where convectional boundary conditions are prescribed.

The difference from time dependent analysis of current density and void volume is trivial. Since quasi-static analysis of current density is obtained in each time step of transient analysis as well, the current density analysis (depend on electrical potential U) remains the same as described in equations (2.4), (2.5), and (2.6). The void volume, according to the atomic conservation, can be directly calculated using equation (1.4).

2.8 Summary

Coupled physics systems are defined in this chapter, thereby domains different physics systems can be clearly specified. To model the boundary of a generated void and copper, several methods are thoroughly discussed and compared. Phase field method is chosen according to the characteristics of the practical demand of the EM analysis. Four physics systems, electrical current density (derived from electrical potential U), temperature

T, copper-void boundary (derived from phase field ϕ), and hydrostatic stress σ are described in math forms. The interaction between hydrostatic and copper-void boundary movement is discussed. As a result, a regulation factor k_R is introduced into the phase field governing equation such that the simulated void volume can match the stress distribution as specified in equation (1.4). In addition to the transient forms, steady analysis are also discussed and equations are derived accordingly.

Chapter 3

Finite Element Method

In this chapter, a finite element method is briefly introduced, followed by the derivation of all the weak forms related to the simulated physics systems as described in chapter 2. Various topics, including time discretization, phase field conservation, and stationary analysis, are also discussed.

3.1   Introduction

Using Poisson equation as an example, the finite element method is briefly introduced as following. The example function $u(x)$ prescribed by

$$-\nabla \cdot (\alpha \nabla u) = f \quad \text{in } \Omega, \tag{3.1}$$

$$u = g \quad \text{on } \partial\Omega \cap \Gamma_g, \tag{3.2}$$

$$\nabla u \cdot \vec{n} = h \quad \text{on } \partial\Omega \cap \Gamma_h, \tag{3.3}$$
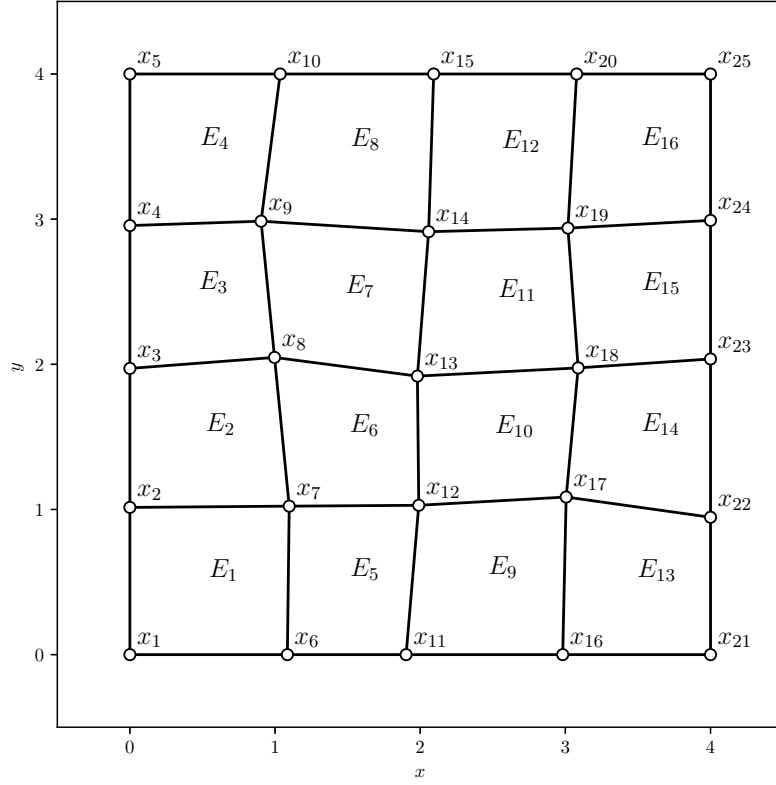
Figure 3.1: A coarse quadrilateral mesh.

where α and f, defined in Ω, are dependent on spacial coordinate x; g and h are respectively defined on Γ_g and Γ_h, prescribing Dirichlet and Neumann boundary conditions. To solve this function u, domain Ω needs firstly to be meshed. As shown in figure 3.1, a domain with arbitrary shape can be meshed into elements E_1 through E_{16}, resulting discrete element vertices x_1 through x_{25}. In practical finite element applications, finer meshing is usually applied in order to achieve desired solution accuracy.

The solution of function u_H, defined by the combination of *shape functions*, is an approximation of the original, true function u. Here, shape functions are defined corresponding to each vertex. For vertex x_i, the corresponding shape function of φ_i can be

(a) Shape function φ_{10}.   (b) Shape function φ_{18}.   (c) Shape function φ_{22}.
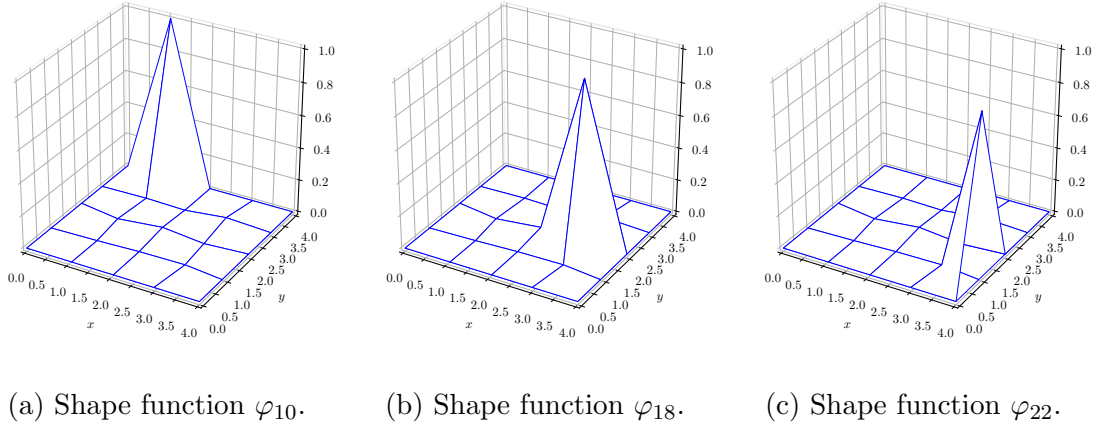
Figure 3.2: Linear shape functions on a coarse quadrilateral mesh.

defined in many ways. It can be linear or in higher orders. In general, shape functions satisfy the following equations:

$$\varphi_i(x_j) = \delta_{ij}, \tag{3.4}$$

$$\sum_{i:\ x_i \text{ on } \partial E} \varphi_i(x) = 1 \quad \text{for all } x \in E, \tag{3.5}$$

where δ_{ij} is Kronecker delta. Figure 3.2 shows linear shape functions φ_{10}, φ_{18}, and φ_{22}.

Using shape functions, a hypothesis solution u_H can then be represented by

$$u_H = \sum_{i=1}^{M} U_i \varphi_i \tag{3.6}$$

where M is the number of vertices, $U_1, U_2, \ldots, U_M$ are discrete values to be solved. In fact, shape functions $\varphi_1, \varphi_2, \ldots, \varphi_M$ spans a finite function space where the original function u can be projected to, or approximated in, as u_H. Figure 3.3 demonstrates how an arbitrary original function is approximated by projecting to the finite function space spanned by the shape functions. Thereby we can build the weaker equity

$$\nabla \cdot (\alpha \nabla u_H) + f = 0 \tag{3.7}$$

(a) Original.   (b) Projected.
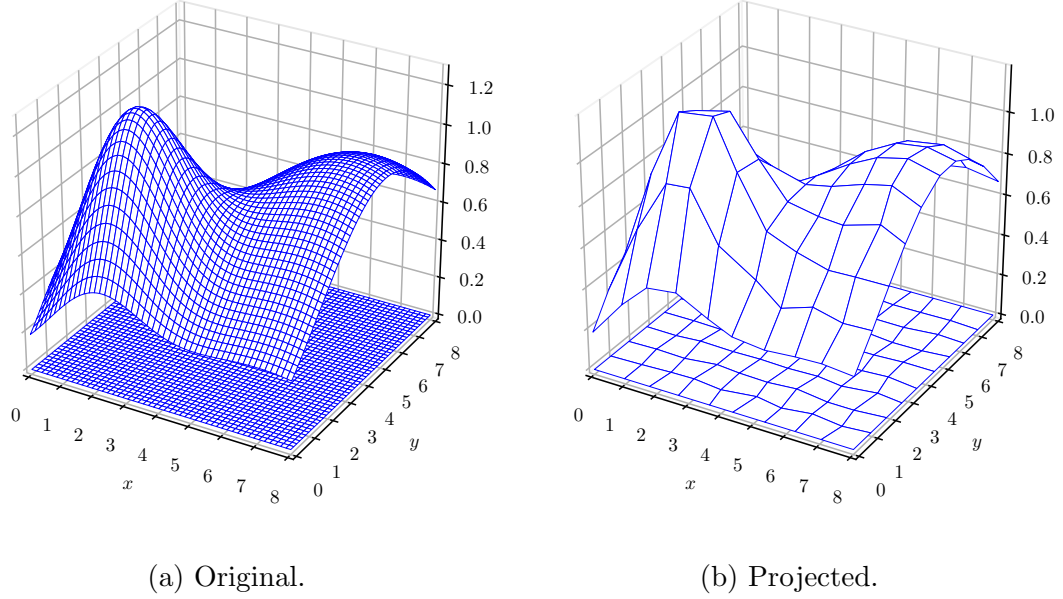
Figure 3.3: An arbitrary function projecting to the function space spanned by linear mesh shape functions.

as suggested in the strong form equation (3.1). (3.7) values to zero implies that for an arbitrary test function v_H[1] in the finite function space, the inner product measure along v_H

$$(\nabla \cdot (\alpha \nabla u_H) + f, v_H) \tag{3.8}$$

should be zero as well.

If the inner product is defined as a uniform weighted integration of euclidean dot product over the domain as

$$(a, b) = \int_\Omega a \cdot b \, dV, \tag{3.9}$$

---

[1]In fact, v_H is not fully arbitrary. It values to zero at the boundaries where Dirichlet conditions are enforced

$$v_H(x) = 0 \quad \text{for all } x \text{ on } \Gamma_g.$$

This is because values of u_H on Dirichlet boundaries will be eventually forced to be the prescribed values, which does not require a "measure" on these locations regarding whether (3.1) is satisfied.

equation (3.7) can be written as

$$-\int_\Omega \nabla \cdot (\alpha \nabla u_H) \cdot v_H \, dV = \int_\Omega f \cdot v_H \, dV, \tag{3.10}$$

which can be simplified by applying divergence theorem on the left hand side, into

$$\int_\Omega \alpha \nabla u_H \cdot \nabla v_H \, dV = \int_\Omega f \cdot v_H \, dV + \int_{\partial\Omega} \alpha \nabla u_H \cdot v_H \cdot \vec{n} \, dS, \tag{3.11}$$

Plug in the Neumann boundary equation (3.3), the weak form of the Poisson problem is acquired[2]

$$\int_\Omega \alpha \nabla u_H \cdot \nabla v_H \, dV = \int_\Omega f \cdot v_H \, dV + \int_{\partial\Omega \cap \Gamma_h} \alpha h v_H \, dS, \tag{3.12}$$

Substitute u_H and v_H with the summations of nodal values in equation (3.12), we have

$$\int_\Omega \left(\sum_{i=1}^{M} \alpha U_i \nabla \varphi_i \right) \cdot \left(\sum_{j=1}^{M} V_j \nabla \varphi_j \right) \, dV = \int_\Omega \left(\sum_{j=1}^{M} f V_j \nabla \varphi_j \right) \, dV$$
$$+ \int_{\partial\Omega \cap \Gamma_h} \left(\sum_{j=1}^{M} \alpha h V_j \nabla \varphi_j \right) \, dS, \tag{3.13}$$

which is equivalent to

$$\sum_{i=1}^{M} \sum_{j=1}^{M} V_j \left(\int_\Omega \alpha \nabla \varphi_i \nabla \varphi_j \, dV \right) U_i = \sum_{j=1}^{M} V_j \left(\int_\Omega f \nabla \varphi_j \, dV + \int_{\partial\Omega \cap \Gamma_h} \alpha h \nabla \varphi_j \, dS \right) \tag{3.14}$$

---

[2]On domain boundary, when there is no Dirichlet or Neumann conditions specified, the conventional assumption is the "zero-flux" constraint presents, which is actually a zero-valued Neumann condition

$$\nabla u \cdot \vec{n} = 0 \quad \text{on } \partial\Omega - \Gamma_g - \Gamma_h.$$

Therefore the substitution is valid given

$$\int_{\partial\Omega} \alpha \nabla u_H \cdot v_H \cdot \vec{n} \, dS = \int_{\partial\Omega \cap \Gamma_g} \alpha \nabla u_H \cdot v_H \cdot \vec{n} \, dS + \int_{\partial\Omega \cap \Gamma_h} \alpha \nabla u_H \cdot v_H \cdot \vec{n} \, dS + \int_{\partial\Omega - \Gamma_g - \Gamma_h} \alpha \nabla u_H \cdot v_H \cdot \vec{n} \, dS$$
$$= \int_{\partial\Omega \cap \Gamma_g} \alpha \nabla u_H \cdot 0 \cdot \vec{n} \, dS + \int_{\partial\Omega \cap \Gamma_h} \alpha h v_H \cdot \, dS + \int_{\partial\Omega - \Gamma_g - \Gamma_h} \alpha \cdot 0 \cdot v_H \, dS$$
$$= \int_{\partial\Omega \cap \Gamma_h} \alpha h v_H \cdot \, dS.$$

given that U_i and V_j are constants. The above equation can be reorganized in matrix form of

$$V^T A U = V^T F, \tag{3.15}$$

where

$$A_{ij} = \int_\Omega \alpha \nabla \varphi_i \nabla \varphi_j \, dV \tag{3.16}$$

$$F_i = \int_\Omega f \nabla \varphi_i \, dV + \int_{\partial\Omega \cap \Gamma_h} \alpha h \nabla \varphi_i \, dS, \tag{3.17}$$

which holds true for any $V = [V_1, V_2, \cdots, V_M]$. It is very obvious that such equity implies

$$AU = F, \tag{3.18}$$

from where vector U can be solved by calculating $U = A^{-1}F$, and projected solution u_H can be thereby determined, satisfying the weak form in equation 3.12.

Assembling matrices A and F, called system matrices, can be divided into calculating each components of them, which requires integration of products of shape functions and other prescribed variables depending on spatial coordinates. Not only for linear shape functions demonstrated above, but also any higher order shape functions, well developed numerical integration techniques, such as Gaussian quadrature rule, will facilitate the assembling process in a computational efficient way, while achieving fully controllable accuracy.

3.2 Variational Weak Forms

As described above, variational weak form is obviously the bridge that connects the mathematical strong form of variables such as ϕ and σ and the software implementation of solving them. Therefore we list the weak forms of the simulated systems in the rest of

this section. We list the derived weak form equations of electrical potential U, phase field ϕ, temperature T, and hydrostatic stress σ in the following.

3.2.1 Weak Form Notations

While in variational analysis the integral over domain Ω of a inner product is expressed as $(\vec{a}, \vec{b})_\Omega$, in this work, for the sake of explicitness, we instead use the conventional form of

$$\int_\Omega \vec{a} \cdot \vec{b} \, dV \tag{3.19}$$

to denote the integral over a 3D domain. Similarly, boundary integral is expressed in

$$\int_{\partial\Omega} \vec{a} \cdot \vec{b} \, dS. \tag{3.20}$$

Here, $\partial\Omega$ denotes the boundary of domain Ω, dS denotes the infinitesimal facet integrand on the domain boundary. In the later use, expression (3.19) may refer the integral in 1D, 2D, or 3D, where correspondingly (3.20) will refer to the integral on the domain boundary of 0D, 1D, or 2D, respectively.

In the time dependent weak form equations, we use $x^{(n)}$ to denote the value of x at the nth simulated time step.

3.2.2 Electrical Potential U

Since EM analysis is done on a time scale of at least days, we treat the electrical potential distribution U as quasi static so it can be described by the strong form Laplace equation (2.4). Multiply shape function φ on both side of equation (2.4) and integrate over

the conductor domain Ω_C, we have

$$\int_{\Omega_C} \nabla \cdot \left(\frac{1}{\rho}\nabla U\right) \varphi \, dV = 0. \tag{3.21}$$

Expand the left hand side by applying the rule of integration by parts and get

$$\int_{\Omega_C} \nabla \cdot \left(\frac{1}{\rho}\nabla U \varphi\right) dV - \int_{\Omega_C} \frac{1}{\rho}\nabla U \cdot \nabla \varphi \, dV = 0, \tag{3.22}$$

which can be simplified by substituting $\int_{\Omega_C} \nabla \cdot \left(\frac{1}{\rho}\nabla U \varphi\right) dV$ with $\int_{\partial \Omega_C} \frac{1}{\rho}\nabla U \varphi \cdot \vec{n} \, dS$, as

$$\int_{\Omega_C} \frac{1}{\rho}\nabla U \cdot \nabla \varphi \, dV = \int_{\partial \Omega_C} \frac{1}{\rho}\nabla U \varphi \cdot \vec{n} \, dS. \tag{3.23}$$

Multiply ρ on both side and plugin equation (2.6), the final weak form for electrical potential is derived as

$$\int_{\Omega_C} \nabla U \cdot \nabla \varphi \, dV = \int_{\partial \Omega_C} g_V \varphi \, dS. \tag{3.24}$$

3.2.3 Phase Field ϕ

Starting from equation (2.26), we respectively apply the approximations of

$$\frac{\partial \phi}{\partial t} \approx \frac{1}{\Delta t}\left(\phi^{(n)} - \phi^{(n-1)}\right) \tag{3.25}$$

to the time variable t,

$$\phi \approx \theta\phi^{(n)} + (1-\theta)\phi^{(n-1)} \tag{3.26}$$

to the diffusion term $\nabla^2 \phi$,

$$\phi \approx \phi^{(n-1)} \tag{3.27}$$

on the rest of terms containing ϕ, and

$$\sigma \approx \sigma^{(n-1)} \tag{3.28}$$

to get

$$\frac{1}{\Delta t}\left(\phi^{(n)} - \phi^{(n-1)}\right)$$

$$= \frac{1}{\tau_\phi}\left(\frac{\delta^2}{2}\nabla^2\left(\theta\phi^{(n)} + (1-\theta)\phi^{(n-1)}\right) + \phi^{(n-1)}\left(1 - \left(\phi^{(n-1)}\right)^2\right)\right) \tag{3.29}$$

$$- k_R\frac{D_a\Omega}{k_B T}\nabla\sigma^{(n-1)}\cdot\phi^{(n-1)}.$$

Collect $\phi^{(n)}$ to the left hand, multiply test function φ to the both hands and integrate over

the conductor domain Ω_W, we have

$$\int_{\Omega_W}\phi^{(n)}\varphi\,dV - \int_{\Omega_W}\frac{\delta^2\theta\Delta t}{2\tau_\phi}\nabla^2\phi^{(n)}\varphi\,dV$$

$$= \int_{\Omega_W}\phi^{(n-1)}\varphi\,dV + \int_{\Omega_W}\frac{\delta^2(1-\theta)\Delta t}{2\tau_\phi}\nabla^2\phi^{(n-1)}\varphi\,dV$$

$$+ \int_{\Omega_W}\frac{\Delta t}{\tau_\phi}\phi^{(n-1)}\left(1 - \left(\phi^{(n-1)}\right)^2\right)\varphi\,dV \tag{3.30}$$

$$- \int_{\Omega_W}\Delta t k_R\frac{D_a\Omega}{k_B T}\nabla\sigma^{(n-1)}\cdot\phi^{(n-1)}\varphi\,dV,$$

which can be simplified by plugging in

$$\int_{\Omega_W}\nabla^2\phi\varphi\,dV = \int_{\Omega_W}\nabla\cdot(\nabla\phi\varphi)\,dV - \int_{\Omega_W}\nabla\phi\cdot\nabla\varphi\,dV$$

$$= \int_{\partial\Omega_W}\nabla\phi\varphi\cdot\vec{n}\,dS - \int_{\Omega_W}\nabla\phi\cdot\nabla\varphi\,dV \tag{3.31}$$

$$= -\int_{\Omega_W}\nabla\phi\cdot\nabla\varphi\,dV$$

and turn into

$$\int_{\Omega_W}\phi^{(n)}\varphi\,dV + \int_{\Omega_W}\frac{\delta^2\theta\Delta t}{2\tau_\phi}\nabla\phi^{(n)}\cdot\nabla\varphi\,dV$$

$$= \int_{\Omega_W}\phi^{(n-1)}\varphi\,dV - \int_{\Omega_W}\frac{\delta^2(1-\theta)\Delta t}{2\tau_\phi}\nabla\phi^{(n-1)}\cdot\nabla\varphi\,dV$$

$$+ \int_{\Omega_W}\frac{\Delta t}{\tau_\phi}(\phi^{(n-1)} - (\phi^{(n-1)})^3)\varphi\,dV \tag{3.32}$$

$$- \int_{\Omega_W}\Delta t k_R\frac{D_a\Omega}{k_B T}\nabla\sigma^{(n-1)}\cdot\nabla\phi^{(n-1)}\varphi\,dV.$$

50

To discretize time variable t in equation (2.26), we respectively substitute $\partial\phi/\partial t$ with $\frac{1}{\Delta t}\left(\phi^{(n)} - \phi^{(n-1)}\right)$, ϕ with $\theta\phi^{(n)} + (1-\theta)\phi^{(n-1)}$ to numerically solve the equation based on Euler method. The superscript n is for time step index. By applying divergence theorem and collect $\phi^{(n)}$ related terms on the left hand side, we have the time-discretized weak form of equation (2.26)

Here θ is the Euler method controller ($\theta = 0$: backward Euler; $\theta = 1$: forward Euler; $\theta = \frac{1}{2}$: mixed Euler or Crank-Nicolson method). We use Crank-Nicolson method ($\theta = \frac{1}{2}$) since it has the stability like backward Euler method and also higher (second order) accuracy. Δt is the time discretization step which is a knob for controlling solution accuracy.

3.2.4 Temperature T

By applying similar approximations

$$\frac{\partial T}{\partial t} \approx \frac{1}{\Delta t}\left(T^{(n)} - T^{(n-1)}\right) \tag{3.33}$$

and

$$T \approx \theta T^{(n)} + (1-\theta)T^{(n-1)}, \tag{3.34}$$

equation (2.7) becomes

$$\rho_M C_p \frac{T^{(n)} - T^{(n-1)}}{\Delta t} - \left(\theta\nabla\cdot(\kappa\nabla T^{(n)}) + (1-\theta)\nabla\cdot(\kappa\nabla T^{(n-1)})\right) = g_T \tag{3.35}$$

which can be rearranged by collecting $T^{(n)}$ to the left hand side as

$$\rho_M C_p T^{(n)} - \theta\Delta t\nabla\cdot(\kappa\nabla T^{(n)}) = \rho_M C_p T^{(n-1)} + (1-\theta)\Delta t\nabla\cdot(\kappa\nabla T^{(n-1)}) + \Delta t g_T. \tag{3.36}$$

Multiply the testing function φ to the both hands of the above equation and integrate over the temperature domain Ω_T, we have

$$
\int_{\Omega_T} \rho_M C_p T^{(n)} \varphi \, dV - \int_{\Omega_T} \theta \Delta t \nabla \cdot (\kappa \nabla T^{(n)}) \varphi \, dV
$$

$$
= \int_{\Omega_T} \rho_M C_p T^{(n-1)} \varphi \, dV + \int_{\Omega_T} (1 - \theta) \Delta t \nabla \cdot (\kappa \nabla T^{(n-1)}) \varphi \, dV + \int_{\Omega_T} \Delta t g_T \varphi \, dV. \tag{3.37}
$$

Noticing that the rule of integration by part and divergence theorem result the transformation of

$$
\int_{\Omega_T} \nabla \cdot (\kappa \nabla T) \, \varphi \, dV
$$

$$
= \int_{\Omega_T} \nabla \cdot (\kappa \nabla T \varphi) \, dV - \int_{\Omega_T} \kappa \nabla T \cdot \nabla \varphi \, dV \tag{3.38}
$$

$$
= \int_{\partial \Omega_T} \kappa \nabla T \varphi \cdot \vec{n} \, dS - \int_{\Omega_T} \kappa \nabla T \cdot \nabla \varphi \, dV,
$$

which can be simplified by plugging in the Robin boundary condition (2.8) as

$$
\int_{\Omega_T} \nabla \cdot (\kappa \nabla T) \, \varphi \, dV
$$

$$
= \int_{\partial \Omega_T} \kappa \nabla T \varphi \cdot \vec{n} \, dS - \int_{\Omega_T} \kappa \nabla T \cdot \nabla \varphi \, dV \tag{3.39}
$$

$$
= \int_{\partial \Omega_T} h(T_{\text{ext}} - T) \varphi \, dS - \int_{\Omega_T} \kappa \nabla T \cdot \nabla \varphi \, dV,
$$

the temperature weak form can be further transformed into

$$
\int_{\Omega_T} \rho_M C_p T^{(n)} \varphi \, dV - \theta \Delta t \left(\int_{\partial \Omega_T} h(T_{\text{ext}} - T) \varphi \, dS - \int_{\Omega_T} \kappa \nabla T \cdot \nabla \varphi \, dV \right)
$$

$$
= \int_{\Omega_T} \rho_M C_p T^{(n-1)} \varphi \, dV + (1 - \theta) \Delta t \left(\int_{\partial \Omega_T} h(T_{\text{ext}} - T) \varphi \, dS - \int_{\Omega_T} \kappa \nabla T \cdot \nabla \varphi \, dV \right) \tag{3.40}
$$

$$
+ \int_{\Omega_T} \Delta t g_T \varphi \, dV.
$$

According to Joule's law, the heat source g_T is expressed as

$$
g_T = j^2 \rho \approx \left(j^{(n-1)} \right)^2 \rho \left(\phi^{(n-1)}, T^{(n-1)} \right) \tag{3.41}
$$

where the current density magnitude $\vec{j} = -\nabla U/\rho$ is derived from electrical potential and the electrical resistivity ρ is a function of temperature T and the copper-void phase field ϕ. Thus the time-discretized weak form of temperature variable T is derived:

$$
\begin{aligned}
&\int_{\Omega_T} \rho_M c_p T^{(n)} \varphi dV + \int_{\Omega_T} k_T \Delta t \theta \nabla T^{(n)} \cdot \nabla \varphi dV + \int_{\partial \Omega_T} \Delta t \theta h T^{(n)} \phi dS \\
&= \int_{\Omega_T} \rho_M c_p T^{(n-1)} \varphi dV - \int_{\Omega_T} k_T \Delta t (1-\theta) \nabla T^{(n-1)} \cdot \nabla \varphi dV \\
&\quad - \int_{\partial \Omega_T} \Delta t (1-\theta) h T^{(n-1)} \phi dS + \int_{\partial \Omega_T} \Delta t h T_{\text{ext}} \phi dS \\
&\quad + \int_{\Omega_T} \left(j^{(n-1)} \right)^2 \rho \left(\phi^{(n-1)}, T^{(n-1)} \right) \Delta t \varphi dV.
\end{aligned}
\tag{3.42}
$$

3.2.5 Hydrostatic Stress σ

We start from the reorganized equation (2.32)

$$
\frac{\partial \sigma}{\partial t} = (w_B + w_C) \nabla \cdot \left(\frac{D_a B \Omega}{k_B T} \nabla \sigma \right) - \left(\frac{w_V}{\tau_{\text{void}}} + \frac{w_B v k_\tau}{\delta} \right) \sigma,
\tag{3.43}
$$

which can be approximated as

$$
\begin{aligned}
\frac{\sigma^{(n)} - \sigma^{(n-1)}}{\Delta t} &= \theta(w_B + w_C) \nabla \cdot \left(\frac{D_a B \Omega}{k_B T} \nabla \sigma^{(n)} \right) \\
&\quad + (1-\theta)(w_B + w_C) \nabla \cdot \left(\frac{D_a B \Omega}{k_B T} \nabla \sigma^{(n-1)} \right) \\
&\quad - \left(\frac{w_V}{\tau_{\text{void}}} + \frac{w_B v k_\tau}{\delta} \right) \left(\theta \sigma^{(n)} + (1-\theta) \sigma^{(n-1)} \right)
\end{aligned}
\tag{3.44}
$$

by plugging in

$$
\frac{\partial \sigma}{\partial t} \approx \frac{\sigma^{(n)} - \sigma^{(n-1)}}{\Delta t}
\tag{3.45}
$$

and

$$
\sigma \approx \theta \sigma^{(n)} + (1-\theta) \sigma^{(n-1)}.
\tag{3.46}
$$

Multiply testing function φ on both sides and integrate over the stress domain Ω_C we have

$$
\int_{\Omega_C} \frac{\sigma^{(n)} - \sigma^{(n-1)}}{\Delta t} \varphi \, dV = \int_{\Omega_C} \theta(w_B + w_C) \nabla \cdot \left(\frac{D_a B\Omega}{k_B T} \nabla \sigma^{(n)} \right) \varphi \, dV
$$
$$
+ \int_{\Omega_C} (1 - \theta)(w_B + w_C) \nabla \cdot \left(\frac{D_a B\Omega}{k_B T} \nabla \sigma^{(n-1)} \right) \varphi \, dV \qquad (3.47)
$$
$$
- \int_{\Omega_C} \left(\frac{w_V}{\tau_{\text{void}}} + \frac{w_B v k_\tau}{\delta} \right) \left(\theta \sigma^{(n)} + (1 - \theta)\sigma^{(n-1)} \right) \varphi \, dV.
$$

By applying the rule of integration by parts and divergence theorem, the diffusion (second derivative) term can be simplified as

$$
\int_{\Omega_C} \nabla \cdot \left(\frac{D_a B\Omega}{k_B T} \nabla \sigma \right) \varphi \, dV
$$
$$
= \int_{\Omega_C} \nabla \cdot \left(\frac{D_a B\Omega}{k_B T} \nabla \sigma \varphi \right) dV - \int_{\Omega_C} \frac{D_a B\Omega}{k_B T} \nabla \sigma \cdot \nabla \varphi \, dV
$$
$$
= \int_{\partial\Omega_C} \frac{D_a B\Omega}{k_B T} \nabla \sigma \varphi \cdot \vec{n} \, dS - \int_{\Omega_C} \frac{D_a B\Omega}{k_B T} \nabla \sigma \cdot \nabla \varphi \, dV \qquad (3.48)
$$
$$
= \int_{\partial\Omega_C} \frac{D_a B}{k_B T} e Z \rho j_N \varphi \, dS - \int_{\Omega_C} \frac{D_a B\Omega}{k_B T} \nabla \sigma \cdot \nabla \varphi \, dV,
$$

where the last step is done by plugging in the current density induced hydrostatic stress boundary condition (1.2). Substitute the diffusion term and simplify equation (3.47) we

54

have

$$\int_{\Omega_C} \sigma^{(n)} \varphi \, dV - \int_{\Omega_C} \sigma^{(n-1)} \varphi \, dV$$

$$= \Delta t \theta \int_{\Omega_C} (w_B + w_C) \nabla \cdot \left(\frac{D_a B \Omega}{k_B T} \nabla \sigma^{(n)} \right) \varphi \, dV$$

$$+ \Delta t (1 - \theta) \int_{\Omega_C} (w_B + w_C) \nabla \cdot \left(\frac{D_a B \Omega}{k_B T} \nabla \sigma^{(n-1)} \right) \varphi \, dV$$

$$- \Delta t \int_{\Omega_C} \left(\frac{w_V}{\tau_{\text{void}}} + \frac{w_B v k_\tau}{\delta} \right) \left(\theta \sigma^{(n)} + (1 - \theta) \sigma^{(n-1)} \right) \varphi \, dV$$

$$= \Delta t \theta \left(\int_{\partial \Omega_C} (w_B + w_C) \frac{D_a B}{k_B T} e Z \rho j_N \varphi \, dS - \int_{\Omega_C} (w_B + w_C) \frac{D_a B \Omega}{k_B T} \nabla \sigma^{(n)} \cdot \nabla \varphi \, dV \right)$$

$$+ \Delta t (1 - \theta) \left(\int_{\partial \Omega_C} (w_B + w_C) \frac{D_a B}{k_B T} e Z \rho j_N \varphi \, dS - \int_{\Omega_C} (w_B + w_C) \frac{D_a B \Omega}{k_B T} \nabla \sigma^{(n-1)} \cdot \nabla \varphi \, dV \right)$$

$$- \Delta t \int_{\Omega_C} \left(\frac{w_V}{\tau_{\text{void}}} + \frac{w_B v k_\tau}{\delta} \right) \left(\theta \sigma^{(n)} + (1 - \theta) \sigma^{(n-1)} \right) \varphi \, dV$$

$$= \Delta t \int_{\partial \Omega_C} (w_B + w_C) \frac{D_a B}{k_B T} e Z \rho j_N \varphi \, dS$$

$$- \Delta t \theta \int_{\Omega_C} (w_B + w_C) \frac{D_a B \Omega}{k_B T} \nabla \sigma^{(n)} \cdot \nabla \varphi \, dV$$

$$- \Delta t (1 - \theta) \int_{\Omega_C} (w_B + w_C) \frac{D_a B \Omega}{k_B T} \nabla \sigma^{(n-1)} \cdot \nabla \varphi \, dV$$

$$- \Delta t \int_{\Omega_C} \left(\frac{w_V}{\tau_{\text{void}}} + \frac{w_B v k_\tau}{\delta} \right) \left(\theta \sigma^{(n)} + (1 - \theta) \sigma^{(n-1)} \right) \varphi \, dV,$$

$$(3.49)$$

which can be simplified by collecting the terms relating to $\sigma^{(n)}$ to the left hand side and acquire the final weak form of hydrostatic stress σ:

$$
\begin{aligned}
& \int_{\Omega_C} \left(1 + \Delta t \theta \left(\frac{w_V}{\tau_{\text{void}}} + \frac{w_B v k_\tau}{\delta}\right)\right) \sigma^{(n)} \varphi \, dV \\
& + \Delta t \theta \int_{\Omega_C} (w_B + w_C) \frac{D_a B \Omega}{k_B T} \nabla \sigma^{(n)} \cdot \nabla \varphi \, dV \\
& = \int_{\Omega_C} \left(1 - \Delta t (1 - \theta) \left(\frac{w_V}{\tau_{\text{void}}} + \frac{w_B v k_\tau}{\delta}\right)\right) \sigma^{(n)} \varphi \, dV \\
& - \Delta t (1 - \theta) \int_{\Omega_C} (w_B + w_C) \frac{D_a B \Omega}{k_B T} \nabla \sigma^{(n-1)} \cdot \nabla \varphi \, dV \\
& + \Delta t \int_{\partial \Omega_C} (w_B + w_C) \frac{D_a B}{k_B T} e Z \rho j_N \varphi \, dS.
\end{aligned}
\tag{3.50}
$$

3.3 Time Discretization

When solving for the time dependent physics systems, the time discretization is driven by the difference between the tempted solution and the last step solution. The time dependent solver starts from a user configured time step $\Delta t^{(0)}$ and changes according to the solution difference. From time step $\Delta t^{(n-1)}$ to $\Delta t^{(n)}$, the default change rule is to increase by a prescribed ratio

$$
\Delta t^{(n)} = \lambda_{\text{inc}} \Delta t^{(n-1)}
\tag{3.51}
$$

where $\lambda_{\text{inc}} > 0$ is a prescribed increasing factor. If such time step $\Delta t^{(n)}$ leads to a solution that is growing too fast, then the solver tries time step size

$$
\Delta t^{(n)} = \lambda_{\text{inc}} \lambda_{\text{dec}}^p \Delta t^{(n-1)}
\tag{3.52}
$$

where $0 < \lambda_{\text{dec}} < 1$ is a prescribed decreasing factor, to acquire another solution and expects a smaller difference. This process is repetitive from $p = 1$ onward, until a solution that is

close enough to the last step solution is acquired. [3]

To precisely define the vague term *close enough* used above, let

$$U^{(n)} = [U_1^{(n)}, U_2^{(n)}, \cdots, U_m^{(n)}]$$

$$\phi^{(n)} = [\phi_1^{(n)}, \phi_2^{(n)}, \cdots, \phi_m^{(n)}]$$

$$\sigma^{(n)} = [\sigma_1^{(n)}, \sigma_2^{(n)}, \cdots, \sigma_m^{(n)}]$$

$$T^{(n)} = [T_1^{(n)}, T_2^{(n)}, \cdots, T_m^{(n)}]$$

(3.53)

respectively denote the four physics systems at time step n, where the underscripted components are nodal values of spatial discretized finite elements, the following method is used to determine the closeness between two solutions that are adjacent in time, for example, $\phi^{(n-1)}$ and $\phi^{(n)}$. Two solutions $u^{(n-1)}$ and $u^{(n)}$ are defined as *close enough* if

$$q\left(u_i^{(n)}, u^{(n-1)}\right) = 1 \quad \text{for all } 1 \le i \le m,$$

(3.54)

where function q is an element-wise distance function

$$q(u, u') = \begin{cases} 1 & \text{if } |u - u'| < \varepsilon_{\text{abs}} \text{ or } \left|\frac{u-u'}{u}\right| < \varepsilon_{\text{rel}}, \\ 0 & \text{otherwise.} \end{cases}$$

(3.55)

Here, two variables ε_{abs} and ε_{rel} are predefined to control the tolerance of the closeness, separately for each physics systems.

With the above definition, the time discretization is driven and determined by solution trials, which guarantees the smoothness of the solved physics variables.

---

[3]Of course, the time step cannot go neither too small nor too big, the limit of which will be configured by user so that any time step increase or decrease beyond this, implying potential unexpected numerical issues, will incur a warning or termination of simulation.

3.4 Phase Field Conservation

As defined above, the phase field variable ϕ implicitly models the copper void boundary, which is driven by other factors such as the hydrostatic stress σ. To align with realistic boundary behavior, the boundary should remain unchanged when there is nothing driving it. That is, for ϕ governed by

$$\frac{\partial \phi}{\partial t} = \frac{1}{\tau_\phi} \left(\frac{\delta^2}{2} \nabla^2 \phi + (\phi - \phi^3) \right) - \vec{v} \cdot \nabla \phi, \tag{3.56}$$

should satisfy

$$\frac{\partial \phi}{\partial t} = 0 \tag{3.57}$$

when $\vec{v} = 0$. This means everywhere on variable ϕ over space remains unchanged at time t. In implementation, we maintain a weaker constraint

$$\int_{\Omega_C} \frac{\partial \phi}{\partial t} \, dV = \frac{\partial}{\partial t} \int_{\Omega_C} \phi \, dV = 0, \tag{3.58}$$

when there is no driving source exists. The consequence is that the implied void volume does not change at time t, which is an important requirement to build an accurate void evolution tool.

Ideally, the above-mentioned constraint requires

$$\frac{\partial \phi}{\partial t} = \frac{1}{\tau_\phi} \left(\frac{\delta^2}{2} \nabla^2 \phi + (\phi - \phi^3) \right) = 0, \tag{3.59}$$

which will be examined in the following. We will first show that

$$\int_{\Omega_C} \nabla^2 \phi \, dV = 0 \tag{3.60}$$

is true, and then demonstrate the cases where

$$\int_{\Omega_C} (\phi - \phi^3) \, dV = 0 \tag{3.61}$$

58
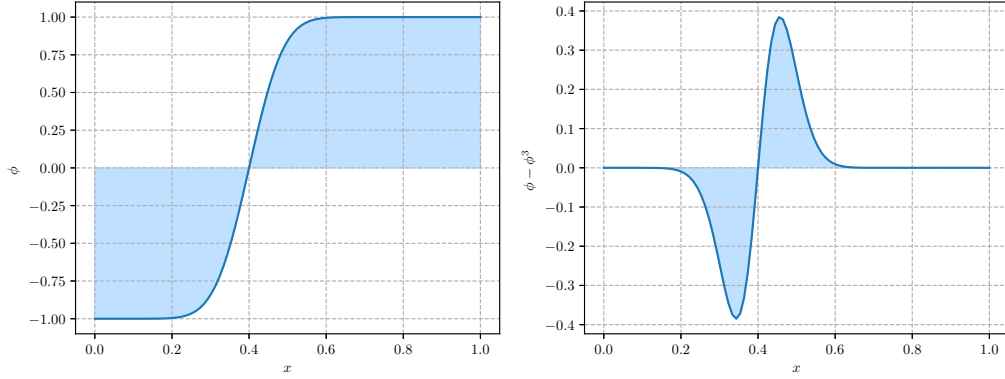
Figure 3.4: 1D phase field conservation.

does not always hold in 2D and 3D dimensions. Eventually a complement term is derived and added to (3.59) for it to remain true in 2D and 3D as well.

Apply divergence theorem to (3.60), we have

$$\int_{\Omega_C} \nabla^2 \phi \, dV = \int_{\partial\Omega_C} \nabla\phi \cdot \vec{n} \, dS = 0 \tag{3.62}$$

as $\nabla\phi \cdot \vec{n}$ is prescribed to 0 everywhere on the boundary. Being similar to thermal energy conservation within a isolated domain, this holds true in arbitrary number of dimensions. Comparing to this, equation (3.61) generally only hold true in 1D, but not in 2D or 3D. As shown in figure 3.4, a perfect complemented peak and valley pair is induced by the ϕ slope. In this case, the integral remains zero.

In 2D where copper-void boundary is not straight, as shown in figure 3.5, equation (3.61) does not hold in general. As demonstrated, the ϕ slope induces non-zero-valued bands with asymmetric circumferences, and consequently different areas. This will lead to non zero integral[4] of term $\phi - \phi^3$ over the domain. These asymmetric bands can also be

---

[4]The numerical result is

$$\int_0^1 \int_0^1 (\phi - \phi^3) \, dx \, dy \approx 6.8 \times 10^{-3}$$
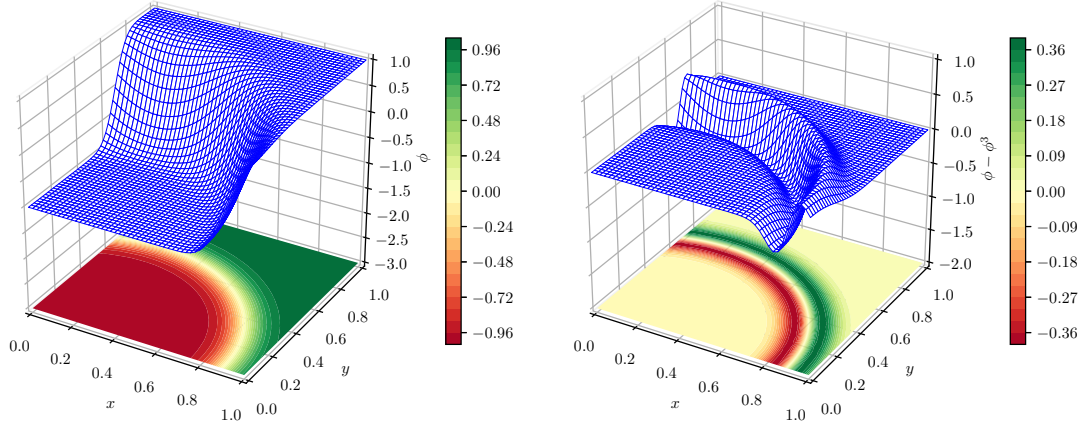
Figure 3.5: 2D phase field conservation.

observed in 3D cases. Therefore, the integral will be larger than zero in the demonstration, which contributes to the time differential and eventually leads to a shrinking void, instead of the expected, fixed-volume void.

As a complement to cancel such silent self driving component, a term ϕ_C is introduced such that

$$\int_{\Omega_C} \left(\phi - \phi^3 - \phi_C \right) dV = 0 \tag{3.63}$$

holds true in any number of dimensions, where

$$\phi_C = \frac{\int_{\Omega_C} \left(\phi - \phi^3 \right) dV}{\int_{\Omega_C} dV}. \tag{3.64}$$

With this modification, equation (3.59) becomes

$$\frac{\partial \phi}{\partial t} = \frac{1}{\tau_\phi} \left(\frac{\delta^2}{2} \nabla^2 \phi + \phi - \phi^3 - \frac{\int_{\Omega_C} \left(\phi - \phi^3 \right) dV}{\int_{\Omega_C} dV} \right) - \vec{v} \cdot \nabla \phi, \tag{3.65}$$

for the ϕ used in figure 3.5, where

$$\phi = \mathrm{erf} \left(10 \left(\sqrt{(x - 0.2)^2 + y^2} - 0.7 \right) \right).$$

which values zero in arbitrary number of dimensions when $\vec{v} = 0$. Note that this modification is tightly coupled with finite element implementation so the complement term ϕ_C is omitted in most of the context throughout this work, while it remains critical for maintaining a conservative phase field.

3.5 Stationary Analysis

As described in section 2.7, time derivatives of T and σ vanish at steady state. Therefore, in stationary analysis, temperature and hydrostatic stress weak forms are modified accordingly.

3.5.1 Stationary Weak Form of Temperature T

The steady state form described in equation (2.43) and (2.44) is a simple Laplace problem with Robin boundary condition. Starting from (2.43) multiplied by test function φ and integrated over domain Ω_T

$$- \int_{\Omega_T} \kappa \nabla^2 T \varphi \, dV = \int_{\Omega_T} j^2 \rho_M \, dV, \tag{3.66}$$

which can be simplified, by applying the rule of integration by parts and divergence theory on the left hand size, into

$$- \int_{\Omega_T} \kappa \nabla^2 T \varphi \, dV = - \int_{\partial\Omega_T} \kappa \nabla T \cdot \vec{n} \varphi \, dS + \int_{\Omega_T} \kappa \nabla T \cdot \nabla \varphi \, dV = \int_{\Omega_T} j^2 \rho_M \, dV. \tag{3.67}$$

It can be further simplified by substituting the boundary integral term using the Robin boundary condition (2.44) into the final weak form

$$\int_{\partial\Omega_T \cap \Gamma_T} hT\varphi \, dS + \int_{\Omega_T} \kappa \nabla T \cdot \nabla \varphi \, dV = \int_{\partial\Omega_T \cap \Gamma_T} hT_{\text{ext}}\varphi \, dS + \int_{\Omega_T} j^2 \rho_M \, dV. \tag{3.68}$$

3.5.2 Stationary Weak Form of Hydrostatic Stress σ

Similar to the stationary temperature analysis, the hydrostatic steady state analysis is a Laplace problem, but with Neumann boundary conditions. Apply divergence theorem and integration by parts on equation (2.39), we have the integral equity for arbitrary test function in the solution space

$$\int_{\Omega_C} \frac{D_a B \Omega}{k_B T} \nabla \sigma \cdot \nabla \varphi \, dV = \int_{\partial \Omega_C} \frac{D_a B \Omega}{k_B T} \nabla \sigma \cdot \vec{n} \, dS, \tag{3.69}$$

which can be simplified by plugging in the boundary conditions (2.40) and (2.41) into the final steady state weak form

$$\int_{\Omega_C} \frac{D_a B \Omega}{k_B T} \nabla \sigma \cdot \nabla \varphi \, dV = \int_{\partial \Omega_C \cap \Gamma_I} \frac{D_a B \Omega}{k_B T} j \, dS, \tag{3.70}$$

where j is the current density magnitude prescribed in the copper interconnect terminals.

3.5.3 Adaptive local mesh refinement

When solving for FEM problems, adaptive local mesh refinement is crucial to achieve desired precision in detail-rich parts of a domain without wasting computation resources in portions with relatively smoother solutions. Essentially, adaptive local mesh refinement allows relatively coarse initial mesh, and then refines the mesh according to the analysis of the solution of each time step.

On each time step, which is solved using backward Euler method, each physical system is updated according to the dependencies. We use Rothe's method [41] to discretize the time variable first and then the spatial variables (meshing). Given the nature of the moving copper-void boundary (implied by the phase field), it requires varying the location of

highly locally refined mesh over time. Therefore method of lines [59] which discretize space variables first and then time variable, not allowing time-dependent local mesh refinement, is not suitable for the implementation of the proposed method.

Kelly estimator [43] is used to do posterior error estimation on finite elements. According to the estimated error, adaptive refinement is conducted on the original coarse grid, which is meshed by Gmsh[25]. Since the three systems have solutions in very different scales, the synthesized error can easily be dominated by one system. We use a normalize-then-scale method to mitigate this problem and provide an adjustable knob to fine-tune the adaptive mesh refinement. Respectively, let $\hat{\epsilon}_P$, $\hat{\epsilon}_E$, and $\hat{\epsilon}_S$ denote the estimated error vector of copper-void phase field, electrical potential, and hydrostatic stress, which are all normalized to $[0, 1]$, we synthesis the final error vector using a weighted sum

$$\bar{\epsilon} = c_P \hat{\epsilon}_P + c_E \hat{\epsilon}_E + c_S \hat{\epsilon}_S, \tag{3.71}$$

where c_P, c_E, and c_S are weighting factors for the physics systems.

(a) Before refinement   (b) After refinement
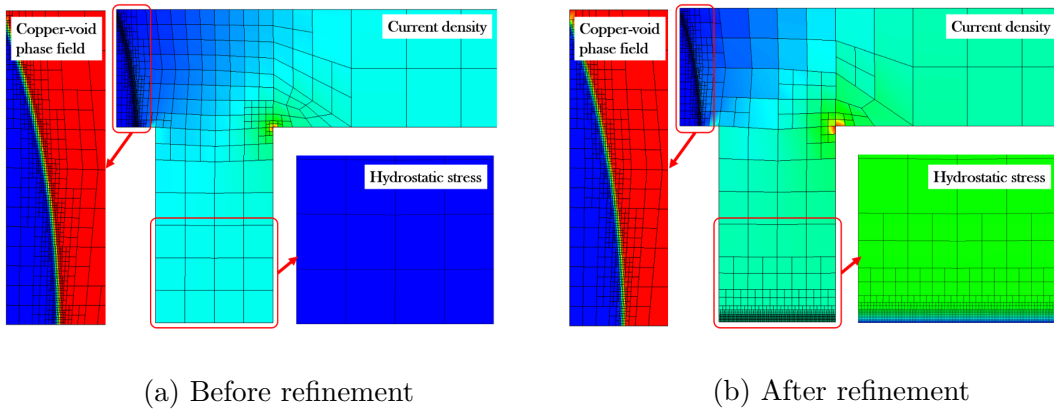
Figure 3.6: Adaptive local refinement of hydrostatic stress solution with phase field fixed.

Figure 3.6 shows the outcome of the refinement. Comparing to the previous refine-

ment step ($t = 0$), hydrostatic stress is evenly distributed over the copper so the mesh does not get coarsened except for the copper-void boundary and a high current density corner. When the stress starts to grow, the normalize-then-scale algorithm synthesizes the errors generated by Kelly estimator and effectively indicates the correct position to be refined. The refinement behavior is determined by the solution while time-dependent analysis is being performed. This is helpful as the characteristics of underlying physical systems are not required to be considered when generating the initial mesh. Instead, only very coarse mesh that merely describes the structure is needed. It is beneficial since the interfacing to automatic interconnect structure generation can be dramatically simplified.

3.6 Summary

This chapter presents the finite element method employed in this thesis. Finite element method is introduced by transforming a sample Poisson equation into a solvable finite element equations system. The major part of this chapter is the derivation of variational weak forms of the four physics systems as introduced in chapter 2, followed by the description of time discretization and a minor fix in order to satisfy phase field conservation. Stationary analysis equations of the weak forms are then derived.

Chapter 4

Software Implementation

Software architecture will firstly be discussed, where modularized and loosely coupled physics systems are focused on. The user interface is thoroughly discussed since it is critical for such a technology computer aided design (TCAD) software can be scripted and automated to couple with upstream and downstream software, as well as easily operated by ad-hoc simulations initiated by humans. The user interface discussion will include the software working flow, input configuration composing, and input geometry construction methods.

4.1 Software Architecture

In this section two different implementations based on deal.II and FEniCS are separately discussed regarding their working flow and simulated physics systems.
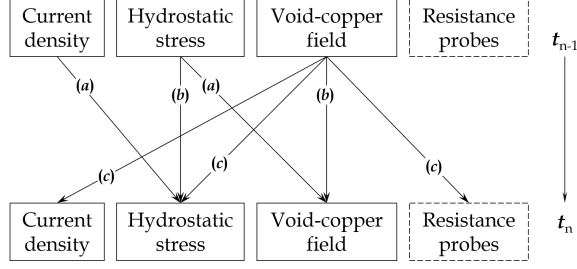
† Labels in parenthesis denote the type of major contribution: (a) source term; (b)

diffusion term; (c) material indicator.

Figure 4.1: Block diagram of time dependent FEM systems coupling.
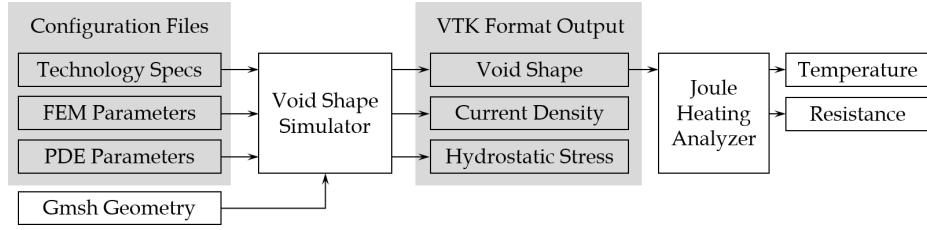
Figure 4.2: The algorithm flow of the proposed postvoiding EM FEM simulator.

4.1.1 Deal.II Based Implementation

The proposed method is implemented using the deal.II FEM library[2]. The algorithm flow of the proposed postvoiding EM FEM solver is shown in figure 4.2. The input mesh is in Gmsh [25] format version 2.2. Only a coarse mesh that is enough to describe the geometry structure is required. The mesh is then refined according to the analysis of the solution. All the parameters, including technology parameters, finite element configurations, solver setups, and PDE configurations, can be easily adjusted by specifying in the input parameter file. The simulation result is in VTK format [77], which is widely used in scientific computing community for scalable and efficient post-processing and rendering.

The software is modularized and divided into the following physical systems with minimized coupling: electrical potential system, phase field system, hydrostatic stress system, and resistance probes system. The resistance probes system are similar to electrical potential system but they only apply voltage on certain user-specified boundary pairs in order to measure resistance, without affecting the hydrostatic stress.

4.1.2   FEniCS Based Implementation

All four coupled physics variables — hydrostatic stress σ, material phase field ϕ, temperature T, and voltage U — are designed to be depend on each others only by means of data transferring. But they are highly decoupled in terms of logic organization, allowing any number of resistance probes, acting as same-level physics variables as the four, to be easily configured and plugged in. The data dependency between each time step is depicted in figure 4.3, where at time step n, all the systems (related physics equation) depend on the previous time step $n-1$ to solve. Take temperature as an example, it serves as a necessary prerequisite of itself because of the thermal diffusion nature over time. Temperature also has impacts on voltage (current density analysis and all resistance probes) by affecting material electrical conductivity. It also has impacts on hydrostatic stress by affecting material activation energy and diffusion coefficient, and on phase field by affecting material diffusion coefficient in a similar way.

As mentioned above, user is able to setup any number of resistance probes by composing corresponding configuration inputs. The simulator will set aside a separate physical system, only focusing on the wire domain Ω_W, and calculating the resistance between the specified anode and cathode. For each configured resistance probe, the simulator will ground

the cathode and apply a fixed inward normal current density on the anode, and then calculate the effective resistance by applying Ohm's law. As the arrows implies in figure 4.3, all resistance calculations will be separately done. Thereby their Joule heating effect will not be coupled with the temperature variable, nor their electromigration effect with the hydrostatic stress variable.
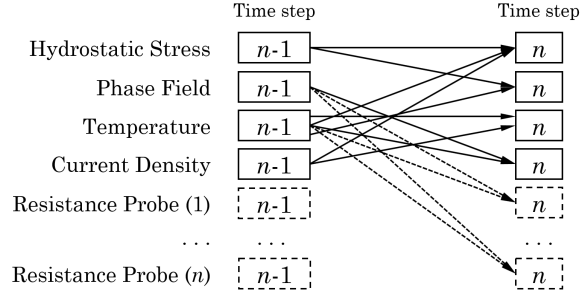
Figure 4.3: Block diagram of time dependent FEM systems coupling.

The proposed method is implemented using FEniCS FEM library [1], programmed in Python 3. The work flow can be divided into the following parts:

1. The user prepares for the input structure in XML format supported by FEniCS. FEniCS supports conversion from a variety of widely-used geometry description format to the XML format that it supports. Currently in the experiments, Gmsh [25] format version 2.2 is used to describe the input structure and create mesh. Thanks to the `Include` directive that Gmsh supports, the input geometry is allowed to be separated into two parts, dimension definition and actual code describing the geometry and physical entities. This allows the geometry to be programmatically controlled for a parameter sweep if needed.

2. The user prepares for the configuration for the simulator. The configuration file

contains all the adjustable parameters, include boundary conditions of the system

PDEs, resistance probe deployments, a large number of material parameters such as

bulk modulus B and resistance ρ, and finite element method parameters such as time

step control, solver selection, error control, etc. The user configuration is also designed

to allow nesting and overwriting, which is for the sake of the automation of parameter

sweep as well.

3. The user runs the simulator program. The simulator can be configured to period-
   ically generate output as well as snapshots. The output is in VTK format [77],
   which is widely used in scientific computing community for scalable and efficient post-
   processing and rendering. The snapshots are designed to allow user save-and-load the
   simulation if to rerun the simulator with different parameters, such as finer time dis-
   critization.

4. The user performs analysis on the results. Geometry dependent results (scalar/vector
   data attached to mesh nodes) are stored as VTK format for further analysis or post-
   process.

4.2 Geometry Construction

Both Deal.II and FEniCS implementations use Gmsh file format to describe in-

put geometries. Gmsh format is capable to describe geometries in 2D and 3D. [1] User is

responsible for creating geometry files, as well as defining/labeling domains and boundaries.

---

[1] In fact, Gmsh format always use three components x, y, and z in point coordinates. Thus 1D and 2D geometries are just those with unused coordinates zeroed out on all points.

The reason Gmsh file format is chosen as the input geometry description method has several folds:

1. Gmsh file format specification is open. This allows it to be easily integrated with other software to serve as a compatible data interface.

2. Gmsh file is human readable. This allows easy modification on existing geometries in any depth of detail, comparing to those formats relying on graphics front-end editors. Although large geometries can require great effort to compose, manageable and scalable code style can help to make it practical.

3. Gmsh file format supports flexible programming compatibilities. Gmsh file format allows user-defined variable, condition statement, loop statement, etc. The support of user-defined variables enables templated geometries, using which user can instantiate a template by feeding different key dimensions to get different result geometries.

4. Gmsh file is in ASCII format, which helps user version control a simulation configuration. [2]

A Gmsh file format can be constructed using two different methods, bottom up and top down. These two methods have their own advantages and disadvantages, which are described and summarized in the following.

---

[2]For example, COMSOL supported/exported geometries [46] are in binary format that is not capable for version controlling and back-tracing.

4.2.1 Bottom Up Construction

As the title implies, bottom up construction means the description starts from defining discrete points, then lines or curves connecting them, then surfaces surrounded by curves, and finally volumes surrounded by surfaces. By building from bottom up, user has the full control of parameterize the coordinate and characteristic length[3] of every key point in the geometry. Furthermore, as the geometry is built up from points through curves, surfaces to volumes, user also can assign detailed mesh specifications. The reason is that, all the sub-entities (points, curves, surfaces), can be explicitly labeled, so user can refer any previous labeled sub-entities and assign properties onto them.

Figure 4.4 and 4.5 are examples of building 13 segment interconnects from bottom up. The geometry is templated, accepting dimensions of every segments widths and lengths. These two geometries share one templates that requiring major effort, while the instantiation code is nothing but solely specifying the required dimensions.

The disadvantage of this method is, for complexed geometries, especially in 3D cases, it is extremely tedious to write a templated geometry. The first reason is that user has to explicitly number the sub-entities, which can become very unmanageable if the user does not carefully follow a clear and scalable coding convention. The second reason is that for 3D geometries, a complexed volume might be composited by many different surfaces, while user has to specify a volume by using all surrounding surfaces. This can be difficult while picturing the actual entity in mind while coding.

As templating and instantiating example, code list 4.1 shows the Gmsh template

---

[3]Characteristic length is a property of Gmsh points guiding the size of meshed elements.

Figure 4.4: 2D example of unified width interconnect.

of geometry shown in figure 4.6, and code list 4.2 demonstrates how the template is instan-

tiated.

Listing 4.1: Gmsh template of an interconnect with 3 vias.

```
1  // DERIVED INTERMEDIATE GEOMETRY VARIABLES
2  x_via1_off = x_padl + th_lin + x_res;
3  x_via2_off = x_via1_off + x_via + th_lin + x_arc1;
4  x_via3_off = x_via2_off + x_via + x_arc2;
5  x_max = x_via3_off + x_via + x_padr;
6
7  // GEOMETRY
8
9  // Bottom of dielectric
10 Y = 0;
11 Point(1) = {0, Y, 0, lc_b};
12 Point(2) = {x_max, Y, 0, lc_b};
13
14 // Bottom of capping I
15 Y += th_di1;
16 Point(3) = {0, Y, 0, lc_a};
```

Figure 4.5: 2D example of customized width interconnect.

```
17  Point(4) = {x_via1_off - th_lin, Y, 0, lc_a};
18  Point(5) = {x_via1_off + x_via + th_lin, Y, 0, lc_a};
19  Point(6) = {x_via1_off + x_via + th_lin + x_aoi, Y, 0, lc_a};
20
21  // Bottom of vias
22  Y += th_lin;
23  Point(7) = {x_via1_off, Y, 0, lc_a};
24  Point(8) = {x_via1_off + x_via, Y, 0, lc_a};
25  Point(9) = {x_via2_off, Y, 0, lc_b};
26  Point(10) = {x_via2_off + x_via, Y, 0, lc_b};
27  Point(11) = {x_via3_off, Y, 0, lc_b};
28  Point(12) = {x_via3_off + x_via, Y, 0, lc_b};
29
30  // Top of capping I
31  Y += th_cap - th_lin;
32  Point(13) = {0, Y, 0, lc_b};
33  Point(14) = {x_via1_off - th_lin, Y, 0, lc_b};
34  Point(15) = {x_via1_off + x_via + th_lin, Y, 0, lc_b};
35  Point(16) = {x_via1_off + x_via + th_lin + x_aoi, Y, 0, lc_b};
36
37  // Bottom of liner cliff
```

```
38  Y += th_di2;
39  Point(17) = {x_padl, Y, 0, lc_a};
40  Point(18) = {x_padl + x_res, Y, 0, lc_a};
41  Point(19) = {x_via1_off + x_via + th_lin, Y, 0, lc_a};
42  Point(20) = {x_via1_off + x_via + th_lin + x_aoi, Y, 0, lc_b};
43  Point(21) = {x_via2_off, Y, 0, lc_b};
44  Point(22) = {x_via2_off + x_via, Y, 0, lc_b};
45  Point(23) = {x_via3_off, Y, 0, lc_b};
46  Point(24) = {x_via3_off + x_via, Y, 0, lc_b};
47  Point(25) = {x_max, Y, 0, lc_b};
48
49  // Bottom of copper cliff
50  Y += th_lin;
51  Point(26) = {x_via1_off - x_res, Y, 0, lc_a};
52  Point(27) = {x_via1_off, Y, 0, lc_a};
53  Point(28) = {x_via1_off + x_via, Y, 0, lc_a};
54  Point(29) = {x_via1_off + x_via + th_lin + x_aoi, Y, 0, lc_a};
55
56  // Top of copper
57  Y += th_cu;
58  Point(30) = {0, Y, 0, lc_b};
59  Point(31) = {x_padl, Y, 0, lc_a};
60  Point(32) = {x_padl + th_lin, Y, 0, lc_a};
61  Point(33) = {x_via1_off + x_via + th_lin + x_aoi, Y, 0, lc_a};
62  Point(34) = {x_max, Y, 0, lc_b};
63
64  // Top of capping II
65  Y += th_cap;
66  Point(35) = {0, Y, 0, lc_b};
67  Point(36) = {x_via1_off + x_via + th_lin + x_aoi, Y, 0, lc_b};
68
69  // Top
70  Y += th_di3;
71  Point(37) = {0, Y, 0, lc_b};
72  Point(38) = {x_max, Y, 0, lc_b};
73
74  // END OF HAND WRITTEN GEOMETRY
```

```
Line(1) = {1, 2};        Line(2) = {2, 25};       Line(3) = {25, 34};
Line(4) = {34, 38};      Line(5) = {38, 37};      Line(6) = {37, 35};
Line(7) = {35, 30};      Line(8) = {30, 13};      Line(9) = {13, 3};
Line(10) = {3, 1};       Line(11) = {3, 4};       Line(12) = {4, 5};
Line(13) = {5, 6};       Line(14) = {6, 16};      Line(15) = {16, 15};
Line(16) = {15, 5};      Line(17) = {4, 14};      Line(18) = {14, 13};
Line(19) = {30, 31};     Line(20) = {31, 17};     Line(21) = {17, 18};
Line(22) = {18, 14};     Line(23) = {15, 19};     Line(24) = {19, 20};
Line(25) = {20, 29};     Line(26) = {29, 33};     Line(27) = {33, 36};
Line(28) = {36, 35};     Line(29) = {31, 32};     Line(30) = {32, 33};
Line(31) = {26, 32};     Line(32) = {26, 27};     Line(33) = {27, 7};
Line(34) = {7, 8};       Line(35) = {8, 28};      Line(36) = {28, 29};
Line(37) = {20, 21};     Line(38) = {21, 9};      Line(39) = {9, 10};
Line(40) = {10, 22};     Line(41) = {22, 23};     Line(42) = {23, 11};
Line(43) = {11, 12};     Line(44) = {12, 24};     Line(45) = {24, 25};
Line(46) = {33, 34};

Line Loop(1) = {10, 1, 2, -45, -44, -43, -42, -41, -40, -39, -38,
    -37, -24, -23, -15, -14, -13, -12, -11};
Plane Surface(1) = {1};

Line Loop(2) = {9, 11, 17, 18};
Plane Surface(2) = {2};

Line Loop(3) = {16, 13, 14, 15};
Plane Surface(3) = {3};

Line Loop(4) = {12, -16, 23, 24, 25, -36, -35, -34, -33, -32, 31,
    -29, 20, 21, 22, -17};
Plane Surface(4) = {4};

Line Loop(5) = {8, -18, -22, -21, -20, -19};
Plane Surface(5) = {5};

Line Loop(6) = {33, 34, 35, 36, 26, -30, -31, 32};
Plane Surface(6) = {6};
```

```
112
113  Line Loop(7) = {19, 29, 30, 27, 28, 7};
114  Plane Surface(7) = {7};
115
116  Line Loop(8) = {26, 46, -3, -45, -44, -43, -42, -41, -40, -39, -38, -37, 25};
117  Plane Surface(8) = {8};
118
119  Line Loop(9) = {4, 5, 6, -28, -27, 46};
120  Plane Surface(9) = {9};
121
122  Physical Surface("di") = {9, 5, 1};
123  Physical Surface("cap") = {7, 2, 3};
124  Physical Surface("lin") = {4};
125  Physical Surface("cu1") = {6};
126  Physical Surface("cu2") = {8};
127
128  Physical Line("heat_sink") = {5};
129  Physical Line("endpoint_1") = {20};
130  Physical Line("endpoint_2") = {3};
131  Physical Line("endpoint_3") = {12};
132  Physical Line("endpoint_4") = {39};
133  Physical Line("endpoint_5") = {43};
134  Physical Line("probe_1") = {26, 25};
```
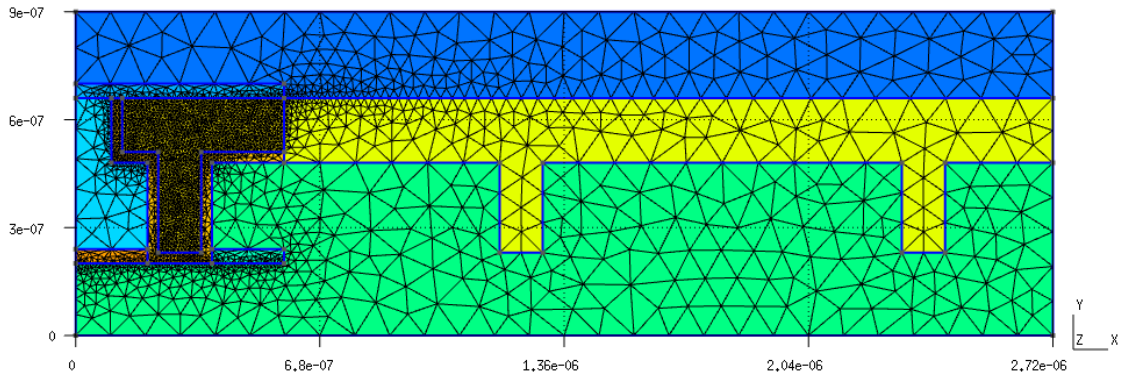

Figure 4.6: Example instantiation of a templated 2D interconnect with 3 vias.

Listing 4.2: Instantiation of a short 3D interconnect with 3 vias.

```
1  U = 1E-9; // Unit: ONE nanometer
2
3  // Cap layer thickness
4  th_cap = 40*U;
5
6  // Liner thickness
7  th_lin = 30*U;
8
9  // Copper thickness
10 th_cu = 150*U;
11
12 // Dielectric
13 th_di1 = 200*U; th_di2 = 240*U; th_di3 = 200*U;
14
15 // Paddings
16 x_padl = 100*U; x_padr = 300*U;
17
18 // Via size
19 x_via = 120*U; y_via = 180*U;
20
21 // Reservior
22 x_res = 100*U;
23
24 // Area of interest
25 x_aoi = 200*U;
26
27 // Arc lengths
28 x_arc1 = 800 * U; x_arc2 = 1000 * U;
29
30 lc_a = 10 * U;
31 lc_b = x_via;
32
33 Include "Templates/3via.geo";
```

4.2.2 Top Down Construction

In contract to the bottom up construction, user can also choose top down construction to build geometries. Instead of starting from defining vertices, user defines entities[4] first and perform operations on them. By using the OpenCASCADE [60] engine integrated in Gmsh, user can later do modifications on the previous defined simple entities by means including combining, translating, scaling, or boolean operations[5].

The advantage of this method is that user does not need to specify all compositing sub-entities to construct a entity[6]. This significantly simplifies the process of defining a 3D volume, especially in complex shape. While the compositing labels are not required, user can still acquire those labels at the end of defining the volume. This is also true for the operations mentioned above. The disadvantage is, it is usually difficult for a user to know which numbered label is corresponding to which sub-entity[7]without using the Gmsh graphical front-end.

For example, when constructing a two-segment, three-via interconnect surrounding by dielectrics (shown in figure 4.7 and 4.8), this method is preferred. User can define overlapped entities first and perform boolean operations on them to describe the buried copper interconnect, the liner, and the capping layers. Gmsh has the capability to optionally preserve the internal entities[8] after boolean operations as configured.

---

[4]Here, an entity refers a non-trivial shape. For instance a rectangle or or circle in 2D, or a cube or cylinder in 3D.

[5]Gmsh version 3.0 supports four boolean operations: union, or, difference, and fraction.

[6]For example, user does not need to specify all six plane surface labels to define a cube. Instead, the user just need the coordinates of diagonal vertices.

[7]For example, when user subtract a smaller cube from a bigger cube surrounding it, the generated inner surface label numbers are able to be acquired, but usually difficult to know which surface is corresponding to which label by only analysing the geometry code. This is true especially when the operands (the two cubes) are rotated before the operation.

[8]For example, the union of two overlapped cubes will leave keeping the buried surfaces an option.

Top down construction method is also friendly when a geometry is required to be parameterized and templated. Figure 4.9 is another instantiation using same template as figure 4.7 and 4.8, but with different dimensions (segment lengths).
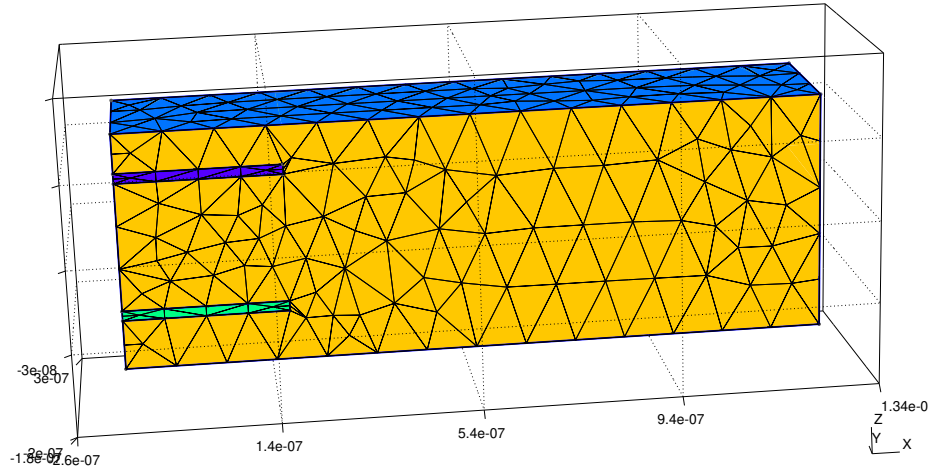
Figure 4.7: 3D example of a 3-via interconnect with all materials.

It should be remarked that when constructing geometries from top down, to avoid numeric accuracy issues[9], the geometries should be constructed and boolean-operated in normal scale first and then scaled down, if micro-meter or nano-meter scale geometries are intended.

As templating and instantiating example, code list 4.3 shows the Gmsh template of geometry shown in figure 4.10, and code list 4.4 demonstrates how the template is instantiated.

Listing 4.3: Gmsh template of an interconnect with 4 vias.

```
1  SetFactory("OpenCASCADE");
```

---

[9]As of version 3.0, Gmsh sometimes can not correctly perform boolean operations on nano-scale 3D entities.
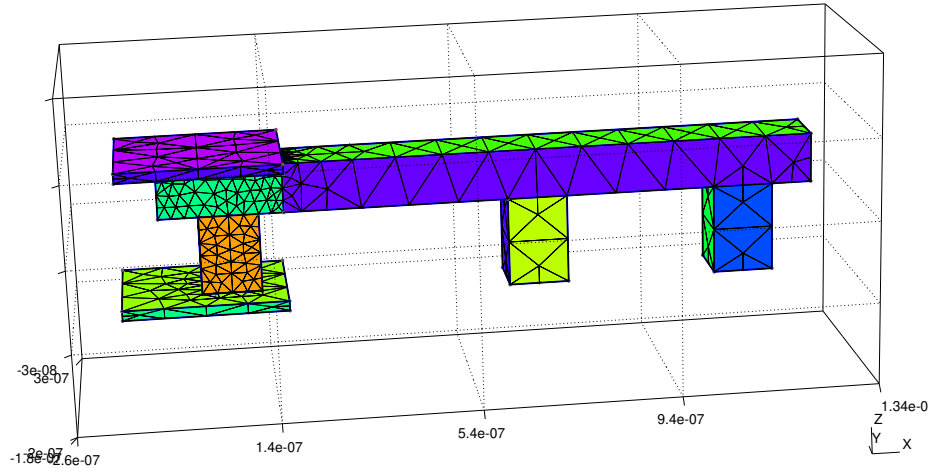
Figure 4.8: 3D example of a 3-via interconnect with dielectric removed.
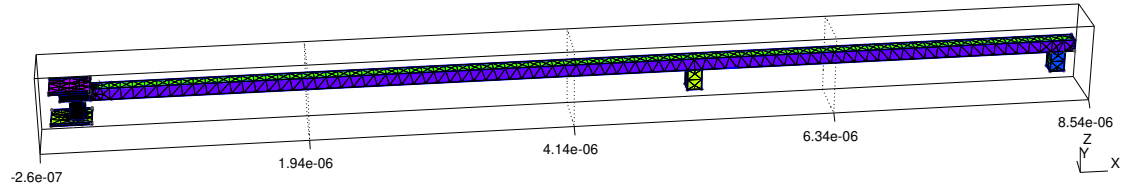
Figure 4.9: 3D example of longer version of 3-via interconnect.

```
2
3  Block(1) = {-X_RES, 0, Z_VIA,
4              X_RES + X_VIA + X_ARC1 + X_VIA + X_ARC2 + X_VIA + X_RES,
5              Y_VIA, Z_CU};
6  Block(2) = {0, 0, 0, X_VIA, Y_VIA, Z_VIA};
7  Block(3) = {X_VIA + X_ARC1, 0, 0, X_VIA, Y_VIA, Z_VIA};
8  Block(4) = {X_VIA + X_ARC1 + X_VIA + X_ARC2, 0, 0, X_VIA, Y_VIA, Z_VIA};
9
10 Block(5) = {X_VIA + X_ARC1, Y_VIA + Y_ARC3, 0, X_VIA, Y_VIA, Z_VIA};
11 Block(6) = {X_VIA + X_ARC1, Y_VIA, Z_VIA, X_CU, Y_ARC3 + Y_VIA + Y_RES, Z_CU};
12
13 Block(9) = {-X_RES, 0, Z_VIA, X_AOI, Y_VIA, Z_CU};
14
```

```
15  // Conductor outline
16  BooleanUnion(10) = {Volume {1}; Delete;}{Volume {2, 3, 4, 5, 6}; Delete;};
17
18  Block(20) = {TH_LIN, TH_LIN, TH_LIN,
19              X_VIA - TH_LIN * 2, Y_VIA - TH_LIN * 2, Z_VIA};
20  Block(21) = {-X_RES + TH_LIN, TH_LIN, Z_VIA + TH_LIN,
21              X_AOI - TH_LIN, Y_CU - TH_LIN * 2, Z_CU - TH_LIN};
22
23  // Area of interest
24  BooleanUnion(30) = {Volume {20}; Delete;} {Volume {21}; Delete;};
25
26  // Capping layers
27  Block(40) = {-X_RES - X_PAD, -Y_PAD, 0,
28              X_RES + X_AOI, Y_CU + Y_PAD * 2, TH_CAP};
29  Block(41) = {-X_RES - X_PAD, -Y_PAD, Z_VIA + Z_CU,
30              X_RES + X_AOI, Y_CU + Y_PAD * 2, TH_CAP};
31
32  // Dielectric
33  Block(50) = {-X_RES - X_PAD, -Y_PAD, -Z_PAD,
34              X_PAD + X_RES + X_VIA + X_ARC1 + X_VIA + X_ARC2 + X_VIA + X_RES,
35              Y_CU + Y_PAD * 2,
36              Z_VIA + Z_CU + Z_PAD * 2};
37  Block(51) = {X_VIA + X_ARC1 - X_PAD, Y_PAD + Y_VIA, -Z_PAD,
38              X_PAD * 2 + X_CU,
39              Y_ARC3 + Y_VIA + Y_RES,
40              Z_VIA + Z_CU + Z_PAD * 2};
41  BooleanUnion(60) = {Volume {50}; Delete;} {Volume{51}; Delete;};
42
43  finish() = BooleanFragments {Volume {60}; Delete;}
44                              {Volume {9, 10, 30, 40, 41}; Delete;};
45
46  // We scale the geo later because the boolean operations in the current version
47  // does not work well at very small scale
48  Dilate {{0, 0, 0}, {SCALE, SCALE, SCALE}} {
49      Volume{finish()};
50  }
51
```

```
52  Characteristic Length{PointsOf{Volume {46, 48, 44};}} = 30*SCALE;
53  Characteristic Length{PointsOf{Volume {47, 49, 50};}} = 10*SCALE;
54
55  Mesh.CharacteristicLengthMax = 100*SCALE;
56
57  Physical Volume("di") = {42};
58  Physical Volume("cap") = {41, 43};
59  Physical Volume("lin") = {46, 48, 44};
60  Physical Volume("cu") = {45};
61  Physical Volume("aio") = {47, 49, 50};
62
63  Physical Surface("heat_sink") = {120, 107};
64  Physical Surface("endpoint_1") = {122};
65  Physical Surface("endpoint_2") = {169};
66  Physical Surface("endpoint_3") = {142};
67  Physical Surface("endpoint_4") = {112};
68  Physical Surface("endpoint_5") = {144};
69  Physical Surface("endpoint_6") = {147};
70  Physical Surface("endpoint_7") = {154};
71  Physical Surface("probe_1") = {170, 171};
```

Listing 4.4: Instantiation of a short 3D interconnect with 4 vias.

```
1   SCALE = 1E-9;
2
3   X_VIA = 120; Y_VIA = 120; Z_VIA = 180;
4   TH_LIN = 10; TH_CAP = 20;
5   X_ARC1 = 500; X_ARC2 = 300; Y_ARC3 = 400;
6   X_RES = 80; Y_RES = 60;
7   X_CU = 120; Y_CU = 120; Z_CU = 100;
8   X_AOI = 250;
9   X_PAD = 80; Y_PAD = 80; Z_PAD = 100;
10
11  Include "4via_template.geo";
```
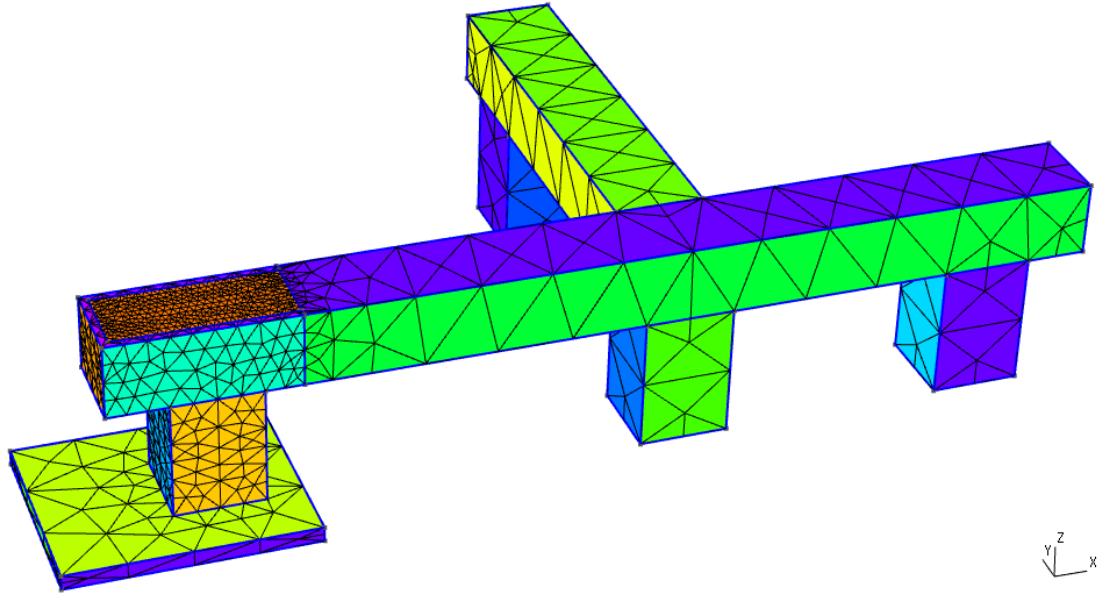
† Dielectric material and the upper capping layer are removed for the sake of structural demonstration.

Figure 4.10: Example instantiation of a templated 3D interconnect with 4 vias.

4.2.3 Comparison Between Two Geometry Description Styles

As discussed in sections 4.2.1 and 4.2.2, the advantages and disadvantages are listed in table 4.1.

4.3 Simplifications

In both the implementations using Deal.II or FEniCS, a number of simplifications are applied to the methods derived in chapter 3, for the sake of computational efficiency. These simplifications include but not limited to approximations on weight function and hydrostatic stress according to the characteristics of finite element method, and modifica-

	Bottom Up	Top Down
Boolean operations	Not required	Often required
Complex geometries preference	Low	High
Description flexibility	High	Not as high
Detailed mesh control	Easy	Hard
Gmsh frontend	Not required	Often required
OpenCASCADE engine	Not required	Required
Simple geometries preference	High	Low
Verbosity	High	Low

Table 4.1: Comparison between two Gmsh geometry composing styles.

tions on time-dependent solver in observance of coupling pattern of physics variables in electromigraion process.

Applied simplifications are described below in more details.

4.3.1 Weight Functions

Weight functions w_B, w_C, and w_V as described in (2.20) are critical in indicating the underlying material (boundary, copper, or void). For the sake of smoothness and continuity, the weight functions are defined by softmax of likelihood functions z_B, z_C, and z_V with a sharpness variable k_S controlling the steepness of the transition between 0 and 1.

In coupled analysis on physics systems, physics variables can have very different scales from $1 \times 10^{-9} \mathrm{m} \cdot \mathrm{s}^{-1}$ for boundary velocity, up to 1×10^9 Pa for critical hydrostatic stress level. As softmax function uses natural exponential function contrast to approximate 0 and 1, the magnitude of near zero values are indirectly controlled by the sharpness variable k_S. To achieve a good zero approximation that can be safely multiplied by a large scale value (such as hydrostatic stress or its gradient on copper-void boundary) to still acquire a good zero approximation comparing to small scales (such as boundary velocity), a large

k_S should be carefully selected. In practice, k_S should be at least larger than 50 to have a safely approximated zero (void indicator) to ensure correct void behavior.

It can be observed that, with such a high k_S, the continuous version of weight functions has even narrower transition bands than the mesh element sizes in a practical meshing. This implies that for an element, it is highly possible that it covers the transition area of the weight functions, with all its nodes valued at either 0 or 1. According to this observation, in software implementation, we can simplify the weight functions into the following:

$$w_B(\phi) = \begin{cases} 1 & \text{if } |\phi| < \phi_{\text{th}} \\ 0 & \text{otherwise} \end{cases}, \tag{4.1}$$

$$w_C(\phi) = \begin{cases} 1 & \text{if } \phi >= \phi_{\text{th}} \\ 0 & \text{if } \phi < \phi_{\text{th}} \end{cases}, \tag{4.2}$$

and

$$w_V(\phi) = \begin{cases} 1 & \text{if } \phi >= \phi_{\text{th}} \\ 0 & \text{if } \phi < -\phi_{\text{th}} \end{cases}, \tag{4.3}$$

where ϕ is the material phase field as described previously, and ϕ_{th} is the threshold of phase field ϕ level. This approach has two major benefits. First, it reduces the computational cost when calculating the weights by eliminating exponential function calculations. Furthermore, it is only nodal operation which is also an lighter operation than projecting the continuous functions. The second benefit is that it eliminates the process of tuning the sharpness factor k_S, which might cause inaccurate simulation if not done carefully.

4.3.2 Hydrostatic Stress on Void

As previously derived, hydrostatic stress is prescribed by equation (2.32), which is composited by different boundary-stress interactions on three regions indicated by material phase field. We will a simplification on the hydrostatic stress calculation on the void area, after which the simulated hydrostatic will be expressed in

$$\sigma_{\text{true}} = \frac{1}{2}(1 + \phi)\sigma. \tag{4.4}$$

This ensures that the simulated true stress σ_{true} will be forced to zero in the void, where $\phi = 0$, and equal to σ on copper. Indeed, this is another perfect match with realistic stress distribution cases, except stress is treated as zero in the void, where mentioning hydrostatic stress is not meaningful. As a result, σ can have another governing equation

$$\frac{\partial \sigma}{\partial t} = \nabla \cdot \left(\frac{D_a B \Omega}{k_B T} \nabla \sigma \right) - \frac{w_B \sigma v k_\tau}{\delta}. \tag{4.5}$$

Comparing to (2.32), equation (4.5) does not have the zero-locking term for the void area (the term with w_V), and extends the hydrostatic stress diffusion term

$$\nabla \cdot \left(\frac{D_a B \Omega}{k_B T} \nabla \sigma \right) \tag{4.6}$$

to the void as well[10]. The validity of this approximation is supported by the following reasons.

1. The void volume is small comparing to the interconnect in all practical cases[11]. Therefore the change of hydrostatic stress - buck volume integral 1.4 can be safely neglected,

---

[10]Note that the extended diffusion term should be explicitly multiplied by all three weight functions

$$(w_B + w_C + w_V)\nabla \cdot \left(\frac{D_a B \Omega}{k_B T} \nabla \sigma \right),$$

which can be simplified by plugging in $w_B + w_C + w_V = 1$.
[11]An interconnect will usually fail before a growing void becomes large enough.

which implies the atom conservation can still be satisfied.

2. Removal of the zero-locking term in the void also eliminates extra hydrostatic stress sources terms other than those introduced by electrical current flux. This ensures the hydrostatic stress conserve within separate copper interconnect when no current flux is applied.

3. Non-zero hydrostatic stress in the void will be masked to zero by equation (4.4) such that the simulated true stress σ_{true} still matches the realistic stress distribution.

4.3.3 Non-Linear Components in Phase Field

When solving for the phase field variable ϕ, it can be noticed that a non-linear term exists in the governing equation 2.13

$$\phi - \phi^3. \tag{4.7}$$

This non-linear term requires Newton-Raphson iterations to acquire precise solution on each time step. Since the phase field is expected to distribute between -1 and 1, and there is an diffusion term in equation 2.13 to ensure stability, for the sake of efficient computation, we treat the non-linear terms 4.7 as a constant (values acquired from last time step) when assembling the finite element system matrix. As shown in the weak form equation 3.32, the backward Euler method guarantees that the phase field will end up with stable solution. Therefore such relaxed linear approximation of phase field solving is effective and valid.

4.3.4 Updating Coupled Physics Systems

As mentioned before, the Deal.II based implementation takes care of physics variables σ (hydrostatic stress), ϕ (material phase field), and U (electrical potential for current density $\vec{j}$; and the FEniCS includes one more variable T (temperature). In both the implementations, let the relationship between time step n and $n-1$ respectively be

$$[\sigma^{(n)}, \phi^{(n)}, \vec{j}^{(n)}] = \boldsymbol{f}_{\text{Deal.II}}([\sigma^{(n-1)}, \phi^{(n-1)}, \vec{j}^{(n-1)}]) \tag{4.8}$$

and

$$[\sigma^{(n)}, \phi^{(n)}, \vec{j}^{(n)}, T^{(n)}] = \boldsymbol{f}_{\text{FEniCS}}([\sigma^{(n-1)}, \phi^{(n-1)}, \vec{j}^{(n-1)}, T^{(n-1)}]), \tag{4.9}$$

evaluating function $\boldsymbol{f}$ involves resolving the system matrix inverse where all physics variables coupled together. We relax the interactions between the variables at time step n and the simplified relationships becomes

$$\sigma^{(n)} = \boldsymbol{f}_{\text{Deal.II},\sigma}([\sigma^{(n-1)}, \phi^{(n-1)}, \vec{j}^{(n-1)}, T^{(n-1)}]),$$

$$\phi^{(n)} = \boldsymbol{f}_{\text{Deal.II},\phi}([\sigma^{(n-1)}, \phi^{(n-1)}, \vec{j}^{(n-1)}, T^{(n-1)}]), \tag{4.10}$$

$$\vec{j}^{(n)} = \boldsymbol{f}_{\text{Deal.II},j}([\sigma^{(n-1)}, \phi^{(n-1)}, \vec{j}^{(n-1)}, T^{(n-1)}]),$$

for the Deal.II implementation, and FEniCS definition is similarly simplified.

4.3.5 Static Mesh Size in FEniCS Implementation

Comparing to the local mesh refinement approach in Deal.II as described in section 3.5.3, FEniCS integrates the mesh refinement into its solver module. This implies less flexibility of user-defined local mesh refinement flexibility comparing to Deal.II. In the FEniCS implementation, preconfigured meshing strategy is applied and no automatic local

mesh refinement is conducted on the go. As shown in figure 4.11 and 4.12, a preconfigured refined mesh is assigned to the area where void is anticipated to generate or grow in. In the rest parts, mesh size is configured to be relatively larger to yield computational resources to the voiding part.
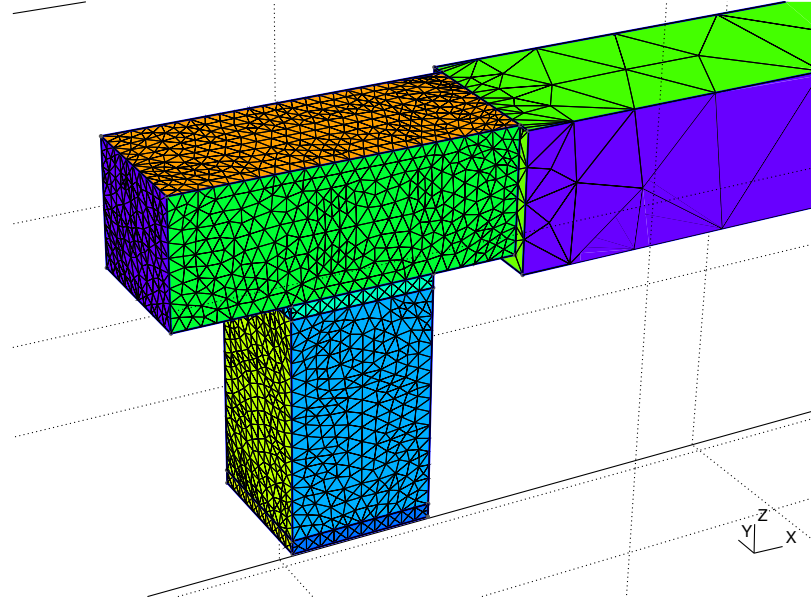
Figure 4.11: Simplified near-void copper.

With the preconfigured mesh size applied, the void generating location should be well-anticipated. Given this true, as shown inf figure 4.11, liner-constrained copper interconnect geometry can also be further simplified. This is divided into the following aspects:

1. While area of high void activity is anticipated, liner material can only be preserved at such area. The reason is in two folds. First, in the rest area, major electrical current is conducted by the copper and the resistivity change caused by adding surrounding liner material is negligible. Second, Joule heating effect can still be analyzed and since
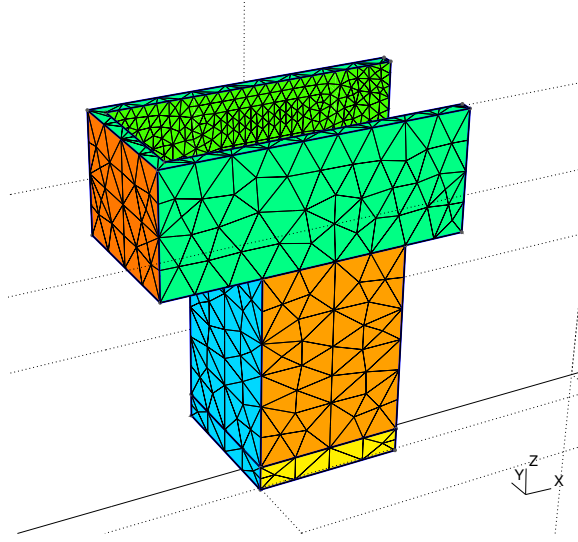
Figure 4.12: Simplified near-void liner.

the near-void liner is preserved and the current-shunting can still take effect when the conducting copper is depleted in the voiding process.

2. The remaining liner material can be meshed into elements with relatively larger size, as shown in figure 4.12. The reason is that we only include the material phase field in the copper domain such that the liner domain does not need to be well meshed to conform the sharp copper-void boundary.

3. Capping layer is shrunk to the near voiding area only, as another simplification. As the capping layer is very thin, it requires a large number of well-conditioned elements to mesh the capping layer. Smaller capping layer area will effectively reduce the number of meshed elements. Figure 4.9 is an example of longer segments where liner and capping layer simplification is performed. As it demonstrates, elements that are away from the anticipated voiding position have relatively larger sizes, effectively reducing

90

the finite element system matrix size.

4.4 Input Configuration

4.4.1 Input Configuration in Deal.II Based Simulator

Google protobuf [11] is integrated to describe the parameters for the simulator. It is divided down to three sections: FEM config, PDE config, and technology specification, as shown in code piece 4.5.

Finite element method configuration (`fem_config`) section contains mesh refinement control, solver specifications, and time step control details.

dimension specifies in which dimension the simulator will be run in. This should meet the dimension of the input geometry file.

refinement-ratio and coarsening-ratio specify the ratio (0 to 1) of the total elements to be refined/coarsened at each adaptive local refinement steps.

n-initial-refinements configures the number of refinements on the initial solutions of phase field and hydrostatic stress.

max-refinement-level and min-refinement-level specify the maximum/minimum refinement depth counted start from the initially meshed elements.

n-global-refinements configures the fixed number of initial global refinements which will apply to all the mesh elements unconditionally.

solver-configs is a Protobuf map configuring solvers for each physics system, specified by

key. *absolute-diff* and *relative-diff* respectively specifies the time discretization parameters described in section 3.3. *solver-type*, *solver-tolerance*, and *solver-iterations* correspondingly configures the solver as their names implies. *refinement-weight* configures the refinement weight coefficients ϵ as introduced in 3.5.3.

renumber-dofs controls whether to renumber finite element node indices in order to improve the sparse system matrix pattern in terms of lower solving complexity.

time-stepping-config configures time discretization parameters as described in section 3.3.

max-simulated-time sets the limit of total simulated time (not simulator run time) in seconds.

steps-per-refinement sets the period of performing adaptive local refinement, in the number of time advancing steps.

max-cells controls the limit total number of finite element cells, which is another control knob on the adaptive mesh refinement.

Partial differential equation configuration (`pde_config`) section sets the voltage and current boundary conditions on boundary IDs as appointed in the input Gmsh file. Under subsection *initial-condition*, field *init-void* sets the initial void[12] placement by prescribing a math function as the initial material phase field. Under *miscellaneous* subsection, *regulation-resolution* corresponds to variable c_R as described in section 2.5.2, controlling the boundary velocity regulation sensitivity. *boundary-thickness*, *phase-threshold*, *voiding-time-*

---

[12]A preexisted void is usually nucleated in the cooling down phase after a chip is manufactured.

const, and *phase-convergence-tau* respectively correspond to variables δ, ϕ_{th}, τ_{void}, and τ_ϕ, as described in chapter 2. Under subsection `abstracted_dim`, *line-width* is the cross-sectional area on the co-dimensions for those simulations that are not 3D.

Technology section specifies all the technology-dependent parameters. For the simulator implemented using deal.II, only copper material specifications are supported.

Listing 4.5: An EM simulator configuration file in Protobuf text format.

```
1  fem_config: {
2      dimension: 2
3      refinement_ratio: 0.60
4      coarsening_ratio: 0.03
5      n_initial_refinements: 6
6      max_refinement_level: 5
7      min_refinement_level: 2
8      n_global_refinements: 2
9      solver_configs: {
10         key: "electrical_potential"
11         value: {
12             absolute_diff: 0
13             relative_diff: 0
14             solver_type: CONJUGATE_GRADIENT
15             solver_tolerance: 1e-6
16             solver_iterations: 1000
17             refinement_weight: 1
18         }
19     }
20     solver_configs: {
21         key: "electrical"
22         value: {
23             absolute_diff: 0
24             relative_diff: 0
25             solver_type: CONJUGATE_GRADIENT
26             solver_tolerance: 1e-3
27             solver_iterations: 1000
```

```
28          refinement_weight: 0
29        }
30      }
31    solver_configs: {
32        key: "korhonen_stress"
33        value: {
34            absolute_diff: 2e6
35            relative_diff: 0.05
36            solver_type: CONJUGATE_GRADIENT
37            solver_tolerance: 1e-65
38            solver_iterations: 1000
39            refinement_weight: 1
40        }
41    }
42    solver_configs: {
43        key: "copper_void_field"
44        value: {
45            absolute_diff: 0.1
46            relative_diff: 0.1
47            solver_type: CONJUGATE_GRADIENT
48            solver_tolerance: 1e-35
49            solver_iterations: 1000
50            refinement_weight: 1
51        }
52    }
53    renumber_dofs: false
54    time_stepping_config: {
55        min_time_step: 1e-6
56        max_time_step: 1000
57        init_time_step: 5e-3
58        growth_factor: 1.5
59        shrinking_factor: 0.5
60    }
61    max_simulated_time: 8640000
62    steps_per_refinement: 10
63    max_cells: 10000
64  }
```

```
65
66  pde_config: {
67      boundary_condition: {
68          voltage_bc: {
69              id: 1
70              value: 0
71          }
72          voltage_bc: {
73              id: 2
74              value: 3e-4
75          }
76      }
77
78      initial_condition: {
79          init_void: {
80              id: 0
81              function: "erf((dist(p, {-4e-7, 8e-7}) - 4.5e-7) / 2e-9)";
82          }
83      }
84
85      miscellaneous: {
86          regulation_resolution: 1e-21        # m^3
87          boundary_thickness: 5.0e-9          # m
88          phase_threshold: 0.9                # (1)
89          voiding_time_const: 1               # s
90          phase_convergence_tau: 1e3          # s
91      }
92
93      abstracted_dim: {
94          line_width: 200E-9                  # m
95      }
96  }
97
98  technology: {
99      material_spec: {
100         key: "main"
101         value: {
```

```
102            type: "copper"
103            electrical_conductivity: 59.6e6    # S/m
104            thermal_conductivity: 401          # W/(m*K)
105            effective_charge_n: -5             # (1)
106            diffusion_coefficient: 5.2e-6      # m^2/s
107            em_activation_energy: 1.4e-19      # J
108            atomic_volume: 1.18e-29            # m^3
109            korhonen_modulus: 110e9            # Pa
110            initial_stress: 0                  # Pa
111            critical_stress: 800e6             # Pa
112        }
113    }
114
115    environment: {
116        temperature: 350    # K
117    }
118 }
119
120 probe_config: {
121 }
```

4.4.2   Input Configuration in FEniCS Based Simulator

JSON [20] file format is used in the FEniCS based simulator. Cascaded configurations are allowed, which requires user to provide the configuration files in command line arguments in order. Firstly provided configuration file will be override by later configurations. This enables different simulation configuration files sharing same technology and finite element configurations. Code 4.6, 4.7, and 4.8 demonstrates such ability. Configuration 4.6 specifies the verbose configurations regarding technology, finite element parameters that are similar to section fem_config and pdf_config in the deal.II configuration files. These options have relatively lower possibility to change among different simulations. The second

configuration, as shown in code 4.7[13]assigns different materials and boundary conditions to specific domains as described by the geometry files, which is marked as `TBD` on purpose for the later overriding. This facilitates the simulation on similar geometries instantiated from a same template using same electrical configurations. As shown in configuration code 4.8, it merely provides paths the input geometry files, which serves as a minimal changing set when user needs to do a simulation scan on a group of similar geometries, such as a set of interconnects with same structure but different segment widths, thicknesses, or lengths.

Listing 4.6: The default configuration file for FEniCS EM simulator.

```
1  {
2      "material_config":
3      {
4          "main": "copper",
5          "liner": "tantalum_nitride",
6          "capping": "silicon_carbonitride",
7          "dielectric": "silicon_dioxide"
8      },
9      "material_repo":
10     {
11         "copper":
12         {
13             "electrical_conductivity": 59.6E+6,
14             "thermal_conductivity": 385.0,
15             "specific_heat_capacity": 385,
16             "mass_density": 8.96E+3,
17             "diffusion_coefficient": 5.2E-6,
18             "charge_number": -5,
```

---

[13]The over wide line specifying *init-shape* under section *phase-field-config* prescribes an pre-existing void with function

$$\phi(x) = \mathrm{erf}\left(\frac{\sqrt{(x_1 - 160 \times 10^{-9})^2 + (x_2 - 80 \times 10^{-9})^2 + (x_3 - 380 \times 10^{-9})^2} - 240 \times 10^{-9}}{20 \times 10^{-9}}\right)$$

which sets the center at $(160, 80, 380)$nm with radius of 240nm and boundary thickness of roughly 20nm.

```json
19          "activation_energy": 1.44E-19,
20          "bulk_modulus": 110E+9,
21          "atomic_volume": 11.82E-30
22      },
23      "tantalum_nitride":
24      {
25          "electrical_conductivity": 1.25E+6,
26          "thermal_conductivity": 4.0,
27          "specific_heat_capacity": 420,
28          "mass_density": 13.7E+3
29      },
30      "silicon_carbonitride":
31      {
32          "thermal_conductivity": 3.0,
33          "specific_heat_capacity": 660,
34          "mass_density": 3.75E+3
35      },
36      "silicon_dioxide":
37      {
38          "thermal_conductivity": 1.4,
39          "specific_heat_capacity": 680,
40          "mass_density": 2.65E+3
41      }
42  },
43  "constants":
44  {
45      "boltzmann": 1.38E-23,
46      "electron_charge": -1.602E-19
47  },
48  "fem_config":
49  {
50      "init_timestep": 0.1,
51      "backward_euler_factor": 0.5,
52      "boundary_thickness": 20E-9,
53      "steps_per_outout": 10,
54      "steps_per_snapshot": 10,
55      "boundary_speedup": 100
```

```
56      },
57      "phase_field_config":
58      {
59          "stabilizing_time": 600,
60          "absolute_growth_limit": 0.2,
61          "relative_growth_limit": 0.2,
62          "boundary_extent": {
63              "2d": 20E-9,
64              "3d": 400E-18
65          }
66      },
67      "kstress_config":
68      {
69          "init_stress": 0,
70          "vanishing_factor": 1,
71          "absolute_growth_limit": 1E+6,
72          "relative_growth_limit": 0.05
73      },
74      "thermal_config":
75      {
76          "heat_transfer_coefficient": 1E+6,
77          "external_temperature": 350,
78          "absolute_growth_limit": 1,
79          "relative_growth_limit": 0.1
80      },
81      "electrical_config":
82      {
83          "absolute_growth_limit": null,
84          "relative_growth_limit": null
85      }
86 }
```

Listing 4.7: A cascaded overidden configuration file for FEniCS EM simulator.

```
1 {
2      "dimension": 3,
3      "geometry":
```

```
 4      {
 5          "mesh": "TBD",
 6          "facet": "TBD",
 7          "domain": "TBD"
 8      },
 9      "electrical_config":
10      {
11          "domain": { "main": [4, 5], "liner": [3] },
12          "voltage_bc": {
13              "gnd": { "value": 0.0, "ids": [10] }
14          },
15          "current_bc":
16          {
17              "isource_1": { "value": 2E+10, "ids": [11] },
18              "isource_2": { "value": 1E+10, "ids": [12] },
19              "isource_3": { "value": 1E+10, "ids": [13] }
20          },
21          "resistance_probe":
22          [
23              {
24                  "name": "hotspot",
25                  "domain": [3, 5],
26                  "anode_ids": [14],
27                  "cathode_ids": [10]
28              }
29          ]
30      },
31      "thermal_config":
32      {
33          "domain": {
34              "main": [4, 5],
35              "liner": [3],
36              "capping": [2],
37              "dielectric": [1]
38          },
39          "heat_flux_ids": [6],
40          "heat_flux_density": 100e+3,
```

```
41        "init_temperature": 350
42      },
43      "phase_field_config":
44      {
45          "domain": [4, 5],
46          "init_shape": "erf((sqrt((x[0] + 160E-9)*(x[0] + 160E-9) + (x[1] + 80E-9)*(x[1]
47      },
48      "kstress_config":
49      {
50          "domain": [4, 5]
51      },
52      "fem_config":
53      {
54          "steps_per_outout": 1
55      }
56  }
```

Listing 4.8: A cascaded overriding configuration file for FEniCS EM simulator.

```
1  {
2      "geometry":
3      {
4          "mesh": "Geometries/3d/4via_short.xml",
5          "facet": "Geometries/3d/4via_short_facet_region.xml",
6          "domain": "Geometries/3d/4via_short_physical_region.xml"
7      }
8  }
```

Configuration files for steady state simulation has different format since many parameters in transient modeling are not required here. Code 4.9 is an example of steady state analysis configuration. This configuration file mainly contains the electrical current prescription on each terminal, along with some necessary parameters required in Korhonen's equation, as specified in subsection *parameters*.

Listing 4.9: A 13-segment steady state analysis configuration file for FEniCS EM simulator.

```
1  {
2      "geometry": "Geometries/2d/13seg_vlen",
3      "ground": {
4          "ids": [2],
5          "phys_name": "ep_1"
6      },
7
8      "current_densities": [
9          { "ids": [3], "phys_name": "ep_2", "value": 5E+9 },
10         { "ids": [4], "phys_name": "ep_3", "value": -5E+9 },
11         { "ids": [5], "phys_name": "ep_3_4", "value": -5E+9 },
12         { "ids": [6], "phys_name": "ep_5_4", "value": 5E+9 },
13         { "ids": [7], "phys_name": "ep_5_6", "value": 5E+9 },
14         { "ids": [8], "phys_name": "ep_10_11", "value": 0 },
15         { "ids": [9], "phys_name": "ep_13", "value": -5E+9 },
16         { "ids": [10], "phys_name": "ep_7", "value": 5E+9 },
17         { "ids": [11], "phys_name": "ep_7_8", "value": 10E+9 },
18         { "ids": [12], "phys_name": "ep_9", "value": -5E+9 },
19         { "ids": [13], "phys_name": "ep_8", "value": -15E+9 },
20         { "ids": [14], "phys_name": "ep_12_11", "value": 0 },
21         { "ids": [15], "phys_name": "ep_12", "value": -5E9 }
22     ],
23
24     "parameters": {
25         "eZ": 1.609E-18,
26         "rho": 3E-8,
27         "omega": 1.18E-29,
28         "B": 1E+11,
29         "thickness": 100E-9
30     }
31 }
```

4.5   Summary

Software architecture is described, showing a decoupled modular design with all data interfaces conforming open standards. Implementations based on Deal.II and FEniCS finite element libraries are separately introduced. Various simplifications are described to reduce the computational complexity and improve numerical stability. The simplifications include nodal discretization of weight functions, allowing hydrostatic stress diffusion in voided area, solver relaxation of non-linear components in phase field equation, relaxation of coupled physics systems solving between time steps, and static mesh size in FEniCS implementation. For aspects of data interface, scripting and automation support, geometry construction and input configuration formats are demonstrated by providing practical examples.

Chapter 5

Experiment Result

In cases that void shapes are allowed to be abstracted into lower dimensional representations, 1D and 2D simulations have advantages of their relatively lighter computational demand. Therefore it is important to verify the sanity of void size, stress evolution, as well as the reasonableness of adaptive local mesh refinement. In the simulated copper domain, since the void size is regulated by the integral of the hydrostatic stress, it allows us to simulate the interconnects with different numbers of dimensions abstracted out without losing consistency between the void size and the distribution of hydrostatic stress. On the other hand, inward and outward electrical current on the interconnect boundary respectively provides hydrostatic stress sink and source, which do not lose their generality because of dimensional abstraction (from 3D towards 2D or 1D).

5.1 Time Dependent Post-Voiding Analysis on Copper Only Domains
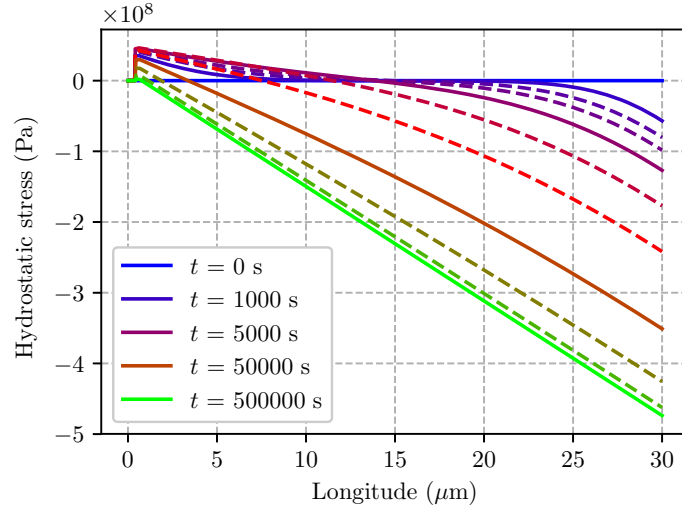
5.1.1 1D Hydrostatic Stress Analysis

Figure 5.1: Hydrostatic stress evolution on a 30 μm copper interconnect.

One-dimensional simulations are conducted on a 30 μm copper interconnect. As shown in figure 5.1 (zoomed details in figure 5.2), it is easier to visualize the stress evolution over the simulated interconnect over time. Under a steady current applied toward the $x-$ direction, the stress converges at a steady state where the linear transition starts from zero near the cathode/void boundary to its minimum value (maximum compressive stress). The increase-then-decrease hydrostatic stress near the cathode meets the 2D simulation as well. In the zoomed view in figure 5.2, the left-most section with zero stress indicates the void. It is clear to see that the copper-void boundary, which is indicated by the following increasing edges, advances towards the $x+$ direction. It is important to mention that the void size does
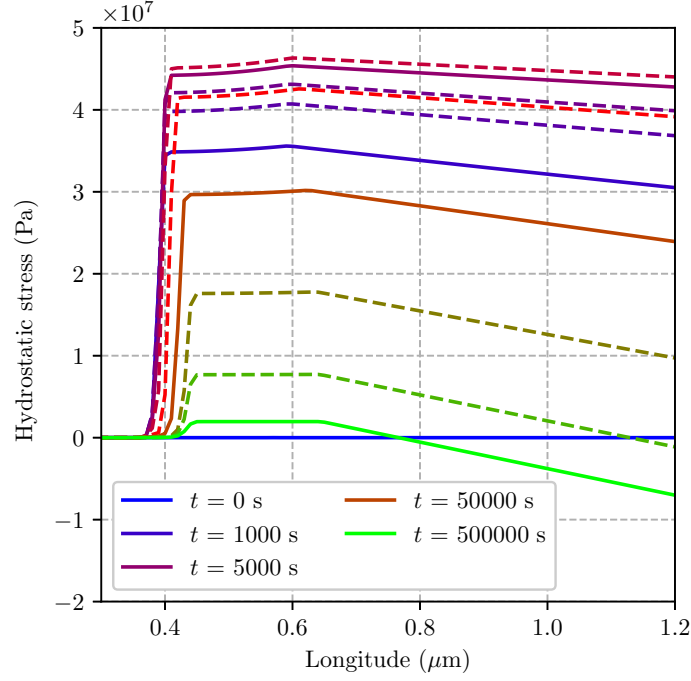
Figure 5.2: Hydrostatic stress evolution on a 30 μm copper interconnect zoomed in at near cathode/void position.

not increase until the cathode hydrostatic stress builds to a critical level at the cathode, which is a strong evidence that the void nucleation phase and growth phase are integrated as a whole in the proposed FEM simulation method.

5.1.2 2D Hydrostatic Stress and Current Density Analysis

Figure 5.3 and 5.4 illustrate the snapshots taken on void shape, hydrostatic stress, and current density of two 2D simulations on a 20 μm upstream-configured copper interconnect. Because of the nature of the phase field representation of void-copper material, as the void boundary moves, it can move across vertices in the simulated geometry without altering the finite element system matrix structure. As shown in the figure, hydrostatic

106

stress grows at the cathode end until it reaches the critical stress and then the void starts to grow. After the void started to grow, the moving copper-void boundary introduces a hydrostatic stress sink due to the material migration, as mentioned in equation (2.30). This is the reason that the cathode hydrostatic stress is decreasing later. Therefore the cathode hydrostatic stress is observed to be first increasing then decreasing, which agrees with the behavior described in nucleation phase and growth phase.

5.1.3 3D Hydrostatic Stress and Current Density Analysis

Figure 5.5 shows a 3D growing void simulation using the proposed method. It is shown that once a void has nucleated, which is typically at or near a terminal node, the stress around the void will immediately become zero (where element entities are removed to better demonstrate the void shape). However, the stress around the void will be close to the same stress level as immediately prior to the nucleation [45, 68]. A very large stress gradient will be formed around the void at nucleation time, which can be described by [45]. As a result, at steady state, hydrostatic stress around the cathode and the void will be at near-zero level because of the stress relaxation introduced by the advancing copper-void boundary. Mesh is locally refined near the void boundary to deliver better accuracy of void shape calculation.

As shown in figure 5.5, local mesh refinement is reasonably applied in different locations in the domain so that all three physical system can have smooth solutions. On copper-void boundary where phase field has large gradient the mesh gets refined to acquire better solution accuracy. On the corner where current density is high, mesh gets refined but still remains relatively coarse compared to the copper-void boundary, as configured
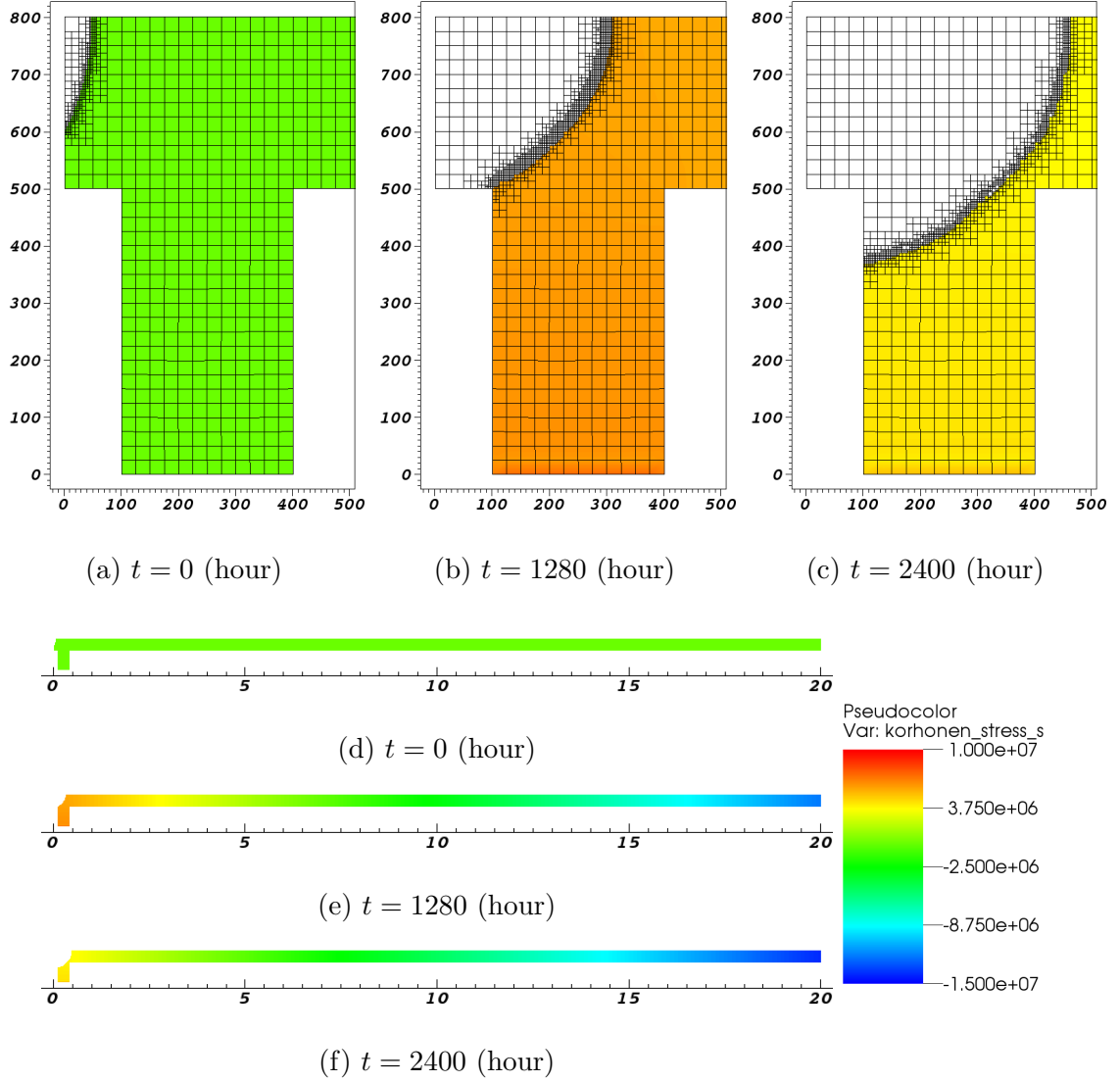
by parameter c_E. We use such a configuration since the current density accuracy is not as prioritized as the phase field and hydrostatic stress. At the cathode end, where the mesh was previously refined (as demonstrated in figure 3.6 at $t = 0$), is now dramatically coarsened since the stress has been evolved to be smooth enough at the point. On the rest of the area where solutions of all three systems are relatively smooth in space, the mesh remains coarse. This leads to smaller sparse system matrix requiring less computational resources.

(a) $t = 0$ (hour)   (b) $t = 1990$ (hour)   (c) $t = 3200$ (hour)

(d) $t = 4130$ (hour)   (e) $t = 4160$ (hour)
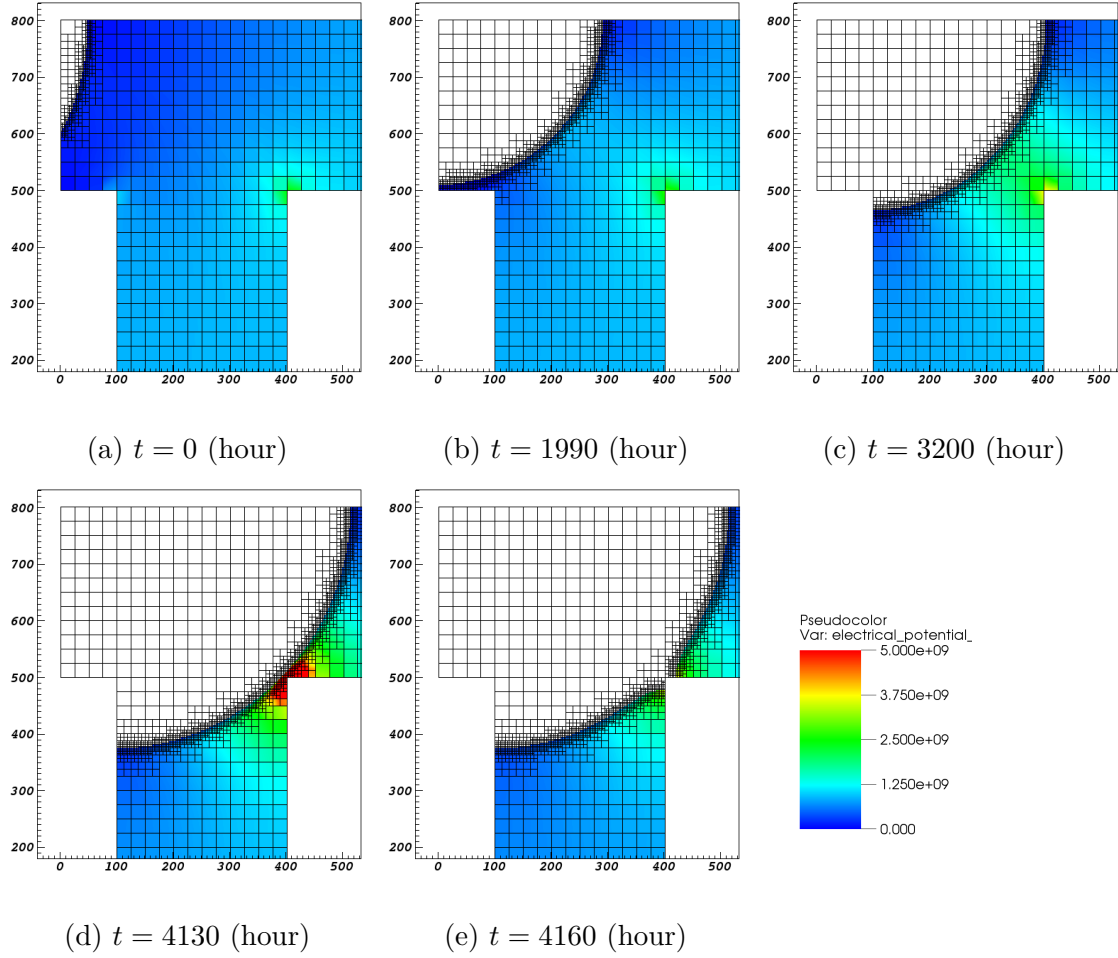
† Views are zoomed at the cathode end, axis unit: nm.

Figure 5.3: Current density and void growth simulation of a 20μm upstream-configured copper interconnect.

(a) $t = 0$ (hour)   (b) $t = 1280$ (hour)   (c) $t = 2400$ (hour)

(d) $t = 0$ (hour)

(e) $t = 1280$ (hour)

(f) $t = 2400$ (hour)

† Zoomed views (a – c) axis unit: nm), overviews (d – f) axis unit: μm.

Figure 5.4: Hydrostatic stress and void growth simulation of a 20μm upstream-configured copper interconnect.

110
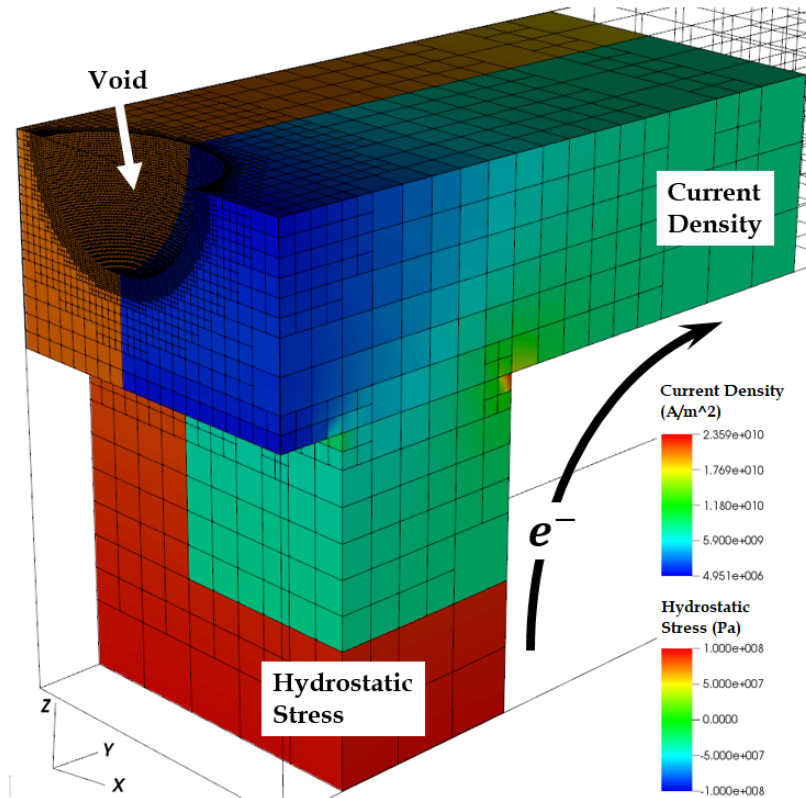
[†] The interconnect is cut into two portions for visualization: front showing current density, and back showing hydrostatic stress.

Figure 5.5: 3D void growth simulation of a 20μm upstream-configured copper interconnect.

5.2 3D Time Dependent Post-Voiding Analysis on All Domains

In this section, we use two examples to demonstrate the effectiveness of the proposed multi-physics electromigration post-voiding simulation method. Both the simulations are visualized in 3D overview of different physics systems combined together, and also in sliced 2D for the sake of details.

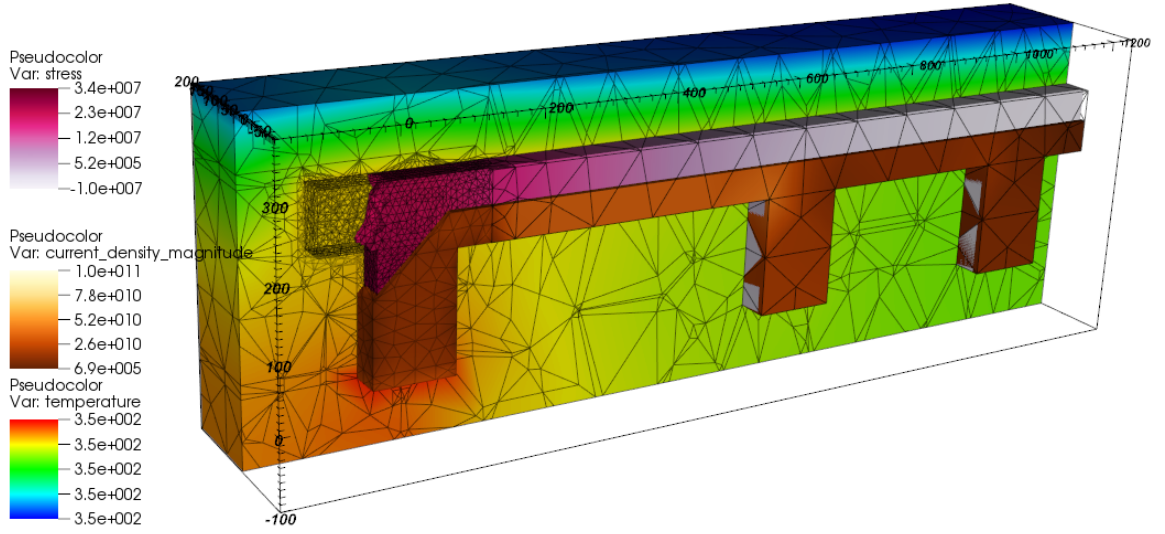### 5.2.1 Two-segment simulation with saturated void

Figure 5.6: Perspective view of the two-segment simulation result with saturated void.

The first example is a two-segment interconnect wire as shown in figure 5.6. The leftmost via is connected to ground and the rest of the two vias are respectively prescribed inward normal current density of $2 \times 10^{10} \mathrm{A \cdot m^{-2}}$ and $1 \times 10^{10} \mathrm{A \cdot m^{-2}}$. An initial small void is placed at the reservoir on the cathode (left) node. The wire lengths are respectively 500nm

and 300nm, in which case the void will saturate at the cathode end (near the via connected to ground) and not cause failure of resistance increase. Figure 5.6 shows the perspective view of all the four simulated physics variables (stress, current density and temperature). The phase field is illustrated by removing the elements where $\phi < 0$ on the stress (σ) plot.
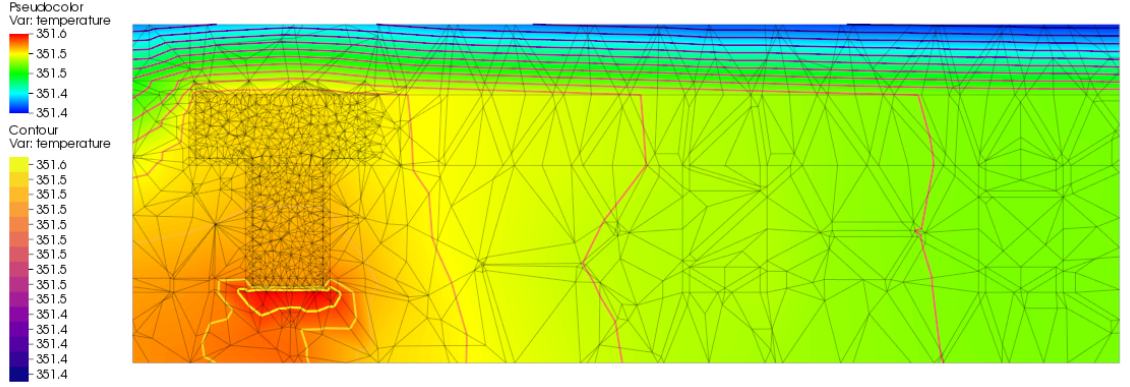

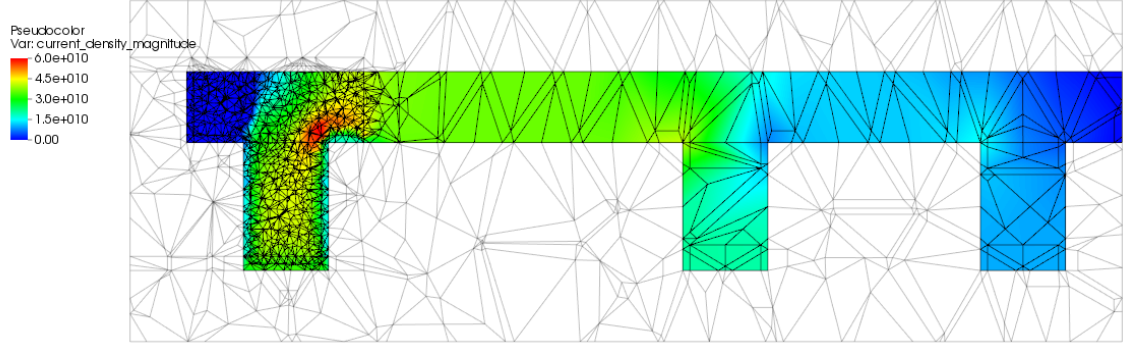Figure 5.7: Temperature vertical slice of the two-segment simulation.

Figure 5.8: Current density vertical slice of the two-segment simulation.

Figure 5.7 shows simulated temperature distribution for the two-segment. As we can see, the highest temperature is around bottom of the via as current flows through the barrier, which has higher resistance than the copper, into bottom wire.
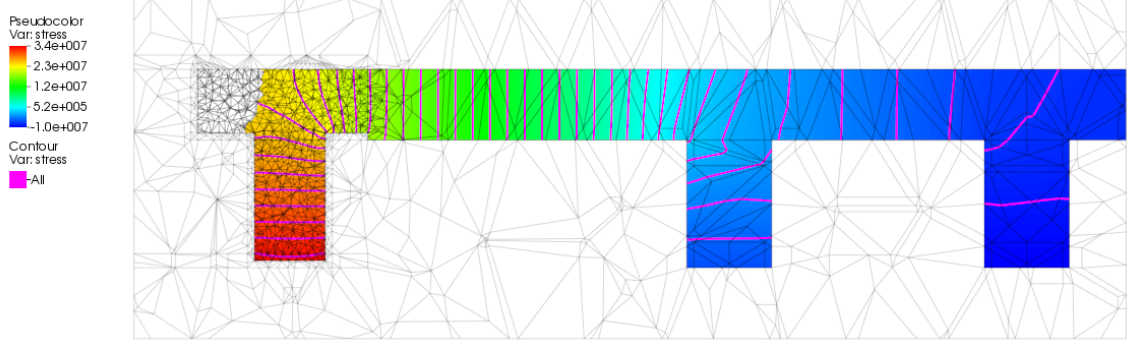
113

Figure 5.9: Hydrostatic vertical slice of the two-segment simulation.

Figure 5.8 shows the current density results. As we can see, the highest current density happens around the corner of via and main wire branch, which is typical local field enhancement field. Figure 5.9 shows the simulated stress distribution. As we can see, near-void area has the highest stress, and the stress around and inside void are zero. In this example, the void is saturated at a relatively small volume, which does not have a strong impact the current density distribution. Therefore the wire resistance remained roughly same as void-less situation.

5.2.2 Three-segment simulation with saturated void

The second example is a T-shape three-segment interconnect as shown in Figure 5.10. For the sake of the completeness of containing the plots of physics variables in their figures, the wire lengths are chosen to be relatively short as well. The three branches have lengths 500nm, 400nm, and 300nm. Inward normal current densities at vias are prescribed as $2 \times 10^{10} \text{A} \cdot \text{m}^{-2}$, $1 \times 10^{10} \text{A} \cdot \text{m}^{-2}$, and $1 \times 10^{10} \text{A} \cdot \text{m}^{-2}$. Similar to the two-segment experiment, this configuration will also result an immortal wire. Figure 5.10 shows the
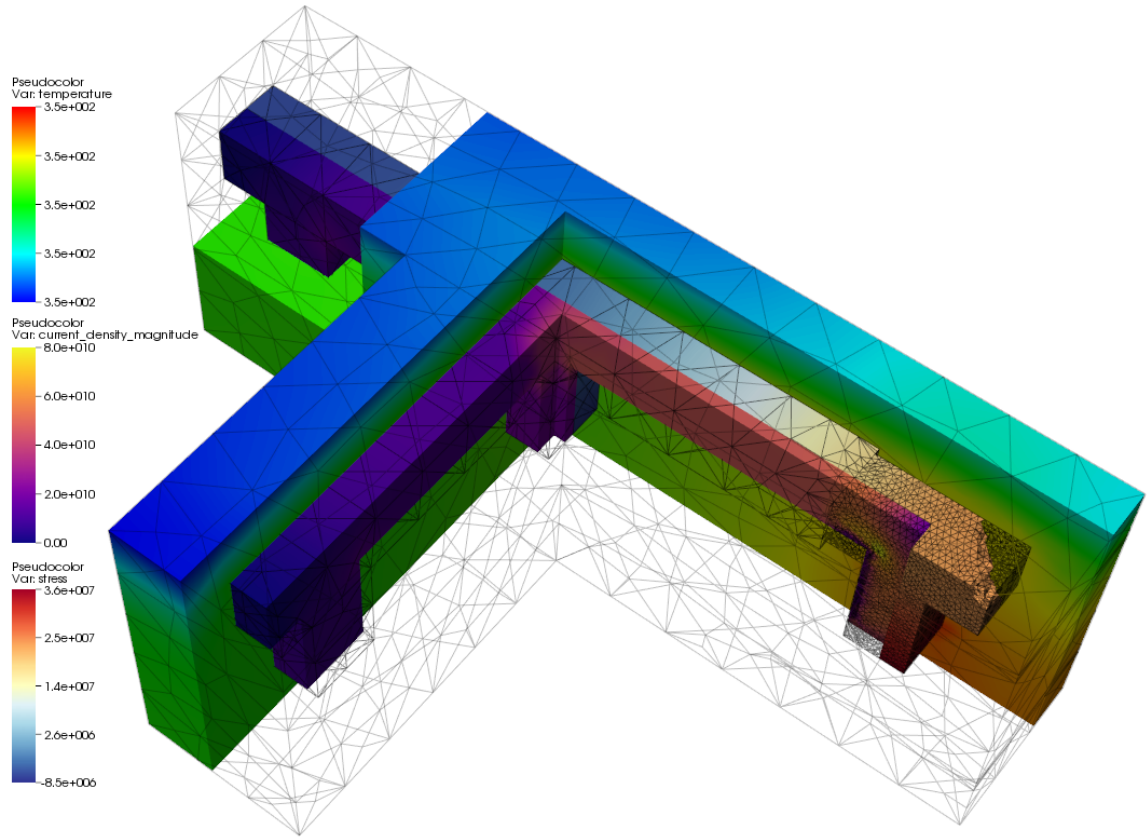
Figure 5.10: Perspective view of the three-segment simulation result with saturated void.

perspective view of the simulation results. Figures 5.11, 5.12, and 5.13 provide horizontal
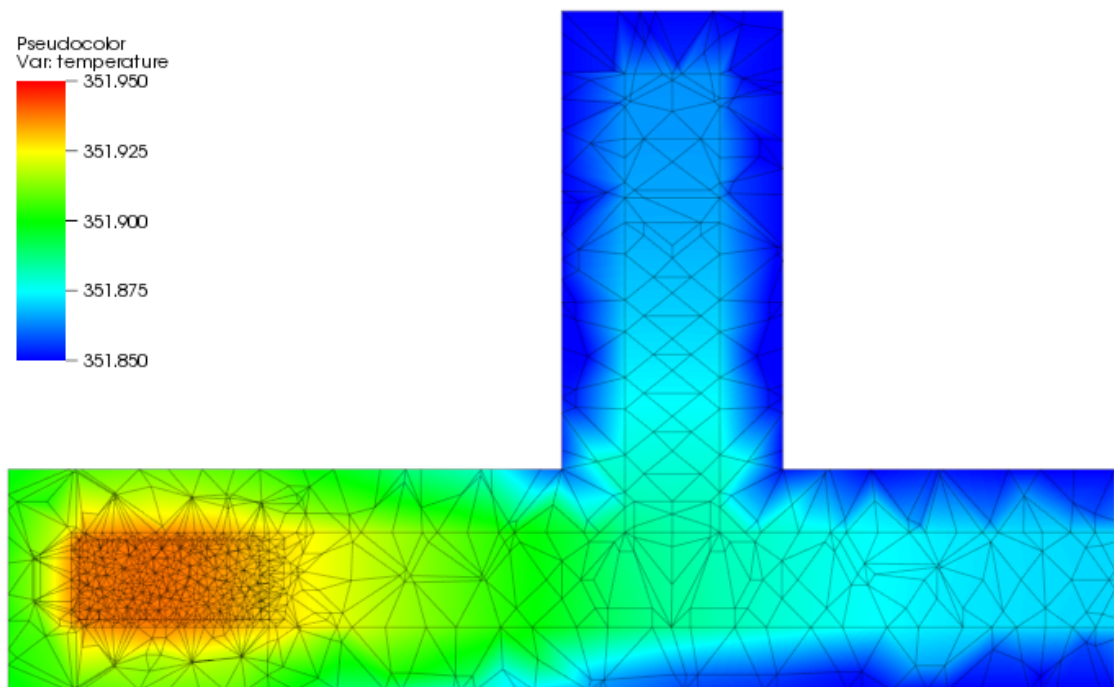
sliced results for more details.

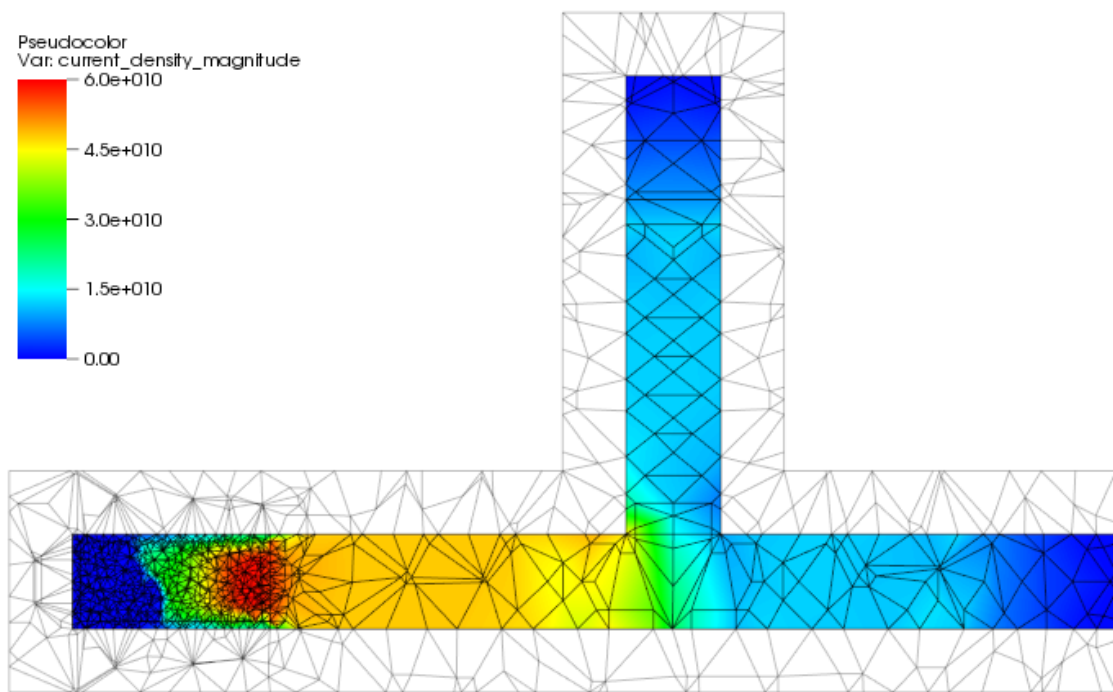Figure 5.11: Temperature vertical slice of the three-segment simulation.

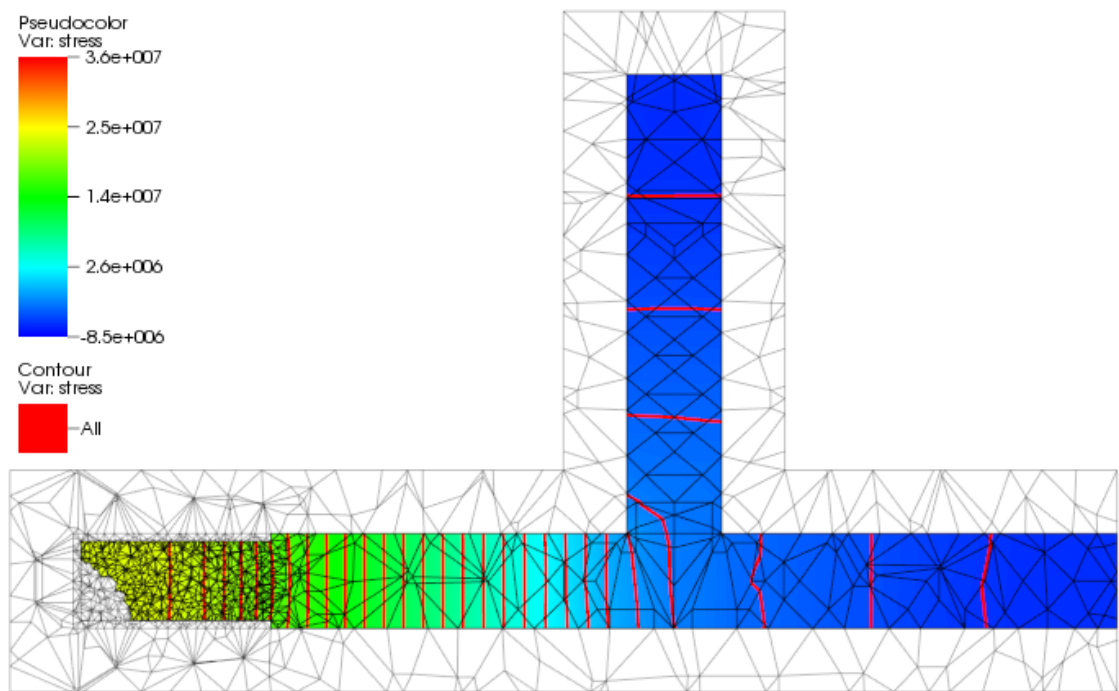Figure 5.12: Current density vertical slice of the three-segment simulation.

Figure 5.13: Hydrostatic vertical slice of the two-segment simulation.

5.3 Joule Heating Analysis

In this section, we investigate the wire resistance change pattern over time when the void is formed and the void grows over the critical volume. We will compare the resistance change against patterns observed in real silicon experiments.

It was observed that the in the failure process of a electromigration, vulnerable interconnect that is upstream-configured, its resistance growth follows a certain pattern [81]. Figure 5.14 shows such resistance change patterns, in which the resistance starts with a very slow increasing speed, followed by a sharp jump, and then increases approximately in a linear manner regarding time. The reason of the sharp resistance jump is due to the Joule heating induced resistance jump.
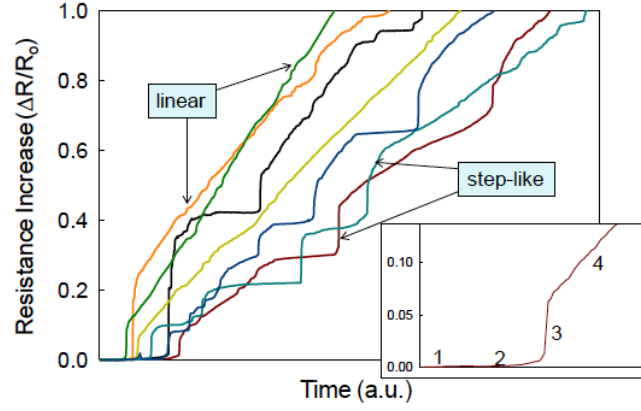
(a) Normalized resistance change over time due to EM failure process. Courtesy of [81].
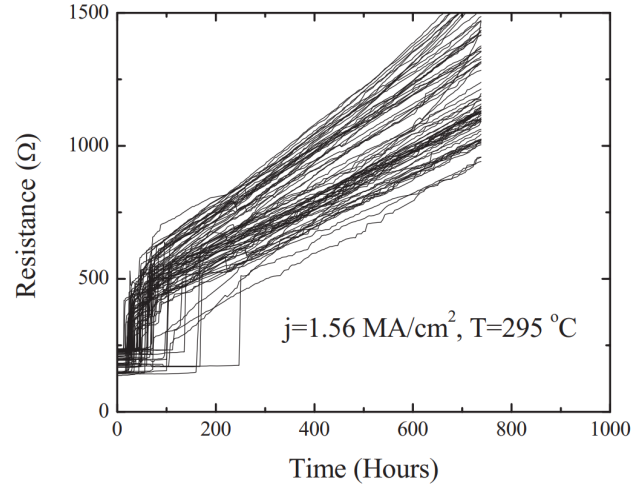
(b) Resistance change of copper interconnects over time measured in experiment. Courtesy of [24].
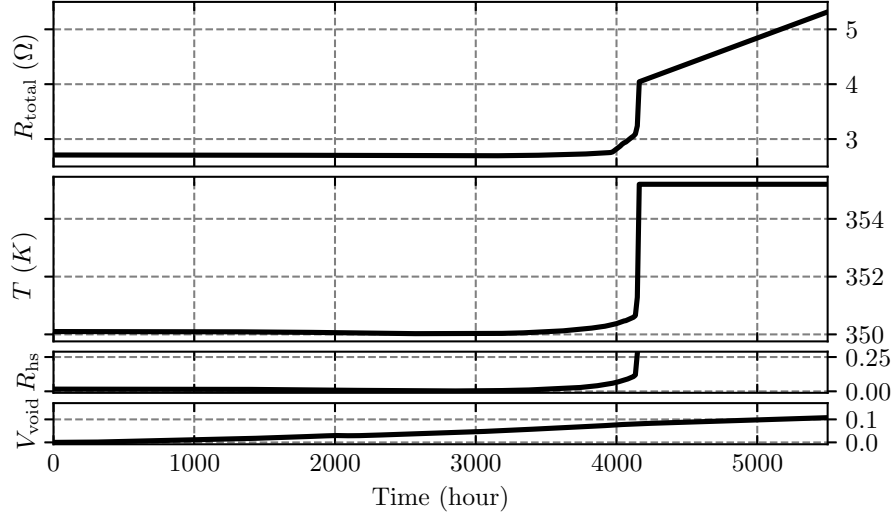
Figure 5.14: Resistance change patterns measured in real silicon experiments.

5.3.1 Joule Heating Analysis on Copper Only Domains

To validate the proposed FEM analysis method, in this subsection, we show that the proposed EM analysis can predict the wire resistance change patterns over time and Joule heating effects, which were observed experimentally[24, 81].

† R_{total} denotes the total wire resistance; T denotes the temperature at the hotspot; R_{hs}, in Ω, denotes copper resistance at the hotspot; V_{void}, in μm^3, denotes simulated void size.

Figure 5.15: Simulation of Joule heating effect.

Figure 5.14 shows that the measured time dependent resistance change of different interconnect wires during test stressing. As we can see, wire resistance initially does not change until an abrupt resistance jump. This happens when all the current starts to flow over the barrier layers as void reaches a critical size, by the time the growing void covers the via and separates the via copper from the main conducting part. Also, the abrupt resistance change is believed to be due to Joule heating in the barrier layers. Since barrier material

has higher resistance and very limited cross section area conducting current, an increment of total effective resistance and temperature should be observed. We collect simulation data from a upstream-configured 20 μm copper interconnect, stressed under a constant current density of 4×10^{10}A/m^2 to expect early failure. As shown in figure 5.15, temperature and resistance change can be divided into four stages.

In the first stage (t/hour < 3000), only void size increasing can be observed while total effective resistance change is negligible. This is because the growing void is still be majorly contained by the designated reservoir and does not cut into the main current-conducting channel. In this stage, non-trivial temperature increase is not observed either.

In the second stage ($3000 < t$/hour < 4150), void starts to cut into and occupy the main channel. As a consequence, increasing of both hotspot resistance and effective total interconnect resistance is observed. Before the hotspot resistance reaches infinity (copper open circuit), barrier material shunts an increasing portion of total current in parallel with the copper at the hotspot.

The third stage (t/hour ≈ 4150), is the instant, or the relatively short time span during the total simulated EM failure process, the copper interconnect is separated by the void at the hotspot. Consequently, the hotspot barrier carries all the prescribed current and generates significantly more Joule heat compared to the case without the void. Since the hotspot is a small volume, having high resistance and limited dissipation area, significant resistance and temperature jump is observed, which matches the analysis in section 2.6.

In the fourth stage (t/hour > 4150), void size keeps growing into the metal layer copper branch. As copper is voided, the current is shunted onto the adjacent barrier ma-

terial, which leads to the increasing of the total effective resistance. Despite this, at this phase, we do not observe noticeable temperature change. This is because the later-added resistance corresponds to relatively large portion of barrier material, which has enough dissipation area to dissipate heat into the surrounding metal.

The above observation matches the analysis in section 2.6 and the experiment data collected in [24]. This validates that the simulation method is consistent with the practical EM failure process.

5.3.2 Joule Heating Analysis on All Domains

In this example, we investigate the wire resistance change pattern over time when the void is formed and the void grows over the critical volume. We will compare the resistance change against patterns observed in real silicon experiments.

(a) $t = 18$ (b) $t = 60.1$ (c) $t = 60.15$ (d) $t = 65$

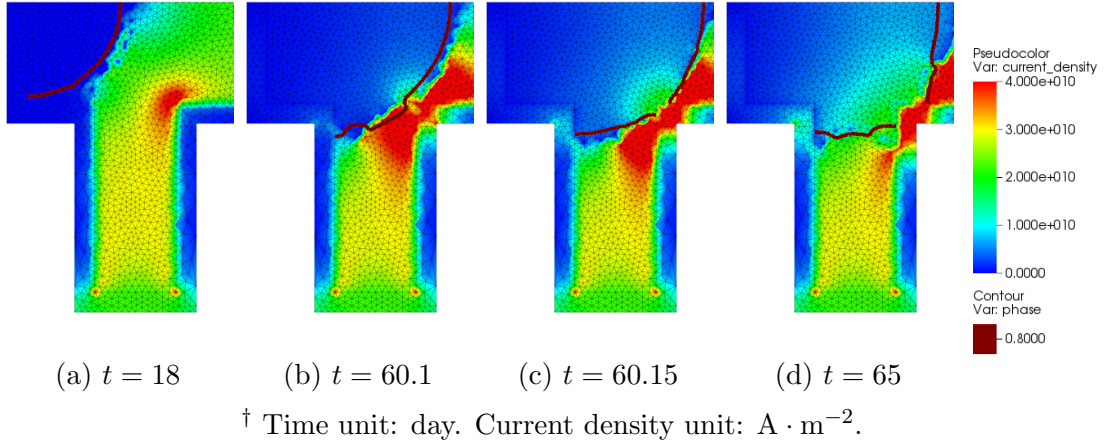† Time unit: day. Current density unit: $A \cdot m^{-2}$.

Figure 5.16: Current density as void growing.

It was observed that the in the failure process of a electromigration, vulnerable interconnect that is upstream-configured, its resistance growth follows a certain pattern [81].

Figure 5.14 shows such resistance change patterns, in which the resistance starts with a very slow increasing speed, followed by a sharp jump, and then increases approximately in a linear manner regarding time. The reason of the sharp resistance jump is due to the Joule heating induced resistance jump. This happens when the void completely cuts the copper wire into two separate parts and all the current has to suddenly be shunted to the liner, which has a much higher resistance as shown in figure 5.16.
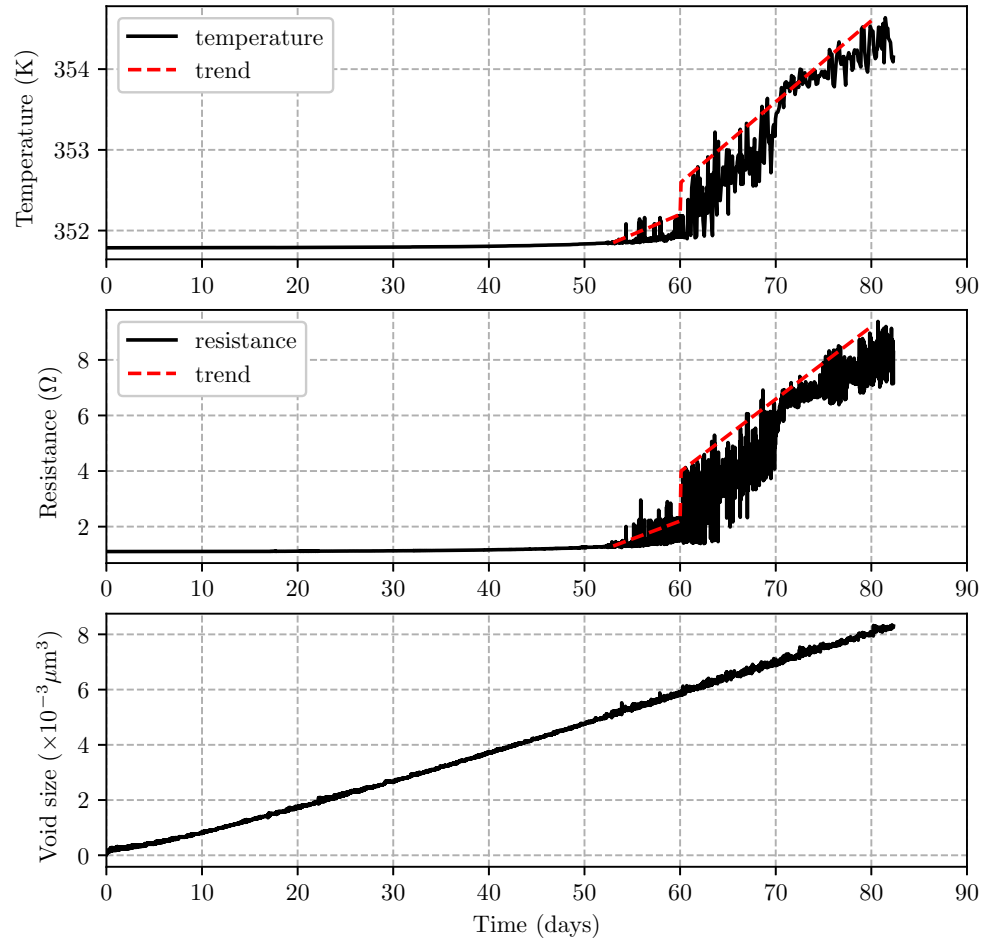

Figure 5.17: Temperature and resistance change over time.

(a) $t = 18$ (b) $t = 60.1$ (c) $t = 60.15$ (d) $t = 65$

† Time unit: day. Power density unit: $W \cdot m^{-3}$.
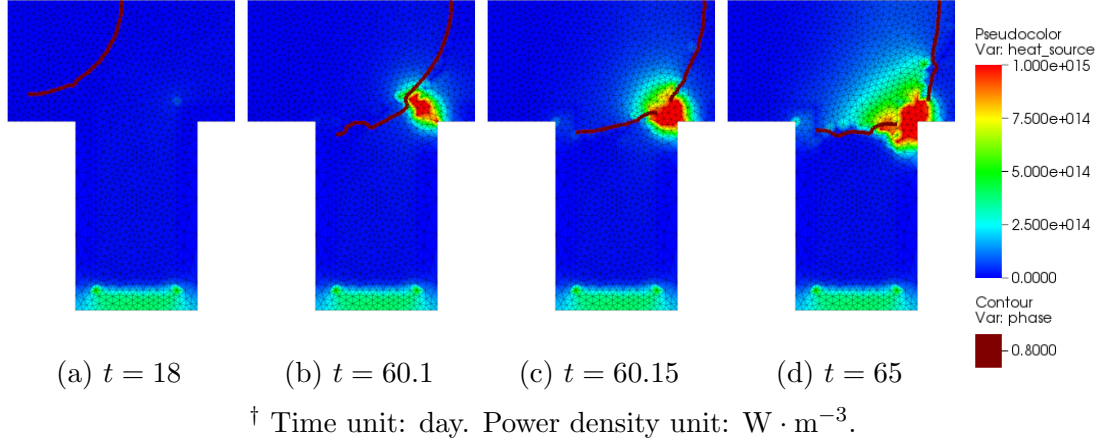
Figure 5.18: Power density (for Joule heating) change as void growing.

(a) $t = 60.1$ (b) $t = 60.15$

† Time unit: day. Temperature unit: K.
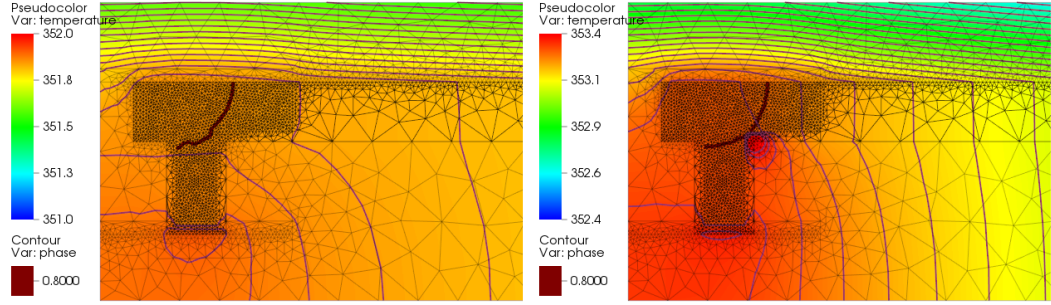
Figure 5.19: Temperature distribution and contour as void growing.

In the simulation, in order to have an interconnect that will end up with high resistance, we use a straight-connected two-segment structure similar as in the one used in section 5.2.1, but with wire lengths $20\mu m$ and $10\mu m$ and same current densities of $2 \times 10^{10} A \cdot m^{-2}$ and $1 \times 10^{10} A \cdot m^{-2}$. As shown in figure 5.17, as void volume increases, a sharp resistance jump is observed at approximately 60 days of simulated time. Local operating temperature is also increased because of the Joule heating effect.

Figure 5.16 and 5.18 respectively show the current density and Joule heating power density distribution change as the void grows to cause failure. Notice that in the figures copper is surrounded by liner material, which can be differentiated by recognizing the area where the current density is relatively low. At the moment the void cuts the copper interconnect, all current are forced to flow through liner material, which causes the rapid resistance increase. Figure 5.18 also explains the reason of temperature increase. Initially, there is only one outstanding heat source located at the bottom of the via where the current has to flow through the thin liner between layers M1 and M2. After the void gains a significant volume, another heat source is created at the liner where the current is shunting through, as depicted by the temperature contour shown in figure 5.19. Therefore, with essentially another heat source introduced by void growing, a local temperature boost is also observed, which contribute to the resistance jump shown in figure 5.17.

5.4 MTTF Analysis

In this section, we present time to failure (TTF) simulation results and comparison with and existing modeling methods and two real-chip experiments.

With different current densities applied, void saturates at different sizes, as shown in figure 5.20. Compared to experimental results, the simulated hydrostatic stress is expected to grow fast at early stage because of large current flux, and slowly drops because of the advancement of copper-void boundary. We can successfully observe that the hydrostatic stress accumulated at the void end is absorbed by the advancing copper-void boundary and stops on a relatively low level when the void is saturated. Under extreme condition of

current density applied on a copper line having enough length, we observe EM open-circuit failure before the void saturates.

The widely used statistical MTTF estimation for EM wearout is given by the Black's equation: [8]:

$$\text{MTTF} \approx A j^{-n} \exp\left(-\frac{E_a}{k_B T}\right), \tag{5.1}$$

which empirically shows relationship between current density and MTTF. Under certain operating temperature, the simulated MTTF should satisfy equation (5.1). To check the consistency with the mean time to failure experimental relationship $\text{MTTF} \propto j^{-n}$, we fit the exponent n using least squares method.
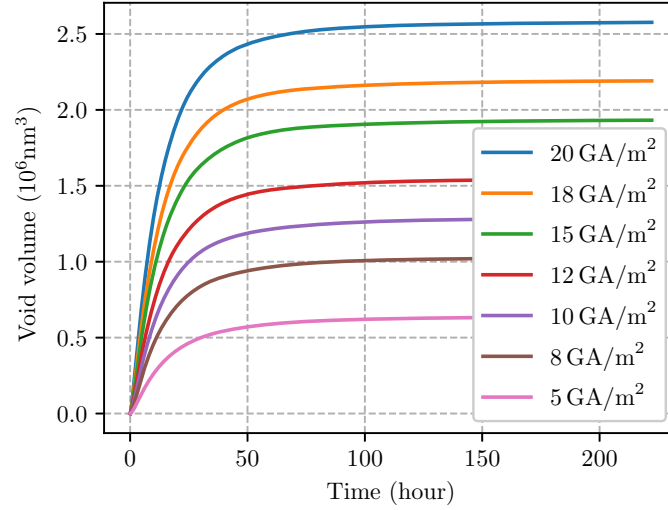
† GA: 10^9A.

Figure 5.20: Void size vs. time with different current densities applied.

Figure 5.21 shows the fitted exponent n under different operating temperature compared with two experimental results measured in [31] and [32]. We also compare the our result against a recently published work [68], which focuses on the stress development
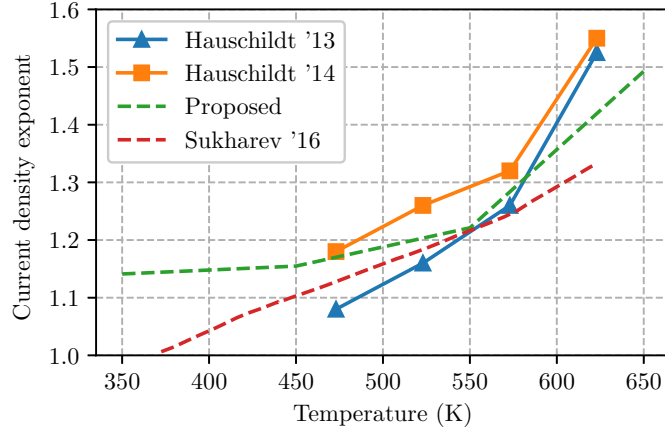
127

Figure 5.21: Extracted current density exponent n.

in the copper wire without considering the their impacts on the void volume. As we can see, the proposed FEM analysis method fits better against the two measured results than the recently published work specially at the high temperature [68].

5.5 Void Growth Speed

Figure 5.17 shows that the void volume grows at a roughly constant speed at $1.15 \times 10^{-9} \mu\mathrm{m}^3 \cdot \mathrm{s}^{-1}$. We compare this result with the void boundary velocity model proposed in work [36], where the boundary velocity (unit: $\mu\mathrm{m} \cdot \mathrm{s}^{-1}$) is calculated by

$$v_{\text{boundary}} = \frac{D_0 e^{-\frac{E_a}{k_B T}}}{k_B T} eZ\rho j. \tag{5.2}$$

To compare between the results here, we need to provide the cross section area of copper interconnect, which is $0.2\mu\mathrm{m} \times 0.15\mu\mathrm{m} = 0.03\mu\mathrm{m}^2$, to reconcile the unit difference. In our 3D FEM simulation, the activity energy used is $E_a = 0.9\mathrm{eV}$. With this E_a applied to equation (5.2), it will result an equivalent void boundary speed of $1.72 \times 10^{-9} \mu\mathrm{m}^3 \cdot \mathrm{s}^{-1}$.

128

This is significantly larger than the result $(1.15 \times 10^{-9} \mu m^3 \cdot s^{-1})$ obtained by the proposed simulator, which implies that the proposed method will deliver electromigration induced life time analysis that not as pessimistic as the existing modeling work [36].

5.6 Stationary Analysis on Copper Only Domains
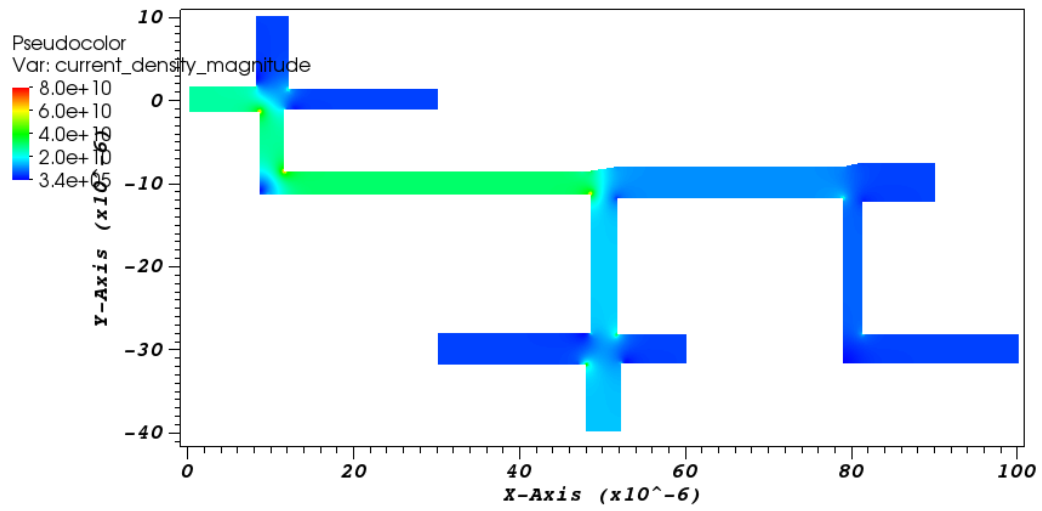
Figure 5.22: Current density analysis on 13-segment copper interconnect.

A 2D steady state simulation is performed on a 13-segment copper interconnect. Since it is stationary analysis, less transient effects are expected to be observed on the materials other than the copper. Therefore in the simulation, liner material, capping layers, and dielectric material are omitted. Figure 5.22 and 5.23 shows the current density and hydrostatic stress analysis. According to the stress distribution result as shown in the figure, the highest stressed region can there by determined by the simulator. In this case, the highest hydrostatic stress is higher than the critical stress of void growth, which implies void will grow at the corresponding site, as shown by the material phase field in figure 5.24.
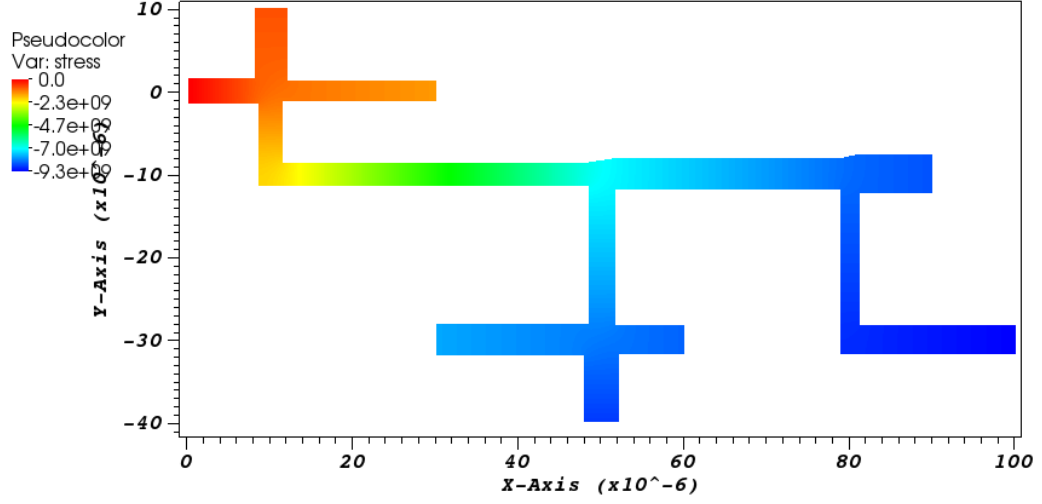
Figure 5.23: Hydrostatic stress analysis on 13-segment copper interconnect.

5.7 Summary

This chapter focuses on providing result for the validation of the proposed method. Simulated results are acquired from the implemented EM post-voiding analyzer and are compared against existing EM models as well as experiment data observed from real silicon tests. Time dependent post-void analyses are conducted on different shapes of interconnects. Results show that the hydrostatic stress distribution and void growth or saturation are as expected. Joule heating effect is observed from the proposed analyzer, showing an abrupt temperature and interconnect resistance jump while the growing void covers the main conducting copper, which matches real silicon test results. Compared to existing models, MTTF analysis derived from the proposed method is closer to the real silicon measurements. With the constraint (1.4) applied, simulated void growth speed is slower than a recently proposed EM void growth model, which implies less pessimistic EM life time estimation and meets the expectation.
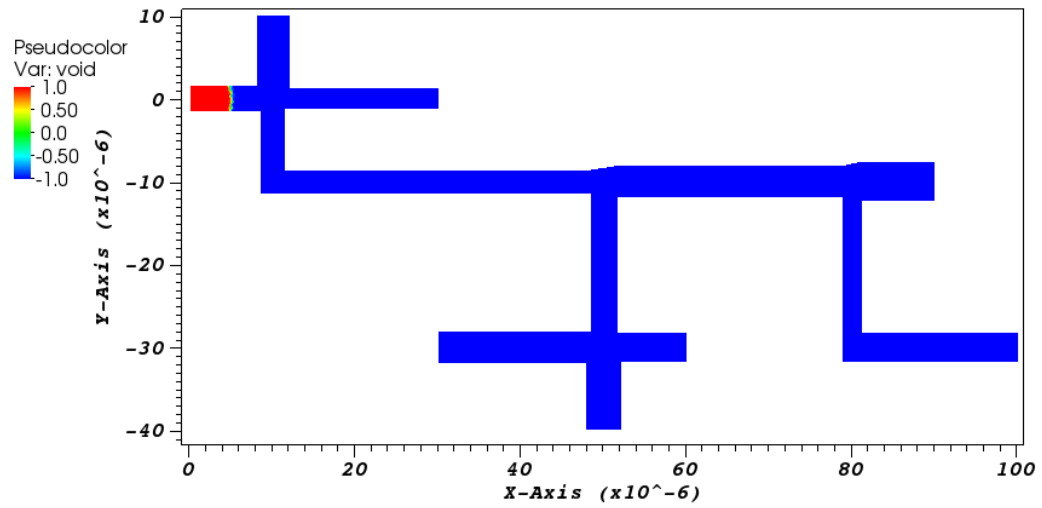
Figure 5.24: Copper-void phase field on 13-segment copper interconnect.

Chapter 6

Conclusion

6.1 Contribution of This Work

In this work, a comprehensive electromigration analysis method is presented, designed and implemented using modern finite element methods and libraries, and validated with experiment results observed in real silicon tests.

To explicitly model the void boundary movement, a phase field variable governed by the partial differential equations is resulted, which results in two coupled dynamic systems to describe the whole post-voiding stress evolution process. To incorporate the inherent physical interaction between the void volume and stress distribution in a confined interconnect wire due to atomic conservation, a regulation factor for void growth velocity is introduced such that the void boundary movement speed is controlled to ensure the atomic conservation.

A finite element (FEM) multiphysics simulation method is proposed to investigate the electromigration effect and its failure characteristics in copper interconnects in mod-

ern integrated circuits. This work focuses on void growth simulation of the post-voiding stage of the electromigration induced failure process. Three physics systems — hydrostatic stress, current density, and temperature — are taken into account while the void growth is simulated.

The software is designed to be easy to configure, scriptable, and highly compatible with the widely-used input and output formats.It facilitates modern IC designs regarding electromigration related failure and life time analysis by providing complete analysis of void growth, hydrostatic stress distribution, electrical current density distribution, and Joule heating effect. Practical simplifications are discussed and implemented in order to efficiently simulate the electromigration effect and the induced failures.

Experiments confirms that the proposed simulation method is capable to handle arbitrary interconnect shapes. Compared with real data observed in interconnect wear-out experiments, the presented simulator can correctly predict the resistance jump and Joule heating effect in the failure process. It is showed that the simulated void growth speed is less conservative than recently proposed compact EM model. Thereby the software presented in this work is validated and effective in the electromigration lifetime analysis in modern IC designs.

6.2   Future Works

During electromigration induced failure process, void behavior is addressed in this work and implemented to simulate the volume and shape change. While it is not a critical factor affecting the simulation accuracy, copper grain boundary plays an important role

guiding the void shape changing, as mentioned in chapter 1. In the future works, copper grain boundary can be modeled by using a varying effective copper atom diffusion factor as a function of spatial coordinates $D_0(\boldsymbol{x})$. With the feature of configurable initial condition provided in the implementation, such a diffusion factor can be setup and automated to configure arbitrary grain boundary for the copper interconnect.

Another possibility to further improve this work is to extend the input module to allow automatically constructing input geometry from a routed circuit design. This would strengthen the capability of the software's integration from other upstream CAD softwares, as well as widen it's applications.

6.3 Related Publications

By the time this thesis is completed, parts of the work of this thesis have been presented as poster session at *International Integrated Reliability Workshop 2017*, published as [83] by *IEEE Transactions On Very Large Scale Integration (VLSI) Systems 2018*, and accepted as [82] by *International Conference On Computer Aided Design (ICCAD) 2018*.

Bibliography

[1] Martin S Alnæs, Jan Blechta, Johan Hake, August Johansson, Benjamin Kehlet, Anders Logg, Chris Richardson, Johannes Ring, Marie E Rognes, and Garth N Wells. The fenics project version 1.5. *Archive of Numerical Software*, 3(100):9–23, 2015.

[2] Daniel Arndt, Wolfgang Bangerth, Denis Davydov, Timo Heister, Luca Heltai, Martin Kronbichler, Matthias Maier, Jean-Paul Pelteret, Bruno Turcksin, and David Wells. The deal.ii library, version 8.5. *Journal of Numerical Mathematics*, 25(3):137–145, 2017.

[3] B Bailey. Thermally challenged. *Semiconductor Engineering*, pages 1–8, 2013.

[4] Eberhard Bänsch. Local mesh refinement in 2 and 3 dimensions. *IMPACT of Computing in Science and Engineering*, 3(3):181–191, 1991.

[5] Theodore L Bergman, Frank P Incropera, David P DeWitt, and Adrienne S Lavine. *Fundamentals of heat and mass transfer*. John Wiley & Sons, 2011.

[6] Deepali N Bhate, Allan F Bower, and Ashish Kumar. A phase field model for failure in interconnect lines due to coupled diffusion mechanisms. *Journal of the Mechanics and Physics of Solids*, 50(10):2057–2083, 2002.

[7] Deepali N Bhate, Ashish Kumar, and Allan F Bower. Diffuse interface model for electromigration and stress voiding. *Journal of Applied Physics*, 87(4):1712–1721, 2000.

[8] James R Black. Electromigration—a brief survey and some recent results. *IEEE Transactions on Electron Devices*, 16(4):338–347, 1969.

[9] Illan A Blech. Electromigration in thin aluminum films on titanium nitride. *Journal of Applied Physics*, 47(4):1203–1208, 1976.

[10] DS Boudreaux, F Williams, and AJ Nozik. Hot carrier injection at semiconductor-electrolyte junctions. *Journal of Applied Physics*, 51(4):2158–2163, 1980.

[11] Protobuf—Protocol Buffers—G. Protocol buffers. *online)(Retrieved on Aug. 2, 2013) Retrieved from internet: http://code. google. com/p/protobuf*, 2012.

[12] Brook Chao, Seung-Hyun Chae, Xuefeng Zhang, Kuan-Hsun Lu, Jay Im, and Paul S Ho. Investigation of diffusion and electromigration parameters for cu–sn intermetallic compounds in pb-free solders using simulated annealing. *Acta Materialia*, 55(8):2805–2814, 2007.

[13] Hai-Bao Chen, Sheldon X-D Tan, Xin Huang, Taeyoung Kim, and Valeriy Sukharev. Analytical modeling and characterization of electromigration effects for multibranch interconnect trees. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 35(11):1811–1824, 2016.

[14] Hai-Bao Chen, Sheldon X-D Tan, Xin Huang, and Valeriy Sukharev. New electromigration modeling and analysis considering time-varying temperature and current densities. In *Design Automation Conference (ASP-DAC), 2015 20th Asia and South Pacific*, pages 352–357. IEEE, 2015.

[15] Hai-Bao Chen, Sheldon X-D Tan, Jiangtao Peng, Taeyoung Kim, and Jie Chen. Analytical modeling of electromigration failure for vlsi interconnect tree considering temperature and segment length effects. *IEEE Transactions on Device and Materials Reliability*, 17(4):653–666, 2017.

[16] Long-Qing Chen. Phase-field models for microstructure evolution. *Annual review of materials research*, 32(1):113–140, 2002.

[17] Yi-Kan Cheng, Ching-Han Tsai, Chin-Chi Teng, and Sung-Mo Steve Kang. *Electrothermal analysis of VLSI systems*. Springer Science & Business Media, 2000.

[18] Chase Cook, Zeyu Sun, Ertugrul Demircan, Mehul D Shroff, and Sheldon X-D Tan. Fast electromigration stress evolution analysis for interconnect trees using krylov subspace method. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 26(5):969–980, 2018.

[19] Chase Cook, Zeyu Sun, Taeyoung Kim, and Sheldon X-D Tan. Finite difference method for electromigration analysis of multi-branch interconnects. In *Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD), 2016 13th International Conference on*, pages 1–4. IEEE, 2016.

[20] Douglas Crockford. The application/json media type for javascript object notation (json). Technical report, 2006.

[21] RL De Orio, Hajdin Ceric, and Siegfried Selberherr. Physically based models of electromigration: From black's equation to modern tcad models. *Microelectronics Reliability*, 50(6):775–789, 2010.

[22] Roberto Lacerda de Orio. *Electromigration modeling and simulation*. na, 2010.

[23] Charles M Elliott and Zheng Songmu. On the cahn-hilliard equation. *Archive for Rational Mechanics and Analysis*, 96(4):339–357, 1986.

[24] RG Filippi, P-C Wang, A Brendler, K Chanda, and JR Lloyd. Implications of a threshold failure time and void nucleation on electromigration of copper interconnects. *Journal of Applied Physics*, 107(10):103709, 2010.

[25] Christophe Geuzaine and Jean-François Remacle. Gmsh: A 3-d finite element mesh generator with built-in pre-and post-processing facilities. *International journal for numerical methods in engineering*, 79(11):1309–1331, 2009.

[26] A Gladkikh, Y Lereah, E Glickman, M Karpovski, A Palevski, and J Schubert. Hillock formation during electromigration in cu and al thin films: Three-dimensional grain growth. *Applied physics letters*, 66(10):1214–1215, 1995.

[27] RJ Gleixner and WD Nix. A physically based model of electromigration and stress-induced void formation in microelectronic interconnects. *Journal of applied physics*, 86(4):1932–1944, 1999.

[28] Alfred Grill, John Patrick Hummel, Christopher Vincent Jahnes, Vishnubhai Vitthalb-hai Patel, and Katherine Lynn Saenger. Dual damascene processing for semiconductor chip interconnects, October 31 2000. US Patent 6,140,226.

[29] Arvind Halliyal, Mark T Ramsbey, Kuo-Tung Chang, Nicholas H Tripsas, and Robert B Ogle. Use of high-k dielectric materials in modified ono structure for semiconductor devices, January 6 2004. US Patent 6,674,138.

[30] Christine S Hau-Riege. An introduction to cu electromigration. *Microelectronics Reliability*, 44(2):195–205, 2004.

[31] M Hauschildt, M Gall, C Hennesthal, G Talut, O Aubel, KB Yeap, and E Zschech. Electromigration void nucleation and growth analysis using large-scale early failure statistics. In *AIP Conference Proceedings*, volume 1601, pages 89–98. AIP, 2014.

[32] M Hauschildt, C Hennesthal, G Talut, O Aubel, M Gall, KB Yeap, and E Zschech. Electromigration early failure void nucleation and growth phenomena in cu and cu (mn) interconnects. In *Reliability Physics Symposium (IRPS), 2013 IEEE International*, pages 2C–1. IEEE, 2013.

[33] K Hoshino, H Yagi, and H Tsuchikawa. Effect of titanium addition to copper interconnect on electromigration open circuit failure. In *VLSI Multilevel Interconnection Conference, 1990. Proceedings., Seventh International IEEE*, pages 357–359. IEEE, 1990.

[34] JK Howard and RF Ross. Hillocks as structural markers for electromigration rate measurements in thin films. *Applied Physics Letters*, 18(8):344–346, 1971.

[35] C-K Hu, MB Small, and PS Ho. Electromigration in al (cu) two-level structures: Effect of cu and kinetics of damage formation. *Journal of applied physics*, 74(2):969–978, 1993.

[36] Xin Huang, Armen Kteyan, Sheldon X-D Tan, and Valeriy Sukharev. Physics-based electromigration models and full-chip assessment for power grid networks. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 35(11):1848–1861, 2016.

[37] Xin Huang, Valeriy Sukharev, Taeyoung Kim, Haibao Chen, and Sheldon X-D Tan. Electromigration recovery modeling and analysis under time-dependent current and temperature stressing. In *Design Automation Conference (ASP-DAC), 2016 21st Asia and South Pacific*, pages 244–249. IEEE, 2016.

[38] Xin Huang, Yu Tan, Valeriy Sukharev, and Sheldon X-D Tan. Physics-based electromigration assessment for power grid networks. In *Design Automation Conference (DAC), 2014 51st ACM/EDAC/IEEE*, pages 1–6. IEEE, 2014.

[39] JianPing Jing, Lihua Liang, and Guang Meng. Electromigration simulation for metal lines. *Journal of Electronic Packaging*, 132(1):011002, 2010.

[40] Carter W Kaanta, Susuan G Bombardier, William J Cote, William R Hill, Gloria Kerszykowski, Howard S Landis, Dan J Poindexter, Curtis W Pollard, Gilbert H Ross, James G Ryan, et al. Dual damascene: A ulsi wiring technology. In *VLSI Multilevel Interconnection Conference, 1991, Proceedings., Eighth International IEEE*, pages 144–152. IEEE, 1991.

[41] Jozef Kačur. Method of rothe in evolution equations. In *Equadiff 6*, pages 23–34. Springer, 1986.

[42] Alain Karma and Wouter-Jan Rappel. Phase-field method for computationally efficient modeling of solidification with arbitrary interface kinetics. *Physical review E*, 53(4):R3017, 1996.

[43] DW Kelly, De SR Gago, OC Zienkiewicz, I Babuska, et al. A posteriori error analysis and adaptive processes in the finite element method: Part I—error analysis. *International journal for numerical methods in engineering*, 19(11):1593–1619, 1983.

[44] MA Korhonen, P Bo/rgesen, DD Brown, and Che-Yu Li. Microstructure based statistical model of electromigration damage in confined line metallizations in the presence of thermally induced stresses. *Journal of applied physics*, 74(8):4995–5004, 1993.

[45] MA Korhonen, P Bo/rgesen, KN Tu, and Che-Yu Li. Stress evolution due to electromigration in confined metal lines. *Journal of Applied Physics*, 73(8):3790–3799, 1993.

[46] Qi Li, Kazumasa Ito, Zhishen Wu, Christopher S Lowry, and Steven P Loheide II. Comsol multiphysics: A novel approach to ground water modeling. *Groundwater*, 47(4):480–487, 2009.

[47] Zhilin Li, Hongkai Zhao, and Huajian Gao. A numerical study of electro-migration voiding by evolving level set functions on a fixed cartesian grid. *Journal of Computational Physics*, 152(1):281–304, 1999.

[48] Minghui Lin and Cemal Basaran. Electromigration induced stress analysis using fully coupled mechanical–diffusion equations with nonlinear material properties. *Computational Materials Science*, 34(1):82–98, 2005.

[49] Yong Liu, Lihua Liang, Scott Irving, and Timwah Luk. 3d modeling of electromigration combined with thermal–mechanical effect for ic device and package. *Microelectronics Reliability*, 48(6):811–824, 2008.

[50] Yongkun Liu, CL Cox, and RJ Diefendorf. Finite element analysis of the effects of geometry and microstructure on electromigration in confined metal lines. *Journal of applied physics*, 83(7):3600–3608, 1998.

[51] JR Lloyd. Electromigration in integrated circuit conductors. *Journal of Physics D: Applied Physics*, 32(17):R109, 1999.

[52] Joe McPherson, Vijay Reddy, Kaustav Banerjee, and Huy Le. Comparison of e and 1/e tddb models for sio/sub 2/under long-term/low-field test conditions. In *Electron Devices Meeting, 1998. IEDM'98. Technical Digest., International*, pages 171–174. IEEE, 1998.

[53] JW McPherson and HC Mogul. Underlying physics of the thermochemical e model in describing low-field time-dependent dielectric breakdown in sio 2 thin films. *Journal of Applied Physics*, 84(3):1513–1523, 1998.

[54] EA Mechtly. *The International System of Units: Physical Constants and Conversion Factors*, volume 7012. Scientific and Technical Information Division, National Aeronautics and Space Administration, 1964.

[55] Nele Moelans, Bart Blanpain, and Patrick Wollants. An introduction to phase-field modeling of microstructure evolution. *Calphad*, 32(2):268–294, 2008.

[56] Ennis T Ogawa, Ki-Don Lee, Volker A Blaschke, and Paul S Ho. Electromigration reliability issues in dual-damascene cu interconnections. *IEEE Transactions on reliability*, 51(4):403–419, 2002.

[57] Oliver Penrose and Paul C Fife. Thermodynamically consistent models of phase-field type for the kinetic of phase transitions. *Physica D: Nonlinear Phenomena*, 43(1):44–62, 1990.

[58] DG Pierce and PG Brusius. Electromigration: A review. *Microelectronics Reliability*, 37(7):1053–1072, 1997.

[59] Reinhold Pregla. The method of lines. *Analysis of Electromagnetic Fields and Waves: The Method of Lines*, pages 1–13, 1989.

[60] Open CASCADE SAS. Open cascade technology, 3d modeling & numerical simulation, 2008.

[61] M Schimschak and J Krug. Electromigration-driven shape evolution of two-dimensional voids. *Journal of Applied Physics*, 87(2):695–703, 2000.

[62] Dieter K Schroder. Negative bias temperature instability: What do we understand? *Microelectronics Reliability*, 47(6):841–852, 2007.

[63] Dieter K Schroder and Jeff A Babcock. Negative bias temperature instability: Road to cross in deep submicron silicon semiconductor manufacturing. *Journal of applied Physics*, 94(1):1–18, 2003.

[64] Shoso Shingubara, Yasushi Nakasaki, and Hisashi Kaneko. Electromigration in a single crystalline submicron width aluminum interconnection. *Applied physics letters*, 58(1):42–44, 1991.

[65] Ingo Steinbach, Franco Pezzolla, Britta Nestler, Markus Seeßelberg, Robert Prieler, Georg J Schmitz, and Joao LL Rezende. A phase field concept for multiphase systems. *Physica D: Nonlinear Phenomena*, 94(3):135–147, 1996.

[66] Tadeusz Stolarski, Yuji Nakasone, and Shigeka Yoshimoto. *Engineering analysis with ANSYS software*. Butterworth-Heinemann, 2018.

[67] Valeriy Sukharev, Xin Huang, Hai-Bao Chen, and Sheldon X-D Tan. Ir-drop based electromigration assessment: Parametric failure chip-scale analysis. In *Proceedings of the 2014 IEEE/ACM International Conference on Computer-Aided Design*, pages 428–433. IEEE Press, 2014.

[68] Valeriy Sukharev, Armen Kteyan, and Xin Huang. Postvoiding stress evolution in confined metal lines. *IEEE Transactions on Device and Materials Reliability*, 16(1):50–60, 2016.

[69] Valeriy Sukharev, Armen Kteyan, and Ehrenfried Zschech. Physics-based models for em and sm simulation in three-dimensional ic structures. *IEEE Transactions on Device and Materials Reliability*, 12(2):272–284, 2012.

[70] Zeyu Sun, Ertugrul Demircan, Mehul D Shroff, Chase Cook, and Sheldon X-D Tan. Fast electromigration immortality analysis for multi-segment copper interconnect wires. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2018.

[71] Zeyu Sun, Ertugrul Demircan, Mehul D Shroff, Taeyoung Kim, Xin Huang, and Sheldon X-D Tan. Voltage-based electromigration immortality check for general multi-branch interconnects. In *Computer-Aided Design (ICCAD), 2016 IEEE/ACM International Conference on*, pages 1–7. IEEE, 2016.

[72] Naohito Suzumura, Shigehisa Yamamoto, D Kodama, K Makabe, J Komori, Eiichi Murakami, S Maegawa, and K Kubota. A new tddb degradation model based on cu ion drift in cu interconnect dielectrics. In *Reliability Physics Symposium Proceedings, 2006. 44th Annual., IEEE International*, pages 484–489. IEEE, 2006.

[73] Cher Ming Tan, Yuejin Hou, and Wei Li. Revisit to the finite element modeling of electromigration for narrow interconnects. *Journal of applied physics*, 102(3):033705, 2007.

[74] Sheldon X-D Tan, Hussam Amrouch, Taeyoung Kim, Zeyu Sun, Chase Cook, and Joerg Henkel. Recent advances in em and bti induced reliability modeling, analysis and optimization. *Integration, the VLSI Journal*, 2017.

[75] X-D Sheldon Tan and C-J Richard Shi. Fast power/ground network optimization based on equivalent circuit modeling. In *Proceedings of the 38th annual Design Automation Conference*, pages 550–554. ACM, 2001.

[76] Janet M Towner. Electromigration-induced short circuit failure. In *Reliability Physics Symposium, 1985. 23rd Annual*, pages 81–86. IEEE, 1985.

[77] Ruopeng Wang, Thomas Benner, Alma Gregory Sorensen, and Van Jay Wedeen. Diffusion toolkit: a software package for diffusion imaging data processing and tractography. In *Proc Intl Soc Mag Reson Med*, volume 15. Berlin, 2007.

[78] Xiaoyi Wang, Hongyu Wang, Jian He, Sheldon X-D Tan, Yici Cai, and Shengqi Yang. Physics-based electromigration modeling and assessment for multi-segment interconnects in power grid networks. In *Proceedings of the Conference on Design, Automation & Test in Europe*, pages 1731–1736. European Design and Automation Association, 2017.

[79] Kirsten Weide-Zaage, David Dalleau, and Xiaoying Yu. Static and dynamic analysis of failure locations and void formation in interconnects due to various migration mechanisms. *Materials Science in Semiconductor Processing*, 6(1-3):85–92, 2003.

[80] Linda Wilson. International technology roadmap for semiconductors (itrs). *Semiconductor Industry Association*, 2013.

[81] Lijuan Zhang. *Effects of scaling and grain structure on electromigration reliability of cu interconnects*. PhD thesis, 2010.

[82] Hengyang Zhao and Sheldon Tan. Multi-physics-based fem analysis for post-voiding analysis of electromigration failure effects (invited paper). In *Computer-Aided Design (ICCAD), 2018 IEEE/ACM International Conference on*. IEEE, 2018.

[83] Hengyang Zhao and Sheldon X-D Tan. Postvoiding fem analysis for electromigration failure characterization. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, (99):1–11, 2018.