AN EFFICIENT MODIFIED HIERARCHICAL DOMAIN DECOMPOSITION FOR TWO-DIMENSIONAL MAGNETOTELLURIC FORWARD PROBLEMS

TAWAT RUNG-ARUNWAN

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR
THE DEGREE OF MASTER OF SCIENCE (PHYSICS)
FACULTY OF GRADUATE STUDIES
MAHIDOL UNIVERSITY
2010

Thesis

entitled

AN EFFICIENT MODIFIED HIERARCHICAL DOMAIN DECOMPOSITION FOR TWO-DIMENSIONAL MAGNETOTELLURIC FORWARD PROBLEMS

.....
Mr. Tawat Rung-Arunwan
Candidate

.....
Assoc. Prof. Weerachai Siripunvaraporn,
Ph.D. (Geophysics)
Major advisor

.....
Lect. Phichet Kittara,
Ph.D. (Astrophysics)
Co-advisor

.....
Lect. Withoon Chunwachirasiri,
Ph.D. (Physics)
Co-advisor

.....
Prof. Banchong Mahaisavariya,
M.D., Dip Thai Board of Orthopedics
Dean
Faculty of Graduate Studies
Mahidol University

.....
Assoc. Prof. Wannapong Triampo,
Ph.D. (Physics)
Program Director
Master of Science Program
in Physics
Faculty of Science
Mahidol University

Thesis

entitled

AN EFFICIENT MODIFIED HIERARCHICAL DOMAIN DECOMPOSITION FOR TWO-DIMENSIONAL MAGNETOTELLURIC FORWARD PROBLEMS

was submitted to the Faculty of Graduate Studies, Mahidol University
for the degree of Master of Science (Physics)
on
June 23, 2010

.....
Mr. Tawat Rung-Arunwan
Candidate

.....
Lect. Siriporn Chaisri,
Ph.D. (Geophysics)
Chair

.....
Assoc. Prof. Weerachai Siripunvaraporn,
Ph.D. (Geophysics)
Member

.....
Lect. Phichet Kittara,
Ph.D. (Astrophysics)
Member

.....
Lect. Withoon Chunwachirasiri,
Ph.D. (Physics)
Member

.....
Prof. Banchong Mahaisavariya,
M.D., Dip Thai Board of Orthopedics
Dean
Faculty of Graduate Studies
Mahidol University

.....
Prof. Skorn Mongkolsuk,
Ph.D.
Dean
Faculty of Science
Mahidol University

ACKNOWLEDGEMENTS

I would like to express my profound gratitude to my advisor, Assoc. Prof. Weerachai Siripunvarporn, for his kindness, patience and wise suggestion. I am also highly grateful to Dr. Phichet Kittara, Dr. Withoon Chunwachirasiri and Dr. Siriporn Chaisri for their invaluable comments to improve this thesis. I gratefully acknowledge Thailand Center of Excellence in Physics (ThEP) for financial support. I also thank the staff in the Department of Physics, particularly Miss Nipaporn Suwannawong.

I greatly appreciate LaTeX class file by Asst. Prof. Michael Allen and also thank Areeya Chantasri and Achara Seripienlert for their suggestions to survive Thesis format with LaTeX.

For much help and encouragement, I would like to thank the members of Geophysics Research Unit, Mahidol university, Puwis Amatyakul, Songkhun Boonchaisuk, Chatchai Vachiratienchai, Weerachai Sarakorn, Suntaree Unhapipat, Ananya Satitpitakul, Patchawee Nualkhow and undergrad students.

And I thank my family for their kind support.

THANK YOU ALL.

Tawat Rung-Arunwan

AN EFFICIENT MODIFIED HIERARCHICAL DOMAIN DECOMPOSITION FOR TWO-DIMENSIONAL MAGNETOTELLURIC FORWARD PROBLEMS.

TAWAT RUNG-ARUNWAN 5036994 SCPY/M

M.Sc. (PHYSICS)

THESIS ADVISORY COMMITTEE : WEERACHAI SIRIPUNVARAPORN, Ph.D. (GEOPHYSICS), PHICHET KITTARA, Ph.D. (ASTROPHYSICS), WITHOON CHUNWACHIRASIRI, Ph.D. (PHYSICS)

ABSTRACT

A global domain is uniformly partitioned into many smaller non-overlapping sub-domains. Two modifications are made to the hierarchical domain decomposition method to suit two-dimensional magnetotelluric problems. Under the hierarchical decomposition, the unknowns are classified into three types: interior, interface, and intersection. Instead of solving the whole domain, these divided systems are successively solved with the lower and upper (LU) factorization via many smaller sub-systems and a reduced system. By taking advantage of the finite difference method, the first modification is to separate the interface into horizontal and vertical interfaces, and the second modification is to apply a red-black ordering to the horizontal and vertical interface system.

The accuracy of the developed technique is first validated with both synthetic and inverted models. At the proper number of sub-domains, the developed technique is better than solving the whole domain in terms of computational time and memory, and also has a higher efficiency when the model size increases. Pre-estimating the memory used leads to a selection of the optimal number of sub-domains prior to the actual calculation. Thus the trial and error approach for choosing the number of sub-domains can be avoided. The proposed domain decomposition technique is proven to be efficient at solving two-dimensional problems and also shows the potential for solving three-dimensional problems.

KEY WORDS : MAGNETOTELLURIC / FORWARD PROBLEM / DOMAIN DECOMPOSITION

94 pages

การแบ่งโดเมนอย่างมีลำดับขั้นโดยปรับแก้ให้มีประสิทธิภาพสำหรับการแก้ปัญหาไปข้างหน้าของแมกนีโตเทลลูริคในสองมิติ
AN EFFICIENT MODIFIED HIERARCHICAL DOMAIN DECOMPOSITION FOR TWO-DIMENSIONAL MAGNETOTELLURIC FORWARD PROBLEMS

ธวัช รุ่งอรุณวรรณ 5036994 SCPY/M

วท.ม. (ฟิสิกส์)

คณะกรรมการที่ปรึกษาวิทยานิพนธ์ : วีระชัย สิริพันธ์วราภรณ์, Ph.D. (GEOPHYSICS), พิเชษฐ กิจธารา, Ph.D. (ASTROPHYSICS), วิทูร ชื่นวชิรศิริ, Ph.D. (PHYSICS)

บทคัดย่อ

โดเมนทั้งหมดถูกแบ่งเป็นโดเมนย่อยที่มีขนาดเท่ากัน การแบ่งโดเมนอย่างมีลำดับขั้นได้ถูกปรับเปลี่ยนสองอย่างเพื่อให้เหมาะสมกับการแก้ปัญหาไปข้างหน้าของแมกนีโตเทลลูริค ในการแบ่งโดเมนอย่างมีลำดับนั้นตัวแปรถูกแบ่งออกเป็นสามชนิดคือ ตัวแปรภายใน ตัวแปรบนรอยต่อ และตัวแปรบนจุดตัด และโดยอาศัยข้อดีของไฟไนต์ดิฟเฟอร์เรนต์ การปรับเปลี่ยนที่หนึ่งคือการแบ่งรอยต่อออกเป็นรอยต่อในแนวราบและรอยต่อในแนวตั้ง การปรับเปลี่ยนที่สองคือการประยุกต์ใช้การกำหนดสีแดงและดำเพื่อใช้แก้ปัญหาของรอยต่อทั้งในแนวราบและในแนวตั้ง

วิธีที่พัฒนาขึ้นมานี้ถูกนำไปใช้แก้ปัญหาจากแบบจำลองสังเคราะห์และแบบจำลองที่ได้จากการแก้ปัญหาย้อนกลับเพื่อทดสอบความถูกต้อง และเมื่อโดเมนย่อยมีจำนวนเหมาะสม วิธีที่พัฒนาขึ้นมานี้ให้ผลที่ดีกว่าการแก้ปัญหาทั้งโดเมนในแง่ของเวลาและหน่วยความจำที่ใช้ นอกจากนี้วิธีที่พัฒนาขึ้นยังให้ประสิทธิภาพที่ดีขึ้น เมื่อโมเดลมีขนาดใหญ่ จากการที่จำนวนหน่วยความจำที่ใช้สามารถประมาณได้ล่วงหน้าจึงนำไปสู่การเลือกจำนวนโดเมนย่อยที่ให้ผลดีที่สุด ดังนั้นเราจึงสามารถหลีกเลี่ยงการลองผิดลองถูกในการเลือกจำนวนโดเมนย่อยได้ ด้วยเหตุนี้วิธีที่นำเสนอจึงเป็นวิธีที่มีประสิทธิภาพในการแก้ปัญหาสองมิติ และมีศักยภาพที่จะนำไปใช้กับปัญหาสามมิติ

94 หน้า

CONTENTS

CONTENTS (cont.)

LIST OF TABLES

LIST OF FIGURES

Figures **Page**

LIST OF FIGURES (cont.)

Figures **Page**

LIST OF FIGURES (cont.)

Figures **Page**

LIST OF FIGURES (cont.)

Figures **Page**

LIST OF FIGURES (cont.)

Figures **Page**

LIST OF FIGURES (cont.)

Figures **Page**

LIST OF FIGURES (cont.)

Figures **Page**

LIST OF FIGURES (cont.)

Figures **Page**

CHAPTER I

INTRODUCTION

Magnetotellurics (MT) is an increasingly popular method in various geophysical applications such as environmental studies, tectonic and crustal studies, geothermal exploration, and mineral and gas exploration. Thus, the fast inversion programs, the process that finds the model to fit the observed data, are demanded. The MT forward problem is very important, because the inversion requires many forward solutions. The forward algorithm must therefore be improved to gain more efficient inversion. Here, the domain decomposition method is chosen to solve two-dimensional magnetotelluric forward problems as a feasibility test before applying it to three-dimensional problems.

In this thesis, the fundamentals of magnetotellurics is given in Chapter 2. The MT forward problem with the finite difference (FD) approach is described in Chapter 3. Solving the resulting systems with the traditional iterative solvers might be less efficient because the iterative solvers might fail to converge when the models are too large or too complicated. The direct solver, LU factorization, was therefore chosen as a solver, and will be referred to as FDWD.

However, to solve large problems, particularly three-dimensional problems, direct solvers might be impractical (see Streich, 2009) because of memory limitations. Basically, the domain decomposition method turns one system of equations into many smaller sub-systems of interiors and a reduced system of interfaces, the Schur complement. Therefore, applying the direct solver on the decomposed domain is more reasonable.

The domain decomposition method used in this thesis is based on the hierarchical domain decomposition method (HD) which will be described in Chapter 4. Under the hierarchical domain decomposition, the unknowns could be classified into three categories: interior, interface and intersection. The system is then transformed to many smaller sub-systems of interiors, and two reduced systems of interfaces and intersections. From the numerical experiments, HD provides the same accuracy as FDWD and uses lower memory when there are small numbers of sub-domains, but HD did not improve the computational time.

To obtain a higher efficiency, the hierarchical domain decomposition was first modified by separating the interfaces into the horizontal and vertical interfaces,

which will be referred to as MHD and will be described in Chapter 5. This results in a smaller interface system. The numerical results also showed that MHD is superior to HD in terms of time and memory. However, the computation time of MHD is still quite close to that of FDWD.

A second modification was therefore made to the interface system of MHD. Because of the block tridiagonal matrix pattern in some parts of the interface system, a combination of red-black ordering and Schur complement could be implemented. This modification will be referred to as MHDRB and will be described in Chapter 6. From the numerical results, MHDRB is better than HD and MHD in terms of time and memory. Moreover, MHDRB has better efficiency than FDWD at proper numbers of sub-domains.

In Chapter 7, the mesh size is varied to demonstrate the potential of the developed domain decomposition methods. All three domain decomposition solvers become more efficient as the model size increases. Thus the developed technique is proven to be efficient in two-dimensional problems and also shows potential for solving three-dimensional problems. When the hierarchical domain decomposition is modified, the minimized computational loads are provided at different numbers of sub-domains. A changing in the optimal number of sub-domains is also discussed. Moreover, the strategy of choosing the optimal number of sub-domains is also presented and the conclusions of this thesis are given at the last chapter.

CHAPTER II

FUNDAMENTAL OF MAGNETOTELLURICS

2.1 Introduction to magnetotellurics

Magnetotellurics (MT) is one of the techniques of geophysics for imaging the resistivity structure of the subsurface by measuring the natural occurring electromagnetic waves as a source. Due to the broad frequency range of the natural wave source, MT sounding samples a volume of the Earth of depth ranging from tens of meters to several hundreds of kilometers. This is deeper than other active EM methods which are limited by the power of source. MT has found use in environmental applications (e.g. Pellerin *et al.*, 2004; Hautot *et al.*, 2002), in petroleum and mineral exploration (e.g. Mitsuhata *et al.*, 1999; Turkoglu *et al.*, 2009; Tuncer *et al.*, 2006), in geothermal explorations (e.g. Tang *et al.*, 2008), and in crustal and tectonic studies (e.g. Noguera & Rea, 2000; Unsworth, 2010).

To study deep structures, the observing periods must be extended, because the penetration depth of electromagnetic waves is strongly determined by the electromagnetic skin depth,

$$\delta = \sqrt{\frac{2}{\omega\sigma\mu}} = \sqrt{\frac{2\rho}{\omega\mu}}, \tag{2.1}$$

where μ is magnetic permeability, assumed to be that of free space μ_0 for earth studies, σ is the electrical conductivity, and ω is an angular frequency, $\omega = 2\pi f$. Skin depth is defined as the length that an electromagnetic wave decays by a factor of $1/e$ as it propagates into the medium. It depends on frequency f and electrical conductivity σ, the inverse of resistivity ρ, as shown in Figure 2.1. The exponential decay of electromagnetic waves makes them less sensitive to the structure deeper than their skin depths. In MT studies, the investigation depth of each frequency is generally equal to its electromagnetic skin depth.

The sources of electromagnetic waves can be categorized into two groups depending on their frequencies. Those with frequencies less than 1 Hz are generated from the interaction between the solar wind and the ionosphere. The frequency range above 1 Hz comes from electrical phenomena in the atmosphere, e.g., lightning. This local effect can saturate the recording signals.

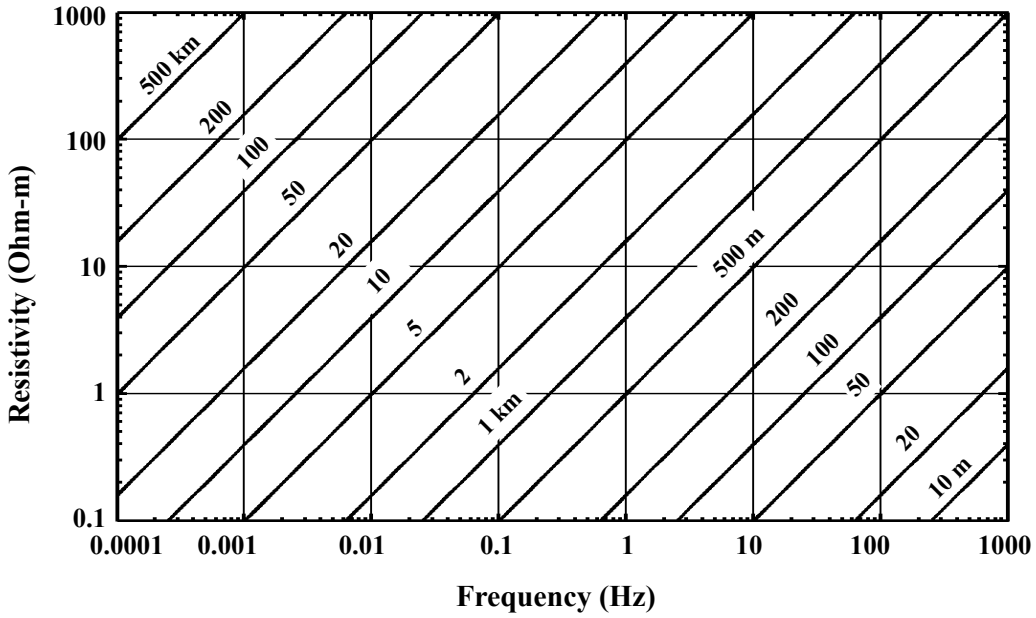


Figure 2.1: Skin depths of electromagnetic waves in a homogeneous medium plotted as a function of frequency and resistivity.

MT work was independently pioneered by Tikhonov (1950) and Cagniard (1953). The theoretical formulation of MT is based on the assumption that electromagnetic waves are considered to be plane waves. Assuming that the wave propagates vertically down into the earth, once it reaches the surface, the magnetic field, which is changing with time, results in varying electric fields associated with currents in the earth, called telluric currents.

By simultaneously recording the orthogonal components of electric and magnetic fields at the surface, the impedances can be obtained. Consequently, the resistivity structure of the earth can be revealed.

2.2 Impedance tensor: MT response and dimensionality

In an MT survey, the electric and magnetic fields are generally measured in the north-south and east-west directions, denoted by x- and y-directions, respectively. This is because the electric and magnetic field components are related in the orthogonal directions. The schematic plane view of an MT station is also shown in Figure 2.2. Examples of time-varying electromagnetic fields are shown in Figure 2.3. E_x is correlated to H_y, and E_y is correlated to H_x. The electric and magnetic fields in the time domain are then transformed into the frequency domain using the Fourier transform.

The electric field **E** and magnetic field **H** are related by the impedance

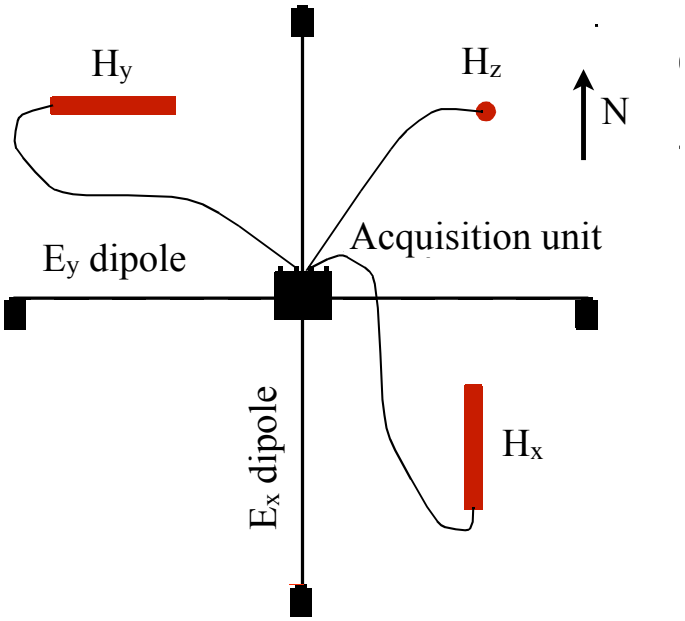


Figure 2.2: A schematic plane view of an MT station. The magnetic field components, H_x, H_y and H_z, are measured by magnetometers. The electric fields, E_x and E_y, are measured by the electric dipoles.

tensor $\mathbf{Z}$ as follows,

$$\mathbf{E} = \mathbf{ZH}, \tag{2.2}$$

or

$$\begin{pmatrix} E_x \\ E_y \end{pmatrix} = \begin{pmatrix} Z_{xx} & Z_{xy} \\ Z_{yx} & Z_{yy} \end{pmatrix} \begin{pmatrix} H_x \\ H_y \end{pmatrix}, \tag{2.3}$$

where $\mathbf{E}$, $\mathbf{H}$, and $\mathbf{Z}$ are functions of the angular frequency ω. The magnetotelluric responses, apparent resistivity ρ_a and phase ϕ, are obtained from

$$\rho_{a,ij} = \frac{1}{\omega\mu}|Z_{ij}|^2, \tag{2.4}$$

and

$$\phi_{ij} = \tan^{-1}\left(\frac{\mathrm{Im}\{Z_{ij}\}}{\mathrm{Re}\{Z_{ij}\}}\right), \tag{2.5}$$

respectively, where i or j denote the x- or y-directions. The apparent resistivity is defined as the average resistivity of an equivalent uniform half-space. The phase represents the relationship between the electric and magnetic fields.

An example of apparent resistivity and phase from the two-layer earth model (Figure 2.4a) is shown in Figure 2.4b. High frequency responses reflect the shallow structure, $\rho_1 = 100\ \Omega\,\mathrm{m}$, and, as decreasing frequencies (or increasing periods), the apparent resistivity curve tends to the resistivity ρ_2 of the second layer. When $\rho_2 = \rho_1 = 100\ \Omega\,\mathrm{m}$, the two-layer earth model becomes a halfspace model, and the apparent resistivity is equal to the true resistivity of the model and it s phase is $-45°$. If the

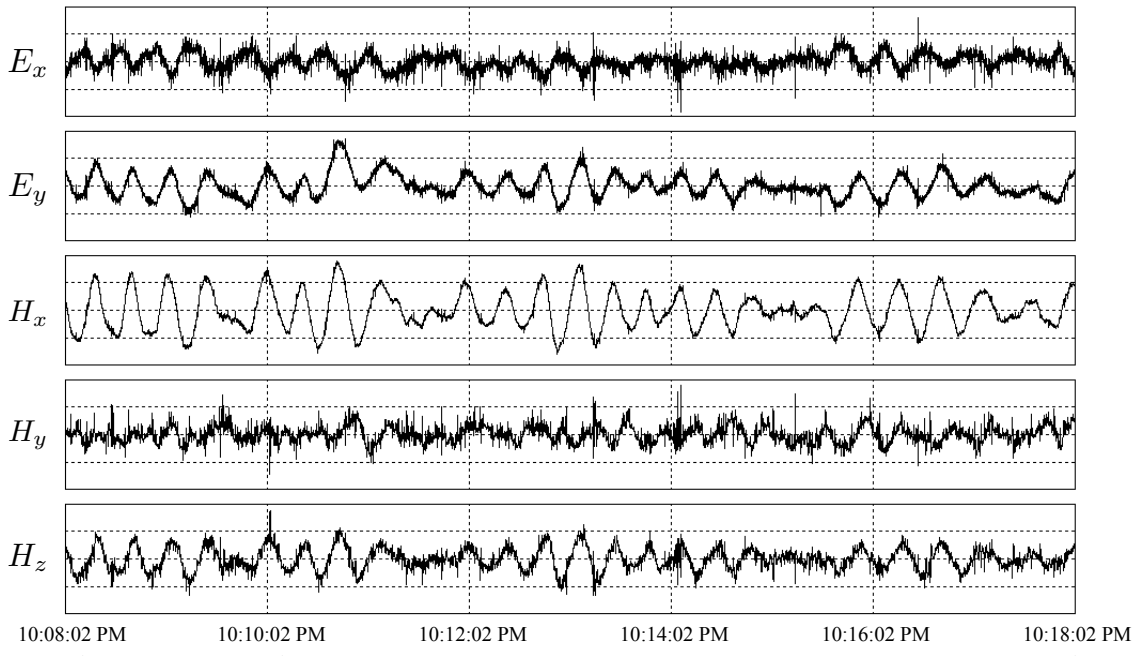


Figure 2.3: The times series data of electric and magnetic field components measured at one MT station.

second layer is more conductive, the phase decreases. However the phase increases with the resistive layer.

The impedance tensor $\mathbf{Z}$ also contains information about dimensionality. For a 1-D Earth (Figure 2.5a), where the conductivity varies with depths only,

$$Z_{xx} = Z_{yy} = 0,$$
$$Z_{xy} = -Z_{yx}.$$

In a 2-D Earth (Figure 2.5b), the conductivity varies in the vertical and one horizontal directions, and the other horizontal direction is called the strike direction. In this case,

$$Z_{xx} = -Z_{yy},$$
$$Z_{xy} \neq -Z_{yx}.$$

An example of a 3-D model is shown in Figure 2.5c. The conductivity varies in all three directions, and so

$$Z_{xx} \neq -Z_{yy},$$
$$Z_{xy} \neq -Z_{yx}.$$

2.3 Governing equations

The magnetotelluric responses can be computed from the ratio of electric to magnetic field strength. The relationship between electric and magnetic fields is governed by Maxwell's equations. Inside a medium, in the absence of free charge, Maxwell's equations are:

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}, \tag{2.6}$$

$$\nabla \times \mathbf{H} = \mathbf{J} + \frac{\partial \mathbf{D}}{\partial t}, \tag{2.7}$$

$$\nabla \cdot \mathbf{B} = 0, \tag{2.8}$$

$$\nabla \cdot \mathbf{D} = 0, \tag{2.9}$$

where $\mathbf{E}$ is the electric field, $\mathbf{B}$ is magnetic induction, $\mathbf{H}$ is the magnetic intensity, $\mathbf{D}$ is electric displacement, and $\mathbf{J}$ is the electric current density.

For an MT study, the electrical permittivity of rock, ϵ, is negligible and the magnetic permeability, μ, is assumed to be that of free space, μ_0 (Kaufman & Keller, 1981).

The Earth is assumed to be a linear isotropic medium, so two further relationships are hold:

$$\mathbf{B} = \mu \mathbf{H}, \tag{2.10}$$

$$\mathbf{D} = \epsilon \mathbf{E}. \tag{2.11}$$

The Earth acts as an ohmic conductor, so

$$\mathbf{J} = \sigma \mathbf{E}. \tag{2.12}$$

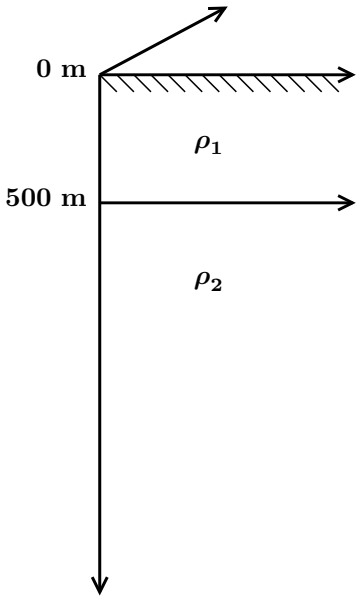
Applying equations (2.10), (2.11) and (2.12) to equations (2.6) to (2.9), Maxwell's equations with the time dependent part assumed to be $e^{-i\omega t}$ are rewritten as,

$$\nabla \times \mathbf{E} = i\omega\mu \mathbf{H}, \tag{2.13}$$

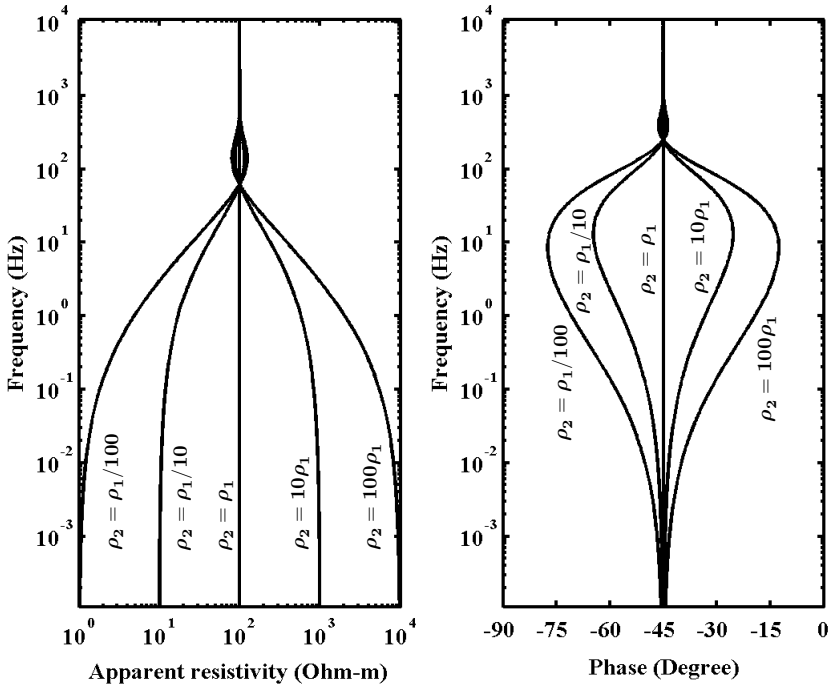
$$\nabla \times \mathbf{H} = \sigma \mathbf{E}, \tag{2.14}$$

where ω is the angular frequency.

From equations (2.13) and (2.14), the electric and magnetic fields depend on the electrical conductivity σ, or inversely on the electrical resistivity ρ. To solve for the electric and magnetic fields, the model is discretized and the finite difference method is applied. This will be described in the next chapter.



(a)



(b)

Figure 2.4: (a) The two-layer model. The conductivity of first layer ρ_1 is 100 Ω m and that of second layer ρ_2 is varied. (b) MT responses from the two-layer model plotted as a function of frequency.

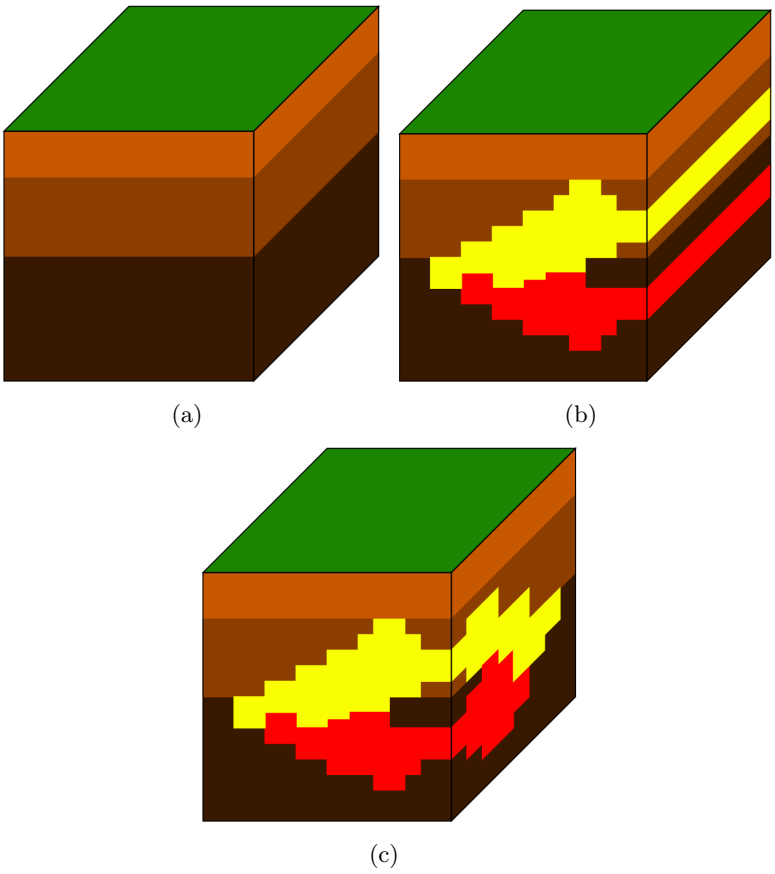


Figure 2.5: Examples of spatial variation of model parameters in (a) 1-D model (b) 2-D model and (c) 3-D model.

CHAPTER III

MAGNETOTELLURIC FORWARD PROBLEM: FINITE DIFFERENCE APPROACH

The forward algorithm is a very important part of the inversion program. It is used to calculate the responses, apparent resistivity ρ_a and phase ϕ, from the known structures. At the surface, the MT response are computed from the ratio of the electric fields and magnetic fields, which are obtained by solving Maxwell's equations.

To solve Maxwell's equations, there are three usual approaches: the finite difference (FD) method (e.g. Mackie *et al.*, 1994; Smith, 1996; Siripunvaraporn *et al.*, 2002, 2005), the finite element (FE) method (e.g. Mitsuhata & Uchida, 2004; Zyserman & Santos, 2000; Zyserman *et al.*, 1999; Wannamaker *et al.*, 1987; Xueming *et al.*, 2010) and the integral equation (IE) method (e.g. Avdeev & Avdeeva, 2009; Xiong, 1992; Wannamaker, 1991). The FD and FE methods are efficient and robust than the IE technique. However, FD is more popular due to its simplicity and accuracy.

In this chapter, the forward modeling code based on the finite difference approach is described in detail. The FD technique used here will be referred to as FDWD, because all unknowns inside the whole domain are solved simultaneously. In the next chapter, the global domain is decomposed into many smaller sub-domains and then solved with the domain decomposition technique.

3.1 Two-dimensional magnetotelluric forward problem

Electric ($\mathbf{E}$) and magnetic ($\mathbf{H}$) fields that constitute an electromagnetic wave are mutually orthogonal. A changing electric field induces the magnetic field which is perpendicular to the electric field. Similarly, a changing magnetic field induces an electric field in the perpendicular direction. Given that the x-direction is the strike direction, Faraday's law (2.13) and Ampere's law (2.14) can be decoupled into two modes: transverse electric fields (TE mode) and transverse magnetic fields (TM mode). The TE mode describes the currents parallel to the strike direction, while the TM mode describes currents perpendicular to the strike. The components of the electric

and magnetic fields of each mode are as follows,

$$(E_x, H_y, H_z) \text{ for TE mode,}$$
$$(H_x, E_y, E_z) \text{ for TM mode.}$$

To calculate the fields for TE mode, it is better to solve for electric field $\mathbf{E}$ from the second order equation,

$$\nabla \times \nabla \times \mathbf{E} = i\omega\mu\sigma\mathbf{E}. \tag{3.1}$$

Equation (3.1) is derived from substituting $\mathbf{H}$ in (2.13) into (2.14). Once E_x is obtained, H_y and H_z are calculated from (2.13). Similarly, to solve for the magnetic fields $\mathbf{H}$ for the TM mode, a second order equation of $\mathbf{H}$,

$$\nabla \times \rho\nabla \times \mathbf{H} = i\omega\mu\mathbf{H}, \tag{3.2}$$

is used. This is derived from substituting $\mathbf{E}$ in (2.14) into (2.13). Once H_x is solved, E_y and E_z can be obtained from (2.14).

Two curl equations of electric field and magnetic field share the same pattern. In order to avoid redundancy, the equations for TE and TM modes will be generally written as

$$\nabla \times \alpha\nabla \times \mathbf{G} = i\omega\mu\beta\mathbf{G}, \tag{3.3}$$

where $\mathbf{G}$, α and β are defined as shown in the following table

Mode	$\mathbf{G}$	α	β
TE	$\mathbf{E}$	1	σ
TM	$\mathbf{H}$	ρ	1

In Cartesian coordinates, $\mathbf{G} = \mathbf{G}(x, y, z)$ and (3.3) can be expanded as

$$\begin{aligned}
\frac{\partial}{\partial y}\left(\alpha\left(\frac{\partial G_y}{\partial x} - \frac{\partial G_x}{\partial y}\right)\right) - \frac{\partial}{\partial z}\left(\alpha\left(\frac{\partial G_x}{\partial z} - \frac{\partial G_z}{\partial x}\right)\right) &= i\omega\mu\beta G_x, \\
\frac{\partial}{\partial z}\left(\alpha\left(\frac{\partial G_z}{\partial y} - \frac{\partial G_y}{\partial z}\right)\right) - \frac{\partial}{\partial x}\left(\alpha\left(\frac{\partial G_y}{\partial x} - \frac{\partial G_x}{\partial y}\right)\right) &= i\omega\mu\beta G_y, \\
\frac{\partial}{\partial x}\left(\alpha\left(\frac{\partial G_x}{\partial z} - \frac{\partial G_z}{\partial x}\right)\right) - \frac{\partial}{\partial y}\left(\alpha\left(\frac{\partial G_z}{\partial y} - \frac{\partial G_y}{\partial z}\right)\right) &= i\omega\mu\beta G_z.
\end{aligned} \tag{3.4}$$

For the two-dimensional problem, given that the x-direction is the strike direction, the resistivity model is varied in the y- and z-directions, and only G_x exists. Equation (3.4) reduces to

$$\frac{\partial}{\partial y}\left(\alpha\frac{\partial G_x}{\partial y}\right) + \frac{\partial}{\partial z}\left(\alpha\frac{\partial G_x}{\partial z}\right) + i\omega\mu\beta G_x = 0. \tag{3.5}$$

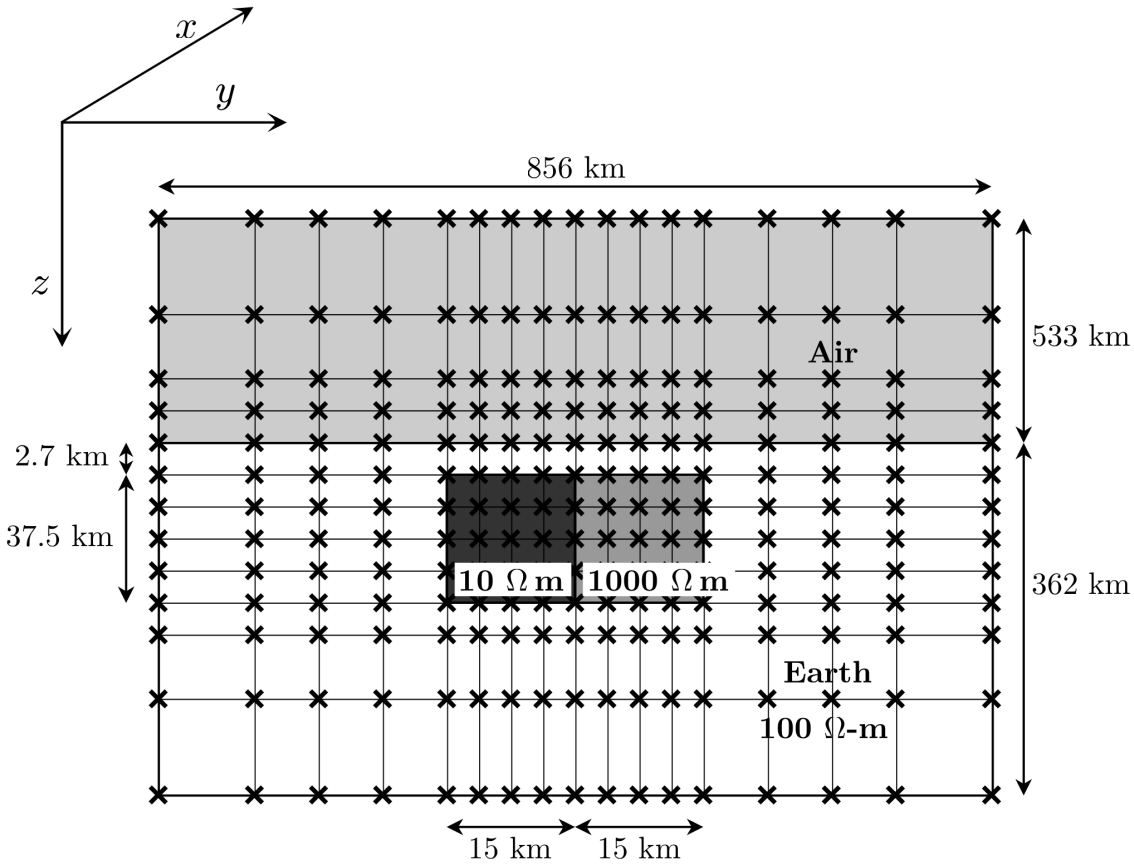


Figure 3.1: Conductive and resistive blocks with vertical contact buried in 100 Ω m half-space. The left and right blocks have resistivity of 10 Ω m and 1000 Ω m, respectively. The model is discretized using 12×16 rectangles. Both electric and magnetic fields are defined at the nodes (junctions denoted by cross symbols). The boundary fields are located at the edges of model.

Before applying the finite difference formulation, the model is first discretized into a rectangular grid. Figure 3.1 illustrates an example of the resistivity model that is non-uniformly discretized into 12×16 rectangles in the z- and y-directions, respectively. Both electric and magnetic fields are defined at the nodes. The indexing discrete field G_x, resistivity ρ, and grid spacings, Δy and Δz, are illustrated in Figure 3.2. The node (i, j) is defined at the top left corner of each rectangle. Resistivity $\rho_{i,j}$ represents the resistivity of the rectangle bounded by nodes (i, j), $(i, j+1)$, $(i+1, j+1)$,

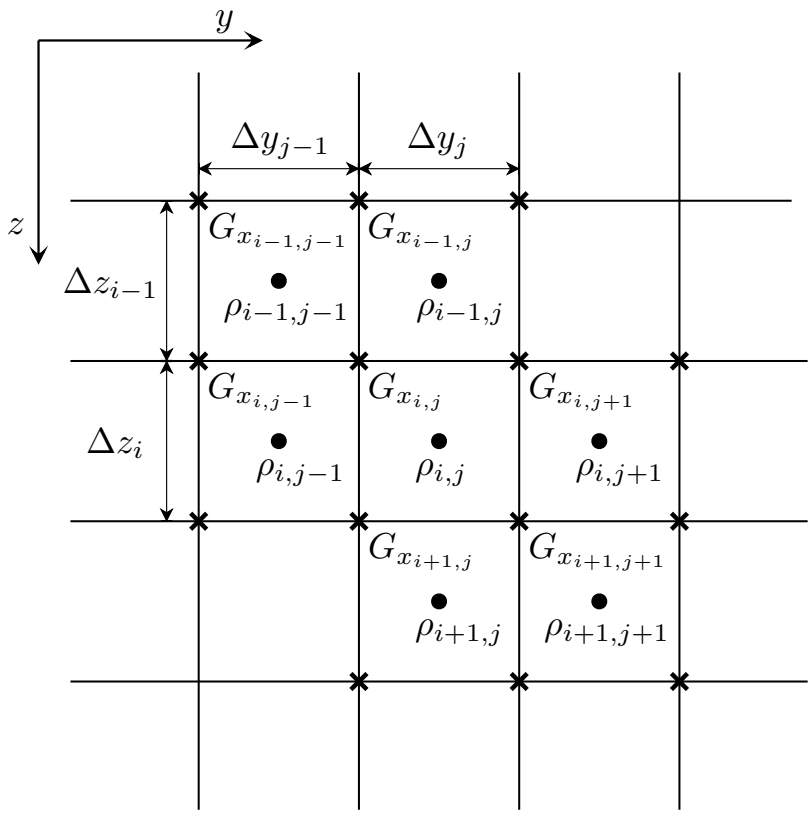


Figure 3.2: Resistivity $\rho_{i,j}$ is defined at the center of each rectangle. Fields $G_{x_{i,j}}$ are defined at the nodes. Mesh sizes in vertical and horizontal directions are Δz_i and Δy_j, respectively.

and $(i+1, j)$. The discrete form of equation (3.5) related to G_x at node (i, j) is

$$\begin{aligned}
&\frac{\frac{\bar{\alpha}_R}{\Delta y_j}(G_{x,i,j+1} - G_{x,i,j}) - \frac{\bar{\alpha}_L}{\Delta y_{j-1}}(G_{x,i,j} - G_{x,i,j-1})}{(\Delta y_{j-1} + \Delta y_j)/2} \\
&\quad + \frac{\frac{\bar{\alpha}_D}{\Delta z_i}(G_{x,i+1,j} - G_{x,i,j}) - \frac{\bar{\alpha}_U}{\Delta z_{i-1}}(G_{x,i,j} - G_{x,i-1,j})}{(\Delta z_{i-1} + \Delta z_i)/2} \\
&\quad + i\omega\mu\bar{\beta}_{avg}G_{x,i,j} = 0,
\end{aligned} \tag{3.6}$$

where $\bar{\beta}_{avg}$, $\bar{\alpha}_U$, $\bar{\alpha_D}$, $\bar{\alpha_L}$, and $\bar{\alpha_R}$ are the average electrical properties defined by

$$\bar{\beta}_{avg} = \frac{\beta_{i-1,j-1}\Delta z_{i-1}\Delta y_{j-1} + \beta_{i-1,j}\Delta z_{i-1}\Delta y_j + \beta_{i,j}\Delta z_i\Delta y_j + \beta_{i,j-1}\Delta z_i\Delta y_{j-1}}{\Delta z_{i-1}\Delta y_{j-1} + \Delta z_{i-1}\Delta y_j + \Delta z_i\Delta y_j + \Delta z_i\Delta y_{j-1}},$$

$$\bar{\alpha}_U = \frac{\alpha_{i-1,j-1}\Delta y_{j-1} + \alpha_{i-1,j}\Delta y_j}{\Delta y_{j-1} + \Delta y_j},$$

$$\bar{\alpha}_D = \frac{\alpha_{i,j-1}\Delta y_{j-1} + \alpha_{i,j}\Delta y_j}{\Delta y_{j-1} + \Delta y_j},$$

$$\bar{\alpha}_L = \frac{\alpha_{i-1,j-1}\Delta z_{i-1} + \alpha_{i,j-1}\Delta z_i}{\Delta z_{i-1} + \Delta z_i},$$

$$\bar{\alpha}_R = \frac{\alpha_{i-1,j}\Delta z_{i-1} + \alpha_{i,j}\Delta z_i}{\Delta z_{i-1} + \Delta z_i}.$$

Rewriting equation (3.6),

$$\begin{aligned}\frac{2\bar{\alpha}_R}{ty_j\Delta y_j}G_{x,i,j+1} + \frac{2\bar{\alpha}_L}{ty_j\Delta y_{j-1}}G_{x,i,j-1} + \frac{2\bar{\alpha}_D}{tz_i\Delta z_i}G_{x,i+1,j} + \frac{2\bar{\alpha}_U}{tz_i\Delta z_{i-1}}G_{x,i-1,j} \\ + \left(i\omega\mu\bar{\beta}_{avg} - \frac{2\bar{\alpha}_R}{ty_j\Delta y_j} - \frac{2\bar{\alpha}_L}{ty_j\Delta y_{j-1}} - \frac{2\bar{\alpha}_D}{tz_i\Delta z_i} - \frac{2\bar{\alpha}_U}{tz_i\Delta z_{i-1}}\right)G_{x,i,j} = 0\end{aligned} \tag{3.7}$$

with tz_i and ty_j defined as,

$$tz_i = \Delta z_{i-1} + \Delta z_i,$$

$$ty_j = \Delta y_{j-1} + \Delta y_j.$$

In order to obtain a symmetric system, we must multiply equation (3.7) by tz_ity_j which gives

$$\begin{aligned}\frac{2\bar{\alpha}_R tz_i}{\Delta y_j}G_{x,i,j+1} + \frac{2\bar{\alpha}_L tz_i}{\Delta y_{j-1}}G_{x,i,j-1} + \frac{2\bar{\alpha}_D ty_j}{\Delta z_i}G_{x,i+1,j} + \frac{2\bar{\alpha}_U ty_j}{\Delta z_{i-1}}G_{x,i-1,j} \\ + \left((tz_ity_j)i\omega\mu\bar{\beta}_{avg} - \frac{2\bar{\alpha}_R tz_i}{\Delta y_j} - \frac{2\bar{\alpha}_L tz_i}{\Delta y_{j-1}} - \frac{2\bar{\alpha}_D ty_j}{\Delta z_i} - \frac{2\bar{\alpha}_U ty_j}{\Delta z_{i-1}}\right)G_{x,i,j} = 0,\end{aligned} \tag{3.8}$$

and then rewrite equation (3.8) as

$$C^R_{i,j}G_{x,i,j+1} + C^L_{i,j}G_{x,i,j-1} + C^D_{i,j}G_{x,i+1,j} + C^U_{i,j}G_{x,i-1,j} + C^C_{i,j}G_{x,i,j} = 0, \tag{3.9}$$

where

$$C^R_{i,j} = \frac{2\bar{\alpha}_R tz_i}{\Delta y_j},$$

$$C^L_{i,j} = \frac{2\bar{\alpha}_L tz_i}{\Delta y_{j-1}},$$

$$C^D_{i,j} = \frac{2\bar{\alpha}_D ty_j}{\Delta z_i},$$

$$C^U_{i,j} = \frac{2\bar{\alpha}_U ty_j}{\Delta z_{i-1}},$$

$$C^C_{i,j} = \left((tz_ity_j)i\omega\mu\bar{\beta}_{avg} - \frac{2\bar{\alpha}_R tz_i}{\Delta y_j} - \frac{2\bar{\alpha}_L tz_i}{\Delta y_{j-1}} - \frac{2\bar{\alpha}_D ty_j}{\Delta z_i} - \frac{2\bar{\alpha}_U ty_j}{\Delta z_{i-1}}\right).$$

The parameters $C^R_{i,j}$, $C^L_{i,j}$, $C^D_{i,j}$, and $C^U_{i,j}$ are the coupling coefficients from nodes $(i, j+1)$, $(i, j-1)$, $(i+1, j)$ and $(i-1, j)$ to node (i, j), respectively, while $C^C_{i,j}$ is the self-coupling coefficient.

Applying equation (3.9) to all interior nodes, we obtain a system of equations,

$$\mathbf{Ax} = \mathbf{b}, \tag{3.10}$$

where $\mathbf{A}$ is a sparse five-banded coefficient matrix, $\mathbf{b}$ is the right-hand side vector associated with the boundary nodes, and $\mathbf{x}$ is the unknown vector which represents either electric or magnetic fields depending on the calculation mode. In the MT problem, $\mathbf{A}$ is symmetric but not hermitian (complex only diagonal elements).

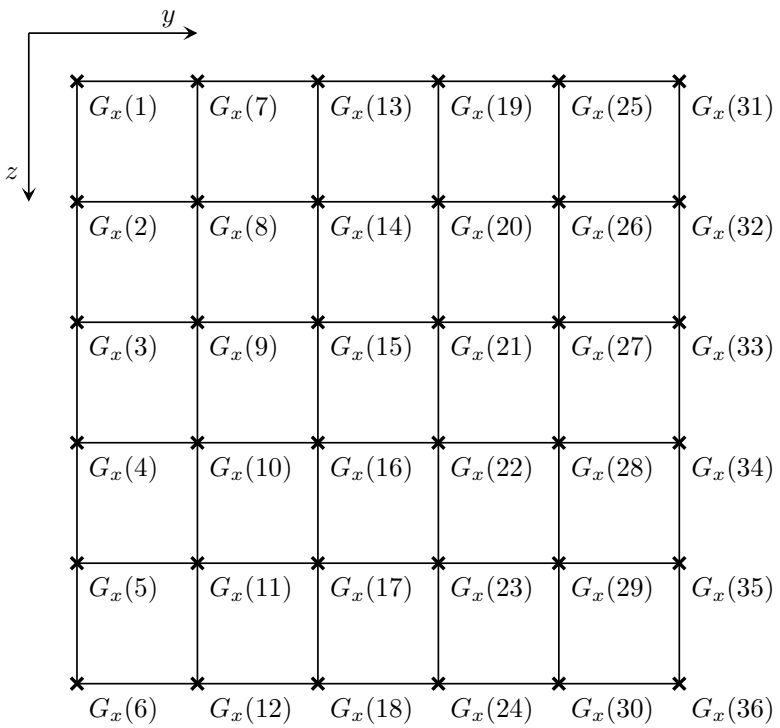


Figure 3.3: An example of a computational model consisting of a 5×5 rectangles, where the non-uniform discretization is omitted. The nodes are defined at the corners. Direction of ordering indices is down first followed by right.

In general, if the model is discretized into $M_z \times M_y$ rectangles in z- and y-directions, respectively, there are $(M_z+1)(M_y+1)$ nodes, in which $(M_z-1)(M_y-1)$ nodes of those are the unknowns. The dimension of $\mathbf{A}$ is therefore $(M_z-1)(M_y-1) \times (M_z-1)(M_y-1)$. For a 5×5 grid, for instance, there are 36 nodes in total, in which 16 nodes are unknowns and 20 nodes are at the boundary, as shown in Figure 3.3. The boundary nodes can be calculated from one-dimensional problems, which will be described in Section 3.2. The resulting system of equations, $\mathbf{Ax} = \mathbf{b}$, can be expressed as equation (3.11) and the sparsity pattern of the coefficient matrix $\mathbf{A}$ is shown in Figure

3.4.

After obtaining the electric or magnetic fields from the second order equations, (3.1) and (3.2), the corresponding fields are solved from the first order equations, Faraday's law (2.13) and Ampere's law (2.14). At the surface, apparent resistivity and phases are then computed from the ratios of electric fields to magnetic fields, which will be described in Section 3.3.

$$
\begin{pmatrix}
C_8^C & C_8^D & & & C_8^R & & & & & & & & & & & \\
C_9^U & C_9^C & C_9^D & & & C_9^R & & & & & & & & & & \\
 & C_{10}^U & C_{10}^C & C_{10}^D & & & C_{10}^R & & & & & & & & & \\
 & & C_{11}^U & C_{11}^C & & & & C_{11}^R & & & & & & & & \\
C_{14}^L & & & & C_{14}^C & C_{14}^D & & & C_{14}^R & & & & & & & \\
 & C_{15}^L & & & C_{15}^U & C_{15}^C & C_{15}^D & & & C_{15}^R & & & & & & \\
 & & C_{16}^L & & & C_{16}^U & C_{16}^C & C_{16}^D & & & C_{16}^R & & & & & \\
 & & & C_{17}^L & & & C_{17}^U & C_{17}^C & & & & C_{17}^R & & & & \\
 & & & & C_{20}^L & & & & C_{20}^C & C_{20}^D & & & C_{20}^R & & & \\
 & & & & & C_{21}^L & & & C_{21}^U & C_{21}^C & C_{21}^D & & & C_{21}^R & & \\
 & & & & & & C_{22}^L & & & C_{22}^U & C_{22}^C & C_{22}^D & & & C_{22}^R & \\
 & & & & & & & C_{23}^L & & & C_{23}^U & C_{23}^C & & & & C_{23}^R \\
 & & & & & & & & C_{26}^L & & & & C_{26}^C & C_{26}^D & & \\
 & & & & & & & & & C_{27}^L & & & C_{27}^U & C_{27}^C & C_{27}^D & \\
 & & & & & & & & & & C_{28}^L & & & C_{28}^U & C_{28}^C & C_{28}^D \\
 & & & & & & & & & & & C_{29}^L & & & C_{29}^U & C_{29}^C
\end{pmatrix}
\begin{pmatrix}
G_x(8) \\ G_x(9) \\ G_x(10) \\ G_x(11) \\ G_x(14) \\ G_x(15) \\ G_x(16) \\ G_x(17) \\ G_x(20) \\ G_x(21) \\ G_x(22) \\ G_x(23) \\ G_x(26) \\ G_x(27) \\ G_x(28) \\ G_x(29)
\end{pmatrix}
=
\begin{pmatrix}
C_8^L G_x(2) + C_8^U G_x(7) \\
C_9^L G_x(3) \\
C_{10}^L G_x(4) \\
C_{11}^L G_x(5) + C_{11}^D G_x(12) \\
C_{14}^U G_x(13) \\
0 \\
0 \\
C_{17}^D G_x 18 \\
C_{20}^U G_x(19) \\
0 \\
0 \\
C_{23}^D G_x 24 \\
C_{26}^R G_x(32) + C_{26}^U G_x(25) \\
C_{27}^R G_x(33) \\
C_{28}^R G_x(34) \\
C_{29}^R G_x(35) + C_{29}^D G_x(30)
\end{pmatrix}
\tag{3.11}
$$

To solve the system of equations $\mathbf{Ax} = \mathbf{b}$, two types of method, namely, direct methods and iterative methods, are normally used. Advantages and disadvantages were discussed in many earlier publications (e.g. Streich, 2009; Operto *et al.*, 2007). Here we briefly summarize them.

A clear advantage of a direct method, e.g., Gaussian elimination, LU-factorization, is at the accuracy of the solutions, because the solution is not approximated. It therefore provides an accurate solution. However, this comes at a large computational cost. Direct solvers usually require large memory to store the coefficients and also a large amount of CPU time to solve them. This is particularly true for large two-dimensional or three-dimensional problems.

An iterative method minimizes the norm of $\mathbf{Ax} - \mathbf{b}$ through many different kinds of iterative schemes. It therefore requires a small amount of memory. Because it

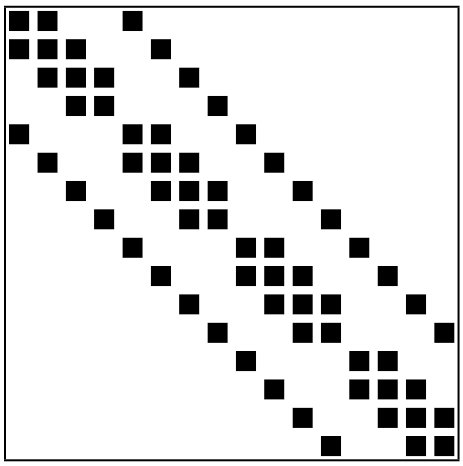


Figure 3.4: The sparsity pattern of coefficient matrix **A** resembles the pattern of the matrix in equation (3.11), which is a sparse five-banded matrix.

successively converges to the solution, it can occasionally break down or become idle, i.e., no convergence, particularly when the system is ill-conditioned. This is caused by models which are too large or too complicated. In MT, long period problems can result in ill-conditioned systems as well.

In reality, the Earth is geologically complex. Consequently, the system of equations becomes ill-conditioned. Thus, to obtain a robust forward solver, LU-factorization is chosen as a solver in this thesis and will be referred to as FDWD. A typical LU-factorization to solve the system consists of two steps:

1. LU-factorization, that decomposes $\mathbf{A} = \mathbf{LU}$, where $\mathbf{L}$ and $\mathbf{U}$ are lower and upper matrices, respectively.

2. Forward and backward substitution using the lower and upper matrices previously decomposed in Step 1

However, a direct solver, like LU-factorization, cannot be used for very large models, particularly in three-dimensional cases, because of the large memory requirement. If the computational domain is broken into many smaller sub-domains, the direct solver becomes reasonable. In this thesis, a domain decomposition technique is therefore studied with the purpose of solving MT forward problems with the direct solver.

3.2 One-dimensional magnetotelluric forward problem

As shown in Figure 3.1, a one-dimensional problem is used as a boundary for a two-dimensional problem. In one-dimensional problems, the fields G_x are defined as a function of depth (z-direction).

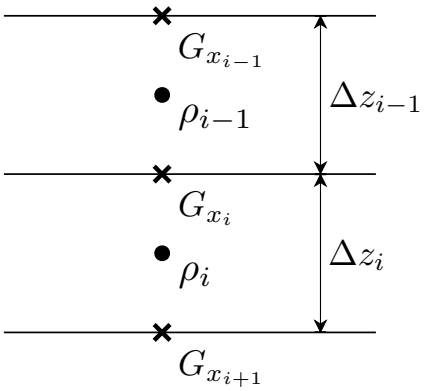


Figure 3.5: One-dimensional model at column j of the two-dimensional model shown in Figure 3.2.

As with two-dimensional problems, we start from the second order Maxwell's equations (3.3) which are expanded to equation (3.4). In one dimensional problems, the resistivity model is varied only in the z-direction. Hence equation (3.5) is reduced to

$$\frac{\partial}{\partial z}\left(\alpha\frac{\partial G_x}{\partial z}\right) + i\omega\mu\beta G_x = 0. \tag{3.12}$$

At the column node j (Figure 3.5), the discrete form of equation (3.12) is

$$\frac{\frac{\alpha_i}{\Delta z_i}(G_{x,i+1} - G_{x,i}) - \frac{\alpha_{i-1}}{\Delta z_{i-1}}(G_{x,i} - G_{x,i-1})}{(\Delta z_{i-1} + \Delta z_i)/2} + i\omega\mu\bar{\beta}_{avg}G_{x,i} = 0, \tag{3.13}$$

where

$$\bar{\beta}_{avg} = \frac{\beta_{i-1}\Delta z_{i-1} + \beta_i \Delta z_i}{\Delta z_{i-1} + \Delta z_i}$$

We rewrite (3.13) as

$$\frac{2\alpha_i}{tz_i\Delta z_i}G_{x,i+1} + \frac{2\alpha_{i-1}}{tz_i\Delta z_{i-1}}G_{x,i-1} + \left(i\omega\mu\bar{\beta}_{avg} - \frac{2\alpha_i}{tz_i\Delta z_i} - \frac{2\alpha_{i-1}}{tz_i\Delta z_{i-1}}\right)G_{x,i} = 0, \tag{3.14}$$

where

$$tz_i = \Delta z_{i-1} + \Delta z_i.$$

In order to yield a symmetric system, we must multiply (3.14) by tz_i,

$$\frac{2\alpha_i}{\Delta z_i}G_{x,i+1} + \frac{2\alpha_{i-1}}{\Delta z_{i-1}}G_{x,i-1} + \left(tz_i i\omega\mu\bar{\beta}_{avg} - \frac{2\alpha_i}{\Delta z_i} - \frac{2\alpha_{i-1}}{\Delta z_{i-1}}\right)G_{x,i} = 0, \tag{3.15}$$

and rewrite equation (3.15) as

$$C_i^D G_{x,i+1} + C_i^U G_{x,i-1} + C_i^C G_{x,i} = 0, \tag{3.16}$$

where

$$C_i^D = \frac{2\alpha_i}{\Delta z_i},$$
$$C_i^U = \frac{2\alpha_{i-1}}{\Delta z_{i-1}},$$
$$C_i^C = \left(tz_i i\omega\mu\bar{\beta}_{avg} - \frac{2\alpha_i}{\Delta z_i} - \frac{2\alpha_{i-1}}{\Delta z_{i-1}} \right).$$

The parameters C_i^D, and C_i^U are coupling coefficients from nodes $i+1$ and $i-1$ to node i, respectively, while C_i^C is self-coupling coefficient.

Applying (3.16) to the one-dimensional model, we obtain a system of equations,

$$\mathbf{A}_1\mathbf{x}_1 = \mathbf{b}_1, \tag{3.17}$$

where the subscript 1 denotes one-dimensional system. For example, the first column in Figure 3.3, node $G_x(1)$ to $G_x(6)$ is solved by applying (3.16). The field at top node $G_x(1)$ and last node $G_x(6)$, are set to one and zero, respectively. The resulting system is

$$\begin{pmatrix} C_2^C & C_2^D & 0 & 0 \\ C_3^U & C_3^C & C_3^D & 0 \\ 0 & C_4^U & C_4^C & C_4^D \\ 0 & 0 & C_5^U & C_5^C \end{pmatrix} \begin{pmatrix} G_x(2) \\ G_x(3) \\ G_x(4) \\ G_x(5) \end{pmatrix} = \begin{pmatrix} C_2^D \\ 0 \\ 0 \\ 0 \end{pmatrix}. \tag{3.18}$$

Its sparsity pattern is shown in Figure 3.6.

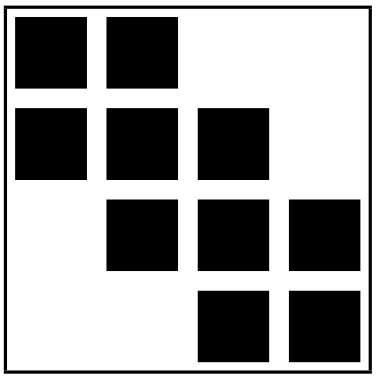


Figure 3.6: The sparsity pattern of matrix $\mathbf{A}_1$ in equation (3.18) is a sparse tridiagonal matrix.

3.3 MT Response calculation

The MT responses which are apparent resistivity and phase can be obtained from the ratios of electric fields to magnetic fields at the surface. As stated in Section 3.1, when x is the strike direction, the components of the fields in TE and TM modes are (E_x, H_y, H_z) and (H_x, E_y, E_z), respectively. After the field in the strike direction is

solved from the second order equation, the corresponding fields are solved from Faraday's law (2.13) for TE mode and Ampere's law (2.14) for TM mode. Solving for these corresponding fields is described in the following.

The first order Maxwell's equations, equations (2.13) and (2.14), both share the same pattern, which will be represented by

$$\nabla \times \mathbf{G} = \alpha \mathbf{F}, \tag{3.19}$$

where $\mathbf{F}$, $\mathbf{G}$ and α are defined as stated in the following table,

Mode	$\mathbf{G}$	$\mathbf{F}$	α
TE	$\mathbf{E}$	$\mathbf{H}$	$i\omega\mu$
TM	$\mathbf{H}$	$\mathbf{E}$	$1/\rho$

On the cartesian coordinate, $\mathbf{F} = \mathbf{F}(x, y, z)$ and $\mathbf{G} = \mathbf{G}(x, y, z)$. Equation (3.19) can then be expanded as

$$\begin{aligned}
\left(\frac{\partial G_z}{\partial y} - \frac{\partial G_y}{\partial z}\right) &= \alpha F_x, \\
\left(\frac{\partial G_x}{\partial z} - \frac{\partial G_z}{\partial x}\right) &= \alpha F_y, \\
\left(\frac{\partial G_y}{\partial x} - \frac{\partial G_x}{\partial y}\right) &= \alpha F_z.
\end{aligned} \tag{3.20}$$

In a two-dimensional problem with the x-direction as the strike direction, only G_x exists. Equations (3.20) then reduce to

$$\frac{\partial G_x}{\partial z} = \alpha F_y, \tag{3.21}$$

$$\frac{\partial G_x}{\partial y} = -\alpha F_z. \tag{3.22}$$

On the discretized model, as shown in Figure 3.7, the field components $F_{y,i,j}$ and $F_{z,i,j}$ are defined at the edges of rectangle with resistivity $\rho_{i,j}$. The discrete forms of equations (3.21) and (3.22) are

$$\frac{G_{x,i+1,j} - G_{x,i,j}}{\Delta z_i} = \bar{\alpha}_z F_{y,i,j}, \tag{3.23}$$

and

$$\frac{G_{x,i,j+1} - G_{x,i,j}}{\Delta y_j} = -\bar{\alpha}_y F_{z,i,j}, \tag{3.24}$$

respectively, where

$$\begin{aligned}
\bar{\alpha}_z &= \frac{\alpha_{i,j-1}\Delta y_{j-1} + \alpha_{i,j}\Delta y_j}{\Delta y_{j-1} + \Delta y_j}, \\
\bar{\alpha}_y &= \frac{\alpha_{i-1,j}\Delta z_{i-1} + \alpha_{i,j}\Delta z_i}{\Delta z_{i-1} + \Delta z_i}.
\end{aligned} \tag{3.25}$$

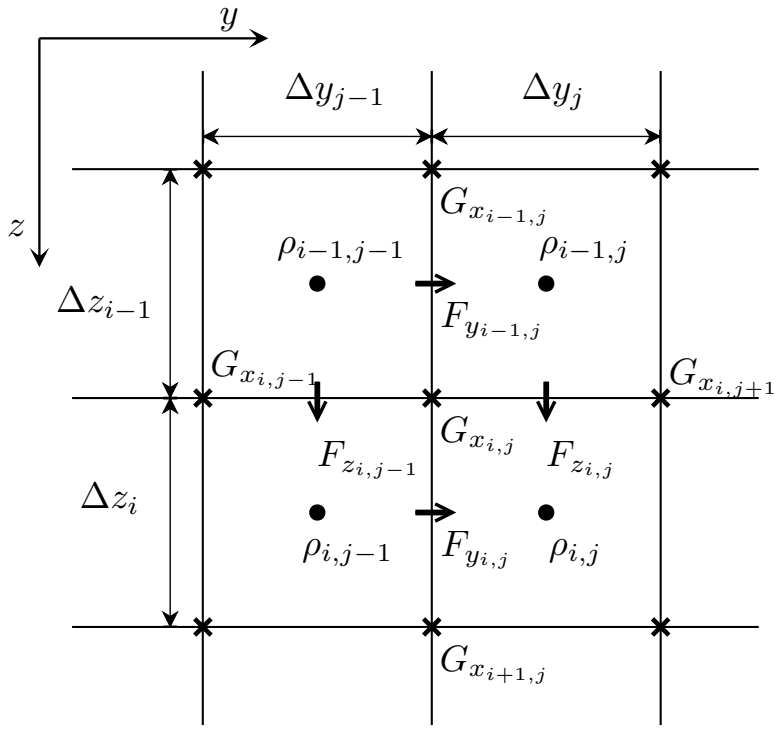


Figure 3.7: $F_{y,i,j}$ and $F_{z,i,j}$ are located at the edge of the rectangle with resistivity $\rho_{i,j}$.

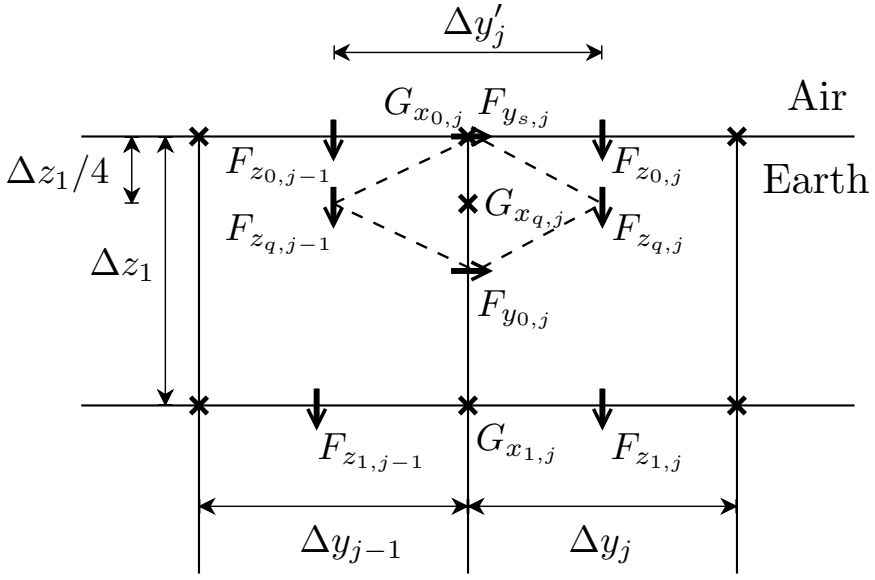


Figure 3.8: At the interface between air and earth, $F_{ys,j}$ is calculated from $G_{xq,j}$, $F_{z,q,j}$, $F_{z,q,j-1}$ and $F_{y,0,j}$, where $G_{xq,j}$, $F_{z,q,j}$, $F_{z,q,j-1}$ are the interpolated fields.

In MT, only the responses at surface are required. The field **F** at the surface, F_{ys}, can be interpolated from

$$\nabla \times \mathbf{F} = \beta \mathbf{G}, \tag{3.26}$$

where β is $1/\rho$ and $i\omega\mu$ for TE and TM mode responses, respectively. Equation (3.26) is expanded as

$$\begin{aligned}
\left(\frac{\partial F_z}{\partial y} - \frac{\partial F_y}{\partial z}\right) &= \beta G_x, \\
\left(\frac{\partial F_x}{\partial z} - \frac{\partial F_z}{\partial x}\right) &= \beta G_y, \\
\left(\frac{\partial F_y}{\partial x} - \frac{\partial F_x}{\partial y}\right) &= \beta G_z.
\end{aligned} \tag{3.27}$$

From equations (3.23) and (3.24), we obtain G_x, F_y, and F_z. Equations (3.27) then reduce to

$$\frac{\partial F_z}{\partial y} - \frac{\partial F_y}{\partial z} = \beta G_x. \tag{3.28}$$

For the first layer of the Earth (Figure 3.8), the discrete form of equation (3.28) is

$$\frac{F_{z,q,j} - F_{z,q,j-1}}{\Delta y'_j} - \frac{F_{y,0,j} - F_{ys,j}}{\Delta z_1/2} = \bar{\beta} G_{xq,j}, \tag{3.29}$$

where $G_{xq,j}$ and $F_{zq,j}$ are the linearly interpolated field at $\frac{1}{4}\Delta z_1$ below the surface, and are defined by

$$G_{xq,j} = G_{x,0,j} + \frac{G_{x,1,j} - G_{x,0,j}}{4},$$

$$F_{zq,j} = F_{z,0,j} + \frac{F_{z,1,j} - F_{z,0,j}}{4},$$

and

$$\bar{\beta} = \frac{\alpha_{i,j-1}\Delta y_{j-1} + \alpha_{i,j}\Delta y_j}{\Delta y_{j-1} + \Delta y_j},$$

$$\Delta y'_j = \frac{\Delta y_j + \Delta y_{j-1}}{2}.$$

Rewriting equation (3.29), $F_{ys,j}$ is obtained from

$$F_{ys,j} = F_{y,0,j} + \frac{\Delta z_1}{2}(\bar{\beta} G_{xq,j} - \frac{F_{z,q,j} - F_{z,q,j-1}}{\Delta y'_j}). \tag{3.30}$$

After the electric and magnetic fields are obtained at the surface, the impedance **Z** is then calculated from the ratio of the electric to magnetic field. From equation (2.3), the impedances for the TE and TM modes are

$$Z_{xy} = \frac{E_x}{H_y}, \quad Z_{yx} = \frac{E_y}{H_x}, \tag{3.31}$$

respectively. For the TE mode, the apparent resistivity $\rho_{a,xy}$ and phase ϕ_{xy} are computed from (2.4) and (2.5) using

$$\rho_{a,xy} = \frac{1}{\omega\mu}|Z_{xy}|^2, \quad \phi_{xy} = \tan^{-1}\left(\frac{\mathrm{Im}\{Z_{xy}\}}{\mathrm{Re}\{Z_{xy}\}}\right). \tag{3.32}$$

Similarly, the TM-mode apparent resistivity $\rho_{a,yx}$ and phase ϕ_{yx} are given by

$$\rho_{a,yx} = \frac{1}{\omega\mu}|Z_{yx}|^2, \quad \phi_{yx} = \tan^{-1}\left(\frac{\mathrm{Im}\{Z_{yx}\}}{\mathrm{Re}\{Z_{yx}\}}\right). \tag{3.33}$$

As examples of MT responses, the apparent resistivity and phases from a buried conductive block model, a buried resistive block model, and a buried conductive-resistive block (two-block) model are shown in Figures 3.9, 3.10, and 3.11, respectively. The responses change as a function of period, where shorter periods reflect shallow structure and the longer periods indicate deeper structure.

For periods of 0.01 and 0.1 seconds, both TE and TM responses sense only the shallow structure because the apparent resistivity is close to 100 Ω m, which is equal to that of the overlying layer over the anomaly. However, at longer periods, the TE and TM modes can sense the anomalies. For example, from the conductive block model (Figure 3.9a), the TE mode shows a lower apparent resistivity about 20 Ω m at 100 seconds, before increasing to 100 Ω m. This indicates that the investigation depth of longer periods is beyond the depth of the conductor.

As with the conductive block model, the TE mode yields the higher apparent resistivity as periods increase for the resistive block model, and then falls back (Figure 3.10). Note that TM responses are more sensitive to the resistor than TE, because TM responses abruptly change at locations near the anomaly. TE responses gradually change along the profile length.

For the two-block model (Figure 3.11a), the curves jump between the conductive and resistive blocks, because of the contrast in electrical resistivity. The jump from the TE mode are less steep than that from the TM mode. However, as seen from the two block model, the TM mode is more sensitive to the resistor than the TE mode.

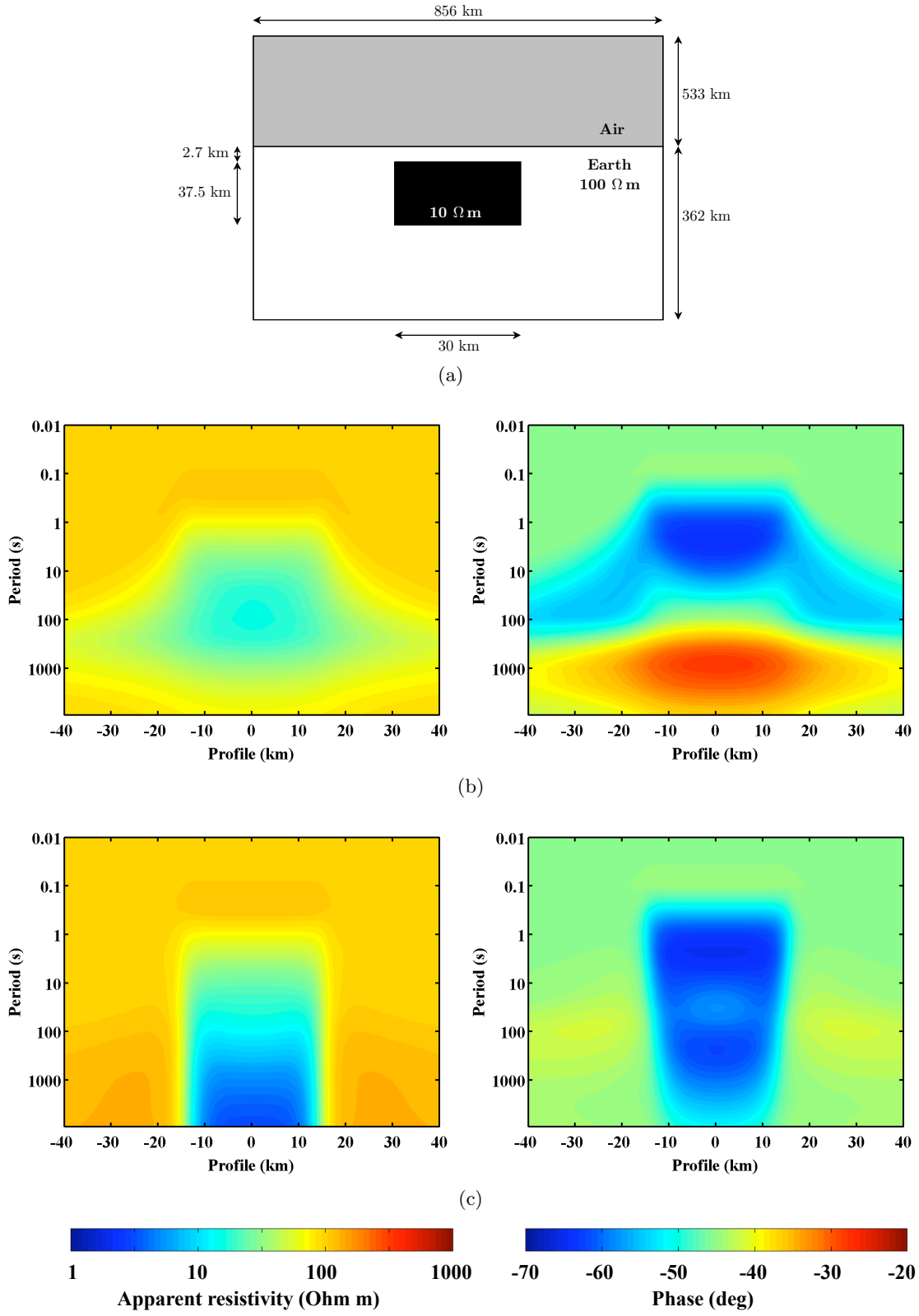


Figure 3.9: (a) A conductive block of 10 Ω m is buried in a 100 Ω m background. The responses, apparent resistivity (left panel) and phase (right panel) from the conductive block model for (b) TE mode and (c) TM mode plotted as a function of period.

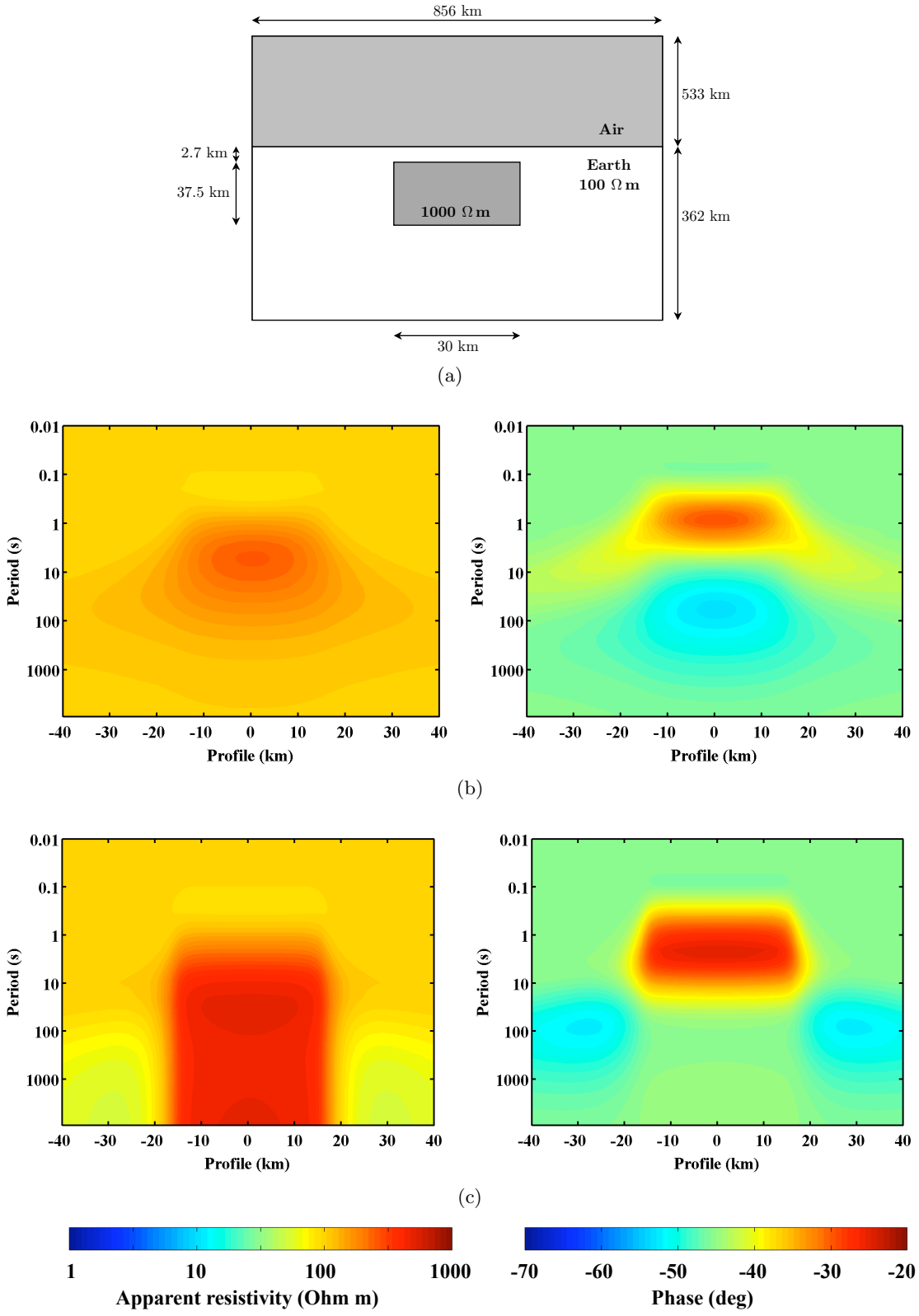


Figure 3.10: (a) A resistive block of 1000 Ω m is buried in a 100 Ω m background. The responses, apparent resistivity (left panel) and phase (right panel) from the resistive block model for (b) TE mode and (c) TM mode plotted as a function of period.

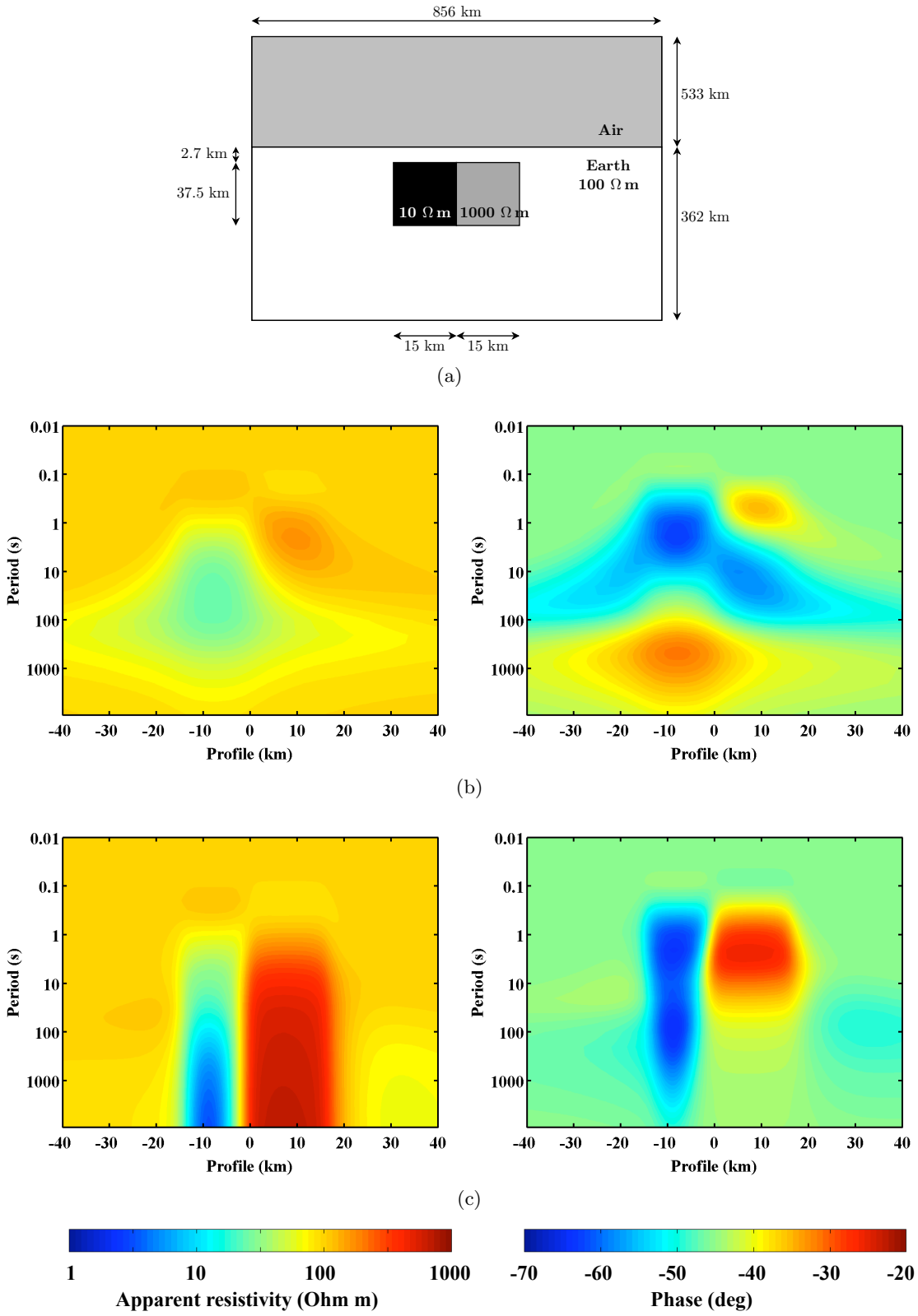


Figure 3.11: (a) A conductive block 10 Ω m and resistive block of 1000 Ω m are buried in a 100 Ω m background (or two block model). The responses, apparent resistivity (left panel) and phase (right panel) from the two block model for (b) TE mode and (c) TM mode plotted as a function of period.

CHAPTER IV

HIERARCHICAL DOMAIN DECOMPOSITION

In the previous section, all unknowns are simultaneously solved. If the models are very large, the direct solver might be impractical. Here, another technique is chosen to solve the same system of equations. The computational domain is decomposed into many smaller sub-domains. This technique is generally known as the domain decomposition technique. The domain decomposition method via the Schur complement is introduced in Section 4.2 and the hierarchical domain decomposition is then described in Section 4.3. Numerical experiments for MT problems are also given in Section 4.4.

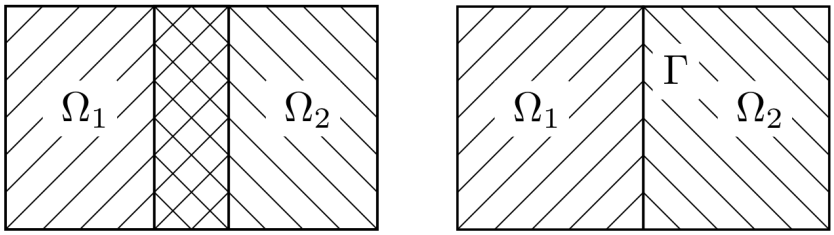
4.1 Introduction

The idea of domain decomposition simply says divide and conquer. Instead of solving the whole domain at once, the domain decomposition method breaks the computational domain into many smaller sub-domains. The system is then solved through a series of smaller sub-systems.

Domain decomposition methods are considered to be one of the most powerful tools for solving large-scale, industrial, ill-conditioned problems arising in various fields such as computational electromagnetism (e.g. Larsson, 1999; Lu *et al.*, 2008; Wang *et al.*, 2008; Lu & Shen, 1997; Yin *et al.*, 2002), engineering problems (e.g. Guo *et al.*, 2006; Basermann *et al.*, 2005; Bitzarakis *et al.*, 1997), and geophysics problems (e.g. Ben-Hadj-Ali *et al.*, 2008; Sourbier *et al.*, 2008; Pain *et al.*, 2002; Xie *et al.*, 2000; Zyserman & Santos, 2000; Zyserman *et al.*, 1999; Xiong, 1999, 1992). Domain decomposition methods are also used as preconditioning techniques (e.g. Pain *et al.*, 2002; Larsson, 1999; Henon & Saad, 2006; Saad & Sosonkina, 1999).

Three reasons for using a domain decomposition technique in many research problems are: (1) a potential for parallelization, (2) an ability to deal with complicated structures, (3) a flexibility to handle different mesh sizes or even mathematical formulations on different parts of the problem.

Domain decomposition methods generally fall into two major categories: an overlapping domain decomposition and a non-overlapping domain decomposition, as illustrated in Figure 4.1. On the overlapping technique, or Schwarz method, sub-domains



(a) (b)

Figure 4.1: A computational domain is divided into two sub-domains, Ω_1 and Ω_2, with (a) an overlapping region, and with (b) no overlapping region.

are updated via the overlapping region (e.g. Xiong, 1999). In the non-overlapping technique, sub-domains communicate among themselves through the interface (e.g. Benamou & Despres, 1997; Rice *et al.*, 2000).

In MT, Zyserman *et al.* (1999) and Zyserman & Santos (2000) applied the non-overlapping domain decomposition techniques to two- and three-dimensional forward problems, respectively. In their works, sub-problems are separately solved, and iteratively updated by enforcing the interfaces with a transmission condition. The memory requirement is significantly diminished due to no appearance of the global matrix. The technique is computationally efficient on a parallel system. However, their technique is sensitive to the model complexity, modes of calculation, and frequencies used.

Xiong (1999) demonstrated solving three-dimensional geo-electromagnetic forward problems by an overlapping technique or Schwarz method. The sub-domain solutions are successively updated from the adjacent sub-domains through the overlapping region. Its convergence depends on the size of the overlapping region. The larger the overlapping region, the faster the convergence, but this comes at a cost of solving larger sub-problems. The numerical experiments show that the overall computational times are higher than solving the whole system with the traditional iterative solvers on a serial machine.

Before applying the domain decomposition method, one must answer these four questions (Saad, 2003), which would help focusing on the scheme used.

1. Type of partitioning. Should the domain be partitioned along edges, or along vertices, or by elements (see Figure 4.2)?

2. Overlap. Should sub-domains overlap or not?

3. Computing of interfaces. Should the interface be solved all at once (direct solution approach, e.g., Schur complement) or successively updated by some constraints (iterative approach)?

4. Sub-domain solution. Should the sub-problems be solved by a direct or iterative method?

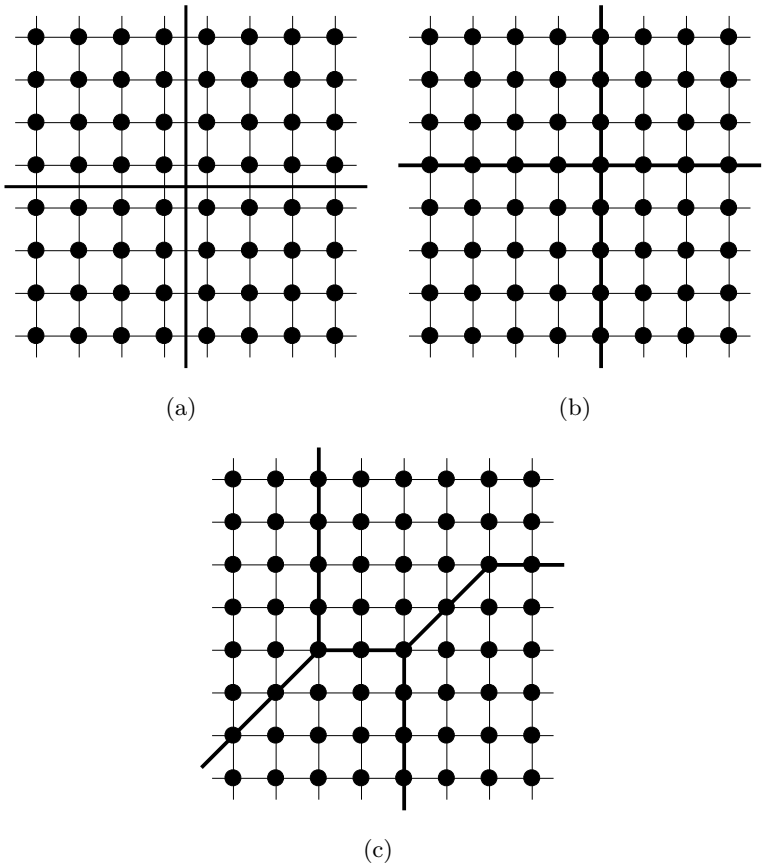


Figure 4.2: The computational domain is divided into four sub-domains (a) by vertex, (b) by element, and (c) by edge (redrawn from Saad, 2003). The thick line indicates boundaries of sub-domains.

In this thesis, element-based decomposition (Figure 4.2b) is chosen, because it uses regularly shaped sub-domains which is consistent with the shape of the MT model. In addition, the number of interfaces is approximately half of that obtained from vertex-based partitioning because sub-domains share common interfaces. When the computational domain is divided along the edges of elements there is no overlapping regions between the sub-domains.

To achieve the goal of obtaining a robust solver for MT forward problems, this thesis focuses on the non-overlapping technique through the Schur complement method. It could be considered as the direct solution approach of computing interfaces, which is in contrast to the iterative domain decomposition scheme (e.g. Zyserman & Santos, 2000; Zyserman *et al.*, 1999; Xiong, 1999; Pain *et al.*, 2002). Furthermore, the sub-problems are all solved with the direct solver, LU factorization. Therefore, the domain decomposition technique applied in this work is not sensitive to the complexity

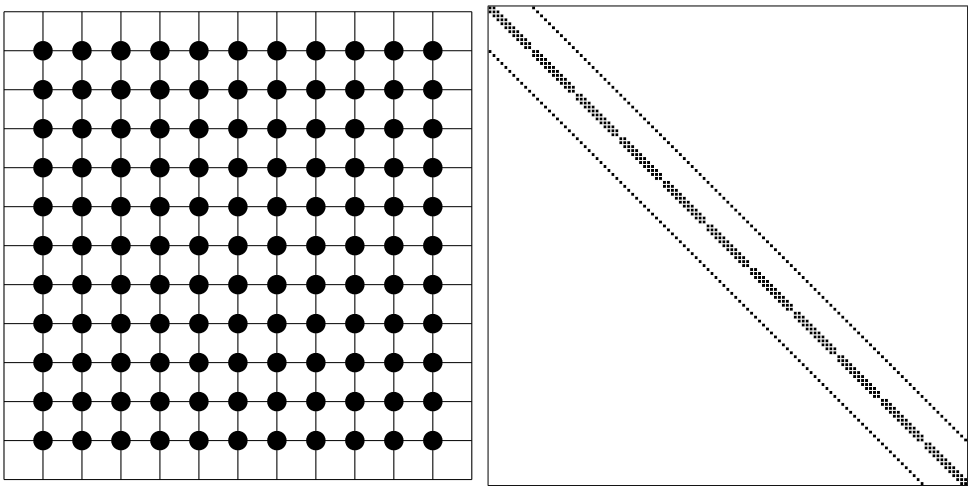
of model, modes of calculation, and periods used.

4.2 Schur complement method

Figure 4.3a shows an example of the discretized model with 11×11 interior nodes. Electric fields or magnetic fields are defined on these nodes. Hence this produces the system of equations, as described in chapter 3,

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \tag{4.1}$$

where $\mathbf{A}$ is the coefficient matrix, $\mathbf{b}$ the right-hand side vector associated with the boundary, and $\mathbf{x}$ the unknown vector which is electric or magnetic field. Matrix $\mathbf{A}$ is a sparse five-banded matrix, as shown in Figure 4.3b.



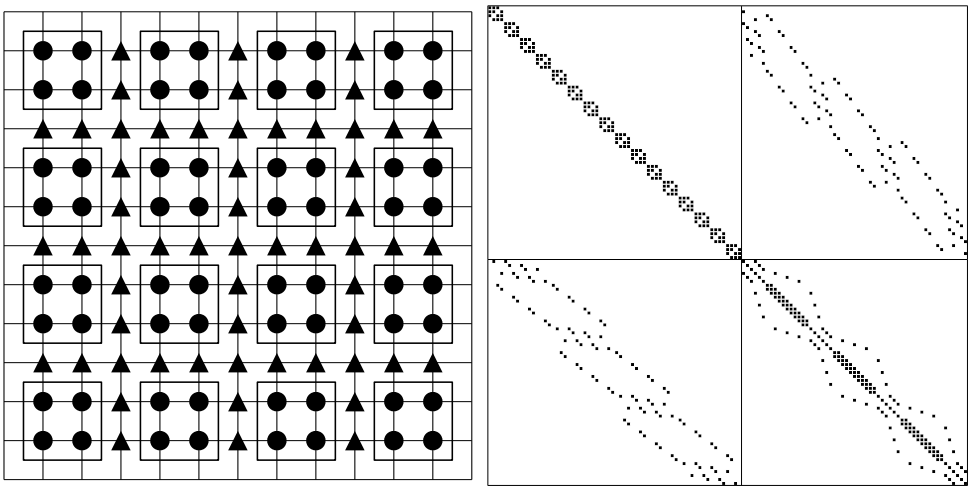
(a) (b)

Figure 4.3: (a) An example of a 12×12 mesh where the interiors are $11 \times 11 = 121$. The non-uniform discretization and boundary nodes are omitted (b) Sparsity pattern of the coefficients of matrix $\mathbf{A}$ obtained from 11×11 node model (dimension of 121×121).

After the computational domain is partitioned into $P_Z \times P_Y$ sub-domains, where P_Z and P_Y are the number of sub-domains in the z- and y-directions, respectively, the total number of sub-domains is $P = P_Z \times P_Y$. An example of a model partitioned into 4×4 sub-domains is shown in Figure 4.4a. Nodes can then be classified into two types: interiors (circles) and interfaces (triangles). The unknown vector $\mathbf{b}$ in equation (4.1) is reordered so that the interiors $\mathbf{u}_i$ of sub-domain i, where $i = 1, ..., p$, come first and the interfaces $\mathbf{v}$ are at the end. Consequently, the system of equations (4.1) can be rearranged as

$$\begin{pmatrix} \mathbf{F}_{11} & \mathbf{0} & \cdots & \mathbf{D}_1 \\ \mathbf{0} & \ddots & \mathbf{0} & \vdots \\ \vdots & \mathbf{0} & \mathbf{F}_{PP} & \mathbf{D}_P \\ \mathbf{D}_1^T & \cdots & \mathbf{D}_P^T & \mathbf{G} \end{pmatrix} \begin{pmatrix} \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_P \\ \mathbf{v} \end{pmatrix} = \begin{pmatrix} \mathbf{f}_1 \\ \vdots \\ \mathbf{f}_P \\ \mathbf{g} \end{pmatrix}, \tag{4.2}$$

where $\mathbf{F}_{ii}$ gives the interior coupling within sub-domain i, $\mathbf{D}_i$ corresponds to coupling from sub-domain i to interfaces, $\mathbf{D}_i^T$ is the reciprocal coupling, $\mathbf{G}$ is the interface coefficient matrix, and vectors $\mathbf{f}_i$ and $\mathbf{g}$ are the corresponding right-hand sides of the sub-domain i and the interface, respectively. Matrices $\mathbf{F}_{ii}$ are sparse five-banded matrices, but their bandwidths are smaller than in the original system. Matrices $\mathbf{G}$, $\mathbf{D}_i$ and $\mathbf{D}_i^T$ are sparse matrices. The sparsity pattern of the reordered matrix is shown in Figure 4.4b.



(a) (b)

Figure 4.4: (a) The model is uniformly partitioned into 4×4 sub-domains. Every sub-domain has the same dimension, 2×2. The circles represent the interior nodes, while the nodes in between (triangles) are the interface nodes. (b) Sparsity pattern of the matrix in equation (4.10). It has a 2×2 block structure.

Instead of applying the direct solver, equation (4.2) can be solved by using block Gaussian elimination (Saad, 2003). Equation (4.2) is first expanded to smaller systems of interiors $\mathbf{u}_i$ and one system of interfaces $\mathbf{v}$ as follows,

$$\begin{aligned} \mathbf{F}_{ii}\mathbf{u}_i + \mathbf{D}_i\mathbf{v} &= \mathbf{f}_i, \\ \sum_{i=1}^{p} \mathbf{D}_i^T\mathbf{u}_i + \mathbf{G}\mathbf{v} &= \mathbf{g}. \end{aligned} \tag{4.3}$$

The interior $\mathbf{u}_i$ can be written as

$$\mathbf{u}_i = \mathbf{F}_{ii}^{-1}(\mathbf{f}_i - \mathbf{D}_i\mathbf{v}). \tag{4.4}$$

Substituting $\mathbf{u}_i$ into the second equation of (4.3), we obtain the system of interface $\mathbf{v}$,

$$(\mathbf{G} - \sum_{i=1}^{p} \mathbf{D}_i^T\mathbf{F}_{ii}^{-1}\mathbf{D}_i)\mathbf{v} = \mathbf{g} - \sum_{i=1}^{p} \mathbf{D}_i^T\mathbf{F}_{ii}^{-1}\mathbf{f}_i, \tag{4.5}$$

or, in short, as

$$\mathbf{S}\mathbf{v} = \mathbf{g}', \tag{4.6}$$

where the Schur complement $\mathbf{S}$ and its right-hand side $\mathbf{g}'$ are defined as

$$\mathbf{S} = \mathbf{G} - \sum_{i=1}^{p} \mathbf{D}_i^T \mathbf{F}_{ii}^{-1} \mathbf{D}_i, \tag{4.7}$$

$$\mathbf{g}' = \mathbf{g} - \sum_{i=1}^{p} \mathbf{D}_i^T \mathbf{F}_{ii}^{-1} \mathbf{f}_i. \tag{4.8}$$

After $\mathbf{v}$ is found from (4.6), the interior $\mathbf{u}_i$ is then obtained from

$$\mathbf{F}_{ii}\mathbf{u}_i = \mathbf{f}_i - \mathbf{D}_i\mathbf{v}, \tag{4.9}$$

where $\mathbf{F}_{ii}$ was earlier factorized in equation (4.6). Therefore, solving the interior $\mathbf{u}_i$ is a forward and backward substitution. Steps of the Schur complement method in case of P sub-domains are summarized in Algorithm 1.

Algorithm 1 Schur complement method in the case of P sub-domains

1: Assign $\mathbf{S} = \mathbf{G}$
2: Assign $\mathbf{g}' = \mathbf{g}$
3: **for** $i = 1$ to P **do**
4: Compute LU-factorization of $\mathbf{F}_{ii}$
5: Compute $\mathbf{S} = \mathbf{S} - \mathbf{D}_i^T \mathbf{F}_{ii}^{-1} \mathbf{D}_i$ and $\mathbf{g}' = \mathbf{g}' - \mathbf{D}_i^T \mathbf{F}_{ii}^{-1} \mathbf{f}_i$
6: **end for**
7: Compute LU-factorization of $\mathbf{S}$
8: Solve $\mathbf{S}\mathbf{v} = \mathbf{g}'$ by using forward/backward substitution
9: **for** $i = 1$ to P **do**
10: Solve $\mathbf{F}_{ii}\mathbf{u}_i = \mathbf{f}_i - \mathbf{D}_i\mathbf{v}$ by using forward/backward substitution
11: **end for**

For simplicity, many block matrices, $\mathbf{F}_{ii}$, $\mathbf{D}_i$, and $\mathbf{D}_i^T$ will be represented by $\mathbf{F}$, $\mathbf{D}$, $\mathbf{D}^T$, respectively, and so the right-hand sides $\mathbf{f}_i$. Equation (4.2) will be written as

$$\begin{pmatrix} \mathbf{F} & \mathbf{D} \\ \mathbf{D}^T & \mathbf{G} \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{g} \end{pmatrix}, \tag{4.10}$$

where $\mathbf{F}$ is a block diagonal matrix, $\mathbf{G}$, $\mathbf{D}$ and $\mathbf{D}^T$ are sparse matrices. The sizes of $\mathbf{F}$ and $\mathbf{G}$ are proportional to the numbers of interiors and interfaces, respectively. The sizes of $\mathbf{D}$ and $\mathbf{D}^T$ are proportional to the number of interiors and interfaces. The system of equations (4.10) can be rewritten as a set of two equations,

$$\begin{aligned} \mathbf{F}\mathbf{u} + \mathbf{D}\mathbf{v} &= \mathbf{f}, \\ \mathbf{D}^T\mathbf{u} + \mathbf{G}\mathbf{v} &= \mathbf{g}. \end{aligned} \tag{4.11}$$

First of all, the interior unknown $\mathbf{u}$ is expressed as

$$\mathbf{u} = \mathbf{F}^{-1}(\mathbf{f} - \mathbf{D}\mathbf{v}). \tag{4.12}$$

Substituting $\mathbf{u}$ into the second equation of (4.11), the reduced system of interfaces $\mathbf{v}$ is

$$\mathbf{S}\mathbf{v} = \mathbf{g}', \tag{4.13}$$

where the Schur complement $\mathbf{S}$ and its right hand side $\mathbf{g}'$ are defined as

$$\mathbf{S} = \mathbf{G} - \mathbf{D}^T\mathbf{F}^{-1}\mathbf{D}, \tag{4.14}$$

and

$$\mathbf{g}' = \mathbf{g} - \mathbf{D}^T\mathbf{F}^{-1}\mathbf{f}. \tag{4.15}$$

After $\mathbf{v}$ is found from (4.13), the interiors $\mathbf{u}$ is then obtained from

$$\mathbf{F}\mathbf{u} = \mathbf{f} - \mathbf{D}\mathbf{v}. \tag{4.16}$$

Forming the Schur complement $\mathbf{S}$ might be not efficient if the inverse of $\mathbf{F}$ is exactly computed. The alternative way of forming the Schur complement is to solve for $\mathbf{D}'$ from

$$\mathbf{F}\mathbf{D}' = \mathbf{D}, \tag{4.17}$$

where $\mathbf{D}'$ is a full matrix with the same dimension as $\mathbf{D}$. The Schur complement and its right hand side could be computed from

$$\mathbf{S} = \mathbf{G} - \mathbf{D}^T\mathbf{D}', \tag{4.18}$$

and

$$\mathbf{g}' = \mathbf{g} - \mathbf{D}'^T\mathbf{f}, \tag{4.19}$$

respectively.

Hence, any systems of equations rewritten in the 2×2 block system could be solved via the Schur complement method. Later in this thesis, this method will be referred to as the standard Schur complement method.

A domain decomposition method via the Schur complement transforms the whole system, equation (4.1), into several sub-systems of interiors $\mathbf{F}_{ii}$ and one reduced system, the Schur complement $\mathbf{S}$ in (4.13). However the Schur complement $\mathbf{S}$ is generally a dense matrix. Therefore it could be prohibitively expensive to store and factorize the Schur complement $\mathbf{S}$, when the number of interface nodes becomes very large (see Papadrakakis & Bitzarakis, 1996; Schöberl, 2001).

Hence the Schur complements are mostly used as a preconditioning technique (e.g. Larsson, 1999). Another application of the Schur complement method is the hybrid direct/iterative solver, in which the direct solver and the iterative solver are applied to the interior problems and the reduced system, respectively (see Ben-Hadj-Ali *et al.*, 2008; Sourbier *et al.*, 2008).

4.3 Hierarchical domain decomposition

Hierarchical domain decomposition is similar to the wirebasket techniques of domain decomposition (see Smith, 1991, 1990). In this method, the intersection, the point that interfaces cross each other, plays more important role than in the standard Schur complement method. It was therefore chosen as a basis of domain decomposition method in this thesis. This will lead to the modification that will be demonstrated in Chapters 5 and 6.

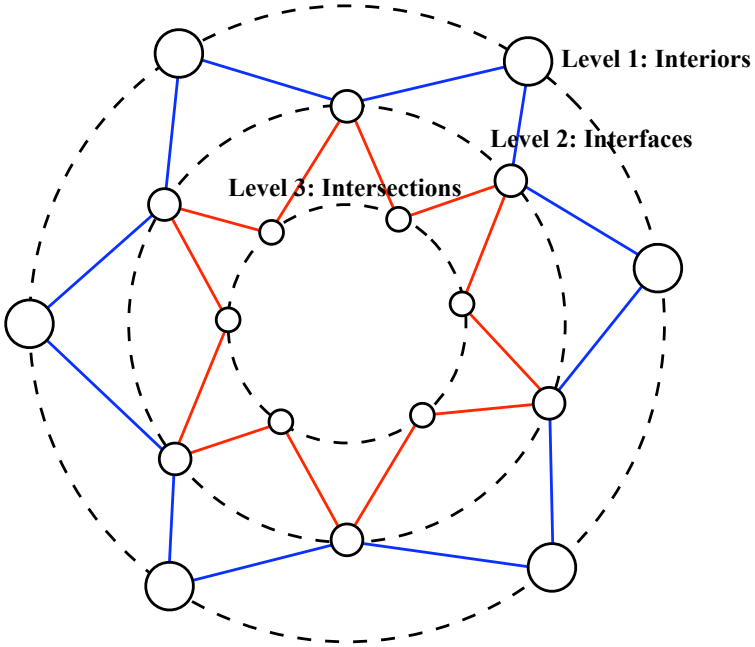
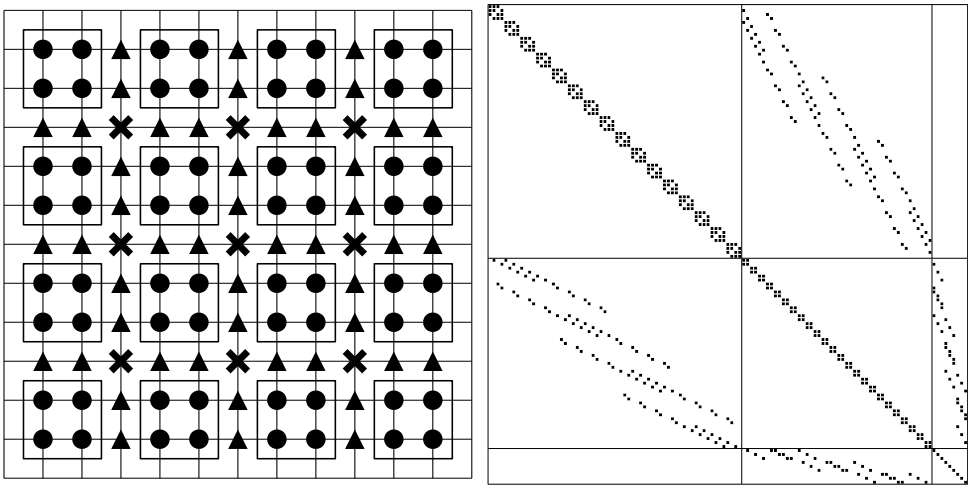


Figure 4.5: Diagram showing hierarchical property in two-dimensional magnetotelluric applications (redrawn from Henon & Saad, 2006). The groups of unknowns (circles) are grouped into three different levels (dashed concentric circles). Within the same level, the unknowns are linked to the adjacencies via the unknowns in the lower level.

On the decomposed domain, the vertices are categorized into different levels. Each level consists of a set of vertices, e.g. interiors and interfaces. The term hierarchical refers to the important property that the set of vertices of any levels is a "separator" for those in a lower level. Consequently, sets of unknowns in the same level have no direct coupling to each other (see Figure 4.5). For instance, once the domain is partitioned, the interfaces will separate the interiors. When horizontal interfaces cross the vertical interfaces, the intersections will act as the separator of the interfaces. The intersections are related to each other via the interfaces, and the interfaces are related to each other via the interiors.

Under the finite difference formulation in two-dimensional MT problems, nodes are related to the neighboring ones only in the up-, down-, left- and right-directions and itself (5-point FD). Intersections therefore have no direct contribution

to interiors but to interfaces (see Figure 4.6a). The intersections can then be excluded from being in the same category as the interface as in the standard Schur complement method. So the nodes can be classified into three levels from lowest to highest: (1) interiors (circles), (2) interfaces (triangles), and (3) intersections (crosses), respectively, as shown in Figure 4.6a.



(a) (b)

Figure 4.6: (a) For the partitioned domain, nodes are classified to three categories: interior (circles), interface (triangles) and intersection (crosses). (b) The sparsity pattern of the coefficient matrix in Figure 4.4b is reordered and becomes a 3×3 block matrix.

The unknowns are also reordered by the level in increasing order from interiors $\mathbf{u}$, interfaces $\mathbf{v}$ and intersections $\mathbf{w}$. For simplicity, the many block matrices, e.g., $\mathbf{F}_{ii}$, $\mathbf{D}_i$ and $\mathbf{D}_i^T$ in equation (4.2), are represented by $\mathbf{F}$, $\mathbf{D}$ and $\mathbf{D}^T$, respectively. The system of equations then becomes the 3×3 block system,

$$\begin{pmatrix} \mathbf{F} & \mathbf{D} & \mathbf{0} \\ \mathbf{D}^T & \mathbf{G} & \mathbf{E} \\ \mathbf{0} & \mathbf{E}^T & \mathbf{H} \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \\ \mathbf{w} \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{g} \\ \mathbf{h} \end{pmatrix}, \tag{4.20}$$

where $\mathbf{F}$, $\mathbf{G}$, and $\mathbf{H}$ are the coefficient matrices of interiors, interfaces, and intersections, respectively. $\mathbf{D}$ and $\mathbf{E}$ represent the contribution from interiors to interfaces and from interfaces to intersections, respectively, while their transposes represent the reciprocal contributions. $\mathbf{F}$ is a block diagonal matrix. In contrast to the standard Schur complement, $\mathbf{G}$ is a block diagonal matrix. $\mathbf{H}$ is a diagonal matrix. The sizes of $\mathbf{F}$, $\mathbf{G}$ and $\mathbf{H}$ are proportional to the numbers of interiors, interfaces, and intersections, respectively. $\mathbf{D}$ and $\mathbf{E}$ and their transposes are sparse matrices. The sizes of $\mathbf{D}$ and its transpose are proportional to the numbers of interiors and interfaces. The sizes of $\mathbf{E}$ and its transpose are proportional to the numbers of interfaces and intersections. On the right-hand side, $\mathbf{f}$ and $\mathbf{g}$ correspond to the boundaries of the interiors and interfaces, respectively. As

shown in Figure 4.6a, there is no boundary associated with the intersections, so $\mathbf{h}$ is zero. The sparsity pattern is shown in Figure 4.6b. The 3×3 block system in equation (4.20) can be expanded as

$$\begin{aligned} \mathbf{F}\mathbf{u} + \mathbf{D}\mathbf{v} &= \mathbf{f}, \\ \mathbf{D}^T\mathbf{u} + \mathbf{G}\mathbf{v} + \mathbf{E}\mathbf{w} &= \mathbf{g}, \\ \mathbf{E}^T\mathbf{v} + \mathbf{H}\mathbf{w} &= \mathbf{h}. \end{aligned} \tag{4.21}$$

Two reduced systems, the interior-interface system and the interface-interior system, can be derived from equation (4.21). The unknowns are then successively solved from higher to lower orders. Thus, intersections are solved first and then interfaces and interiors.

The interior-interface system is derived from the first two equations of (4.21),

$$\begin{pmatrix} \mathbf{F} & \mathbf{D} \\ \mathbf{D}^T & \mathbf{G} \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{g} - \mathbf{E}\mathbf{w} \end{pmatrix}. \tag{4.22}$$

The interface-intersection system is obtained from substituting

$$\mathbf{u} = \mathbf{F}^{-1}(\mathbf{f} - \mathbf{D}\mathbf{v}).$$

into the last two equations of (4.21) and rewriting as,

$$\begin{pmatrix} \mathbf{S} & \mathbf{E} \\ \mathbf{E}^T & \mathbf{H} \end{pmatrix} \begin{pmatrix} \mathbf{v} \\ \mathbf{w} \end{pmatrix} = \begin{pmatrix} \mathbf{g}' \\ \mathbf{h} \end{pmatrix}, \tag{4.23}$$

where the interface Schur complement $\mathbf{S}$ and its right-hand side $\mathbf{g}'$ are given by,

$$\mathbf{S} = \mathbf{G} - \mathbf{D}^T\mathbf{F}^{-1}\mathbf{D}, \tag{4.24}$$

$$\mathbf{g}' = \mathbf{g} - \mathbf{D}^T\mathbf{F}^{-1}\mathbf{f}. \tag{4.25}$$

Equations (4.22) and (4.23) share the 2×2 block structure, so they can be solved by repeating the standard Schur complement method, equations (4.12) to (4.16). The intersections are first solved from

$$\mathbf{H}'\mathbf{w} = \mathbf{h}', \tag{4.26}$$

where the intersection Schur complement $\mathbf{H}'$ and its right-hand side $\mathbf{h}'$ are given by,

$$\mathbf{H}' = \mathbf{H} - \mathbf{E}^T\mathbf{S}^{-1}\mathbf{E}, \tag{4.27}$$

$$\mathbf{h}' = \mathbf{h} - \mathbf{E}^T\mathbf{S}^{-1}\mathbf{g}'. \tag{4.28}$$

$\mathbf{H}'$ is a dense matrix but its size is equal to the number of intersections, which is very small. After the intersections are known, the interfaces $\mathbf{v}$ are then found from

$$\mathbf{S}\mathbf{v} = \mathbf{g}' - \mathbf{E}\mathbf{w}, \tag{4.29}$$

and, eventually, the interiors $\mathbf{u}$ from

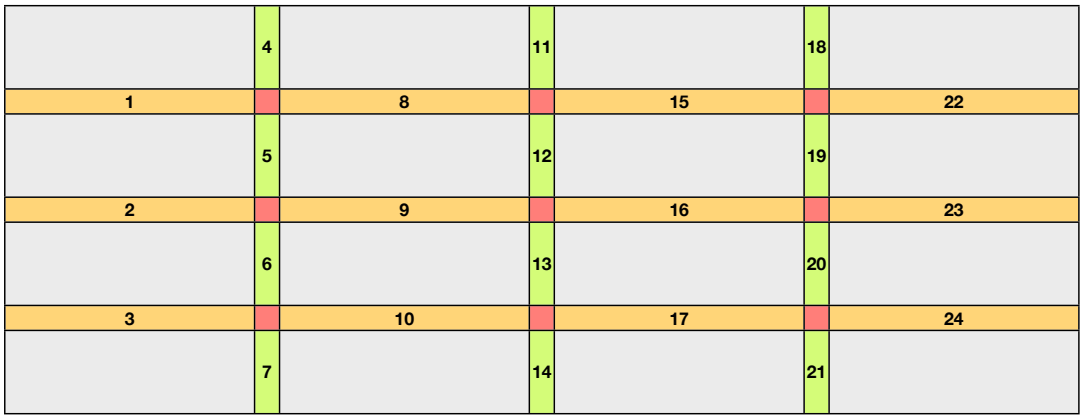
$$\mathbf{Fu} = \mathbf{f} - \mathbf{Dv}, \tag{4.30}$$

where $\mathbf{F}$ is factorized earlier in forming the interface Schur complemen $\mathbf{S}$, equation (4.24). Note that once the interface Schur complement is factorized, it can be used in both forming the intersection Schur complement $\mathbf{H}'$, equation (4.27), and solving the interface system, equation (4.29). Steps of applying the hierarchical domain decomposition, equation (4.20) to (4.30), are summarized in Algorithm 2.

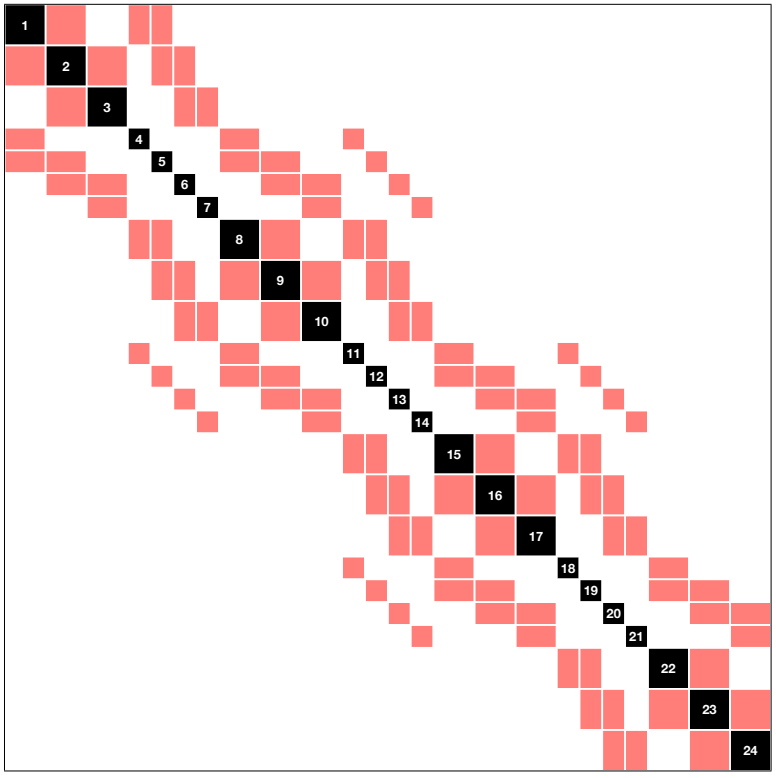
Algorithm 2 Schur complement method under the hierarchical decomposition

1: Assign $\mathbf{S} = \mathbf{G}$
2: Assign $\mathbf{g}' = \mathbf{g}$
3: **for** $i = 1$ to P **do**
4: Compute LU-factorization of $\mathbf{F}_{ii}$
5: Compute $\mathbf{S} = \mathbf{S} - \mathbf{D}_i^T\mathbf{F}_{ii}^{-1}\mathbf{D}_i$ and $\mathbf{g}' = \mathbf{g}' - \mathbf{D}_i^T\mathbf{F}_{ii}^{-1}\mathbf{f}_i$
6: **end for**
7: Compute LU-factorization of $\mathbf{S}$
8: Compute $\mathbf{H}' = \mathbf{H} - \mathbf{E}^T\mathbf{S}^{-1}\mathbf{E}$ and $\mathbf{h}' = \mathbf{h} - \mathbf{E}^T\mathbf{S}^{-1}\mathbf{g}'$
9: Compute LU-factorization of $\mathbf{H}'$
10: Solve $\mathbf{H}'\mathbf{w} = \mathbf{h}'$ by using forward/backward substitution
11: Solve $\mathbf{Sv} = \mathbf{g}' - \mathbf{Ew}$ by using forward/backward substitution
12: **for** $i = 1$ to P **do**
13: Solve $\mathbf{F}_{ii}\mathbf{u}_i = \mathbf{f}_i - \mathbf{D}_i\mathbf{v}$ by using forward/backward substitution
14: **end for**

Contrary to the reduced system in the standard Schur complement, the interface Schur complement $\mathbf{S}$ is a block matrix. However it will be stored as a full matrix due to applying LU factorization. For example, the model is partitioned into 4×4 sub-domains, as shown in Figure 4.7a. Ordering the interface segments from top to bottom and then left to right, as indicated by the numbers results in the interface Schur complement matrix with the block structure shown in Figure 4.7b. The numbers on black-colored diagonal blocks are referred to the numbers of interface segments. Both black- and red-colored blocks are dense matrices, and their sizes are proportional to the corresponding the interface segments. The larger blocks, for example, belong to the horizontal interfaces segments. However, they have different meanings. Blacks are the coupling within interface segments. Reds are the filled-in blocks due to the definition of the interface Schur complement, equation (4.24), which can be viewed as the relationship between the interface segments via the interiors. For example, Interface 1 is related to Interface 4 via the upper interior, and to Interfaces 2 and 5 via the lower interior (see Figure 4.7a), so there are the red blocks in columns 2, 4 and 5 of row 1, respectively (see Figure 4.7b).



(a)



(b)

Figure 4.7: (a) Model is partitioned into 4×4 sub-domains as an example. Grey represents interior segments. There are more horizontal nodes than vertical nodes. Orange and green represent horizontal and vertical interface segments, respectively. The horizontal interface segments are therefore larger than the vertical ones. Red represents the intersections. Every interface segment is numbered. (b) The sparsity pattern of the interface Schur complement $\mathbf{S}$, in which the numbers on the black blocks correspond to the interface segment number in Figure 4.7a.

4.4 Numerical experiments

The hierarchical domain decomposition (HD) was tested and compared to solving the whole domain (FDWD). The hierarchical domain decomposition was first validated with both synthetic and inverted models by comparing apparent resistivity and phase. The numerical efficiency of HD is shown by relative CPU time and memory. The direct solver used is LAPACK's LU factorization subroutine.

4.4.1 Validation

Responses of HD and FDWD are calculated from the three synthetic models: the conductive block (Figure 3.9a), the resistive block (Figure 3.10a), and the two blocks (Figure 3.11a), and the inverted model from Siripunvaraporn & Egbert (2000). The calculation is performed in both TE and TM modes at six periods, 0.01, 0.1, 1, 10, 100 and 1000 seconds. The synthetic models are non-uniformly discretized into 120×360 grids in the z- and y-directions, respectively. The computational domain is decomposed into 4×8 sub-domains with uniform partitioning. Every sub-domain is therefore at 39×44 interiors. The inverted model is non-uniformly discretized into 84×200 grids, and then uniformly partitioned into 4×8 sub-domains.

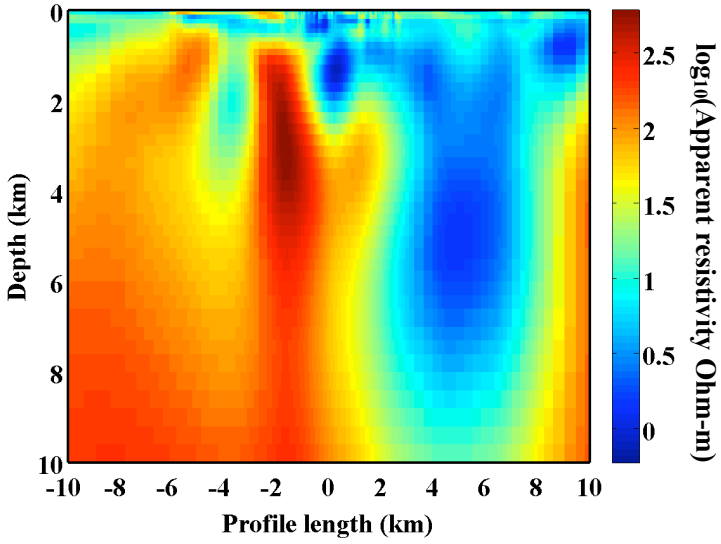


Figure 4.8: Inverted model from Siripunvaraporn & Egbert (2000). The actual model size is larger than shown.

The results from the three synthetic models are shown in Figures 4.11 - 4.14, and those from the inverted model are shown in Figures 4.15 and 4.16. These figures show that responses from both FDWD and HD are mostly identical. The difference in solutions is at the round-off level. This is as expected since both methods solve the same system of equations except that HD breaks the computational domain into

many smaller sub-domains and solves the problem through a series of smaller systems. HD is also valid for other combinations of sub-domains other than 4×8 sub-domains. In addition to uniform partitioning, the developed code also handles non-uniformly partitioned models. This therefore validates the accuracy of solutions obtained from hierarchical domain decomposition.

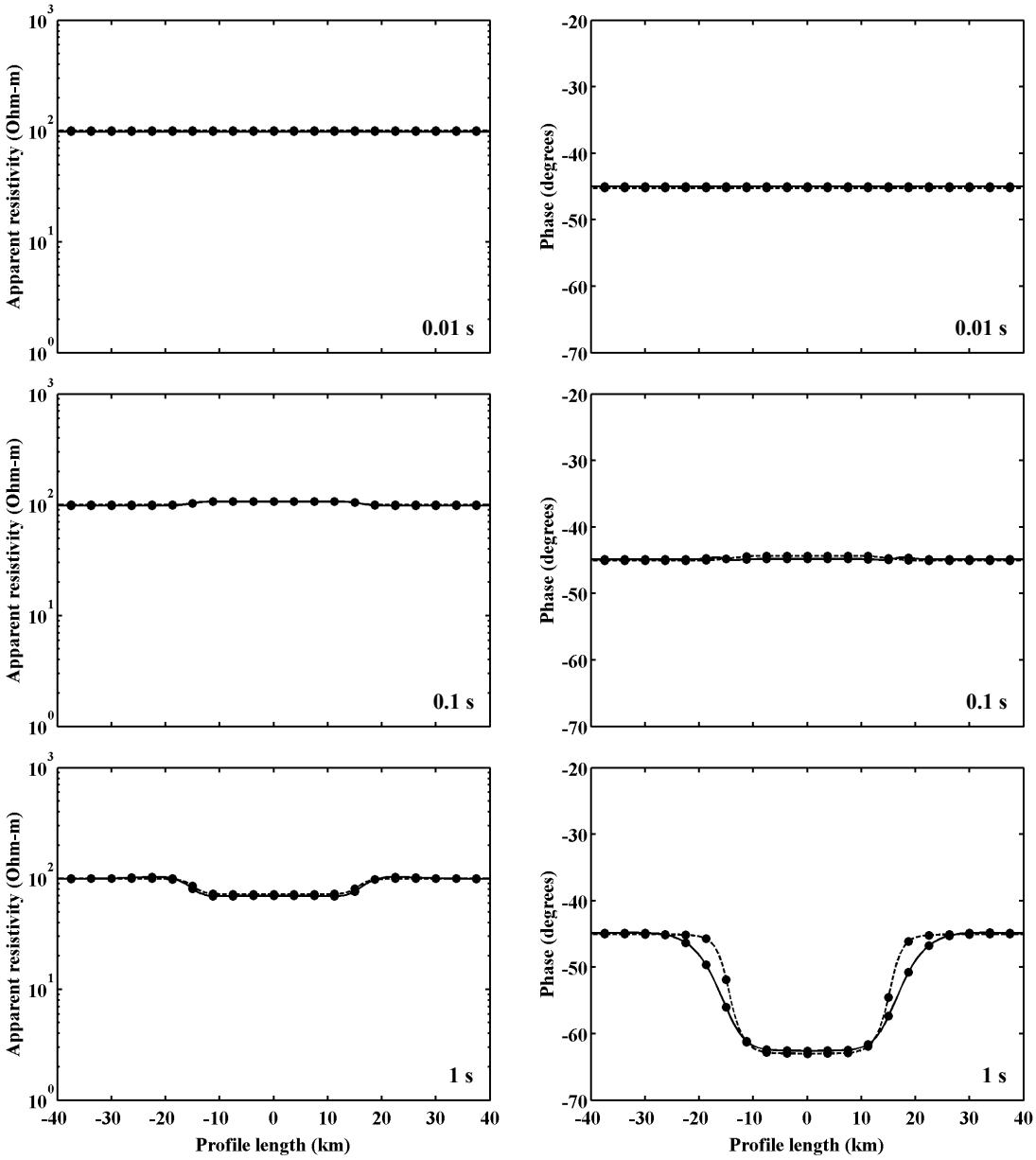


Figure 4.9: Apparent resistivity (left panel) and phase (right panel) from the conductive-block model (Figure 3.9a) for TE mode (solid line) and TM mode (dashed line) at periods of 0.01, 0.1 and 1 seconds. The responses are computed using FDWD (solid and dashed line) and HD (circles).

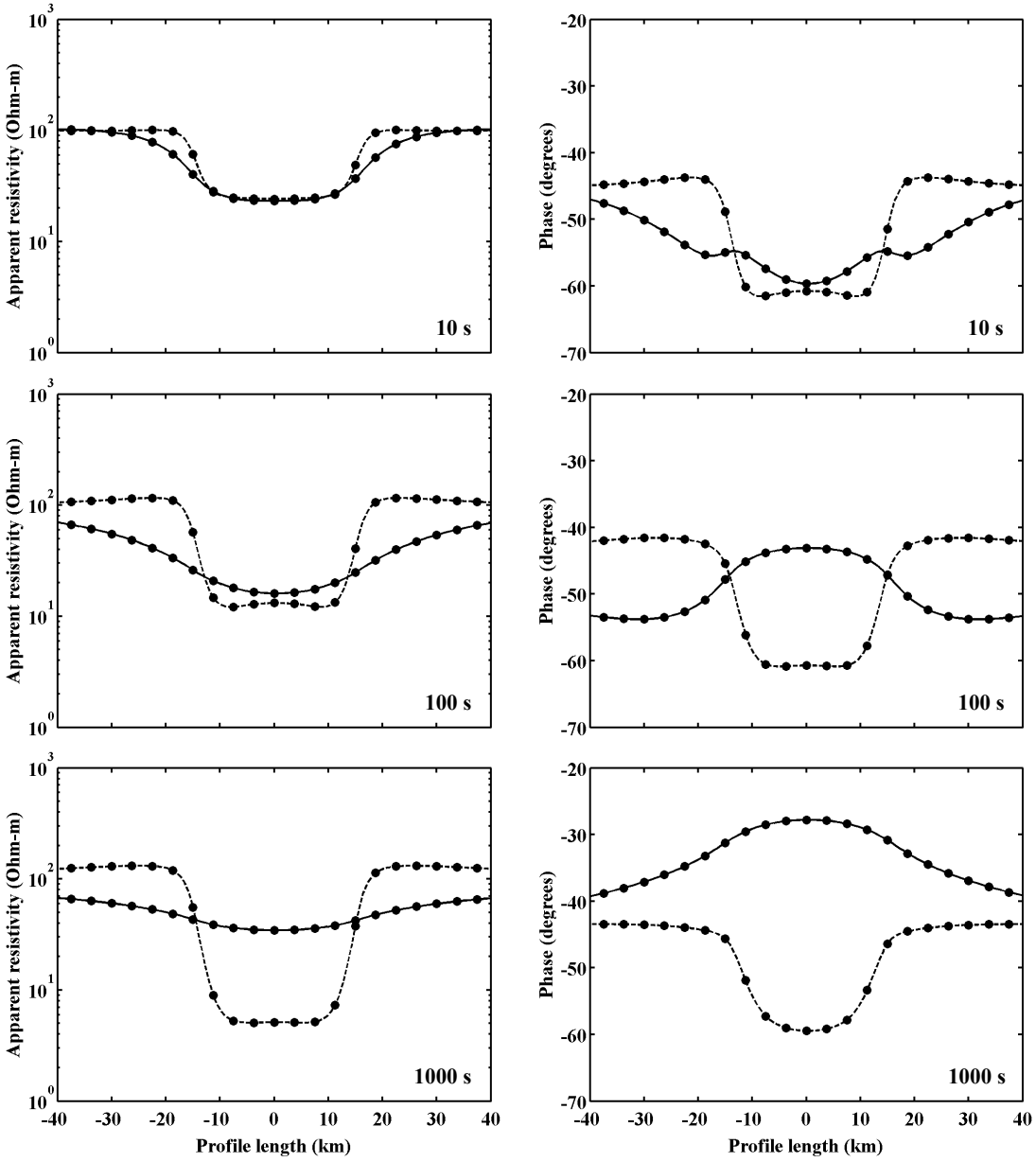


Figure 4.10: Apparent resistivity (left panel) and phase (right panel) from the conductive-block model (Figure 3.9a) for TE mode (solid line) and TM mode (dashed line) at periods of 10, 100 and 1000 seconds. The responses are computed using FDWD (solid and dashed line) and HD (circles).

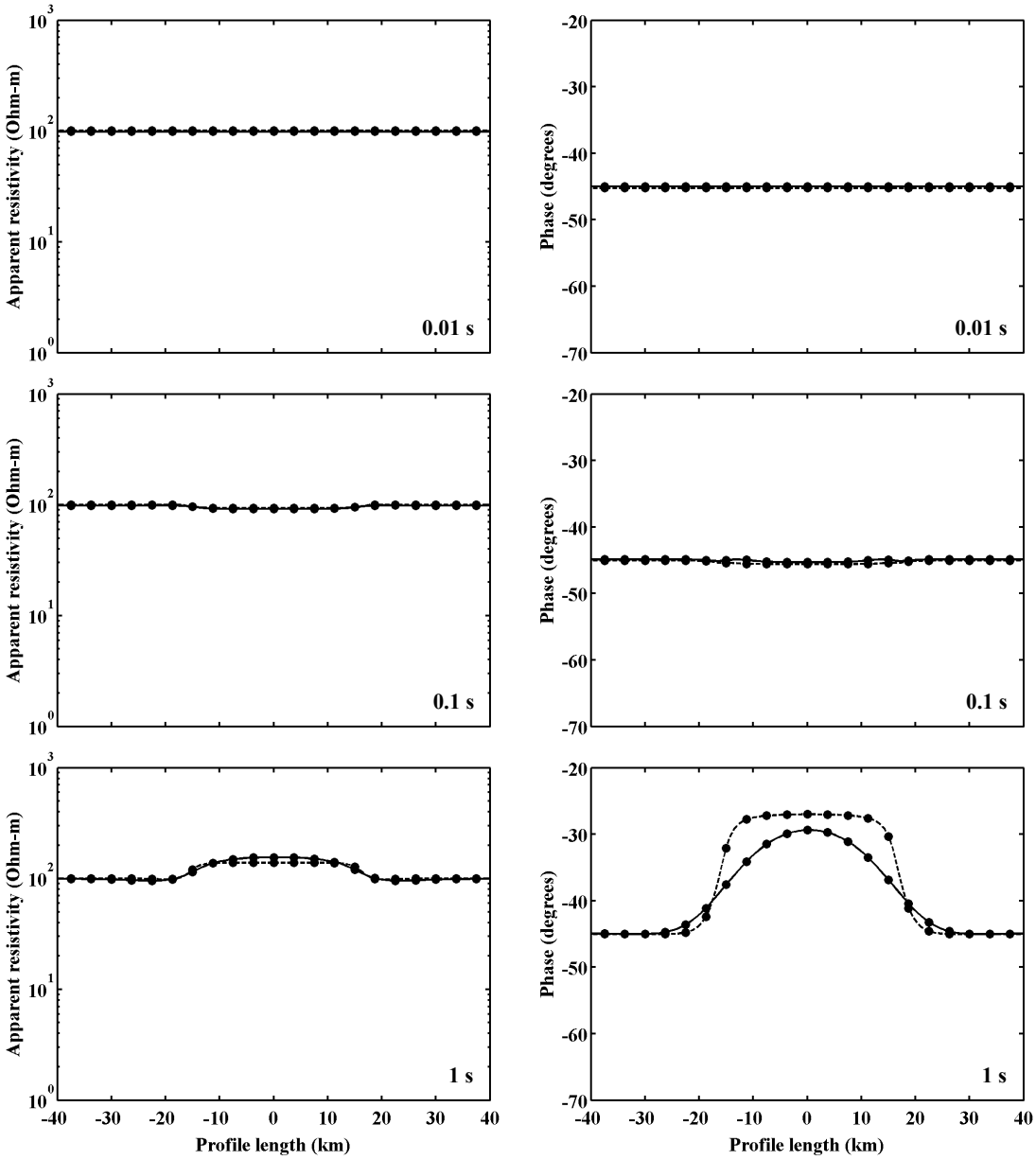


Figure 4.11: Apparent resistivity (left panel) and phase (right panel) from the resistive-block model (Figure 3.10a) for TE mode (solid line) and TM mode (dashed line) at periods of 0.01, 0.1 and 1 seconds. The responses are computed using FDWD (solid and dashed line) and HD (circles).

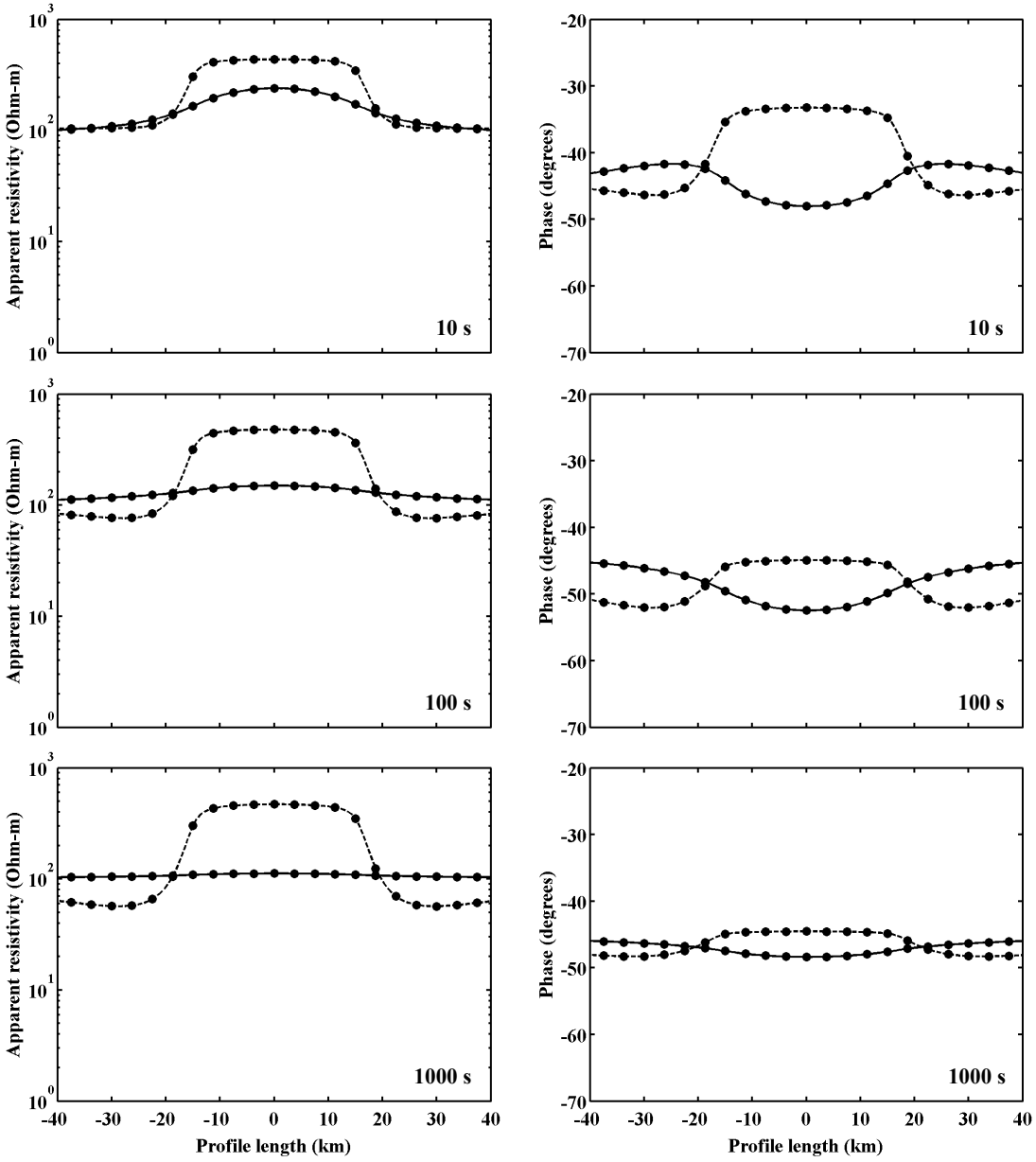


Figure 4.12: Apparent resistivity (left panel) and phase (right panel) from the resistive-block model (Figure 3.10a) for TE mode (solid line) and TM mode (dashed line) at periods of 10, 100 and 1000 seconds. The responses are computed using FDWD (solid and dashed line) and HD (circles).

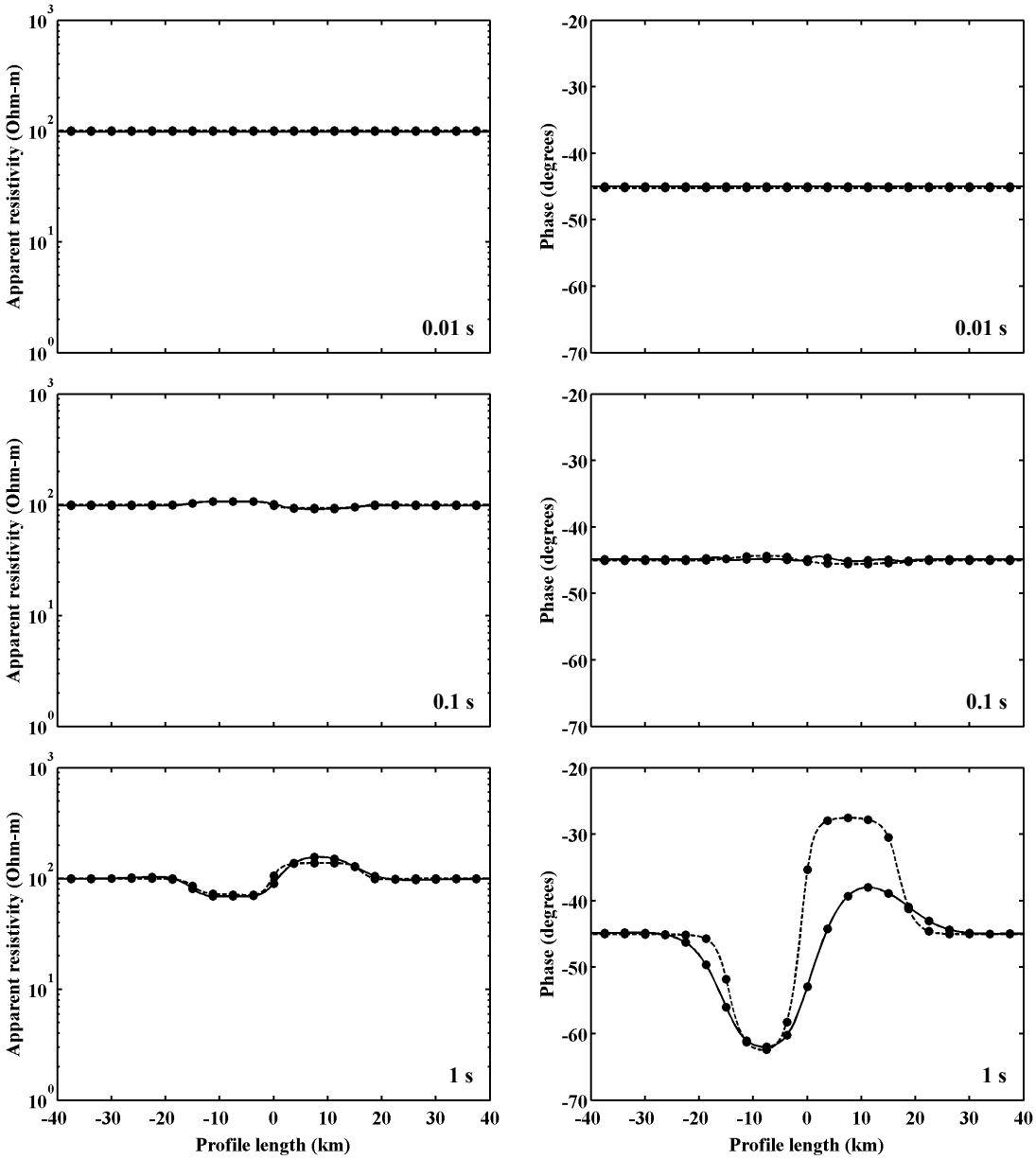


Figure 4.13: Apparent resistivity (left panel) and phase (right panel) from the two-block model (Figure 3.11a) for TE mode (solid line) and TM mode (dashed line) at periods of 0.01, 0.1 and 1 seconds. The responses are computed using FDWD (solid and dashed line) and HD (circles).

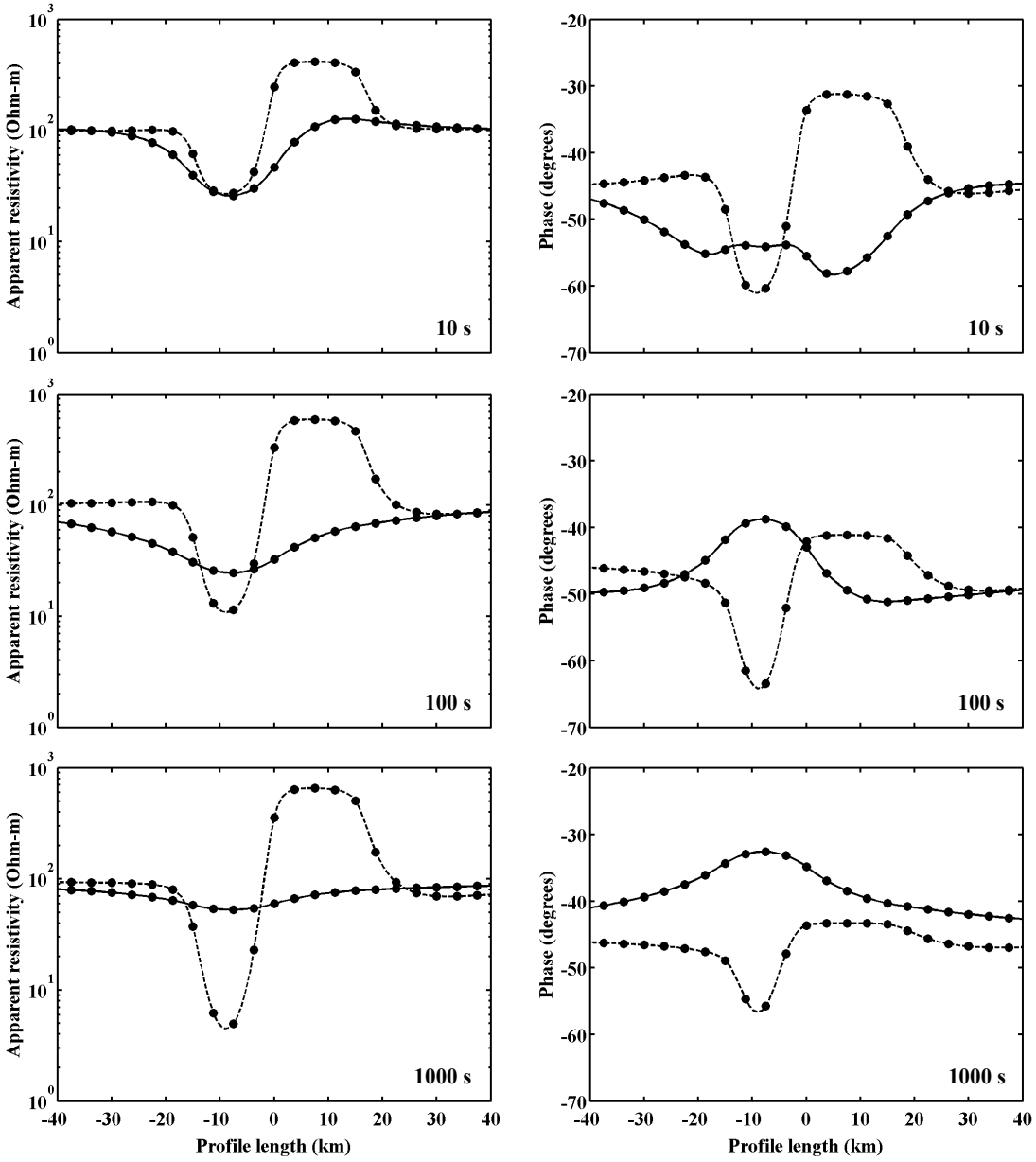


Figure 4.14: Apparent resistivity (left panel) and phase (right panel) from the two-block model (Figure 3.11a) for TE mode (solid line) and TM mode (dashed line) at periods of 10, 100 and 1000 seconds. The responses are computed using FDWD (solid and dashed line) and HD (circles).

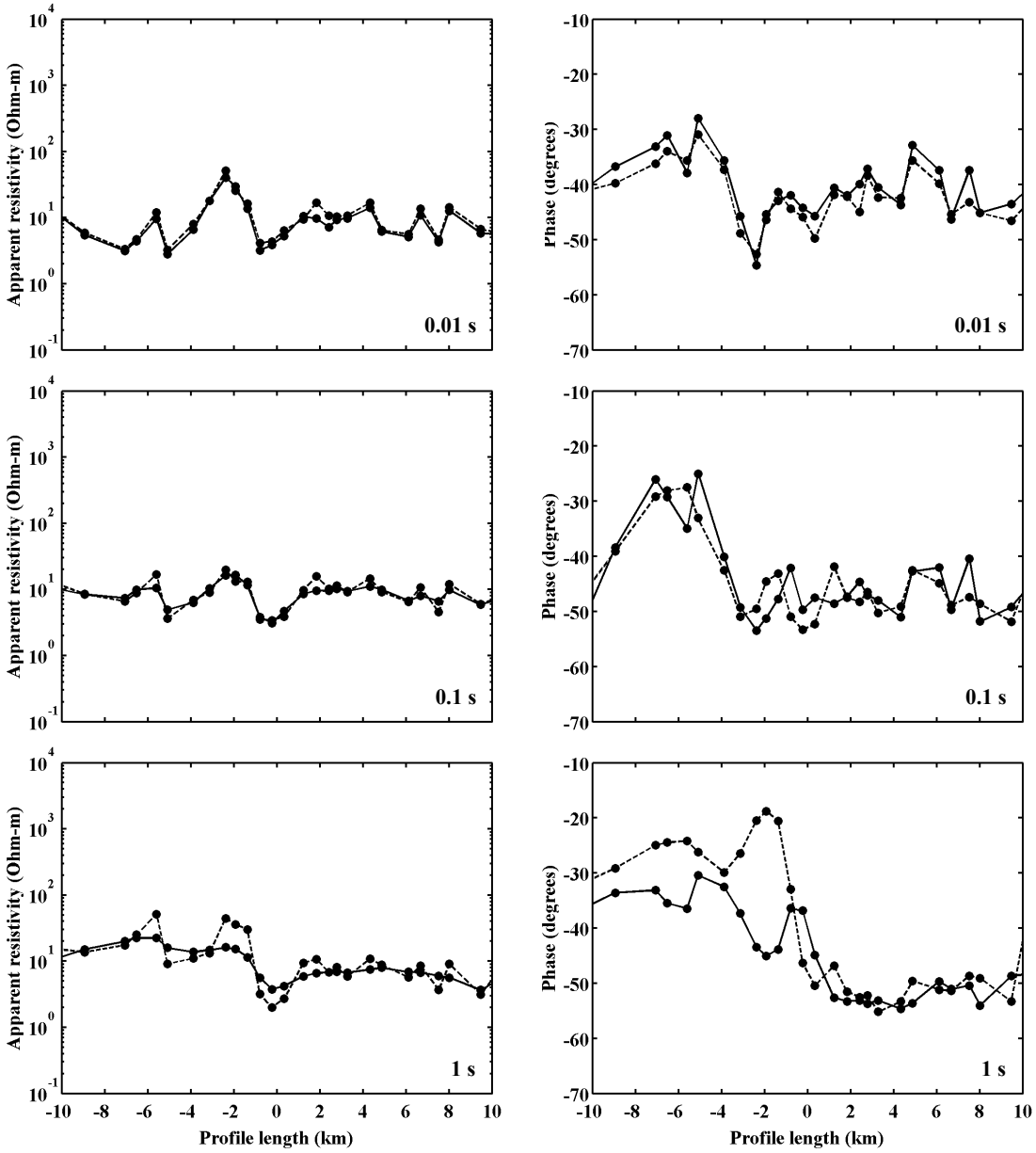


Figure 4.15: Apparent resistivity (left panel) and phase (right panel) from the inverted model (Figure 4.8) for TE mode (solid line) and TM mode (dashed line) at periods of 0.01, 0.1 and 1 seconds. The responses are computed using FDWD (solid and dashed line) and HD (circles).

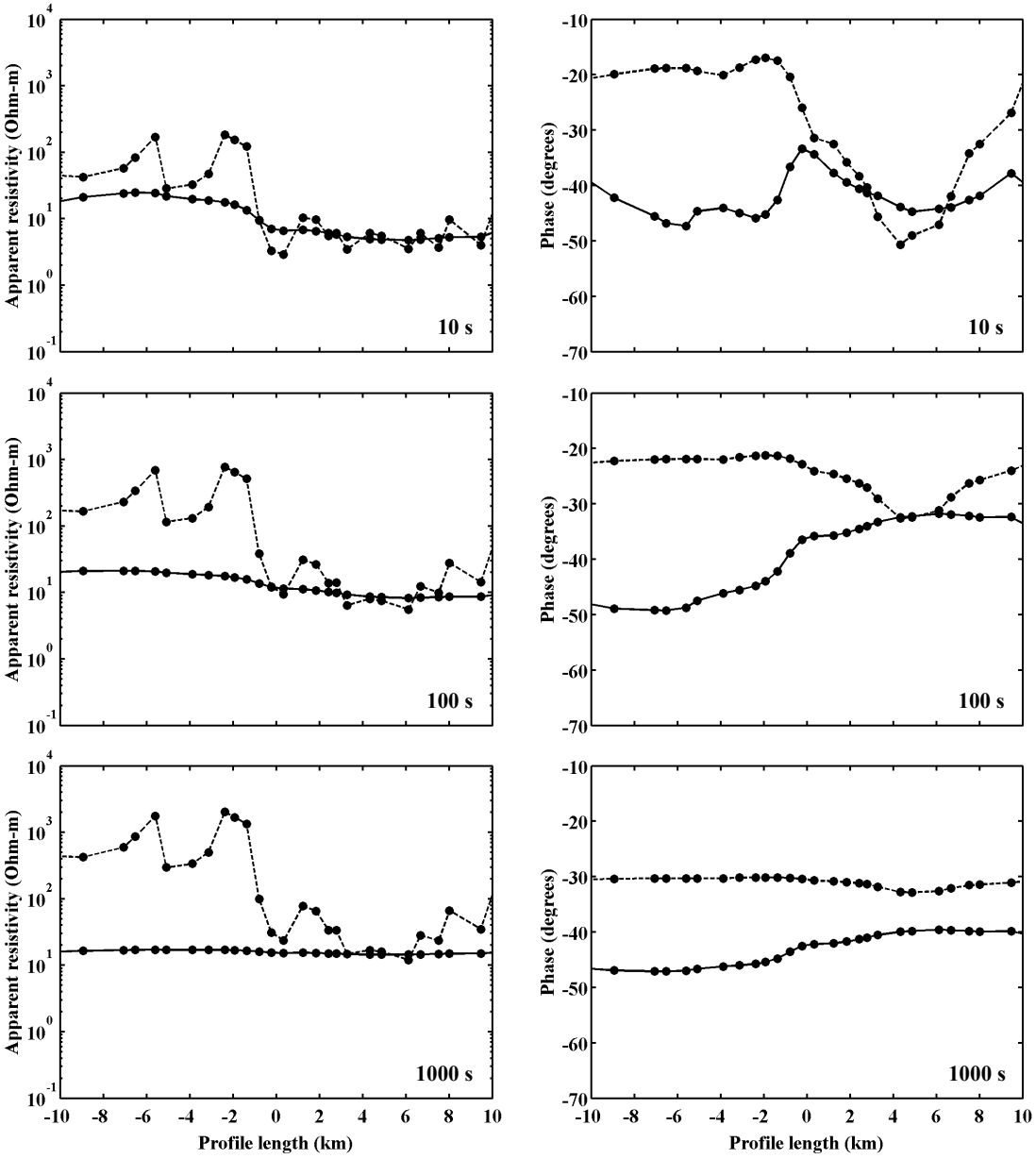


Figure 4.16: Apparent resistivity (left panel) and phase (right panel) from the inverted model (Figure 4.8) for TE mode (solid line) and TM mode (dashed line) at periods of 10, 100 and 1000 seconds. The responses are computed using FDWD (solid and dashed line) and HD (circles).

4.4.2 Numerical efficiency

At the same level of accuracy, the numerical efficiency of the hierarchical domain decomposition (HD) is assessed by CPU time and memory and compared with those obtained from solving the whole domain (FDWD). Memory is estimated from the double precision complex variables declared inside the code. There might be other memory usage by the operating system, but this is beyond our scope. The estimated memory is therefore a lower limit.

In FDWD, matrix $\mathbf{A}$ in equation (3.17) is stored as a banded matrix in LAPACK format. The memory usage of FDWD can be estimated from $(M_z-1)(M_y-1)(3M_z-1)\times 16$, where M_z and M_y are grid sizes in z- and y-directions, respectively. Multiplication with 16 is required because complex double precision is used. If the model is discretized at 120×360 mesh, $(120-1)(360-1)(3\cdot 120-1)\times 16 \approxeq 234$ MBytes of memory is necessary. Although not simple, when the domain is partitioned into $P_Z \times P_Y$ sub-domains, the memory usage of HD is obtained from multiplying the number of elements required (4.31) with 16.

$$
\begin{aligned}
\text{No. of elements}_{\text{HD}} =& 12 + M_Y + M_Y^2 + M_Z - 3M_Y M_Z + M_Z^2 \\
&- \frac{1}{P_Y^2} M_Y^2 \\
&+ \frac{1}{P_Y}(M_Y - M_Y^2) \\
&+ P_Y(-15 - M_Y + 9M_Z - 2M_Y M_Z - 2M_Z^2) \\
&+ P_Y^2(1 + M_Z + M_Z^2) \\
&+ \frac{1}{P_Z}(-M_Z^2 + 3M_Y M_Z^2) \\
&+ \frac{1}{P_Y^2 P_Z} M_Y^2 M_Z \\
&- \frac{1}{P_Y P_Z} M_Y M_Z \\
&- 2\frac{P_Y}{P_Z} M_Z^2 \\
&+ P_Z(-15 + 6M_Y - 2M_Y^2 - M_Z - 2M_Y M_Z) \\
&+ \frac{P_Z}{P_Y} M_Y^2 \\
&+ P_Y P_Z(5 + 4M_Y + 4M_Z + 2M_Y M_Z) \\
&- P_Y^2 P_Z(3 + 3M_Z) \\
&+ P_Z^2(1 + M_Y + M_Y^2) \\
&- P_Y P_Z^2(3 + 3M_Y) \\
&+ 3P_Y^2 P_Z^2.
\end{aligned}
\tag{4.31}
$$

CPU times shown in this thesis are obtained from an Intel Core Two Duo 6400, 2.13

GHz machine with 8 GBytes of RAM. All codes are written in Fortran95. Different machines and programming languages may result in different CPU times.

The first test is performed with the two-block model that is non-uniformly discretized into 120×360 grids, or $119 \times 359 = 42721$ unknowns. The calculation is performed only for the TE mode at period 10 seconds. Because of using LU factorization, the complexity of models, calculation modes and periods used do not significantly affect the computational time. Factors affecting the CPU time are model size and number of sub-domains. For example, when the number of sub-domains is small, the sub-problems $\mathbf{F}_{ii}$ are large, while the interface system $\mathbf{S}$ and intersection system $\mathbf{H}'$ are small. HD was thus tested on various numbers of sub-domains.

In this work, the notation of partitioning is $P_Z \times P_Y$, where P_Z and P_Y are the numbers of sub-domains in z-direction (or vertical) and in y-direction (or horizontal), respectively. The numbers of interiors, horizontal and vertical interfaces, and intersections are related to the combination of numbers of sub-domains, as stated in Table 4.1. The dimension of each segment type is the number of nodes in z-direction by the number of nodes in y-direction. For example, the numbers and sizes of interiors, interfaces, and intersections for the 120×360 mesh model partitioned into 4×4 sub-domains is shown in Table 4.2.

Variable type	Number of segment		Dimension of segment
	in z-direction	in y-direction	
Interior	P_Z	P_Y	$(M_Z/P_Z - 1) \times (M_Y/P_Y - 1)$
H-interface	$P_Z - 1$	P_Y	$1 \times (M_Y/P_Y - 1)$
V-interface	P_Z	$P_Y - 1$	$(M_Z/P_Z - 1) \times 1$
Intersection	$P_Z - 1$	$P_Y - 1$	1×1

Table 4.1: Numbers of interiors, horizontal and vertical interfaces, and intersections when the computation domain $M_Z \times M_Y$ is partitioned into $P_Z \times P_Y$.

Variable type	No. of segments $\times$ No. of nodes per segments	No. of nodes in total
Interior	$16 \times 2,581$	41,296
Interface	$12 \times 89 + 12 \times 29$	1,416
Intersection	9×1	9
Total	-	42,721

Table 4.2: Numbers of interiors, interfaces, and intersections of 120×360 meshes partitionined into 4×4 sub-domains are shown.

The contour plots of the number of nodes at various numbers of sub-domains in the z- and y-directions are shown in Figure 4.17. As expected, the number of interior

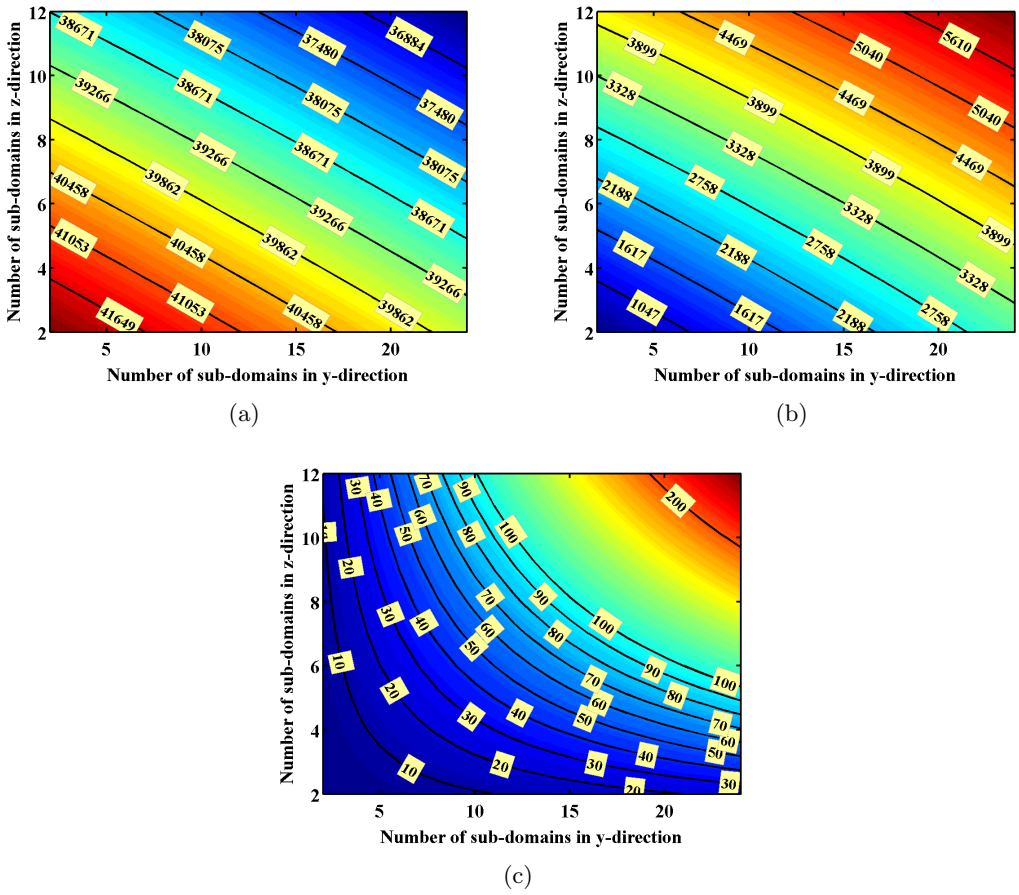


Figure 4.17: From 120×360 mesh model, the numbers of (a) interiors, (b) interfaces and (c) intersections are plotted as a function of the number of sub-domains in the z- and y-directions.

nodes is large when the number of sub-domains is small This is indicated by the red zone in Figure 4.17a. The number of interface nodes becomes larger as the number of sub-domains increases and so does the number of intersections.

To show that the efficiency of HD depends on the numbers of sub-domains, the numbers of sub-domains in the z- and y-directions are 2, 4, 6, 8 and 12, and 2, 4, 9, 18 and 24, respectively. The combination of these numbers will result in a uniform partitioning of the 120×360 mesh model. The contour plots of actual CPU time and memory are shown in Figures 4.18a and 4.18b, respectively. For this model size, FDWD consumes 5.59 seconds of CPU time, and 234 MBytes of memory, as calculated earlier.

From the actual CPU time map (Figure 4.18a), HD tends to consume higher CPU time and memory as the numbers of sub-domains increases. However, memory saving is gained at small numbers of sub-domains. When compared to the map of number of unknowns (Figure 4.17), the computational loads heavily depend on the interiors sub-problems $\mathbf{F}$ in equation (4.20) for small numbers of sub-domains (bottom

left of the map) and then move to the interface Schur complement $\mathbf{S}$ in equation (4.24) at larger numbers of sub-domains (top right of the map). To better demonstrate the efficiency of HD compared to FDWD, the relative CPU time and memory are calculated from (4.32) and (4.33), respectively, and plotted, as shown in Figure 4.19.

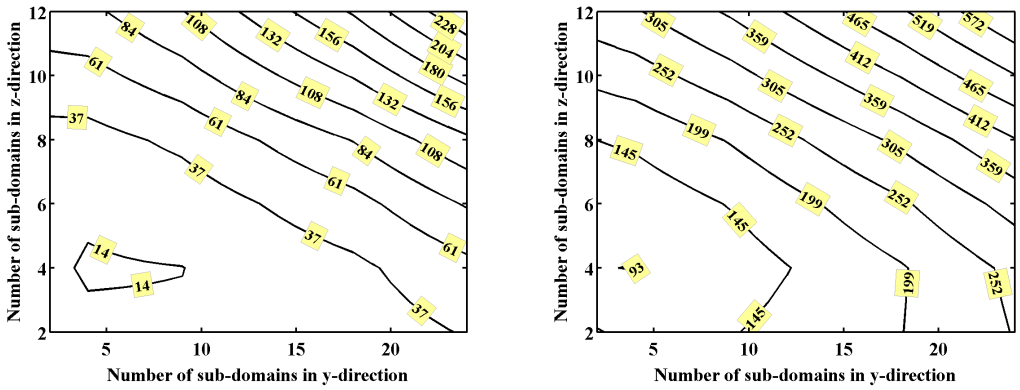
$$\text{Relative CPU time} = \frac{\text{Time}_{\text{HD}} - \text{Time}_{\text{FDWD}}}{\text{Time}_{\text{FDWD}}} \times 100\ \%, \tag{4.32}$$

$$\text{Relative memory} = \frac{\text{Memory}_{\text{HD}} - \text{Memory}_{\text{FDWD}}}{\text{Memory}_{\text{FDWD}}} \times 100\ \%. \tag{4.33}$$

In the map of relative CPU time and memory, the positive sign indicates that FDWD is superior to HD. Inversely, the negative sign shows that HD is superior to FDWD. It is clearly seen that, with no partitioning, HD is better than FDWD in terms of CPU time. The minimum CPU time from HD, 12.45 seconds at 4×4 partitioning, is 127% higher, and the minimum memory is approximately 60% less at 4×4 partitioning.

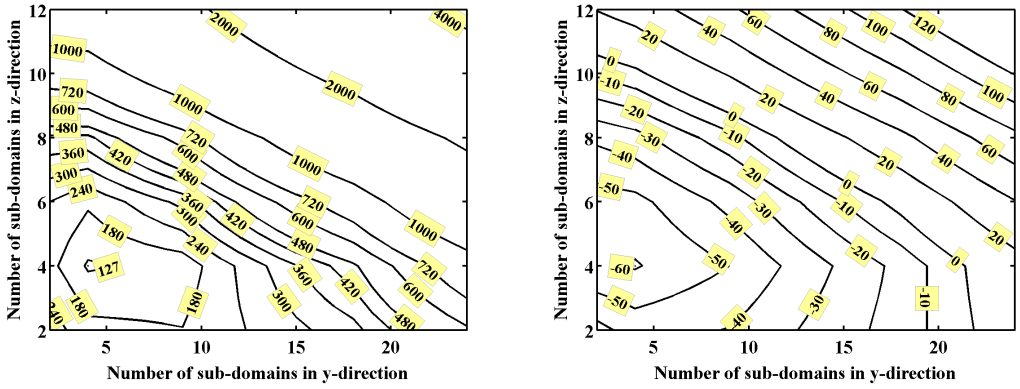
The hierarchical domain decomposition could basically be broken into 5 steps: (1) LU-factorization of interior sub-problems $\mathbf{F}_{ii}$, (2) forming and factorizing the interface Schur complement $\mathbf{S}$ in equation (4.24), (3) solving for intersections, equation (4.26), (4) solving for interfaces, equation (4.29), and (5) solving for interiors, equation (4.30). Note that Step 3, solving for intersections, includes forming and factorizing the intersection Schur complement $\mathbf{H}'$. For example, the times from each step of HD at eight sub-domains in the z-direction are plotted in Figure 4.20. The overall CPU time of HD is dominated by forming and factorizing the interface Schur complement (triangles). When the number of sub-domains increases, the sizes of matrix $\mathbf{D}$, the contribution from interior to interface, and its transpose are larger. Because the definition of the interface Schur complement (4.24) requires multiplication with matrix $\mathbf{D}$ and $\mathbf{D}_T$, the computational time in forming interface Schur complement $\mathbf{S}$ (Line 5 in Algorithm 2) increases. Also, the interface Schur complement is a full matrix.

The system of intersections (diamonds) also increases in complexity, because the definition of $\mathbf{H}'$, equation (4.26), contains the interface Schur complement. Consequently, when the interface size is large, solving for the intersection consumes more CPU time. Additionally, forming the intersection Schur complement $\mathbf{H}'$ (Line 8 in Algorithm 2) is similar to forming the interface Schur complement, but $\mathbf{E}$, the contribution from the interface to the interior, and its transpose are rather smaller. $\mathbf{H}'$ is a dense matrix but is very small. Thus solving the intersections is not as costly as forming and factorizing the interface Schur complement $\mathbf{S}$. After the interface Schur complement is factorized, solving for the interfaces (squares) is just done by forward and backward substitution, so it does not require a lot of CPU time. The load from interiors sub-problems, factorization (circles) and substitution (stars), is very small, because the interior sub-problems, $\mathbf{F}_{ii}$, are banded matrices.



(a) (b)

Figure 4.18: (a) Actual CPU time and (b) memory used by HD to solve 120×360 model at various numbers of sub-domains.



(a) (b)

Figure 4.19: (a) Relative CPU time and (b) memory maps correspond to Figures 4.18a and 4.18b, respectively.

4.5 Summary

Contrary to the standard Schur complement method, the hierarchical domain decomposition solves the problem through a number of smaller sub-systems of interiors and two reduced systems of interface and intersection. At the same level of accuracy, the hierarchical domain decomposition cannot reduce CPU time, but does reduce the memory requirement for some numbers of sub-domains. The overall time of the hierarchical domain decomposition is significantly influenced by forming and factorizing the interface Schur complement. This indicates that the hierarchical domain decomposition is not suitable for solving MT problems. Although the sub-problems are smaller, matrix multiplication is required to form the interface Schur complement. Also, high CPU time is used to factorize the interface Schur complement, which is a full matrix. However, the size of the intersection Schur complement $\mathbf{H}'$ is very small

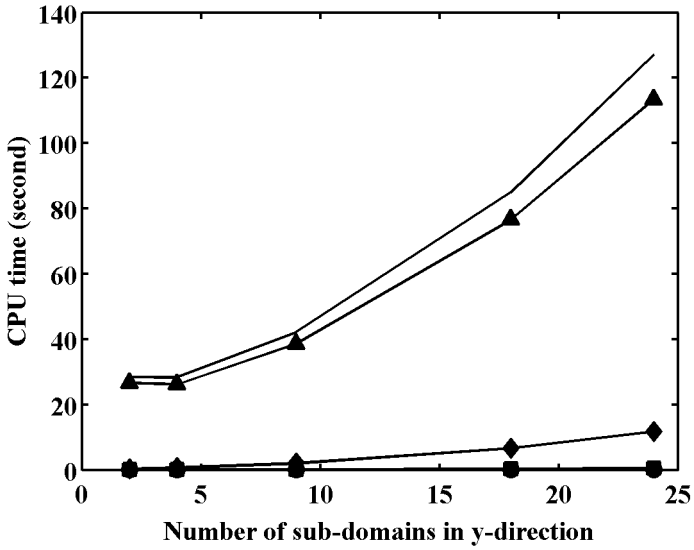


Figure 4.20: Overall time of HD (topmost solid line) consists of five steps: LU factorization of $\mathbf{F}_{ii}$ (circles), forming and factorizing interface Schur complement $\mathbf{S}$ (triangles), solving for the intersections (diamonds), solving for the interfaces (squares) and solving the interiors sub-problems for interiors (stars).

so that using the direct solver is reasonable. Thus, the modification must be applied to the interface Schur complement. This is because under the finite difference method in two-dimensional problems each node contributes only in the horizontal and vertical directions. Therefore, the horizontal and vertical interfaces have no direct coupling to each other. Consequently, the interface Schur complement could be broken by approximately one half.

CHAPTER V
MODIFIED HIERARCHICAL DOMAIN DECOMPOSITION

As demonstrated in the previous chapter, the performance of the hierarchical domain decomposition (HD) is not superior to solving the whole domain with a traditional direct solver. Therefore, a modification is necessary to speed it up. The first modification is to separate horizontal and vertical interfaces, which is described in this chapter, and will be referred to as MHD. Its numerical tests are conducted in a similar way to HD. The second modification is to implement the red-black ordering, which will be explained in the next chapter.

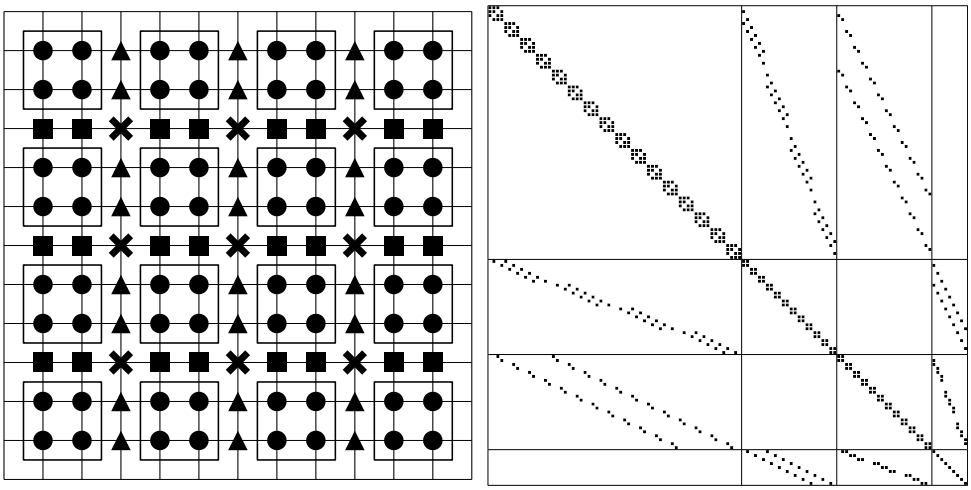
5.1 Modification of the hierarchical domain decomposition

To fully benefit from the hierarchical decomposition, the interfaces are classified into two types: horizontal and vertical interfaces (see Figure 5.1a). This will lead to a separation of the interface Schur complement $\mathbf{S}$ into a 2×2 block Schur complement system, which requires less computing time and memory requirement.

Here, the interface vector $\mathbf{v}$ in a 3×3 block system, equation (4.20), are reordered so that the horizontal interfaces $\mathbf{v}_H$ comes before the vertical interfaces $\mathbf{v}_V$. Equation (4.20) are then rewritten as the 4×4 block system,

$$\begin{pmatrix} \mathbf{F} & \mathbf{D}_H & \mathbf{D}_V & \mathbf{0} \\ \mathbf{D}_H^T & \mathbf{G}_H & \mathbf{0} & \mathbf{E}_H \\ \mathbf{D}_V^T & \mathbf{0} & \mathbf{G}_V & \mathbf{E}_V \\ \mathbf{0} & \mathbf{E}_H^T & \mathbf{E}_V^T & \mathbf{H} \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \mathbf{v}_H \\ \mathbf{v}_V \\ \mathbf{w} \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{g}_H \\ \mathbf{g}_V \\ \mathbf{h} \end{pmatrix}, \tag{5.1}$$

where subscripts H and V represent horizontal and vertical interfaces, respectively. Other variables are defined in a similar way to those in equation (4.20). The main difference from the original is the separation of matrix $\mathbf{G}$ into $\mathbf{G}_V$ and $\mathbf{G}_H$, where $\mathbf{G}_H$ is a coefficient matrix of the horizontal interfaces, and $\mathbf{G}_V$ is for the vertical interfaces. $\mathbf{D}_H$ and $\mathbf{D}_V$ are the coupling from interior to horizontal and vertical interfaces, respectively, and $\mathbf{D}_H^T$ and $\mathbf{D}_V^T$ are their reciprocals. $\mathbf{E}_H$ and $\mathbf{E}_V$ are the coupling from horizontal and vertical interfaces to intersections, respectively, and $\mathbf{E}_H^T$ and $\mathbf{E}_V^T$ are their reciprocals. $\mathbf{F}$, $\mathbf{G}_H$ and $\mathbf{G}_V$ are block diagonal matrices. $\mathbf{H}$ is a diagonal matrix. $\mathbf{D}_H$, $\mathbf{D}_V$, $\mathbf{E}_H$



(a) (b)

Figure 5.1: (a) For the 4×4 sub-domains, nodes can be classified into four types: the interior (circles), the horizontal interface (squares), the vertical interface (triangles) and the intersection (crosses). (b) The sparsity pattern of the reordered matrix in equation (5.1).

and $\mathbf{E}_V$ and their transposes are sparse matrices. As with the hierarchical domain decomposition, the size of $\mathbf{F}$ and $\mathbf{H}$ are proportional to the interiors and intersections, respectively. $\mathbf{G}_H$ and $\mathbf{G}_V$ are proportional to the horizontal interfaces and vertical interfaces, respectively. The sparsity pattern of the resulting matrix in equation (5.1) is shown in Figure 5.1b. For the case of 4×4 sub-domains, it is not essential to set the horizontal interface prior to the vertical interface. However, setting horizontal interfaces prior to vertical interfaces provides some advantages which will be explained later in this chapter.

From equation (5.1), the interface Schur complement can be written in the 2×2 block form as

$$\begin{aligned} \mathbf{S} &= \mathbf{G} - \mathbf{D}^T \mathbf{F}^{-1} \mathbf{D} \\ &= \begin{pmatrix} \mathbf{G}_H & \mathbf{0} \\ \mathbf{0} & \mathbf{G}_V \end{pmatrix} - \begin{pmatrix} \mathbf{D}_H^T \\ \mathbf{D}_V^T \end{pmatrix} \mathbf{F}^{-1} \begin{pmatrix} \mathbf{D}_H & \mathbf{D}_V \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{S}_{HH} & \mathbf{S}_{HV} \\ \mathbf{S}_{VH} & \mathbf{S}_{VV} \end{pmatrix}, \end{aligned} \tag{5.2}$$

where $\mathbf{S}_{HH}$ and $\mathbf{S}_{VV}$ are the interface Schur complements for the horizontal and vertical interfaces, respectively, and $\mathbf{S}_{HV}$ and $\mathbf{S}_{VH}$ are the couplings from horizontal to vertical interfaces and vertical to horizontal interfaces, respectively.

In the modified hierarchical domain decomposition, solving the intersection problem, equation (4.26), and the interior sub-problems, equation (4.30), remains the same. However, forming the intersection Schur complement, equation (4.27), and solving the interface problem, equation (4.29), can be computed via the reduced system of

interfaces, which are explained in the following

To form the intersection Schur complement $\mathbf{H}'$, equation (4.27) can be rewritten as,

$$\begin{aligned}\mathbf{H}' &= \mathbf{H} - \begin{pmatrix}\mathbf{E}_H^T & \mathbf{E}_V^T\end{pmatrix}\begin{pmatrix}\mathbf{S}_{HH} & \mathbf{S}_{HV}\\ \mathbf{S}_{VH} & \mathbf{S}_{VV}\end{pmatrix}^{-1}\begin{pmatrix}\mathbf{E}_H\\ \mathbf{E}_V\end{pmatrix}\\ &= \mathbf{H} - \begin{pmatrix}\mathbf{E}_H^T & \mathbf{E}_V^T\end{pmatrix}\begin{pmatrix}\mathbf{E}'_H\\ \mathbf{E}'_V\end{pmatrix},\end{aligned} \tag{5.3}$$

where $\mathbf{E}'_V$ are first computed from

$$\mathbf{S}'_{VV}\mathbf{E}'_V = \mathbf{E}_V - \mathbf{S}_{VH}\mathbf{S}_{HH}^{-1}\mathbf{E}_H, \tag{5.4}$$

where the reduced system, $\mathbf{S}'_{VV}$, of the interface Schur complement are defined as

$$\mathbf{S}'_{VV} = \mathbf{S}_{VV} - \mathbf{S}_{VH}\mathbf{S}_{HH}^{-1}\mathbf{S}_{HV}. \tag{5.5}$$

$\mathbf{E}'_H$ is then solved from

$$\mathbf{S}_{HH}\mathbf{E}'_H = \mathbf{E}_H - \mathbf{S}_{HV}\mathbf{E}_V. \tag{5.6}$$

As with forming the intersection Schur complement $\mathbf{H}'$, to obtain the interfaces, we start by rewriting (4.29) as,

$$\begin{pmatrix}\mathbf{S}_{HH} & \mathbf{S}_{HV}\\ \mathbf{S}_{VH} & \mathbf{S}_{VV}\end{pmatrix}\begin{pmatrix}\mathbf{v}_H\\ \mathbf{v}_V\end{pmatrix} = \begin{pmatrix}\mathbf{g}'_H\\ \mathbf{g}'_V\end{pmatrix} - \begin{pmatrix}\mathbf{E}_H\\ \mathbf{E}_V\end{pmatrix}\mathbf{w}. \tag{5.7}$$

$\mathbf{v}_V$ is first obtained from

$$\mathbf{S}'_{VV}\mathbf{v}_V = \mathbf{g}'_V - \mathbf{S}_{VH}\mathbf{S}_{HH}^{-1}(\mathbf{g}'_H - \mathbf{E}_H\mathbf{w}), \tag{5.8}$$

where $\mathbf{S}'_{VV}$ was defined in (5.5). $\mathbf{v}_H$ is then solved from

$$\mathbf{S}_{HH}\mathbf{v}_H = \mathbf{g}'_H - \mathbf{E}_H\mathbf{w} - \mathbf{S}_{HV}\mathbf{v}_V. \tag{5.9}$$

Once the $\mathbf{S}_{HH}$ and $\mathbf{S}'_{VV}$ are factorized, they will be used in forming the intersection Schur complement $\mathbf{H}'$, equations (5.4) and (5.6), and solving the interface problem, equations (5.8) and (5.9).

As mentioned earlier, the horizontal interface is set prior to the vertical interface. The dimension of the reduced system $\mathbf{S}'_{VV}$ is then equal to the number of nodes on the vertical interfaces. On the contrary, if the vertical interface is set prior to the horizontal interface, the dimension of the reduced interface system will be equivalent to the number of nodes of the horizontal interface, which is much larger than that of the vertical interface in two-dimensional MT problems.

Figure 5.2a shows the block pattern of the interface Schur complement from setting the horizontal interface prior to vertical interface and the inverse is shown in

Figure 5.2b. The bottom right block matrices, which correspond to the size of the reduced interface system, in this two figures differ in dimension, which is larger for Figure 5.2b than for Figure 5.2a. Setting the horizontal interface prior to the vertical interface is important in this work because the model size of MT problem is generally larger in the horizontal direction than in the vertical direction. With this configuration it will require less computational loads for storing and factorizing the reduced interface system.

In addition to setting the horizontal interface prior to the vertical interface, the order of the interface segments within the same types is crucial. In this work, both horizontal and vertical interface segments are ordered from top to bottom and then left to right. This results in the optimized block structures which will be used in the next chapter.

From the 4×4 partitioned model, where the segment number is shown in Figure 4.7a, the horizontal interfaces, for example, are ordered from the group of numbers 1, 2 and 3, and the group of 8, 9 and 10, and so on. Similarly, the vertical interfaces are ordered from the group of numbers 4, 5, 6, and 7, and the group of 11, 12, 13 and 14, and so on. The resulting block pattern of the interface Schur complement is shown in Figure 5.2a. These result in the upper left block $\mathbf{S}_{HH}$ in the form of a block diagonal matrix, and the lower right block $\mathbf{S}_{VV}$, which is the same for $\mathbf{S}'_{VV}$, in the form a block tridiagonal matrix. Inversely, if both interface types are ordered from left to right and then top to bottom, the block patterns of S_{HH} and S_{VV} are block tridiagonal and block diagonal, respectively, as shown in Figure 5.3.

The advantage of having $\mathbf{S}_{HH}$ in the form of a block diagonal matrix is that LU factorization of the whole $\mathbf{S}_{HH}$ is equivalent to that of its diagonal blocks. The diagonal block of $\mathbf{S}_{HH}$ will be denoted by $\mathbf{S}_{HH,ii}$. Therefore only diagonal blocks of $\mathbf{S}_{HH}$ will be stored. Although $\mathbf{S}'_{VV}$ is a block tridiagonal matrix, it will be stored as a full matrix due to applying LU factorization. Having $\mathbf{S}'_{VV}$ in the form of block tridiagonal matrix and its advantages will be explained in the next chapter. The steps in factorizing the 2×2 interface Schur complement is also summarized in Algorithm 3. In this work, the horizontal interfaces are set to be prior to the vertical interfaces, and both interface types are ordered from top to bottom and then left to right. These will lead to the optimized computational load in storing and factorizing $\mathbf{S}_{HH}$ and $\mathbf{S}'_{VV}$.

From the block structure of the interface Schur complement, only the non-empty blocks of $\mathbf{S}_{HH}$, $\mathbf{S}_{HV}$, $\mathbf{S}_{VH}$ and $\mathbf{S}_{VV}$ are stored. Furthermore, forming the reduced interface system $\mathbf{S}'_{VV}$ in equation (5.5) could be performed block by block by using a block matrix operation. For example, the interface Schur complement for the 4×4

Algorithm 3 Factorizing the 2×2 interface Schur complement

1: **for** $i = 1$ to P_Y **do**
2: Compute LU-factorization of $\mathbf{S}_{HH,ii}$
3: **end for**
4: Compute $\mathbf{S}'_{VV} = \mathbf{S}_{VV} - \mathbf{S}_{VH}\mathbf{S}_{HH}^{-1}\mathbf{S}_{HV}$
5: Compute LU-factorization of $\mathbf{S}'_{VV}$

partitioning, whose sparsity pattern is shown in Figure 5.2a, can be written as

$$\left(\begin{array}{c|c} \mathbf{S}_{HH} & \mathbf{S}_{HV} \\ \hline \mathbf{S}_{VH} & \mathbf{S}_{VV} \end{array}\right) = \left(\begin{array}{cccc|ccc} \mathbf{S}_{HH,11} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{S}_{HV,15} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_{HH,22} & \mathbf{0} & \mathbf{0} & \mathbf{S}_{HV,25} & \mathbf{S}_{HV,26} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{S}_{HH,33} & \mathbf{0} & \mathbf{0} & \mathbf{S}_{HV,36} & \mathbf{S}_{HV,37} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{S}_{HH,44} & \mathbf{0} & \mathbf{0} & \mathbf{S}_{HV,47} \\ \hline \mathbf{S}_{VH,51} & \mathbf{S}_{VH,52} & \mathbf{0} & \mathbf{0} & \mathbf{S}_{VV,55} & \mathbf{S}_{VV,56} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_{VH,62} & \mathbf{S}_{VH,63} & \mathbf{0} & \mathbf{S}_{VV,65} & \mathbf{S}_{VV,66} & \mathbf{S}_{VV,67} \\ \mathbf{0} & \mathbf{0} & \mathbf{S}_{VH,73} & \mathbf{S}_{VH,74} & \mathbf{0} & \mathbf{S}_{VV,76} & \mathbf{S}_{VV,77} \end{array}\right), \tag{5.10}$$

where $\mathbf{S}_{HH}$, $\mathbf{S}_{HV}$, $\mathbf{S}_{VH}$ and $\mathbf{S}_{VV}$ with subscripts represent block matrices in $\mathbf{S}_{HH}$, $\mathbf{S}_{HV}$, $\mathbf{S}_{VH}$ and $\mathbf{S}_{VV}$, respectively. The block matrix operation for computing $\mathbf{S}'_{VV,55}$, for example, could be performed as

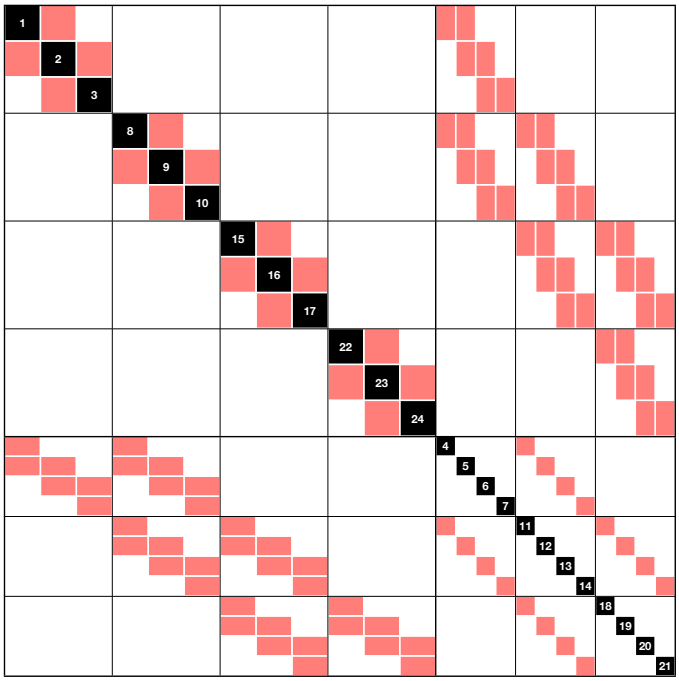
$$\mathbf{S}'_{VV,55} = \mathbf{S}_{VV,55} - \mathbf{S}_{VH,51}\mathbf{S}_{HH,11}^{-1}\mathbf{S}_{HV,15} + \mathbf{S}_{VH,52}\mathbf{S}_{HH,22}^{-1}\mathbf{S}_{HV,25}.$$

Null block matrices in $\mathbf{S}_{HH}$, $\mathbf{S}_{HV}$, $\mathbf{S}_{VH}$ and $\mathbf{S}_{VV}$ are therefore excluded from the calculation. After $\mathbf{S}'_{VV}$ is completely formed, every block will be filled according to its definition, equation (5.5). The block pattern of $\mathbf{S}'_{VV}$, as shown in Figure 5.4, is the same as that of $\mathbf{S}_{VV}$ in Figure 5.2a, but each block is now filled with yellow sub-blocks.

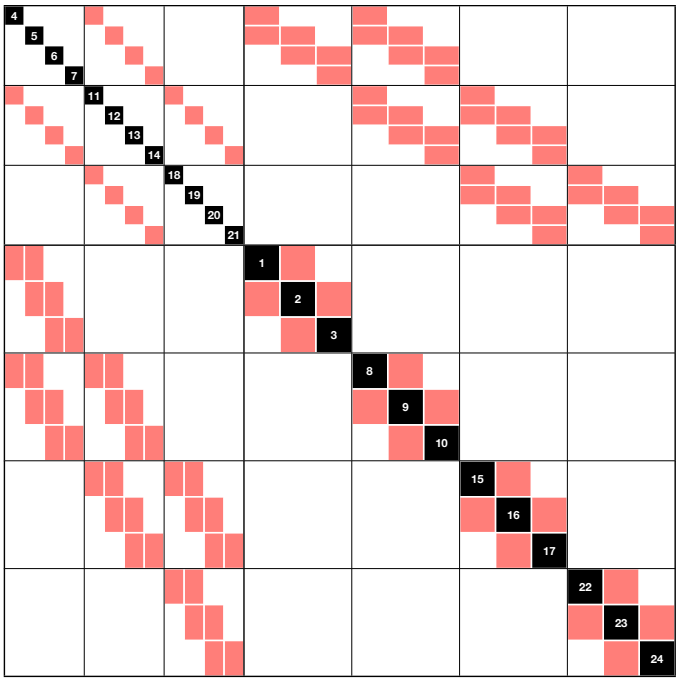
In the hierarchical domain decomposition, the dimension of a square matrix interface Schur complement $\mathbf{S}$ in equation (4.24) is the number of all interface nodes. The 120×360 mesh model partitioned into 4×4 sub-domains, for example, means $\mathbf{S}$ has a size of 1416×1416, which is 30.6 MBytes of memory to be stored.

In the modified hierarchical domain decomposition, the 2×2 interface Schur complement, equation (5.2), consists of $\mathbf{S}_{HH}$, $\mathbf{S}_{HV}$, $\mathbf{S}_{VH}$ and $\mathbf{S}_{VV}$, which are block matrices. In general, the memory used for $\mathbf{S}_{HH}$, $\mathbf{S}_{HV}$, $\mathbf{S}_{VH}$ and $\mathbf{S}_{VV}$ can be estimated from the product of the number of blocks and sub-blocks and the size of the interface segment. The term sub-block refers to the colored-block in the sparsity pattern of the 2×2 interface Schur complement. In Figure 5.2a, $\mathbf{S}_{HH}$, for example, contains 4 blocks, and their sub-block dimensions are 3×3.

Block and sub-block dimensions of $\mathbf{S}_{HH}$ and $\mathbf{S}_{VV}$ are directly related to the numbers of horizontal and vertical interfaces in the z- and y-directions, as stated in Table 5.1. The block and sub-block dimensions of the reduced interface system $\mathbf{S}'_{VV}$ are consistent with those of $\mathbf{S}_{VV}$. For both types of interfaces, sub-block dimension is



(a)



(b)

Figure 5.2: The block pattern of interface Schur complement from different interface ordering. (a) The horizontal interfaces are set to be prior to the vertical interfaces. (b) The vertical interfaces are set to be prior to the horizontal interface. For both (a) and (b), the interface segments with the same alignment are ordered from top to bottom and then left to right. This two block patterns can be viewed as a 2×2 block matrices which are distinguished by the thick line. The numbers on black sub-blocks correspond to those of interface segments in Figure 4.7a. Note that the bottom right block of Figure 5.2b is larger than that of Figure 5.2a.

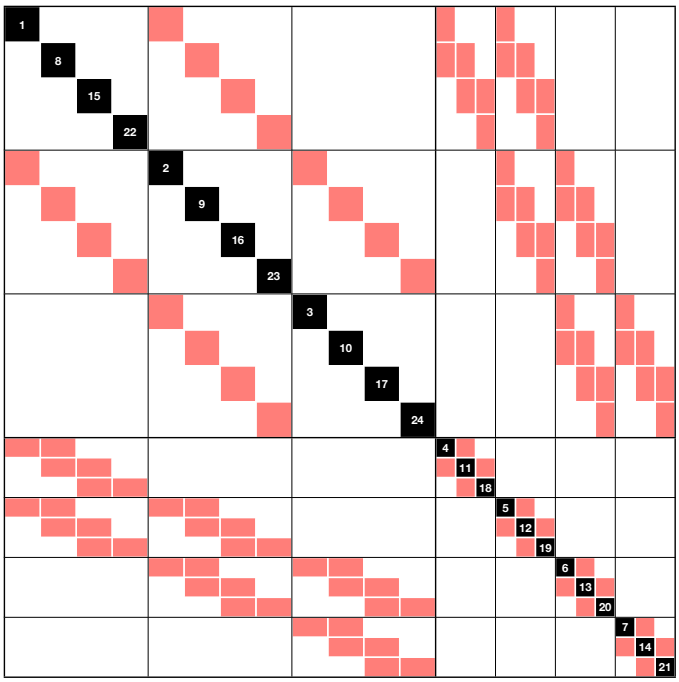


Figure 5.3: As with Figure 5.2a, but both the horizontal and vertical interfaces are ordered from left to right and then top to bottom. This results in a different block pattern that is less optimized, because $\mathbf{S}_{HH}$, the upper right block, is not a block diagonal matrix. Note that the pattern of $\mathbf{S}_{HV}$ and $\mathbf{S}_{VH}$ are also changed.

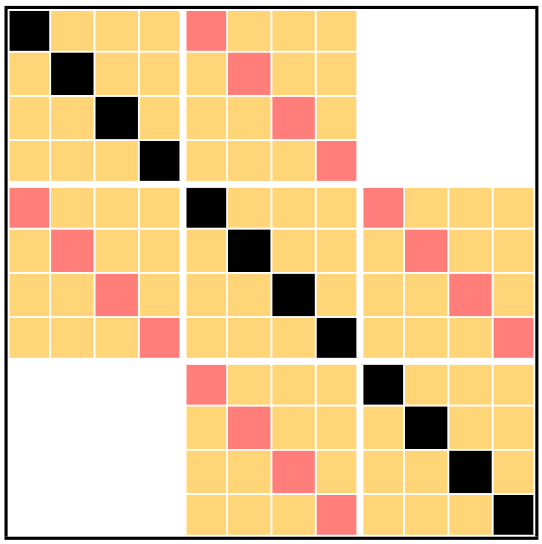


Figure 5.4: The reduced interface system $\mathbf{S}'_{VV}$ obtained from 4×4 partitioning is now filled with yellow sub-blocks according to its definition.

consistent with the number of sub-domains in the z-direction, and block dimension is consistent with the number of sub-domains in the y-direction.

Matrix	Block dimension	Sub-block dimension
$\mathbf{S}_{HH}$	$P_Y \times P_Y$	$P_Z - 1 \times P_Z - 1$
$\mathbf{S}_{HV}$	$P_Y \times P_Y - 1$	$P_Z - 1 \times P_Z$
$\mathbf{S}_{VH}$	$P_Y - 1 \times P_Y$	$P_Z \times P_Z - 1$
$\mathbf{S}_{VV}$	$P_Y - 1 \times P_Y - 1$	$P_Z \times P_Z$

Table 5.1: Block and sub-block dimensions of the interface Schur complement in MHD

The 4×4 partitioned model, for instance, results in an $\mathbf{S}_{HH}$ with block and sub-block dimensions of 4×4 and 3×3, respectively. However, $\mathbf{S}_{HH}$ has only 4 diagonal blocks, and all 3×3 sub-blocks are stored for each. The block and sub-block dimensions of $\mathbf{S}'_{VV}$ are 3×3 and 4×4, respectively, and the whole 3×3 blocks of $\mathbf{S}'_{VV}$ will be stored.

The dimensions of the blocks and sub-blocks affect the memory stored for the 2×2 interface Schur complement. From the 120×360 model partitioned into 4×4 sub-domains, the numbers of elements in $\mathbf{S}_{HH}$, $\mathbf{S}_{HV}$, $\mathbf{S}_{VH}$ and $\mathbf{S}'_{VV}$, shown in Table 5.2, need about 11.9 MBytes of memory, which is 61% less than HD. However, the overall memory reduction is essentially not equal to the reduction of the interface system. The total number of elements to be stored can be computed from (5.11).

Matrix	Number of elements
$\mathbf{S}_{HH}$	$4 \times 3^2 \times 89^2$
$\mathbf{S}_{HV}$	$(2 \cdot 3) \times (3 \cdot 4) \times (89 \cdot 29)$
$\mathbf{S}_{VH}$	$(2 \cdot 3) \times (4 \cdot 3) \times (29 \cdot 89)$
$\mathbf{S}'_{VV}$	$(3^2) \times 4^2 \times 29^2$
Total	777,924

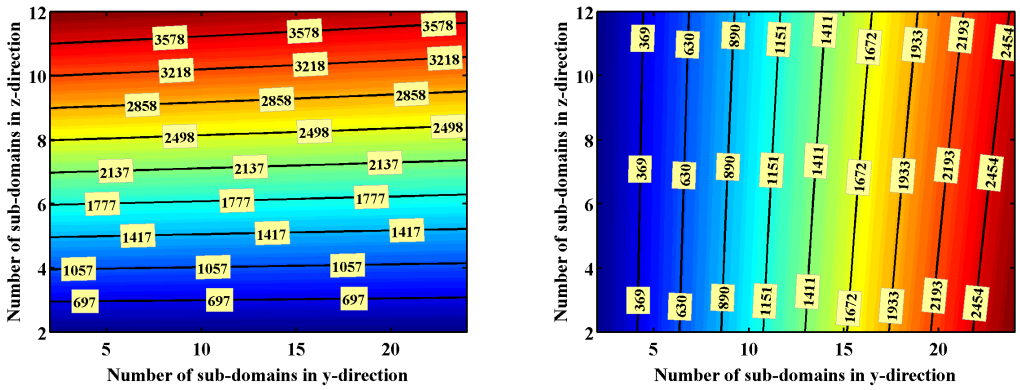
Table 5.2: Number of elements stored for the interface Schur complement in MHD, where the model size is 120×360 and the model is partitioned into 4×4 sub-domains.

$$
\begin{aligned}
\text{No. of elements}_{\text{MHD}} =& 12 - M_Y - 3M_Z - 9M_Y M_Z + M_Z^2 \\
& - \frac{1}{P_Y^2} M_Y^2 \\
& + \frac{1}{P_Y}(M_Y + 4M_Y M_Z) \\
& + P_Y(-14 + M_Y + 15M_Z - 2M_Z^2) \\
& + P_Y^2(-M_Z + M_Z^2) \\
& + \frac{1}{P_Z}(-M_Z^2 + 3M_Y M_Z^2) \\
& + \frac{1}{P_Y^2 P_Z} M_Y^2 M_Z \\
& - \frac{1}{P_Y P_Z} M_Y M_Z \\
& - 2\frac{P_Y}{P_Z} M_Z^2 \\
& + P_Z(-11 + 16M_Y + 3M_Z + 4M_Y M_Z) \\
& - \frac{P_Z}{P_Y}(4M_Y + M_Y^2 + 4M_Y M_Z) \\
& - P_Y P_Z(3 + 2M_Y + 2M_Z) \\
& + P_Y^2 P_Z(1 - M_Z) \\
& - P_Z^2(3 + 7M_Y) \\
& + \frac{P_Z^2}{P_Y}(4M_Y + M_Y^2) \\
& + P_Y P_Z^2(4 + M_Y)
\end{aligned}
\tag{5.11}
$$

5.2 Numerical efficiency

The modified hierarchical domain decomposition (MHD) differs from the hierarchical domain decomposition in that the interfaces are separated into horizontal and vertical interfaces. However, the system of equations remains the same. Thus there is no significant change in accuracy in solutions from MHD. To avoid redundancy, the plots showing validation of MHD are omitted. Numerical tests are conducted to evaluate the efficiency of MHD in terms of CPU time and memory. The model used is the same as in Section 4.4.2 and all partitionings are also the same.

First of all, the contour plot of the number of interfaces is divided into a plot of the horizontal interfaces and a plot of the vertical interfaces, as shown in Figure 5.5a and 5.5b, respectively. The horizontal interfaces keep increasing as the number of sub-domains in the z-direction increases, while the vertical interfaces increase as the number of sub-domains in the y-direction increases. The number of interfaces is directly related to the size of the interface Schur complement $\mathbf{S}$. Therefore, the red zones in 5.5a and



(a) (b)

Figure 5.5: The plot of the numbers of interfaces (Figure 4.17b) is divided into (a) the numbers of horizontal interfaces and (b) the numbers of vertical interfaces.

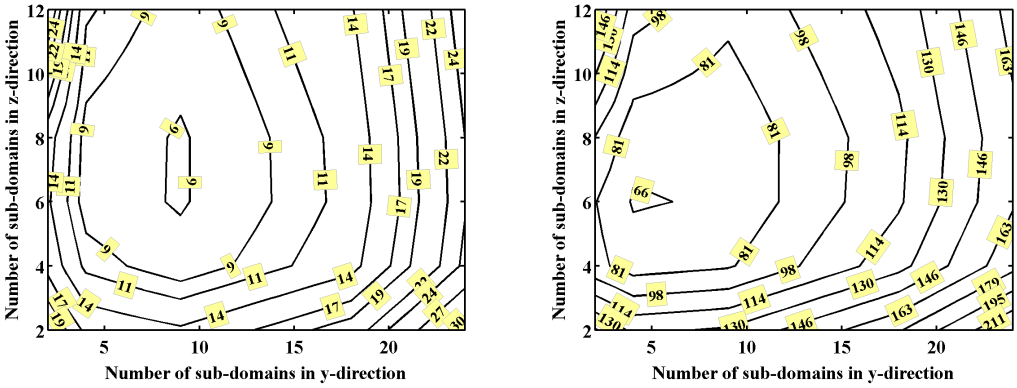
5.5b could be used to specify the boundary of the optimized numbers of sub-domains.

The contour maps of actual CPU time and memory are shown in Figure 5.6, and their relative values in Figure 5.7. At 6×9 sub-domains, the CPU time consumed by MHD is quite close to FDWD, and the memory usage is mostly improved all over the plot with a maximum of 70% reduction at 6×4 partitioning. Both CPU time and memory of MHD are less than those from HD. However, MHD is not yet more efficient than FDWD in terms of CPU time. The minimized zones of CPU time and memory lie on the region bounded by the red zones in Figures 4.17a, 5.5a and 5.5b.

The reason why the overall time is greatly reduced could be explained by plotting CPU time from each steps of MHD (Figure 5.8), which is similar to that of HD (Figure 4.20). Forming and factorizing the interface Schur complement (triangles) still plays an important role in MHD, but relatively less than HD. This indicates that breaking the interface Schur complement into the 2×2 block structure, equation (5.2), helps reduce CPU time. Furthermore, CPU time used in solving for intersections (diamonds) also decreases, because the intersection Schur complement $\mathbf{H}'$ involves the interface Schur complement $\mathbf{S}$ (see equation (5.3) to (5.6)). LU factorization of $\mathbf{F}_{ii}$ (circles) solving for the interfaces (squares) and the interiors (stars) are the same as described in Section 4.4.2.

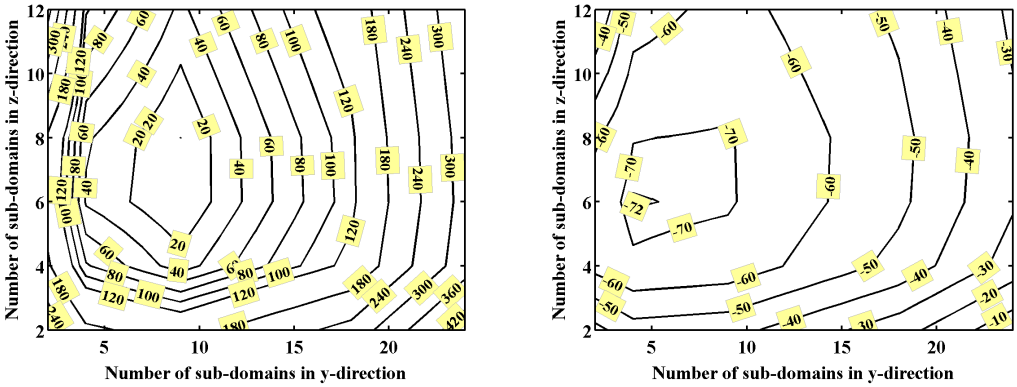
5.3 Summary

In modified hierarchical domain decomposition, the interfaces are separated into two groups, horizontal and vertical interfaces. The interface Schur complement is then written as 2×2 block matrix, equation (5.2). This leads to the reduction in factorization of interface Schur complement, instead of applying LU factorization to the



(a) (b)

Figure 5.6: (a) Actual CPU time and (b) memory used by MHD to solve 120×360 model at various numbers of sub-domains.



(a) (b)

Figure 5.7: (a) Relative CPU time and (b) memory maps correspond to Figure 5.6a and 5.6b, respectively.

whole interface Schur complement like HD. Because $\mathbf{S}_{HH}$, $\mathbf{S}_{HV}$, $\mathbf{S}_{VH}$ and $\mathbf{S}_{VV}$ are block matrices, their null matrices are not stored and excluded from the calculation. MHD thus provides higher efficiency than HD in both CPU time and memory. However, the CPU time of MHD is slightly larger than FDWD.

In this chapter, the advantage of forming the interface reduced system $\mathbf{S}'_{VV}$ as a block tridiagonal matrix is not yet explained. The second modification focuses on this block structure with the expectation that the CPU time will be better than for FDWD.

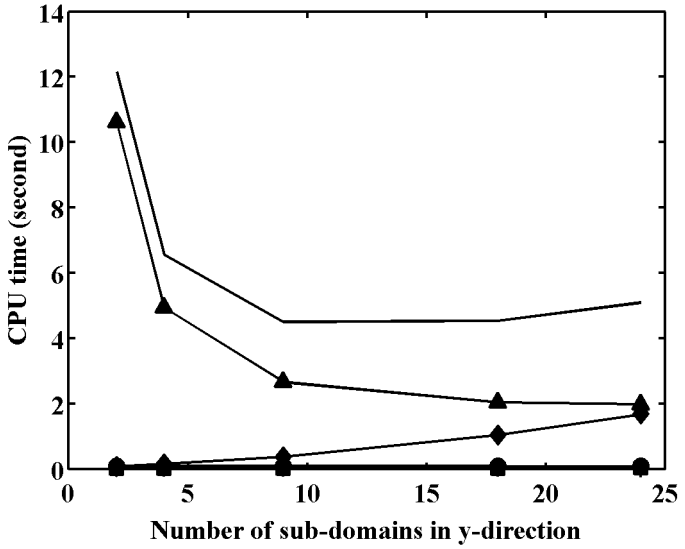


Figure 5.8: Overall time of MHD (topmost solid line) consists of five steps: LU factorization of $\mathbf{F}_{ii}$ (circles), forming and factorizing interface Schur complement $\mathbf{S}$ (triangles), solving for the intersections (diamonds), solving for the interfaces (squares) and solving the interiors sub-problems for interiors (stars).

CHAPTER VI
MODIFIED HIERARCHICAL DOMAIN DECOMPOSITION WITH RED-BLACK COLORING

The first modification made to the original hierarchical domain decomposition (HD) is to separate the horizontal and vertical interfaces and referred to as MHD. This results in reducing the size of the interface Schur complement by approximately one half. From the efficiency tests, MHD is superior to HD but not to solving the whole domain.

Here, the second modification is to repeatedly apply the combination of red-black coloring and the Schur complement. It will be used in the factorization of diagonal blocks of $\mathbf{S}_{HH}$, denoted by $\mathbf{S}_{HH,ii}$, and the reduced system, $\mathbf{S}'_{VV}$, of interface Schur complement. This modification will be referred to as MHDRB. The efficiency tests are also similar to those in Chapters 4 and 5.

6.1 One-dimensional problem and red-black coloring

In the one-dimensional problem of 16 nodes (see Figure 6.1), these nodes are related only in one direction. As an example, if the finite difference approach is applied, we will obtain the system of equations,

$$\mathbf{A}_1\mathbf{x}_1 = \mathbf{b}_1, \tag{6.1}$$

where $\mathbf{A}_1$ is the coefficient matrix, $\mathbf{x}_1$ is the unknown vector, and $\mathbf{b}_1$ is the vector associated with the boundary. The subscript 1 indicates a one-dimensional problem and will be omitted for simplicity. The coefficient matrix $\mathbf{A}$ is a tridiagonal matrix, as shown in Figure 6.2a. The diagonal elements are self-coupling coefficients and the off-diagonal elements represent coupling between neighboring nodes. Direct solvers, e.g., Gaussian elimination, LU-factorization, can solve this system.

However, if these nodes are alternately colored red and black (see Figure 6.1), the system of equations is consequently reordered so that the reds come before the blacks. It then becomes the 2×2 block system (see Figure 6.2b),

$$\begin{pmatrix} \mathbf{A}_{RR} & \mathbf{A}_{RB} \\ \mathbf{A}_{BR} & \mathbf{A}_{BB} \end{pmatrix} \begin{pmatrix} \mathbf{x}_R \\ \mathbf{x}_B \end{pmatrix} = \begin{pmatrix} \mathbf{b}_R \\ \mathbf{b}_B \end{pmatrix}, \tag{6.2}$$

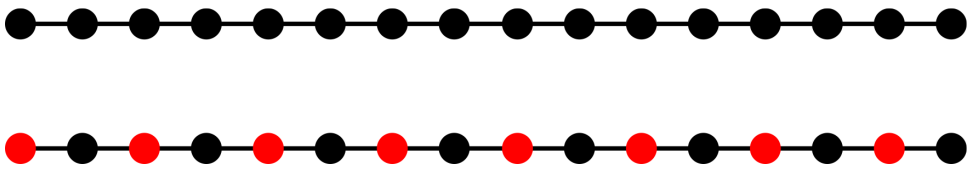
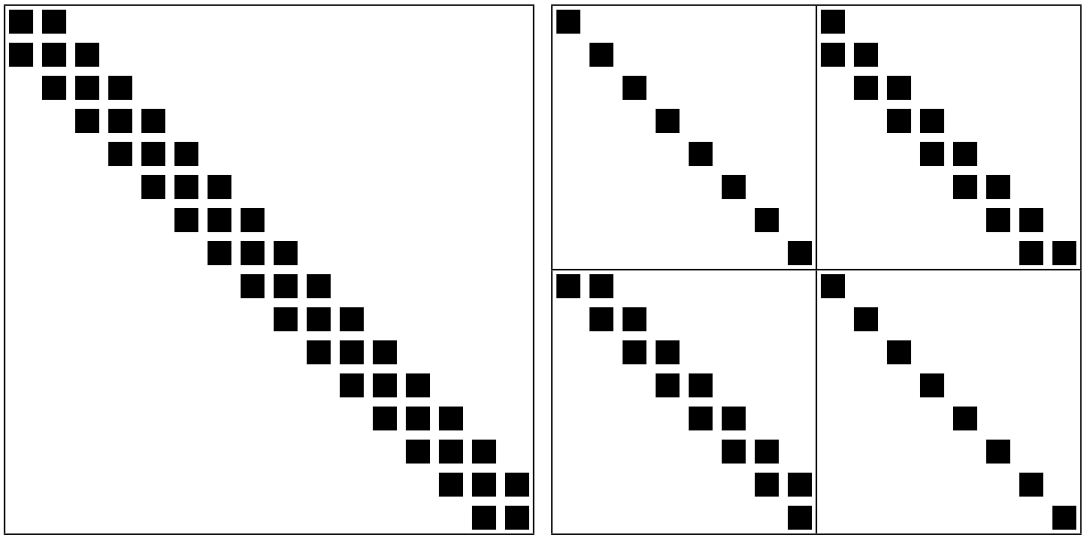


Figure 6.1: One-dimensional problem of 16 nodes, which are originally black-colored nodes (top). These nodes are then alternately colored red and black (bottom).

where the subscripts R and B represent the red-colored and black-colored nodes, respectively. $\mathbf{A}_{RR}$ and $\mathbf{A}_{BB}$ are the coefficient matrices of the reds and blacks, respectively. $\mathbf{A}_{RB}$ is the contribution from the blacks to the reds, and the reciprocal contribution is $\mathbf{A}_{BR}$. $\mathbf{A}$ is a symmetric matrix and so $\mathbf{A}_{BR} = \mathbf{A}_{RB}^T$. $\mathbf{b}_R$ and $\mathbf{b}_B$ are the corresponding right-hand side vectors.



(a) (b)

Figure 6.2: (a) The tridiagonal matrix $\mathbf{A}$ is obtained from one-dimensional problem having 16 nodes (b) After applying red-back coloring, the coefficient matrix $\mathbf{A}$ is re-ordered according to colors so that coefficients of red-colored nodes come first and end with those of black-colored nodes. The sparsity pattern of $\mathbf{A}$ becomes a 2×2 block matrix.

The system of equations (6.2) can be solved through the reduced system (Saad, 2003) by repeating the standard Schur complement, equation (4.10) to (4.16), as shown in the following. The unknown vector $\mathbf{x}_B$ is first solved from

$$\mathbf{A}'_{BB}\mathbf{x}_B = \mathbf{b}'_B, \tag{6.3}$$

where the reduced system, Schur complement, of this system $\mathbf{A}'_{BB}$ and its right-hand side $\mathbf{b}'_B$ are defined by

$$\mathbf{A}'_{BB} = \mathbf{A}_{BB} - \mathbf{A}_{BR}\mathbf{A}_{RR}^{-1}\mathbf{A}_{RB}, \tag{6.4}$$

$$\mathbf{b}'_B = \mathbf{b}_B - \mathbf{A}_{BR}\mathbf{A}_{RR}^{-1}\mathbf{b}_R. \tag{6.5}$$

The unknown vector $\mathbf{x}_R$ is then obtained by solving

$$\mathbf{A}_{RR}\mathbf{x}_R = \mathbf{b}_R - \mathbf{A}_{RB}\mathbf{x}_B, \tag{6.6}$$

where $\mathbf{A}_{RR}$ is earlier factorized in forming the reduced system, equation (6.4).

To fully exploit the combination of red-black coloring and the Schur complement, red-black coloring is reapplied by redefining the black-colored nodes with red-black coloring. This results in a smaller 2×2 block system. Thus, solving through the reduced system could be reapplied by repeating equations (6.4) to (6.6). Moreover, the reducing step could be repeatedly applied so that there is only one black-colored node left, as demonstrated by the diagram in Figure 6.3.

The coefficient matrix $\mathbf{A}$ in each reduction step is shown in Figure 6.4. In the first reduction, the 16 nodes problem is reduced to an 8 nodes problem. The blue elements are then filled to bottom right block (see Figure 6.4b), according to the definition of the reduced system $\mathbf{A}'_{BB}$ (6.4). The blue elements could be considered as the coupling between the black nodes through the red nodes. Due to the filled-in elements, the bottom right block becomes a tridiagonal matrix again, which resembles the original system. The second reduction is then applied, which reduces the problem size from 8 nodes to 4 nodes. As with the blue elements, the green elements are then filled. The reduction step is applied so that the problem is reduced from 4 to 2 and eventually 1, where the last node is a black-colored node. Note that there is no filled-in element at the last reduction step.

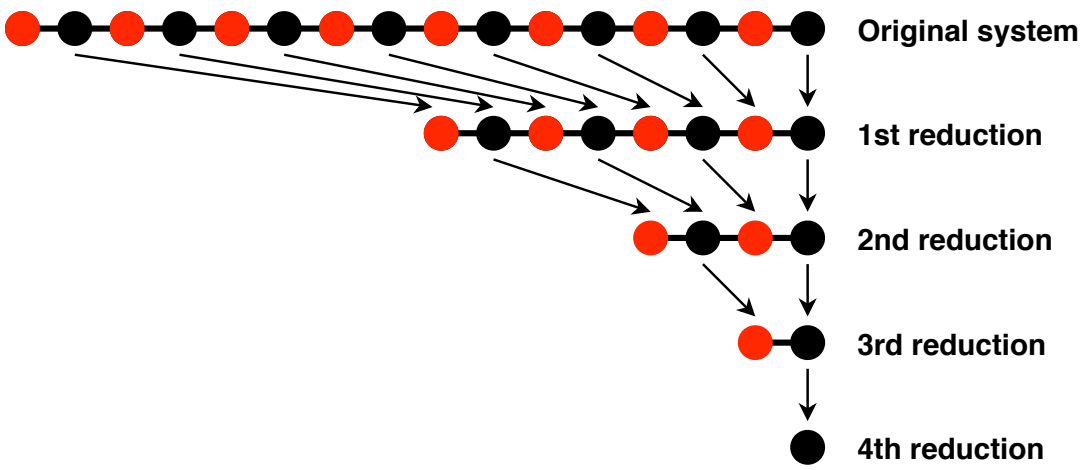
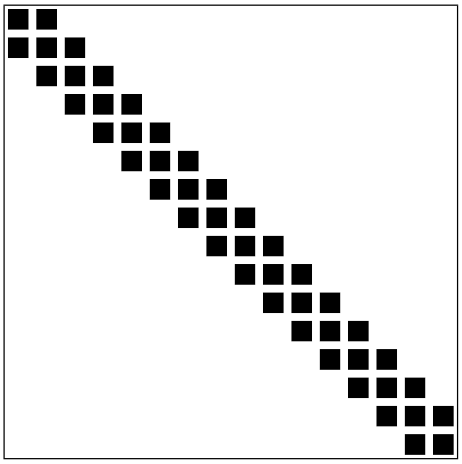
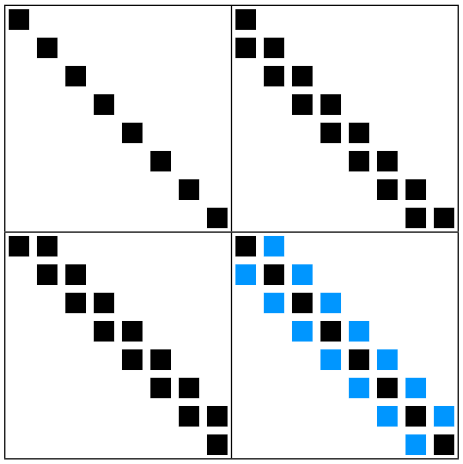


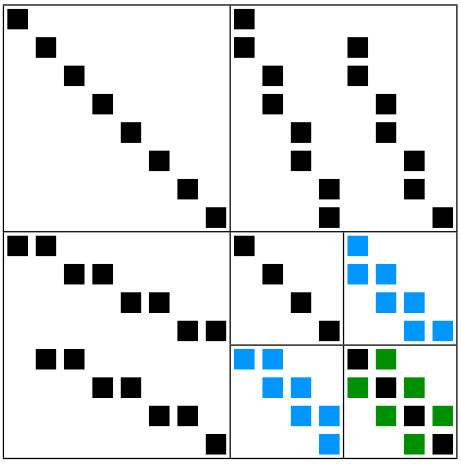
Figure 6.3: The problem is reduced by redefining the black-colored nodes as reds and blacks.



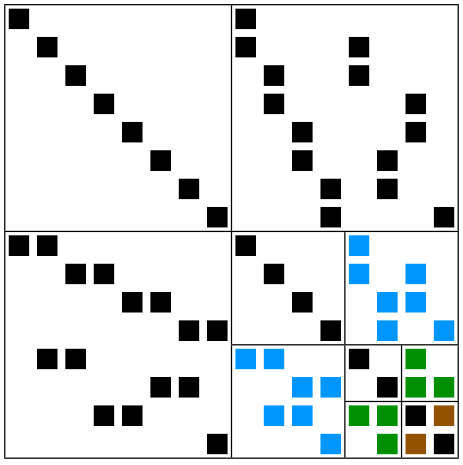
(a) Original system



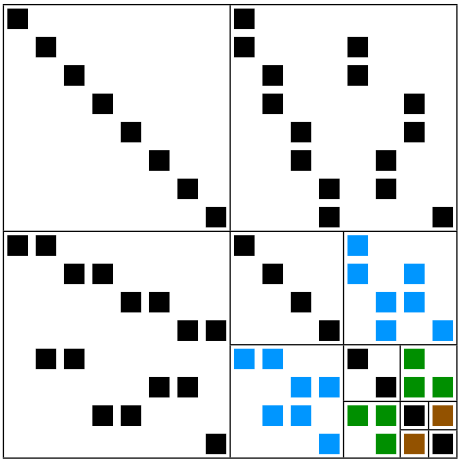
(b) 1st reduction



(c) 2nd reduction



(d) 3rd reduction



(e) 4th reduction

Figure 6.4: The coefficient matrix $\mathbf{A}$ of size 16×16 from the original problem and matrices from each reduction step are shown. Blue, green and brown elements are filled-in elements in the reduction steps 1, 2 and 3, respectively.

6.2 Block matrix reduction by implementing red-black coloring and the Schur complement

In modified hierarchical domain decomposition, $\mathbf{S}_{HH}$ and the reduced system $\mathbf{S}'_{VV}$ are block matrices. Taking a closer look, diagonal blocks of $\mathbf{S}_{HH}$ ($\mathbf{S}_{HH,ii}$) and matrix $\mathbf{S}'_{VV}$ share the block tridiagonal pattern, as illustrated in Figure 5.2a and 5.4. This pattern resembles the coefficient matrix $\mathbf{A}$ from the one-dimensional problem, but in a form of a block matrix, not an element. This pattern will be clearly observed when the number of sub-domains increases. For example, the block pattern of $\mathbf{S}_{HH}$ and $\mathbf{S}'_{VV}$ from 8×9 partitioned model is shown in Figure 6.5. Because of their block tridiagonal pattern, factorizing $\mathbf{S}_{HH,ii}$ and $\mathbf{S}'_{VV}$ (Line 2 and 5 in Algorithm 3) can be computed through a series of smaller sub-systems by repeatedly applying the combination of red-black coloring and the Schur complement. This modification will be referred to as MHDRB.

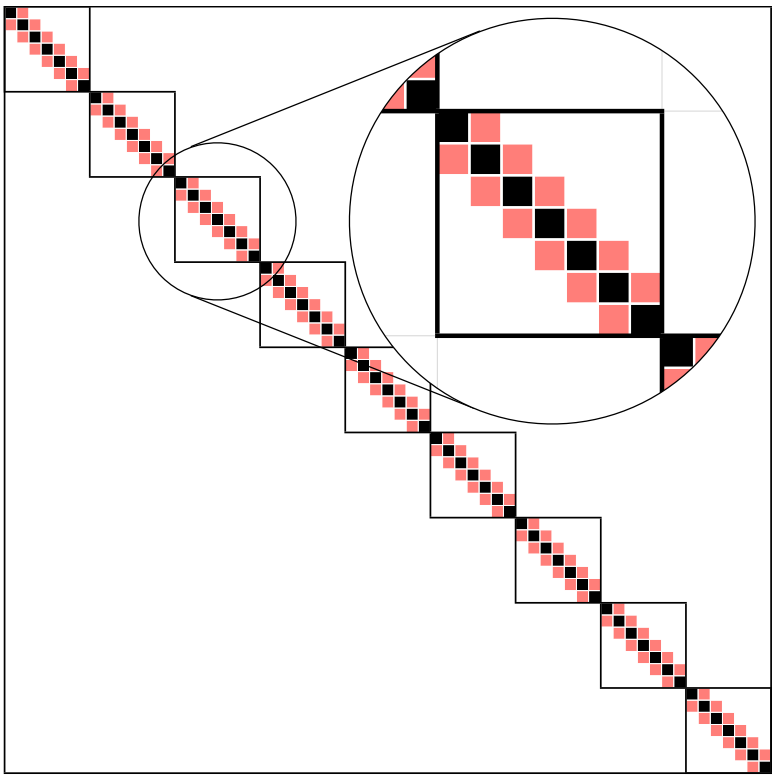
When the computational domain is partitioned into 8×9 sub-domains, the combination of red-black coloring and the Schur complement will be applied to nine $\mathbf{S}_{HH,ii}$ matrices and one $\mathbf{S}'_{VV}$ matrix. If LU-factorization is directly applied as MHD, their null matrices will be included. The sub-block dimension of each $\mathbf{S}_{HH,ii}$ is 7×7 and the block dimension of $\mathbf{S}'_{VV}$ is 8×8. As in Figure 6.4 in the previous section, the 7×7 and 8×8 block matrix reductions are demonstrated in Figures 6.6 and 6.7, respectively. Additional fill-in block matrices are 4 and 10 for 7×7 and 8×8 block matrices, respectively.

Instead of storing all sub-blocks of $\mathbf{S}_{HH,ii}$ like in HD, there are only the original sub-blocks and some additional filled-in blocks to be stored, and similarly for $\mathbf{S}'_{VV}$. The number of filled-in blocks could be counted from the reduction step, e.g., from Figures 6.8 and 6.9. However, storing $\mathbf{S}_{HV}$ and $\mathbf{S}_{VH}$ remains the same as in MHD.

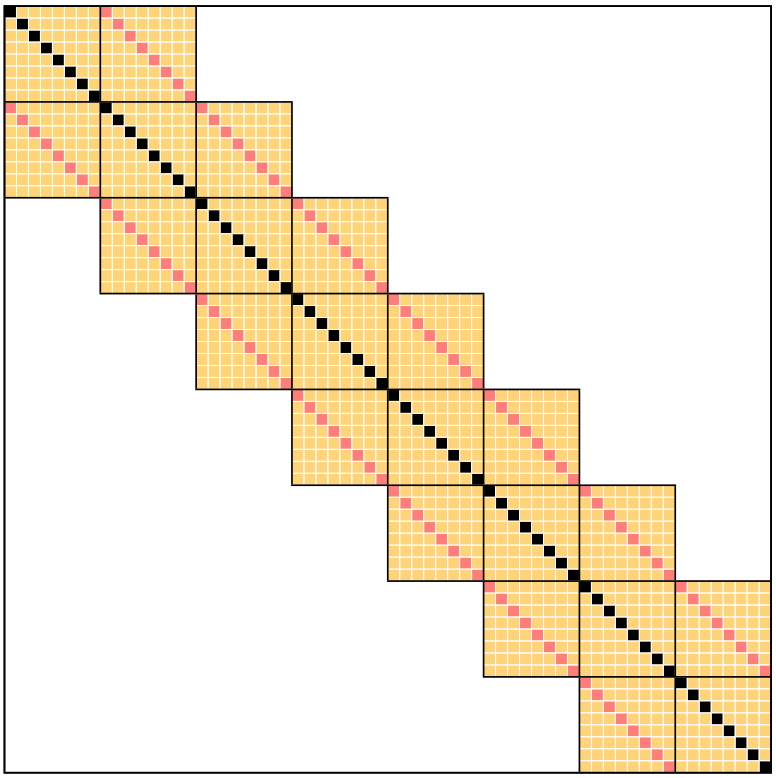
For example, the model with size of 120×360s partitioned into 4×4 sub-domains will use 22% less memory than MHD for storing $\mathbf{S}_{HH}$ and $\mathbf{S}'_{VV}$, where the numbers of elements in $\mathbf{S}_{HH}$ and $\mathbf{S}'_{VV}$ are shown in Table 6.1. The 3×3 and 4×4 block reductions are shown in Figure 6.8 and 6.9, respectively. However, this number does not correctly represent the overall memory reduction. The total number of elements for MHDRB can be approximately obtained from (6.7).

Matrix	Number of elements
$\mathbf{S}_{HH}$	$4 \times (7+0) \times 89^2$
$\mathbf{S}'_{VV}$	$(7+0) \times 4^2 \times 29^2$
Total	315,980

Table 6.1: Number of elements stored for the interface Schur complement in MHDRB when the model size is 120×360 and the model is partitioned into 4×4 sub-domains.

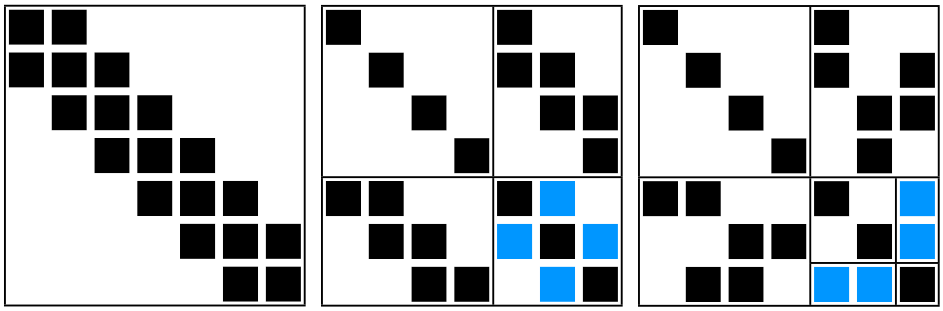


(a)



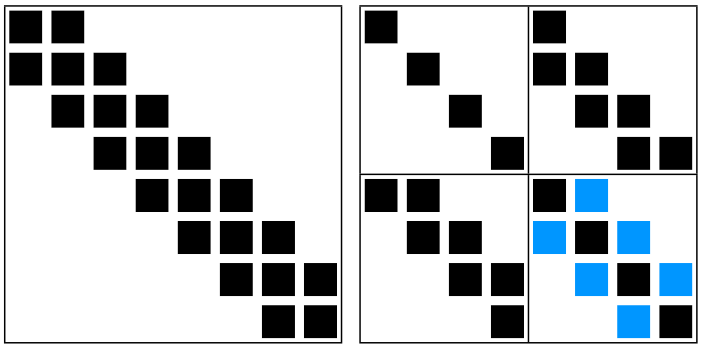
(b)

Figure 6.5: (a) The block pattern of diagonal blocks of $\mathbf{S}_{HH}$. (b) Matrix $\mathbf{S}'_{VV}$ from 8×9 partitioned model. Both have the same pattern of block tridiagonal matrix.

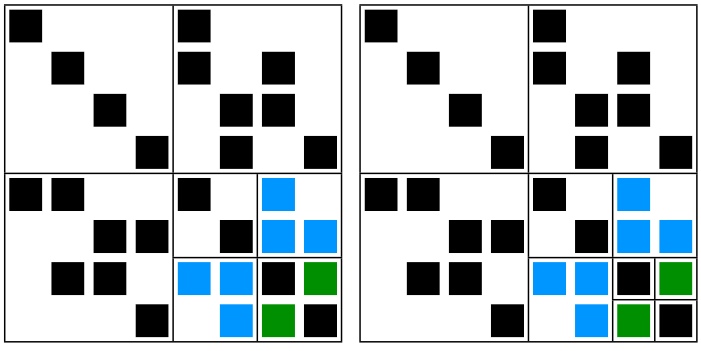


(a) Original system (b) 1st reduction (c) 2nd reduction

Figure 6.6: Original block system with 7×7 blocks repeatedly applied with the combination of red-black coloring and the Schur complement. Blue blocks are filled in due to the definition of the reduced system.



(a) Original system (b) 1st reduction



(c) 2nd reduction (d) 3rd reduction

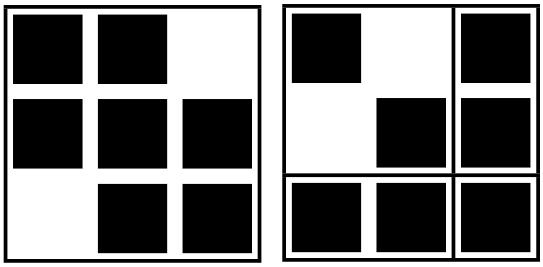
Figure 6.7: Original block system with 8×8 blocks repeatedly applied with the combination of red-black coloring and the Schur complement. Blue and green blocks are filled in due to the definition of the reduced system.

$$
\begin{aligned}
\text{No. of elements}_{\text{MHDRB}} \approx & 15 + \frac{65}{3}M_Y + 2M_Z - 3M_Y M_Z - \frac{25}{3}M_Z^2 \\
& - \frac{1}{P_Y^2}M_Y^2 \\
& - \frac{1}{P_Y}(2M_Y + \frac{28}{3}M_Y^2 + M_Y M_Z) \\
& + P_Y(\frac{-82}{3} + M_Y + 9M_Z + \frac{13}{3}M_Z^2) \\
& - P_Y^2 M_Z \\
& + \frac{1}{P_Z}(-3M_Z - 4M_Y M_Z - M_Z^2 + 3M_Y M_Z^2) \\
& + P_Z(-16 - \frac{8}{3}M_Y + \frac{59}{3}M_Z + 2M_Y M_Z) \\
& - P_Z^2(\frac{31}{3} + 3M_Y) \\
& + \frac{1}{P_Y^2 P_Z}M_Y^2 M_Z \\
& + 2\frac{1}{P_Y P_Z}M_Y M_Z \\
& + \frac{P_Y}{P_Z}(4M_Z - 2M_Z^2) \\
& + \frac{P_Z}{P_Y}(M_Y + \frac{16}{3}M_Y^2 - 2M_Y M_Z) \\
& + P_Y P_Z(\frac{28}{3} - 2M_Y - \frac{38}{3}M_Z) \\
& + P_Y^2 P_Z(1 + M_Z) \\
& + 2\frac{P_Z^2}{P_Y}M_Y \\
& + P_Y P_Z^2(\frac{22}{3} + M_Y) \\
& - P_Y^2 P_Z^2
\end{aligned}
\tag{6.7}
$$

In contrast to MHD, $\mathbf{S}_{HH,ii}$ and $\mathbf{S}'_{VV}$ now are block matrices. Thus, applying the combination of the red-black ordering and the Schur complement could be performed using the block matrix operation. Their null matrices will be excluded from storing and calculation. Consequently, this results in lower computational loads than HD, and hopefully the CPU time will be better than FDWD.

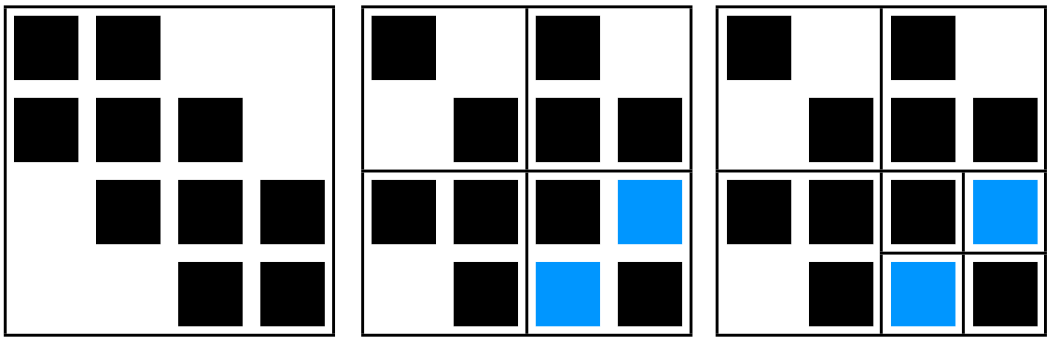
6.3 Numerical efficiency

Because of the second modification, the interface Schur complement system is decomposed into many smaller systems. The block structures of $\mathbf{S}_{HH}$ and $\mathbf{S}'_{VV}$ share the same tridiagonal block matrix. The combination of red-black coloring and the Schur complement could then be applied. As with the first modification, MHDRB does not



(a) Original system (b) 1st reduction

Figure 6.8: Original block system with 3×3 blocks is repeatedly applied with the combination of red-black coloring and the Schur complement, where no additional fill-in blocks are required.



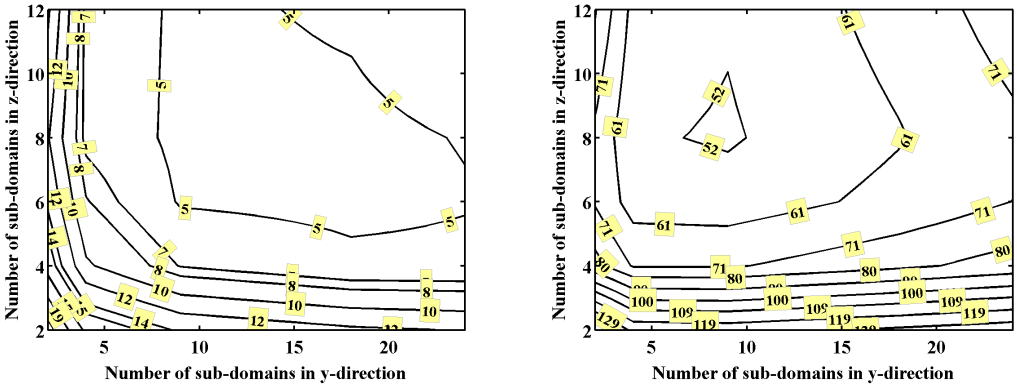
(a) Original system (b) 1st reduction (c) 2nd reduction

Figure 6.9: Original block system with 4×4 blocks repeatedly applied with the combination of red-black coloring and the Schur complement. Blue blocks are filled in due to the definition of the reduced system.

change the system of equations. The same level of accuracy in solutions is therefore obtained. To avoid redundancy, the responses from MHDRB are omitted.

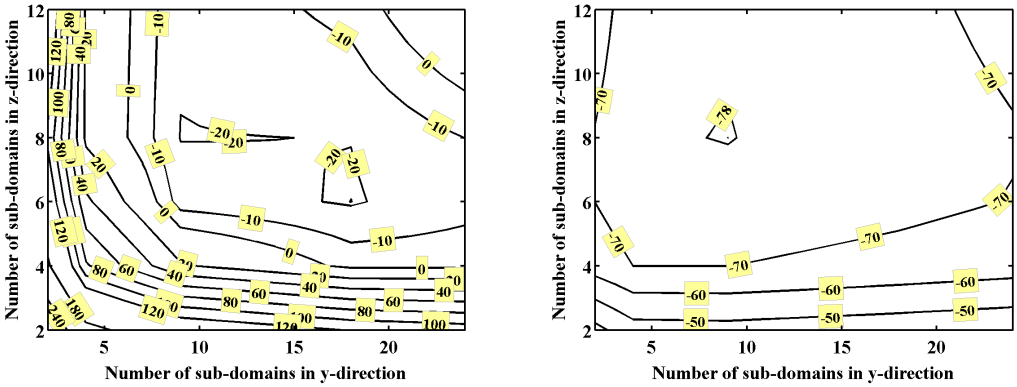
The contour maps of actual CPU time and memory are shown in Figure 6.10 and their relative values in Figure 6.11. The CPU time for MHDRB is much less than for MHD, and the memory used is almost the same. At 8×9 and 6×18 partitionings, MHDRB uses about 20% less CPU time than FDWD. The minimum memory is obtained at 8×9 partitioning. This indicates that MHDRB is superior to FDWD.

In contrast to HD and MHD, the plot of time steps in MHDRB, as shown in Figure 6.12, shows a less steep curve of forming and factorizing the interface Schur complement (triangles) for large numbers of sub-domains. Forming and factorizing the interface Schur complement and solving for the intersections (diamonds) become comparable at larger numbers of sub-domains. This is a result of fully decomposing the interface Schur complement. LU factorization of $\mathbf{F}_{ii}$ (circles), solving for the interfaces (squares) and the interiors (stars) are similar to what was described in Section 4.4.2.



(a) (b)

Figure 6.10: (a) Actual CPU time and (b) memory used by MHDRB to solve 120×360 model at various numbers of sub-domains.



(a) (b)

Figure 6.11: (a) Relative CPU time and (b) memory maps corresponding to Figures 6.10a and 6.10b, respectively.

6.4 Summary

Because of its block pattern, the interface Schur complement is fully decomposed by repeatedly applying the combination of red-black coloring and the Schur complement to the diagonal blocks of $\mathbf{S}_{HH}$ ($\mathbf{S}_{HH,ii}$) and the matrix $\mathbf{S}'_{VV}$. The factorization of them could be computed through a series of smaller systems instead. The computational time and memory are then reduced, because the null matrices in $\mathbf{S}_{HH,ii}$ and $\mathbf{S}'_{VV}$ are no longer included. This is similar to MHD but at different scale. MHD excludes the null block matrices, while MHDRB excludes the null sub-block matrices, which are at the finer scale. In addition to the memory requirement, the second modification is superior to FDWD in terms of CPU time.

The efficiency on larger problems and pre-selecting the optimal numbers of sub-domains of the developed domain decomposition technique will be further discussed

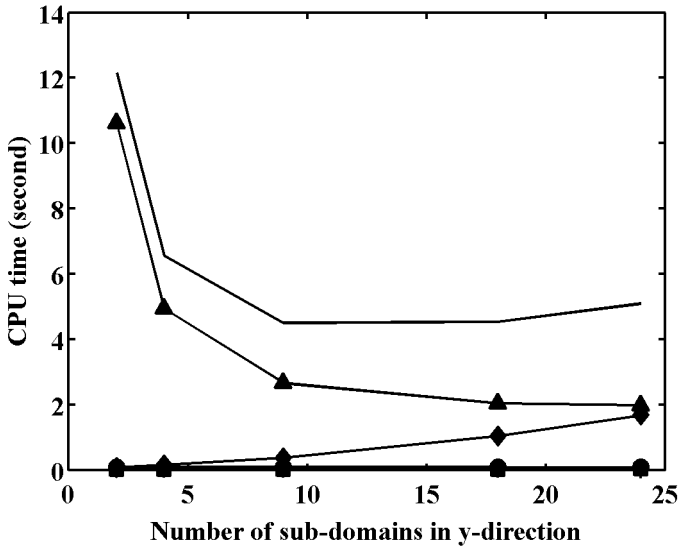


Figure 6.12: Overall time of MHDRB (topmost solid line) consists of five steps: LU factorization of $\mathbf{F}_{ii}$ (circles), forming and factorizing interface Schur complement $\mathbf{S}$ (triangles), solving for the intersections (diamonds), solving for the interfaces (squares) and solving the interiors sub-problems for interiors (stars).

in the next chapter.

CHAPTER VII
POTENTIAL OF DEVELOPED DOMAIN DECOMPOSITION TECHNIQUES

Starting from the hierarchical domain decomposition, a modification was made so that the developed method is superior to solving the whole domain, as demonstrated in Chapters 4, 5 and 6. In this chapter, to demonstrate that the developed method has the potential to solve the large problems, the numerical efficiency will be determined as a function of the number of unknowns. The results of applying the three domain decomposition solvers, HD, MHD and MHDRB, are also compared. Finally, the strategy of choosing the numbers of sub-domains that provide the optimal computational time or memory is presented.

7.1 Solving larger problems

To show the potential of the domain decomposition solvers for solving large problems, the size of computational domain is varied. The same model, two-block model, was used and it was discretized into three different meshes: 80×240, 120×360 and 160×480. Their numbers of unknowns are shown in Table 7.1. Each mesh uses a different combination of numbers of sub-domains $P_Z \times P_Y$. All three domain decomposition solvers, HD, MHD and MHDRB, are then applied to solve the problems.

Mesh	Mesh size	Number of unknowns	P_Z	P_Y
Mesh 1	80×240	18,881	2, 4, 8, 10 and 16	2, 5, 10, 15 and 24
Mesh 2	120×360	42,721	2, 4, 6, 8 and 12	2, 4, 9, 18 and 24
Mesh 3	160×480	76,161	2, 4, 8, 10 and 16	2, 4, 8, 10, 16 and 24

Table 7.1: The two-block model discretized into three different meshes: 80×240, 120×360 and 160×480. P_Z and P_Y are the numbers of sub-domains in z- and y-directions specific to each mesh.

Table 7.2 shows CPU times and memory usages for three meshes from FDWD and the three domain decomposition solvers, HD, MHD and MHDRB. As shown

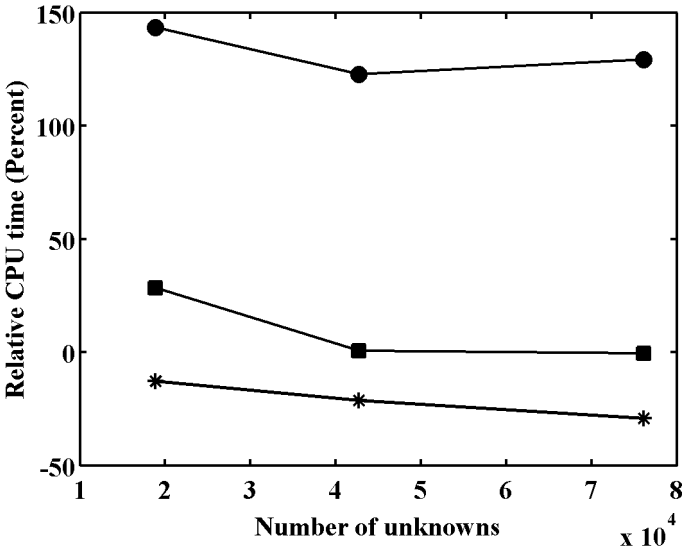
Mesh	Solver	Time (seconds)		Memory (MByte)	
Mesh 1	FDWD	1.12		68.9	
	HD	2.74	at 4×5	31.1	at 4×2
	MHD	1.44	at 4×10	23.0	at 4×5
	MHDRB	0.98	at 8×10	18.3	at 8×5
Mesh 2	FDWD	5.59		234.0	
	HD	12.45	at 4×4	91.6	at 4×4
	MHD	5.63	at 6×9	64.6	at 6×4
	MHDRB	4.37	at 8×9	51.0	at 8×9
Mesh 3	FDWD	15.46		556.7	
	HD	35.45	at 4×4	200.4	at 4×4
	MHD	15.38	at 8×8	136.5	at 8×8
	MHDRB	10.88	at 8×16	102.7	at 10×8

Table 7.2: Computational loads from FDWD, HD, MHD and MHDRB. The minimum CPU time and memory consumption from domain decomposition solvers is provided at different partitioning $P_Z \times P_Y$.

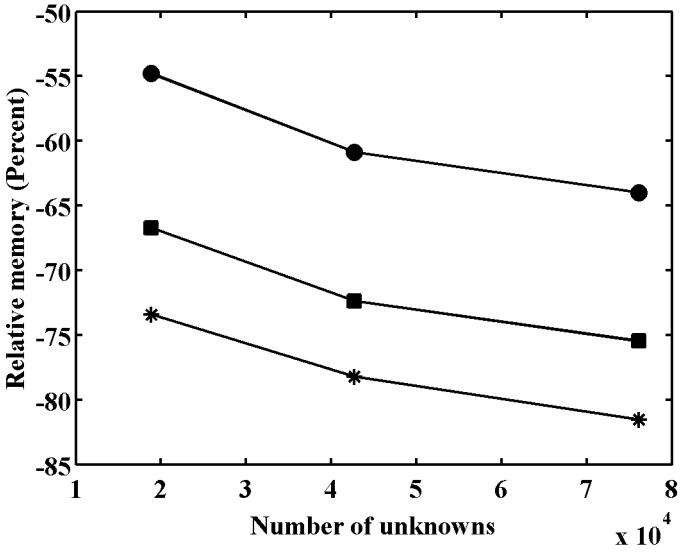
in the previous section, MHDRB is superior to FDWD at the proper numbers of sub-domains, while HD and MHD are not. Note that the partitioned models that provide the minimum CPU time and minimum memory slightly differ in the number of sub-domains. Next, the relative CPU time and memory of each solver are calculated and then plotted as a function of the number of unknowns, as shown in Figure 7.1.

MHD greatly reduces the CPU times compared to HD, while the reduction of CPU time due to MHDRB is smaller. Moreover, MHDRB is superior to FDWD for all problem sizes. However, none of the three solvers show a significant difference in memory requirements, as shown in Figure 7.1b. This might be because no global matrix is declared when the model is decomposed. Thus, the memory saved by the modifications become very small compared to the diminished global matrix. The design curves, the optimal number of sub-domains plotted as a function of number of unknowns, for minimum time and memory are shown in Figure 7.2.

When the problem size is larger, all three domain decomposition solvers show higher numerical efficiency, less CPU time and memory usage. MHDRB makes a saving of up to 80% on memory usage and uses 30% less CPU time. This indicates the potential of MHDRB for solving large problems, particularly three-dimensional problems, on the decomposed domain.

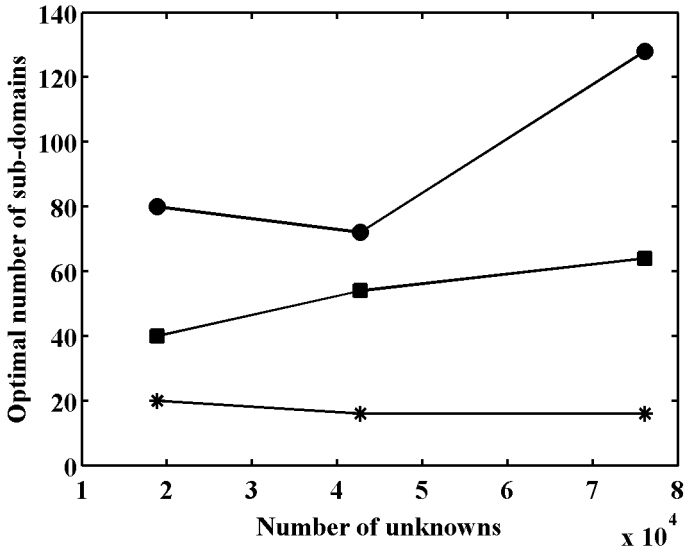


(a)

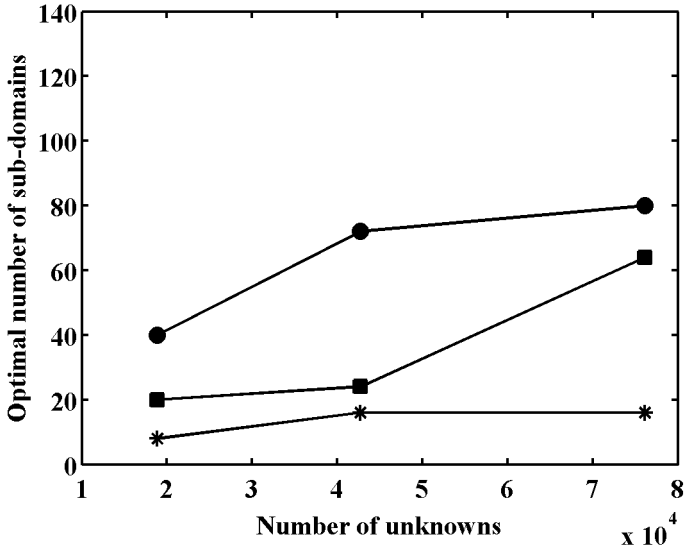


(b)

Figure 7.1: (a) Relative CPU time and (b) relative memory from HD (circles) MHD (squares) and MHDRB (stars) at three different numbers of unknonws.



(a)



(b)

Figure 7.2: The design curves for (a) minimum CPU time and (b) minimum memory from HD (circles) MHD (squares) and MHDRB (stars).

7.2 Numerical efficiency comparison of three domain decomposition solvers

Applying HD, MHD and MHDRB results in a changing pattern of relative CPU time and memory usage, as shown in Figures 7.3, 7.4 and 7.5, respectively. The contour maps in Figure 7.4 are the same as in Figures 4.19, 5.7 and 6.11. The interesting regions in the CPU time map are labeled with letters. The corresponding regions in the memory map are labeled with primed letters.

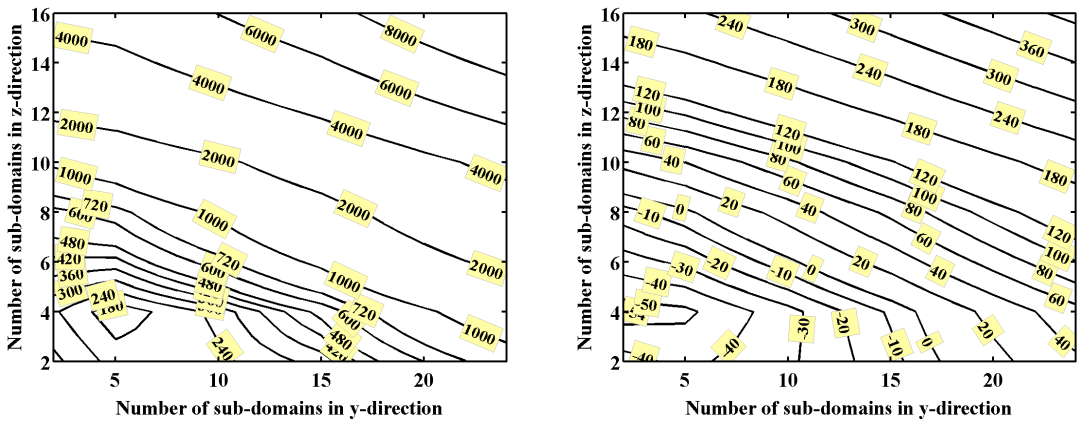
From Figure 7.4a, the minimized zone M_1 of HD is limited to the small numbers of sub-domains. So they are located near the bottom left of the graph. In addition, the minimized zone is quite small.

At small numbers of sub-domains the interiors are large, but the interfaces are small. Therefore, the size of the interface Schur complement S is small. However, the high computational loads in Region A are mostly caused by forming the interface Schur complement because the definition of the interface Schur complement, equation (4.24), requires multiplication with $\mathbf{D}$ and its transpose, the contribution from interior to interfaces and its reciprocal, and their sizes depend on both the numbers of interiors and interfaces. Beyond this region, the computational loads tend to go higher at the top right corner, Region B. This is because of the increasing number of interfaces which is consistent with the red zone in Figure 4.17b.

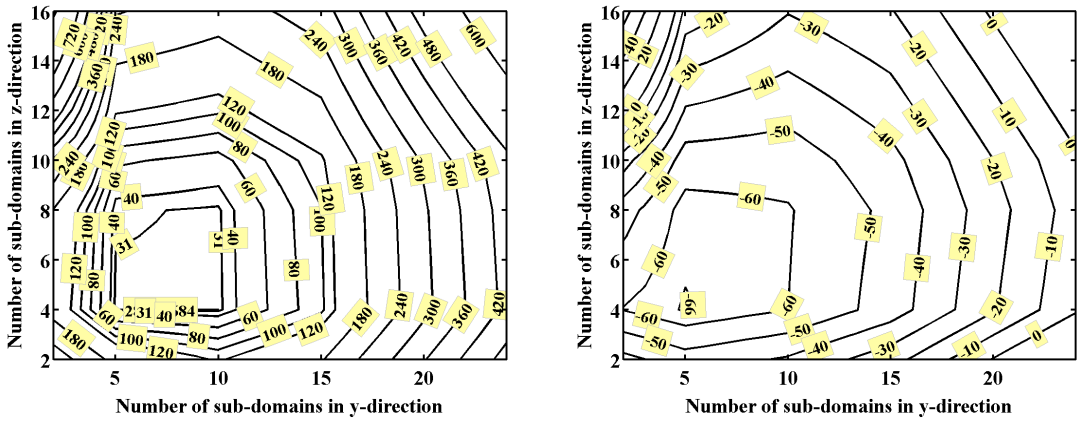
When MHD is applied, the minimized zone M_2 is slightly shifted away from the left because the Region B in Figure 7.4a, which depends on the interfaces, is less severe. This indicates that the huge computational load due to the interface Schur complement $\mathbf{S}$ is reduced. However, the trend of increasing CPU time and memory in Region E of Figure 7.4b is proportional to the vertical interfaces, and increases as the number of sub-domains in the y-direction increases (see Figure 5.5b). This is due to the reduction of the interface Schur complement system $\mathbf{S}$ to the reduced interface system $\mathbf{S}'_{VV}$, which corresponds to the vertical interfaces only. As with Region A in Figure 7.4a, Region C is caused by the large size of the sub-domains, but is relatively higher than Region A. The relative CPU times become larger because the overall times of MHD are less than those of HD.

The minimized zone M_3 of MHDRB is further away because the reduced interface system $\mathbf{S}'_{VV}$ is broken into many smaller sub-matrices. Region E, which constrains the minimized zone M_2 in MHD (Figure 7.4b), is then diminished. Regions F, G and H then become higher relative to MHD.

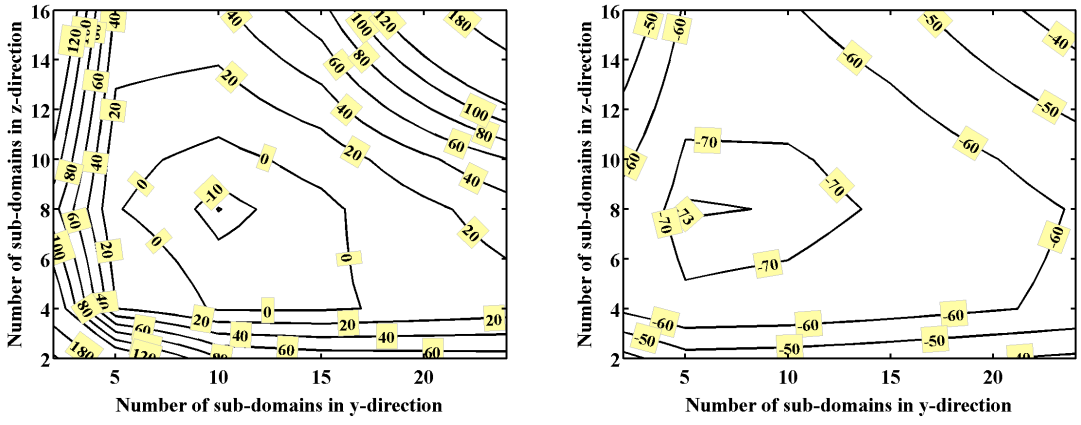
The corresponding minimized zones, M'_1, M'_2 and M'_3, and the corresponding regions, A', B', C', D', E', F', G' and H', in the relative memory map behave in similar way, but are in slightly different positions.



(a)

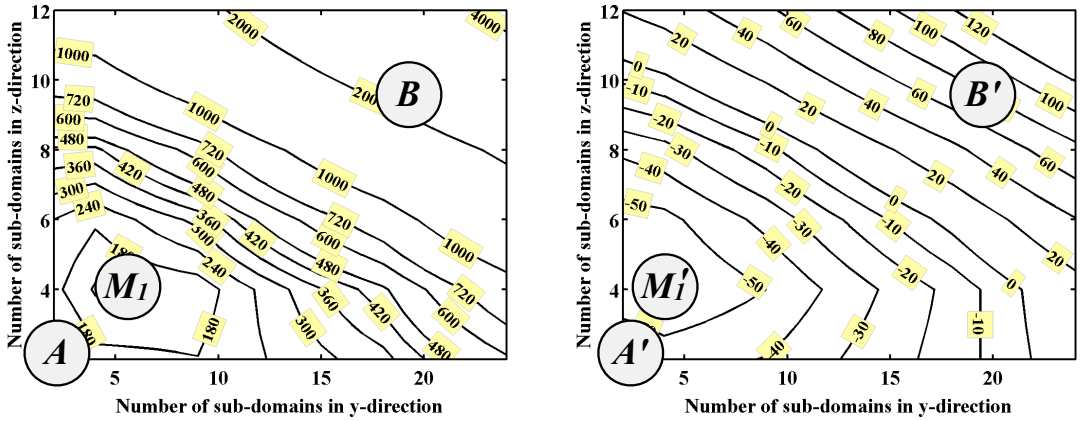


(b)

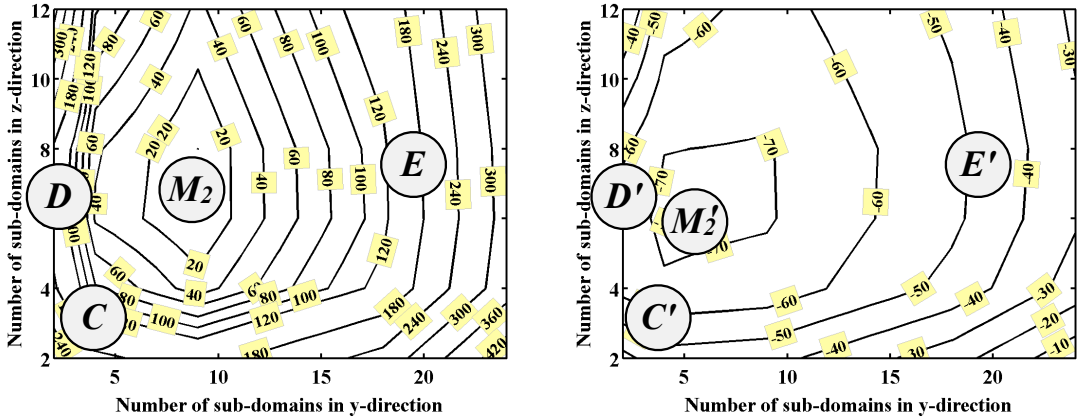


(c)

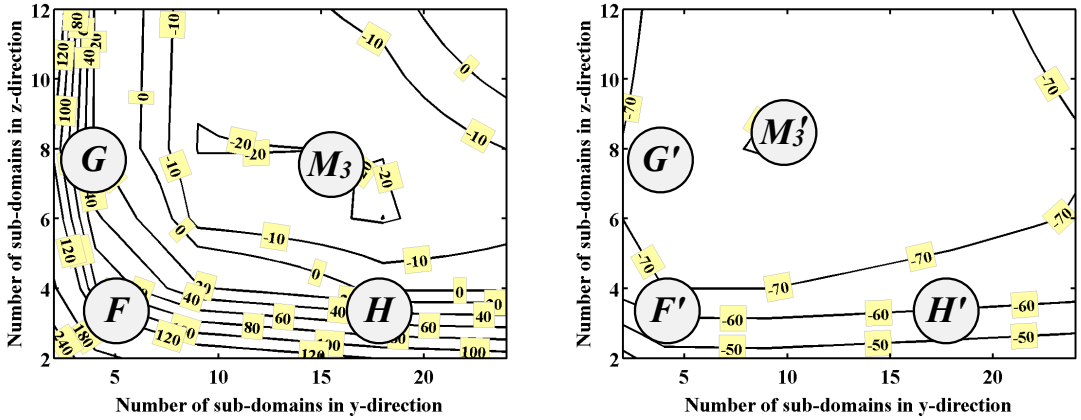
Figure 7.3: The maps of relative CPU time (left) and relative memory (right) obtained from using (a) HD, (b) MHD and (c) MHDRB to solve a model with an 80×240 grid.



(a)

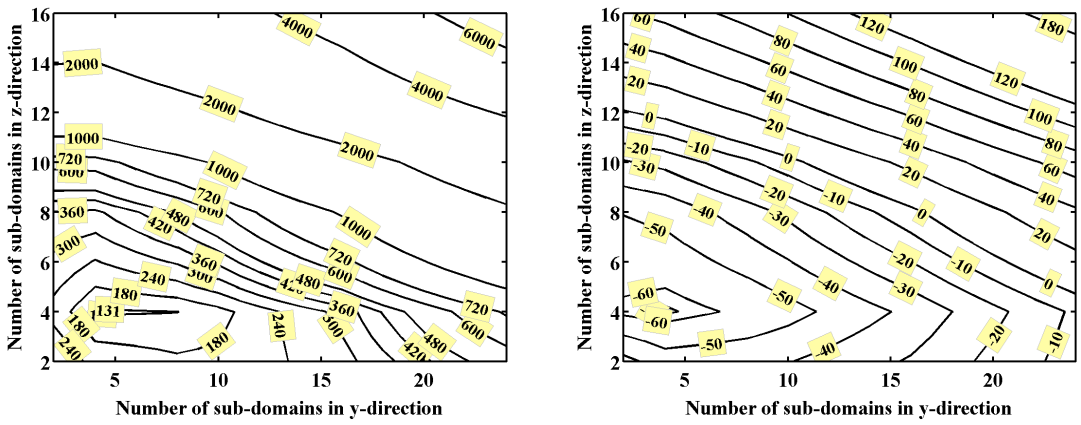


(b)

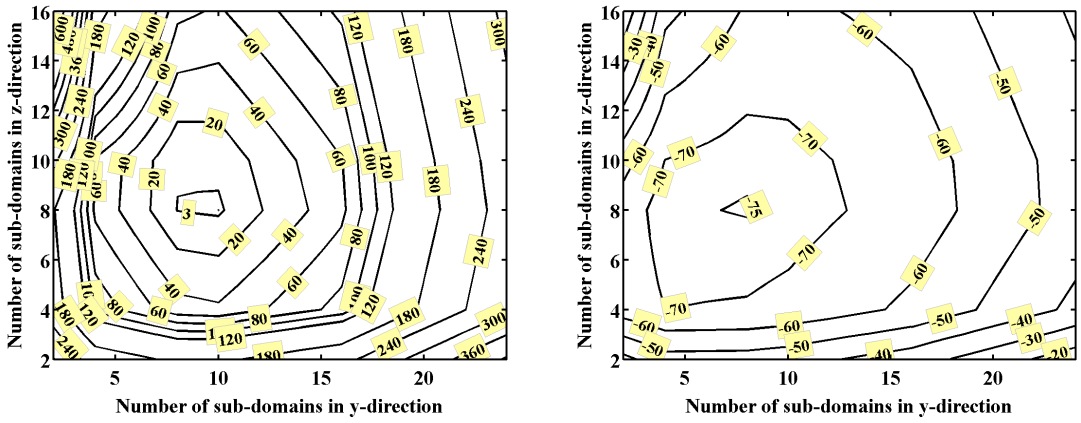


(c)

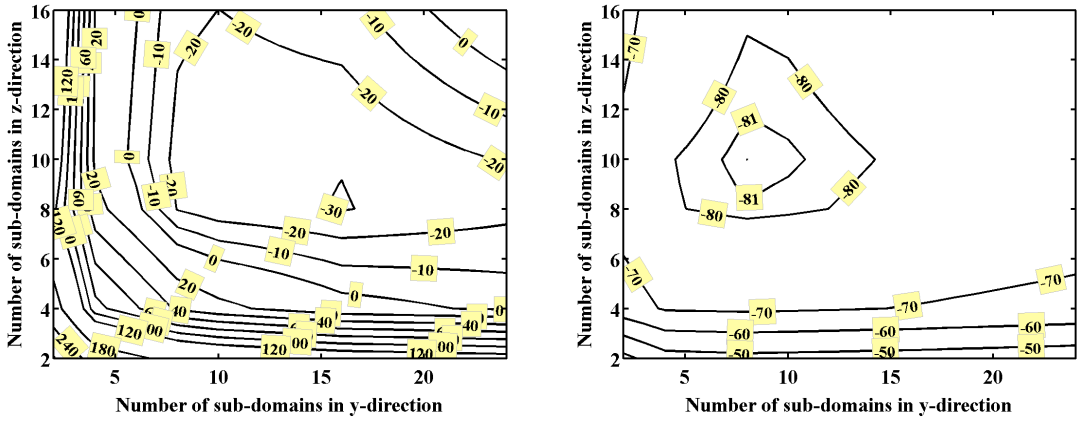
Figure 7.4: The maps of relative CPU time (left) and relative memory (right) obtained from using (a) HD, (b) MHD and (c) MHDRB to solve a model with a 120×360 grid. They are the same as Figures 4.19, 5.7 and 6.11. Labels represent the minimized zones, M_1, M_2 and M_3, and the zone with high computational loads, A to H. The zones, A' to H' and M_1', M_2' and M_3', on the relative memory maps correspond to those on the relative time maps.



(a)



(b)



(c)

Figure 7.5: The maps of relative CPU time (left) and relative memory (right) obtained from using (a) HD, (b) MHD and (c) MHDRB to solve a model with a 160×480 grid.

7.3 Pre-selection of the optimal number of sub-domains

The important question in applying domain decomposition is what is the suitable number of sub-domains? This is because the wrong number of sub-domains leads to an imbalance between the numbers of interior nodes and interfaces. Too many sub-domains results in increasing interface problems, while too few would increase the size of the interior sub-problems.

In this thesis, the strategy of choosing the proper numbers of sub-domains is provided by the maps of relative time and memory. These maps show the changing in computational load, as a function of the numbers of sub-domains. The important feature observed from these maps is the regions of minimized computational time which are located close to these of memory, as shown in Figures 7.3, 7.4 and 7.5. This might be because the number of interiors interfaces and intersections are justified so that solving and storing interior sub-problems $\mathbf{F}_{ii}$ and the reduced systems, $\mathbf{S}$ and $\mathbf{H}'$, are relatively fast and have a lower memory requirement. In addition, the number of sub-domains in the z- and y-directions also control the block pattern of the interface Schur complement, which is a crucial point in applying the combination of red-black coloring and the Schur complement.

In contrast to the CPU time map, the memory map can be generated prior to actual calculations. The agreement between CPU time and memory usage has led to the strategy of pre-selecting the number of sub-domains in the z- and y-directions. To avoid a trial and error approach, the number of sub-domains should first be chosen from the region of minimized memory and then varied around it. This will provide the chance of CPU time better than by solving the whole domain.

CHAPTER VIII
CONCLUSIONS

The domain decomposition method was developed to solve two-dimensional magnetotelluric forward problems formulated with a finite difference approach. In contrast to iterative solvers, solving the resulting system of equations with direct solvers is desirable because they provide a theoretical guarantee of accuracy in solution and the ability to solve multiple right-hand-side problems. In addition, they are not sensitive to ill-conditioned systems, which occurred when the models become geologically complex and large, particularly in the case of three-dimensional problems. In MT, the long period problems also result in ill-conditioned systems. However, the direct solvers might be impractical because of memory limitation. The natural option is therefore to break the computational domain into many smaller sub-domains. Consequently, applying direct solvers becomes reasonable.

To benefit from employing the finite different approach, hierarchical domain decomposition (HD) was chosen. The unknowns are classified into three types: interior, interface and intersection. The system of equations is then solved through many smaller interior sub-systems and two reduced systems of interfaces and intersections. However, using HD is not faster than using direct solvers but its memory consumption is greatly reduced. Another two modifications are therefore applied.

The first modification (MHD) separates the interfaces into horizontal and vertical interfaces. This breaks the interface Schur complement system and then solves through a series of smaller systems and one reduced system. The problem size of the reduced system is approximately half that of the interface system. The second modification (MHDRB) is to repeatedly apply the combination of red-black coloring and the Schur complement to those smaller systems and the reduced interface system. Both modifications exclude null matrices from being stored and calculated. Better numerical efficiency is then gained.

The accuracy of the developed method was verified by comparing both TE and TM responses with those from solving the whole domain on synthetic and inverted models at various frequencies. The responses from the developed domain decomposition method had the same level of accuracy as solving the whole domain because they solve the same system of equations. This validates the developed domain decomposition method.

The efficiency of the developed method was tested by varying the model size and the number of sub-domains. The parameters used to evaluate the efficiency are CPU time and memory compared to those obtained from solving the whole domain.

On the decomposed domain, the memory requirement is substantially diminished because the global matrix is not used. The computational time is less than that for solving the whole domain when both modifications are applied. However, all three domain decomposition solvers have higher efficiency when the size of computational domain increases. On the largest model tested, MHDRB used 80% less memory and was 30% faster. The developed domain decomposition method was thus proven to be efficient. It also shows the potential of using direct solvers to solve larger problems, especially in three-dimensional cases.

In addition, the strategy of choosing the optimal number of sub-domains was also presented. Because the minimized zone of time map is consistent with that of the memory map, which can be generated prior to the actual calculation, the optimal numbers of sub-domains can be pre-selected from the region that provides minimum memory. Hence a trial and error approach for choosing the suitable numbers of sub-domains could be avoided.

REFERENCES

Avdeev, D. & Avdeeva, A., 2009 "3D magnetotelluric inversion using a limited-memory quasi-Newton optimization," *Geophysics* **74**, F45–F57.

Basermann, A., Jaekel, U., Nordhausen, M. & Hachiya, K., 2005 "Parallel iterative solvers for sparse linear systems in circuit simulation," *Future Generation Computer Systems* **21**, 1275–1284.

Ben-Hadj-Ali, H., Operto, S., Virieux, J. & Sourbier, F., 2008 "3D frequency-domain full-waveform tomography based on a domain decomposition forward problem," *SEG Technical Program Expanded Abstracts* **27**, 1945–1949.

Benamou, J. & Despres, B., 1997 "A Domain Decomposition Method for the Helmholtz Equation and Related Optimal Control Problems," *Journal of Computational Physics* **136**, 68–82.

Bitzarakis, S., Papadrakakis, M. & Kotsopulos, A., 1997 "Parallel solution techniques in computational structural mechanics," *Computer Methods in Applied Mechanics and Engineering* **148**, 75–104.

Guo, Y., Jin, X. & Ding, J., 2006 "Parallel numerical simulation with domain decomposition for seismic response analysis of shield tunnel," *Advances in Engineering Software* **37**, 450–456.

Hautot, S., Tarits, P., Perrier, F., Tarits, C. & Trique, M., 2002 "Groundwater electromagnetic imaging in complex geological and topographical regions: A case study of a tectonic boundary in the french alps," *Geophysics* **67**, 1048–1060.

Henon, P. & Saad, Y., 2006 "A Parallel Multistage ILU Factorization Based on a Hierarchical Graph Decomposition," *SIAM Journal on Scientific Computing* **28**, 2266–2293.

Kaufman, A. A. & Keller, G. V., 1981 *The magnetotelluric sounding method*, no. 15 in Methods in geochemistry and geophysics (Elsevier Scientific Publishing Company).

Kocak, S. & Akay, H., 2001 "Parallel Schur complement method for large-scale systems on distributed memory computers," *Applied Mathematical Modelling* **25**, 873–886.

Larsson, E., 1999 "A Domain Decomposition Method for the Helmholtz Equation in a Multilayer Domain," *SIAM Journal on Scientific Computing* **20**, 1713–1731.

Lu, Y. & Shen, C., 1997 "A domain decomposition finite-difference method for parallel numerical implementation of time-dependent Maxwell's equations," *IEEE Transactions on Antennas and Propagation* **45**, 556–562.

Lu, Z., An, X. & Hong, W., 2008 "A fast domain decomposition method for solving three-dimensional large-scale electromagnetic problems," *IEEE Transactions on Antennas and Propagation* **56**, 2200–2210.

Mackie, R., Smith, J. & Madden, T., 1994 "Three-dimensional electromagnetic modeling using finite difference equations: The magnetotelluric example," *Radio Science* **29**, 923–935.

Mitsuhata, Y., Matsuo, K. & Minegishi, M., 1999 "Magnetotelluric survey for exploration of a volcanic-rock reservoir in the Yurihara oil and gas field, Japan," *Geophysical Prospecting* **47**, 195–218.

Mitsuhata, Y. & Uchida, T., 2004 "3d magnetotelluric modeling using the t-omega finite-element method," *Geophysics* **69**, 108–119.

Noguera, A. & Rea, G., 2000 "Deep structure of the Campanian-Lucanian Arc (Southern Apennine, Italy)," *Tectonophysics* **324**, 239–265.

Operto, S., Virieux, J., Amestoy, P., L'Excellent, J.-Y., Giraud, L. & Ben-Hadj-Ali, H., 2007 "3D finite-difference frequency-domain modeling of visco-acoustic wave propagation using a massively parallel direct solver: A feasibility study," *Geophysics* **72**, SM195–SM221.

Pain, C., Herwanger, J., Worthington, M. & De Oliveira, C., 2002 "Effective multidimensional resistivity inversion using finite-element techniques," *Geophysical Journal International* **151**, 710–728.

Papadrakakis, M. & Bitzarakis, S., 1996 "Domain decomposition PCG methods for serial and parallel processing," *Advances in Engineering Software* **25**, 291–307.

Pellerin, L., Alumbaugh, D., Reinemann, D. & Thompson, P., 2004 "Power line induced current in the earth determined by magnetotelluric techniques," *Applied Engineering in Agriculture* **20**, 703–706.

Rice, J., Tsompanopoulou, P. & Vavalis, E., 2000 "Interface relaxation methods for elliptic differential equations," *Applied Numerical Mathematics* **32**, 219–245.

Saad, Y., 2003 *Iterative Methods for Sparse Linear Systems*, 2nd edn. (SIAM).

Saad, Y. & Sosonkina, M., 1999 "Distributed schur complement techniques for general sparse linear systems," *SIAM Journal on Scientific Computing* **21**, 1337–1356.

Schöberl, J., 2001 "Efficient contact solvers based on domain decomposition techniques," *Computers and Mathematics with Applications* **42**, 1217–1228.

Siripunvaraporn, W. & Egbert, G., 2000 "An efficient data-subspace inversion method for 2-D magnetotelluric data," *Geophysics* **65**, 791–803.

Siripunvaraporn, W., Egbert, G. & Lenbury, Y., 2002 "Numerical accuracy of magnetotelluric modeling: A comparison of finite difference approximations," *Earth, Planets and Space* **54**, 721–725.

Siripunvaraporn, W., Egbert, G., Lenbury, Y. & Uyeshima, M., 2005 "Three-dimensional magnetotelluric inversion: Data-space method," *Physics of the Earth and Planetary Interiors* **150**, 3–14.

Smith, B., 1990 "Domain decomposition algorithms for the partial differential equations of linear elasticity," Tech. rep., Department of Computer Science, Courant Institute of Mathematical Sciences, New York University, New York.

Smith, B., 1991 "A domain decomposition algorithm for elliptic problems in three dimensions," *Numerische Mathematik* **60**, 219–234.

Smith, J., 1996 "Conservative modeling of 3-D electromagnetic fields, Part II: Biconjugate gradient solution and an accelerator," *Geophysics* **61**, 1319–1324.

Sourbier, F., Haidar, A., Giraud, L., Operto, S. & Virieux, J., 2008 "Frequency-domain full-waveform modeling using a hybrid direct-iterative solver based on a parallel domain decomposition method: A tool for 3D full-waveform inversion?" *SEG Technical Program Expanded Abstracts* **27**, 2147–2151.

Streich, R., 2009 "3D finite-difference frequency-domain modeling of controlled-source electromagnetic data: Direct solution and optimization for high accuracy," *Geophysics* **74**, F95–F105.

Tang, C., Rial, J. & Lees, J., 2008 "Seismic imaging of the geothermal field at Krafla, Iceland using shear-wave splitting," *Journal of Volcanology and Geothermal Research* **176**, 315–324.

Tuncer, V., Unsworth, M., Siripunvaraporn, W. & Craven, J., 2006 "Audio-magnetotelluric exploration for unconformity uranium deposits in the Athabasca Basin, Saskatachewan, Canada," *SEG Technical Program Expanded Abstracts* **25**, 1328–1332.

Turkoglu, E., Unsworth, M. & Pana, D., 2009 "Deep electrical structure of northern Alberta (Canada): Implications for diamond exploration," *Canadian Journal of Earth Sciences* **46**, 139–154.

Unsworth, M., 2010 "Magnetotelluric studies of active continent-continent collisions," *Surveys in Geophysics* **31**, 137–161.

Wang, B., Mittra, R. & Shao, W., 2008 "A domain decomposition finite-difference method utilizing characteristic basis functions for solving electrostatic problems," *IEEE Transactions on Electromagnetic Compatibility* **50**, 946–952.

Wannamaker, P., 1991 "Advances in three-dimensional magnetotelluric modeling using integral equations," *Geophysics* **56**, 1716–1728.

Wannamaker, P., Stodt, J. & Rijo, L., 1987 "A stable finite element solution for two-dimensional magnetotelluric modelling." *Geophysical Journal - Royal Astronomical Society* **88**, 277–296.

Xie, G., Li, J., Majer, E., Zuo, D. & Oristaglio, M., 2000 "3-D electromagnetic modeling and nonlinear inversion," *Geophysics* **65**, 804–822.

Xiong, Z., 1992 "Electromagnetic modeling of 3-D structures by the method of system iteration using integral equations," *Geophysics* **57**, 1556–1561.

Xiong, Z., 1999 "Domain decomposition for 3D electromagnetic modeling," *Earth, Planets and Space* **51**, 1013–1018.

Xueming, S., Utada, H., Wang, J. & Siripunvaraporn, W., 2010 "Three-dimensional magnetotelluric forward modeling using an improved vector finite-element method combined with divergence corrections based on magnetic fields (VFEH++)," To appear.

Yin, L., Wang, J. & Hong, W., 2002 "A novel algorithm based on the domain-decomposition method for the full-wave analysis of 3-D electromagnetic problems," *IEEE Transactions on Microwave Theory and Techniques* **50**, 2011–2017.

Zyserman, F., Guarracino, L. & Santos, J., 1999 "A hybridized mixed finite element domain decomposed method for two dimensional magnetotelluric modelling," *Earth, Planets and Space* **51**, 297–306.

Zyserman, F. & Santos, J., 2000 "Parallel finite element algorithm with domain decomposition for three-dimensional magnetotelluric modelling," *Journal of Applied Geophysics* **44**, 337–351.

BIOGRAPHY

NAME	Mr. Tawat Rung-Arunwan
DATE OF BIRTH	2nd November 1985
PLACE OF BIRTH	Chonburi, Thailand
INSTITUTIONS ATTENDED	Mahidol University, 2007
	Bachelor of Science (Physics)
	Mahidol University, 2008
	Master of Science (Physics)
SCHOLARSHIPS	Teaching assistant,
	Mahidol University,
	2008-2009
	Research assistant,
	Thailand Center of Excellence in Physics,
	2009-2010
HOME ADDRESS	12/69, Moo 6, Tamboan Bansuan,
	Muang, Chonburi,
	Thailand, 20000
E-MAIL	t.rungarunwan@yahoo.com