

Doctoral Thesis

Geometric multigrid for eddy current problems

Chao Chen

Faculty of Electrical and Information Engineering
Graz University of Technology, Austria

First Examiner:

Univ.-Prof. Dipl.-Ing. Dr.techn. Oszkár Bíró
Graz University of Technology, Austria

Second Examiner:

Univ.-Prof. Dipl.-Ing. Dr.techn. Gundolf Haase
University of Graz, Austria

Graz, December 2012

Kurzfassung

In dieser Arbeit wird das Multigrid-Verfahren für zeitharmonische Wirbelstromprobleme präsentiert. Verschiedene Glättungsalgorithmen werden auf das Multigrid-Verfahren für Gleichungssysteme resultierend aus der $\mathbf{A}, V\text{-}\mathbf{A}$ Formulierung und der $\mathbf{T}, \Phi\text{-}\Phi$ Formulierung angewendet.

Um die Anisotropie des Finite-Elemente-Netzes zu behandeln, wird der planare Glätter für Kantenelemente vorgeschlagen, und auf Finite-Elemente-Formulierungen mit Vektorpotential angewendet. Seine Glättungseigenschaften für mit dünnen finiten Elementen diskretisierte Probleme wird analysiert. Die Leistungsfähigkeit des Multigrid-Verfahrens mit dem planaren Glätter wird für magnetostatische und Wirbelstromprobleme untersucht. Darüber hinaus wird das Multigrid-Verfahren als Vorkonditionierer der Methode der konjugierten Gradienten verwendet, um die Effizienz des Lösers zu verbessern.

Im Falle von nichtlinearen Problemen wird das Multigrid-Verfahren basiert auf einer modifizierten Picard-Banach Methode auf die resultierenden nichtlinearen Gleichungssysteme angewendet. Die Eigenschaften des Multigrid-Lösers werden durch praktische numerische Beispiele demonstriert.

Abstract

This work presents the multigrid method for time-harmonic eddy current problems. Various smoothers for systems of equations resulting from the $\mathbf{A}, V\text{-}\mathbf{A}$ formulation and the $\mathbf{T}, \Phi\text{-}\Phi$ formulation are applied to the multigrid algorithm.

To deal with the anisotropy in the finite element mesh, the plane smoother for edge elements is proposed and applied to finite element formulations with vector potentials. Its smoothing ability for problems discretized by thin finite elements is analyzed. The performance of the multigrid method with the plane smoother for magnetostatic and eddy current problems is investigated. In addition, the multigrid method as the preconditioner of the conjugate gradient method is used for the improvement of the efficiency of the solver.

In the case of nonlinear problems, the multigrid method based on a modified Picard-Banach method is applied to the resulting nonlinear system of equations. The characteristics of the multigrid solver are demonstrated by solutions of practical numerical examples.

Acknowledgement

The research work for this dissertation was initiated by the doctoral school "Numerical Simulations in Technical Sciences" established at the Graz University of Technology and the University of Graz. I am grateful for the financial support by the Austrian Science Fund (FWF).

I would like to express my deepest gratitude to Professor Oszkár Bíró for his continuous guidance throughout the process of performing my research and organizing this dissertation. He always showed me the right direction to finish this work with his knowledge and supported me throughout all the way. In addition, I would like to thank Professor Gundolf Haase for his invaluable help and advices during my doctoral study.

Many thanks to my family for their constant understanding and encouragement, especially to my wife, Shuli for all of her support and love.

Finally, I would like to thank everybody at the Institute for Fundamentals and Theory in Electrical Engineering (IGTE) of the Graz University of Technology for the wonderful time we had during my PhD study.

Graz, December 2012

Chao Chen

Contents

List of abbreviations and symbols	vi
1 Introduction	1
1.1 Introduction	1
1.2 Overview of the multigrid method	2
1.3 Overview of the application to the multigrid method for electro- magnetic problems	3
1.4 Thesis organization	5
2 General principle of the multigrid method	7
2.1 Iterative methods	7
2.1.1 Introduction	7
2.1.2 Symmetric Gauss-Seidel method	9
2.2 The multigrid idea	10
2.3 Two-grid method	12
2.4 Multigrid method	14
2.5 Preconditioned conjugate gradient method	15
3 Multigrid for finite elements	19
3.1 $H^1(\Omega)$ and $\mathbf{H}(\text{curl}, \Omega)$ spaces	19
3.2 Finite elements	21
3.2.1 Introduction to finite elements	21

Contents

3.2.2	Nodal elements	22
3.2.3	Edge elements	23
3.2.4	Relation between the nodal element space and the edge element space	25
3.3	Hierarchy of grids	27
3.4	Restriction and prolongation	29
3.5	Smoother	32
3.5.1	Multiplicative Schwarz iteration	33
3.5.2	Arnold, Falk and Winther smoother	34
3.5.3	Plane smoother	36
3.6	Multigrid preconditioned conjugate gradient method	41
4	The magnetostatic problem	43
4.1	Differential equations and boundary conditions	43
4.1.1	The Φ -formulation	44
4.1.2	The \mathbf{A} -formulation	45
4.2	Tree and Cotree	47
4.3	Smoother for magnetostatic problems	49
4.3.1	The smoother for the \mathbf{A} formulation	49
4.3.2	The plane smoother for thin elements	50
4.4	Numerical examples	54
5	The time-harmonic eddy current problem	59
5.1	Differential equations and boundary conditions	59
5.1.1	The \mathbf{A}, V formulation	61
5.1.2	The \mathbf{T}, Φ formulation	63
5.1.3	Coupling of the potential formulations	65
5.2	Multigrid for nonlinear problems	66
5.3	Smoother for eddy current problems	69
5.3.1	Gauss-Seidel smoother for the \mathbf{A}, V formulation	69
5.3.2	Block smoothers for eddy current problems	71
5.4	Numerical examples	77
5.4.1	Example 1	77
5.4.2	Example 2	81
6	Conclusion	85

List of abbreviations and symbols

GS	Gauss-Seidel
SGS	Symmetric Gauss-Seidel
MG	Multigrid
CG	Conjugate gradient
PCG	Preconditioned conjugate gradient
ICCG	Incomplete Cholesky conjugate gradient
SGSCG	Symmetric Gauss-Seidel preconditioned conjugate gradient
MGCG	Multigrid preconditioned conjugate gradient
MSI	Multiplicative Schwarz iteration
AFW	Arnold, Falk and Winther
MAFW	Modified AFW
$\ker A$	Kernel of the matrix A
$\text{Rank} A$	Rank of the matrix A
S^ν	Smoother with ν steps of smoothing iterations
ν	Number of smoothing iterations in S^ν ; also the magnetic reluctivity $\nu = 1/\mu$
P	Prolongation operator
R	Restriction operator
Ω	Domain in $\mathbb{R}^n (n = 1, 2, 3)$

Contents

Γ	Boundary of Ω
\mathbf{n}	Normal vector
$L^2(\Omega)$	Space of the square integrable functions in Ω
$H^1(\Omega)$	$\{v \in L^2(\Omega); \text{grad}v \in (L^2(\Omega))^3\}$
$H_0^1(\Omega; \Gamma_D)$	$\{v \in L^2(\Omega); \text{grad}v \in (L^2(\Omega))^3; v = 0 \text{ on } \Gamma_D\}$
$\mathbf{H}(\text{curl}; \Omega)$	$\{\boldsymbol{\eta} \in L^2(\Omega)^3; \text{curl}\boldsymbol{\eta} \in (L^2(\Omega))^3\}$
$\mathbf{H}_0(\text{curl}; \Omega; \Gamma_D)$	$\{\boldsymbol{\eta} \in L^2(\Omega)^3; \text{curl}\boldsymbol{\eta} \in (L^2(\Omega))^3; \boldsymbol{\eta} \times \mathbf{n} = \mathbf{0} \text{ on } \Gamma_D\}$
$\mathbf{H}_0^o(\text{curl}; \Omega; \Gamma_D)$	$\{\boldsymbol{\eta} \in \mathbf{H}_0(\text{curl}; \Omega; \Gamma_D); \text{curl}\boldsymbol{\eta} = \mathbf{0}\}$
$\mathbf{H}_0^\perp(\text{curl}; \Omega; \Gamma_D)$	$\{\mathbf{u} \in \mathbf{H}_0(\text{curl}; \Omega; \Gamma_D); \text{div}\mathbf{u} = 0\}$
T	One finite element in \mathbb{R}^3
\mathcal{T}_h	Discretization over the domain $\bar{\Omega}$
h	Characteristic size of the finite element
Σ	Degrees of freedom in one finite element
I_k^h	Interpolation operator that projects $H^1(\Omega)$ onto $\mathcal{S}_k(\mathcal{T}_h)$ denoted by $I_k^h : H^1(\Omega) \mapsto \mathcal{S}_k(\mathcal{T}_h)$
$\mathcal{S}_k(\mathcal{T}_h)$	Space of the nodal elements of order k
$\mathcal{S}_k(\mathcal{T}_h; \Gamma_D)$	$\{\varphi_h \in \mathcal{S}_k(\mathcal{T}_h); \varphi_h = 0 \text{ on } \Gamma_D\}$
\mathcal{S}_h^k	Abbreviation of $\mathcal{S}_k(\mathcal{T}_h)$
$\mathcal{S}_{h,D}^k$	Abbreviation of $\mathcal{S}_k(\mathcal{T}_h; \Gamma_D)$
N_i	Nodal basis functions
Π_k^h	Interpolation operator $\Pi_k^h : \mathbf{H}(\text{curl}; \Omega) \mapsto \mathcal{N}_k(\mathcal{T}_h)$
$\mathcal{N}_k(\mathcal{T}_h)$	Space of the edge elements of order k
$\mathcal{N}_{k,0}(\mathcal{T}_h; \Gamma_D)$	$\{\boldsymbol{\eta}_h \in \mathcal{N}_k(\mathcal{T}_h); \boldsymbol{\eta}_h \times \mathbf{n} = \mathbf{0} \text{ on } \Gamma_D\}$
\mathcal{N}_h^k	Abbreviation of $\mathcal{N}_k(\mathcal{T}_h)$
$\mathcal{N}_{h,D}^k$	Abbreviation of $\mathcal{N}_{k,0}(\mathcal{T}_h; \Gamma_D)$
$\mathcal{N}_h^{k,o}$	$\{\boldsymbol{\eta}_h \in \mathcal{N}_h^k; \text{curl}\boldsymbol{\eta}_h = \mathbf{0}\}$
$\mathcal{N}_{h,D}^{k,\perp}$	$\{\boldsymbol{\eta}_h \in \mathcal{N}_{h,D}^k; (\boldsymbol{\eta}_h, \text{grad}\psi_h) = \mathbf{0} \forall \psi \in \mathcal{S}_{h,D}^k\}$
\mathbf{N}_j	Edge basis functions
(ξ, η, ζ)	Local coordinates of the nodes of the second order hexahedral finite elements
$\mathcal{D}(X)$	Domain of definition of the operator X
G	Incidence matrix of the finite element mesh

l	Level number of the finite element grid
l_{max}	Level number of the finest grid
T_{mac}	Macro element
u_l	Subdivisions of the elements in the grid
X	Space of the solution
δ_{max}	Maximum aspect ratio
n_n	Number of nodes in the finite element grid
n_{n_d}	Number of nodes on the Dirichlet boundary
n_e	Number of edges in the finite element grid
n_{e_d}	Number of edges on the Dirichlet boundary
B	Magnetic flux density
H	Magnetic field intensity
E	Electric field intensity
J	Current density
μ	Magnetic permeability
σ	Electric conductivity
ρ	Electric resistivity
Φ	Magnetic scalar potential
V	Electric scalar potential
A	Magnetic vector potential
T	Current vector potential
δ_p	Penetration depth

1.1 Introduction

Computer-aided simulation is an essential tool in all fields of technical and scientific research. It provides the opportunity of investigating properties of prototypes in advance by means of numerical calculation and thus saves costs. Its applications range from the calculation of characteristics of mobile phones, power transformers, magnetic resonance imaging and turbo machinery to the simulation of weather, tides and avalanches.

An efficient technique for the numerical calculation of electric and magnetic fields is the finite element method by which the problem domain is divided into smaller subregions called finite elements. With the help of this decomposition, the partial differential equations derived from Maxwell's equations can be discretized to set up a system of equations whose solution results in the field quantities.

The number of unknowns in the equation system depends directly on the number of finite elements, which means that for areas or volumes with fine discretization, large systems of equations are usually obtained. Since their corresponding system matrices are primarily populated by zero, they are called sparse systems of equations.

In the finite element method, mainstream applications are using scalar potentials and vector potentials to describe the electric and magnetic fields. The scalar potential is commonly approximated by nodal elements. However, for the discretization of vector potentials, it has been proven that edge elements have advantages. In recent years, the significance of edge elements for the numerical

treatment of magnetostatic problems, eddy current problems and high frequency problems has increased rapidly. They have become indispensable in those fields.

The solution of large sparse systems of equations constitutes a comprehensive topic in numerical mathematics. Since the calculation of the direct solution of a large system of equations is time-consuming or needs big amount of computer memory, algorithms for iterative solutions have become a promising choice. Starting with an initial solution, a new value of the solution is computed and approximates the exact solution better and better. Classical algorithms of iterative methods include the Jacobi iteration, Gauss-Seidel iteration or their corresponding overrelaxation methods. But these iterative methods are no longer used in practice, since their convergence is poor for problems with large condition number, influenced by large number of degrees of freedom, poor mesh quality, etc..

A widely used algorithm is the preconditioned conjugate gradient method which is a faster and more stable method, for which the above classical iterative algorithms can be used as a preconditioner. The most frequently used preconditioner is the incomplete Cholesky decomposition. However, the convergence of the preconditioned conjugate gradient method deteriorates as the number of unknowns increases.

One of the fastest methods for solving large sparse systems of equations is the multigrid (multilevel) method, in which the error of the approximate solution is corrected by a solution based on a coarser finite element or finite difference grid. A brief overview of the history of the multigrid method and its application for electromagnetics are given in the following sections.

1.2 Overview of the multigrid method

The development of the multigrid method is based on three well-known ideas [1]: the smoothing property of iterative methods; calculation of the correction on a coarse grid; nested iterations.

The first description and analysis of the multigrid method is presented by Fedorenko in [2, 3] and by Bakhvalov in [4] in the early 1960s. The authors combined the smoothing error of the Jacobi iteration with the correction of the error on a coarser mesh. The convergence behavior for problems which consist of a unit cubic, as well as complex structures is demonstrated in these works.

The efficiency of the multigrid method was first investigated in [5]. In 1976/77,

1.3 Overview of the application to the multigrid method for electromagnetic problems

Hackbusch developed a multigrid algorithm independent of previous works. In 1977 the multigrid method for solving Navier-Stokes equations was developed by Wesseling, and results for the multigrid method from the group of Trottenberg, Stueben and Malevich have been published since 1980. After 1981, the number of publications on the multigrid method has been growing rapidly.

The first multigrid method was developed for the finite difference method. An important application was computational fluid dynamics. The multigrid algorithm for finite element equations was presented in [6] in 1977, which was used for practical applications starting in 1980 [7]. In the early 1980s, Brandt and Ruge developed the algebraic multigrid method.

A good summary of the history of the multigrid method before 1981 and an introduction to the algorithm can be found in [1]. A guide for the construction of multigrid algorithms is described in [8]. The first monography was published in 1985 by Hackbusch, in which he presented the geometric multigrid method for the finite difference method, as well as for the finite element method [9]. In addition, a detailed introduction to the multigrid method for finite differences can be found in [10]. For the algebraic multigrid method, a general description is given in the book [11], and its application to the finite element method is presented in [11]. The book published by Trottenberg et al. [12] includes an overview of the multigrid method, and the investigation of its efficiency and the optimization and improvement of the multigrid components.

1.3 Overview of the application to the multigrid method for electromagnetic problems

The multigrid method considered in the calculation of electric and magnetic fields discretized by finite elements was first developed for nodal elements using various algorithms as presented in [13]. However, since edge elements are popular and advantageous for the discretization of vector potentials, their corresponding multigrid algorithm have become the focus of research.

The difficulty for solving the curl-curl equation derived from Maxwell's equations discretized by edge elements originates from the kernel of the curl operator [14]. An important publication by Hiptmair in 1998 [15] proposes a new smoother by which the kernel of the curl operator and its orthogonal space are solved individually with the help of a so-called lift operator and nodal basis func-

tions. In 2000, Arnold, Falk and Winther (AFW) resolved the same problem by using a block iteration based on a domain decomposition method as the smoother for the multigrid method [16].

As practical applications, in the case of magnetostatic problems, the above smoothers applied to regular meshes with triangular elements, second order hexahedral elements and nonstructured mesh are presented in [17–19], respectively, as well as the hierarchy of grids, the prolongation and restriction operators for edge elements. It has been found that symmetric Gauss-Seidel iterations instead of block iterations can also be applied to the multigrid as the smoother for edge elements [20–22], which reduces the complexity of the algorithm. In parallel, the algebraic multigrid method has been successfully applied to edge elements [23,24].

In the case of eddy current problems, for the finite element formulation discretized by the magnetic vector potential only, i.e. the \mathbf{A}^* formulation [25], the Hiptmair smoother and the AFW smoother are capable of smoothing errors in the kernel space of the curl operator and thus can be applied to the multigrid method, whereas the Gauss-Seidel smoother failed. But for the formulation with scalar potentials, i.e. the \mathbf{A}, V formulation and the \mathbf{T}, Φ formulation [26], the Gauss-Seidel smoother is able to eliminate the error in both edge and nodal element spaces [27,28]. Moreover, the application of the algebraic multigrid method to eddy current problems is presented in [29,30].

The nonlinearities of material properties and anisotropies of meshes in most cases need to be taken into special account. For the former issue, the multigrid based on a Picard-Banach method for solving nonlinear magnetostatic and eddy current problems are presented in [20,31], respectively. For the latter one, the line smoother and the plane smoother for 2-D and 3-D problems discretized by nodal elements are introduced in [12]. The applications of these smoothers have been extended to edge elements as presented in [32,33].

For complicated problems where the multigrid may lose its efficiency, the combination of the multigrid method with the conjugate gradient method may have the potential of a substantial acceleration [34]. Its application for electromagnetic problems is investigated in [33,35] for the geometric and algebraic multigrid methods, respectively.

1.4 Thesis organization

This work is organized as follows: In the second chapter, iterative methods used in this work are presented. A detailed introduction of the multigrid algorithm is given. The third chapter can be divided into two parts, one of which deals with the nodal elements and edge elements used for the discretization of scalar and vector potentials, and the other focuses on the individual components of the multigrid method (hierarchy of grids, prolongation and restriction, smoother).

Chapter four describes the application of the multigrid method to systems of equations arising from the finite element discretization of magnetostatic problems. Smoothers applied to the Φ -formulation and the \mathbf{A} -formulation are presented, respectively. In particular, the efficiency of the plane smoother designed for problems with thin elements in the finite element mesh is investigated by a test problem. The numerical analysis of a practical example is used to demonstrate advantages of the plane smoother and the improvement of the efficiency of the solver achieved by the multigrid preconditioned conjugate gradient method.

Chapter five is dedicated to time-harmonic eddy current problems. First of all, the \mathbf{A} , $V\text{-}\mathbf{A}$ and \mathbf{T} , $\Phi\text{-}\Phi$ formulations are presented. The second part of this chapter introduces the algorithm for solving nonlinear systems of equations. In the following, the characteristics of smoothers designed for the eddy current formulations are investigated by comparison of the convergence and the solution time for test problems. In the end, two numerical examples illustrate the efficiency of the proposed multigrid algorithms. Finally, there is a brief summary of this work.

The major contributions of this work are listed as follows:

- The geometric MG method has been applied for solving the systems of equations resulting from various finite element formulations.
- The plane smoothing iteration for edge elements has been proposed to deal with problems discretized by anisotropic meshes.
- The geometric MG method based on a modified Picard-Banach method has been used as the solver for nonlinear eddy current problems.
- The characteristics of the MG method with different smoothers are demonstrated by practical numerical examples.

General principle of the multigrid method

Consider the linear system of equations

$$A\mathbf{x} = \mathbf{b} \quad (2.1)$$

where $A \in \mathbb{R}^{n \times n}$ is a symmetric, positive (or negative) semi-definite matrix with coefficients denoted by a_{ij} ($i, j = 1, 2, \dots, n$), and $\mathbf{x}, \mathbf{b} \in \mathbb{R}^n$ are vectors with entries denoted by x_i and b_i ($i = 1, 2, \dots, n$), respectively.

Two important subspaces that are associated with the system matrix A are its *kernel* defined by $\ker A = \{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} = \mathbf{0}\}$ and its *range* defined by $\text{Rank} A = \{A\mathbf{x} : \mathbf{x} \in \mathbb{R}^n\}$. The dimension of the range of A is called the rank of A and is equal to the number of linearly independent equations in (2.1).

The iterative methods used in this work will be briefly introduced in the following sections, the details of which can be found in [36, 37].

2.1 Iterative methods

2.1.1 Introduction

An iterative method that produces iterates $\mathbf{x}^1, \mathbf{x}^2, \dots$, from an arbitrary starting value \mathbf{x}^0 can be described as

$$\mathbf{x}^{m+1} := \Phi(\mathbf{x}^m, \mathbf{b}). \quad (2.2)$$

where Φ depends on A and \mathbf{b} in (2.1) and m denotes the number of iterations.

Definition 1. An iterative method is a mapping

$$\Phi : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^n. \quad (2.3)$$

Definition 2. $\mathbf{x}^* = \mathbf{x}^*(\mathbf{b})$ is called a *fixed point* of the iterative method Φ belonging to $\mathbf{b} \in \mathbb{R}^n$, if

$$\mathbf{x}^* := \Phi(\mathbf{x}^*, \mathbf{b}). \quad (2.4)$$

Definition 3. The iterative method Φ is called *consistent* with the system of equations (2.1), if for all right-hand sides $\mathbf{b} \in \mathbb{R}^n$ any solution \mathbf{x} of (2.1) is a fixed point of Φ with respect to \mathbf{b} .

Definition 4. An iterative method Φ is called *convergent*, if for all right-hand sides $\mathbf{b} \in \mathbb{R}^n$, there is a limit $\mathbf{x}^*(\mathbf{b})$ of iterates \mathbf{x}^m independent of the starting value \mathbf{x}^0 .

Definition 5. An iterative method Φ is called *linear* if $\Phi(\mathbf{x}^m, \mathbf{b})$ is linear in \mathbf{x} and \mathbf{b} , i.e., if there are matrices M and N such that

$$\Phi(\mathbf{x}, \mathbf{b}) = M\mathbf{x} + N\mathbf{b} \quad (2.5)$$

where the matrix M is called the iteration matrix (also called error prolongation matrix) of the iteration Φ .

All linear iterative methods for solving linear equations can be described as [36]:

1. The *first normal form*:

$$\mathbf{x}^{m+1} := M\mathbf{x}^m + N\mathbf{b} \quad m \geq 0. \quad (2.6)$$

2. The *second normal form*:

$$\mathbf{x}^{m+1} := \mathbf{x}^m - N(A\mathbf{x}^m - \mathbf{b}) \quad m \geq 0 \quad (2.7)$$

where the matrix N is called the matrix of the second normal form of Φ .

3. The *third normal form*:

$$W(\mathbf{x}^m - \mathbf{x}^{m+1}) = A\mathbf{x}^m - \mathbf{b} \quad m \geq 0 \quad (2.8)$$

where the matrix W is called the matrix of the third normal form of Φ , or the preconditioner of iteration Φ since (2.8) can be also expressed as

$$\mathbf{x}^{m+1} = \mathbf{x}^m - W^{-1}(A\mathbf{x}^m - \mathbf{b}) \quad m \geq 0. \quad (2.9)$$

A convergence criterion of the consistent iteration Φ is given by a *norm* estimate [36]:

Theorem 1. A sufficient condition for the convergence of an iteration is the estimate of the iteration matrix M :

$$\|M\| \leq 1 \quad (2.10)$$

where $\|\cdot\|$ be a corresponding matrix norm defined by $\|A\| := \sup \left\{ \frac{\|A\mathbf{x}\|}{\|\mathbf{x}\|} : \mathbf{0} \neq \mathbf{x} \in \mathbb{R}^n \right\}$. If the iteration is consistent, the error estimate can be written as

$$\|\mathbf{e}^{m+1}\| \leq \|M\| \|\mathbf{e}^m\| \quad \text{or} \quad \|\mathbf{e}^m\| \leq \|M\|^m \|\mathbf{e}^0\| \quad (2.11)$$

where $\mathbf{e}^m = \mathbf{x}^m - \mathbf{x}$ is defined as the *iteration error*.

2.1.2 Symmetric Gauss-Seidel method

The *Gauss-Seidel* (GS) method is one of iterative methods used to solve the linear system of equations (2.1). The matrix A can be decomposed into the sum

$$A = D - L - U \quad (2.12)$$

where D is a diagonal matrix, L is a strictly lower triangular matrix and U is a strictly upper triangular matrix. The standard GS method, i.e. the forward GS method results from the third normal form (2.8) by choosing

$$W^{\text{FGS}} = D - L. \quad (2.13)$$

Then the iteration matrix M^{FGS} is expressed as

$$M^{\text{FGS}} = (D - L)^{-1}U. \quad (2.14)$$

The backward GS iteration can be obtained by choosing

$$W^{\text{BGS}} = D - U \quad (2.15)$$

instead of (2.13) and

$$M^{\text{BGS}} = (D - U)^{-1}L \quad (2.16)$$

instead of (2.14). The iteration matrix of the symmetric GS (SGS) iteration is defined by the product of iteration matrices of the forward and backward GS iterations:

$$M^{\text{SGS}} = (D - U)^{-1}L(D - L)^{-1}U. \quad (2.17)$$

The matrix W^{SGS} of the third normal form (2.8) is

$$W^{\text{SGS}} = (D - L)D^{-1}(D - U). \quad (2.18)$$

The componentwise description of the SGS iteration can be written as

$$\begin{aligned} x_i^{m+1} &:= \left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{m+1} - \sum_{j=i+1}^n a_{ij}x_j^m \right) / a_{ii} \quad i, j = 1, 2, \dots, n, \\ x_i^{m+1} &:= \left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^m - \sum_{j=i+1}^n a_{ij}x_j^{m+1} \right) / a_{ii} \quad i, j = n, n-1, \dots, 1. \end{aligned} \quad (2.19)$$

2.2 The multigrid idea

The *multigrid* (MG) method is an iterative method that combines the *smoothing* property of certain iterative methods with the scheme called the *coarse grid correction* whose fundamental idea is to calculate the correction of the solution on a coarse finite element or coarse finite difference grid.

The consideration of the MG idea is based on the observation of the smoothing property of classic iterative methods. The highly oscillating (high frequency) components of errors in the first few iterations are greatly reduced, while the components with less local variation (low frequency) are damped slowly.

Let \mathbf{x}^m be the approximation of the solution \mathbf{x} of (2.1) after m steps of smoothing iterations and the error of \mathbf{x}^m be written as $\mathbf{e}^m = \mathbf{x}^m - \mathbf{x}$. The error vector \mathbf{e}^m is composed of high frequency and low frequency components. In Fig. 2.1, the smoothing property of the iterative method against the error \mathbf{e}^m for a one-dimensional problem

$$\begin{aligned} -u''(x) &= f(x), \quad x \in (0, x_L) \\ u(0) &= u(x_L) = 0 \end{aligned} \quad (2.20)$$

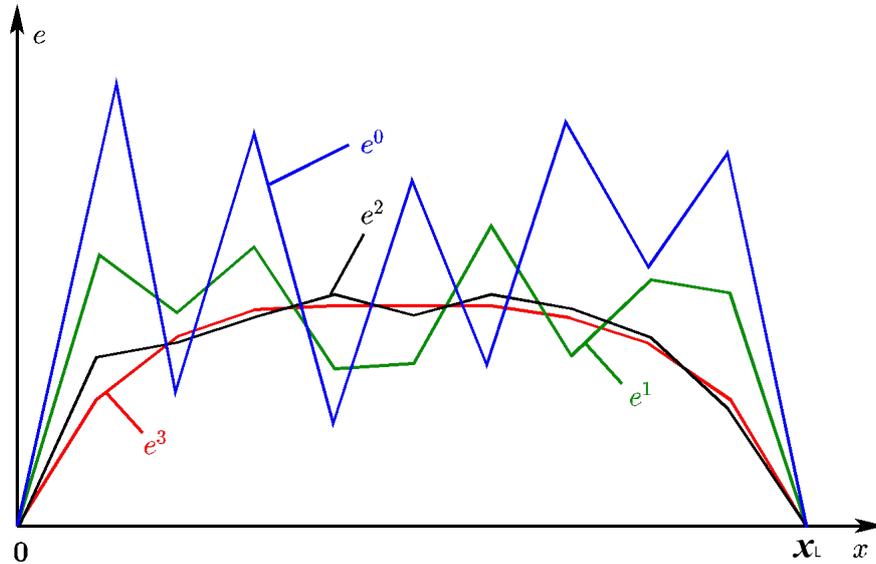


Figure 2.1 The error e^m after m smoothing iterations for a one-dimensional problem.

is schematically illustrated. After one step of the smoothing iteration, high frequency components in the error are greatly damped, and after three iterations the error e^3 has basically only low frequency components, which indicates that e^3 has been successfully smoothed. An iterative method with this property is called *smoother* and can be applied to the MG method.

According to the above observation, since the error consists of mainly low frequency components, it is possible to approximate the error by means of a coarse grid. The idea of the coarse grid correction is to project the *defect* defined as $A\mathbf{x}^m - \mathbf{b}$ onto the coarse grid, solve the error correction scheme thereon, and project the coarse correction again to the original grid.

It can be seen that the interaction between the smoothing property of the iterative method and the coarse grid correction is the key factor of the MG algorithm.

With respect to the setup of the grid hierarchy, there are two types of MG methods: the *algebraic* MG method and the *geometric* MG method. In the geometric MG method, a hierarchy of grids has to be established a priori. To achieve this, a coarse grid is primarily constructed. The following finer grids are obtained either by subdividing the grids into finer grids (*h-method*), or by increasing the order of the shape functions (*p-method*) [38].

The algebraic MG method is a true “black-box” solver for the system of equations discretized by finite differences or finite elements. It constructs a hierarchy of systems of equations and MG operators without a priori knowledge of the coarse grid and the geometric interpretation. The algebraic MG method has its advantage when the coarse grid is too hard to be generated primarily [39].

In this work only the geometric MG method with the hierarchy of grids generated by the h -method will be investigated.

2.3 Two-grid method

For better understanding, the two-grid method is first introduced. The finite element discretization on the fine grid h leads to a system of n_h linear equations:

$$A_h \mathbf{x}_h = \mathbf{b}_h \quad (2.21)$$

where the system matrix is denoted by A_h . After ν_1 smoothing iterations $S^{\nu_1}(\mathbf{x}_h, \mathbf{b}_h)$ the approximation $\tilde{\mathbf{x}}_h$ of the solution of (2.21) can be obtained. The exact correction of $\tilde{\mathbf{x}}_h$ is

$$\mathbf{e}_h = \tilde{\mathbf{x}}_h - \mathbf{x}_h. \quad (2.22)$$

Let \mathbf{d}_h be the defect of $\tilde{\mathbf{x}}_h$,

$$\mathbf{d}_h = A_h \tilde{\mathbf{x}}_h - \mathbf{b}_h. \quad (2.23)$$

From (2.21) and (2.23) one can obtain

$$A_h \mathbf{e}_h = \mathbf{d}_h \quad (2.24)$$

called *defect equation* which would yield $\mathbf{x}_h = \tilde{\mathbf{x}}_h - \mathbf{e}_h$. According to the idea of the coarse grid correction, the approximation to \mathbf{e}_h on a coarse grid H needs to be computed, leading to the equation

$$A_H \mathbf{e}_H = \mathbf{d}_H \quad (2.25)$$

where the solution \mathbf{e}_H is an approximation of \mathbf{e}_h on the coarse grid H . The linear mapping

$$\mathbf{d}_H = R_h \mathbf{d}_h \quad (2.26)$$

is called *restriction* with R_h as the restriction matrix that projects the defect \mathbf{d}_h to the coarse grid. By the *prolongation* operator P_h one can interpolate the error

from the coarse grid to the fine grid:

$$\mathbf{e}_h = P_h \mathbf{e}_H. \quad (2.27)$$

Finally, the new solution $\mathbf{x}_h^{\text{new}}$ is calculated by the coarse grid correction:

$$\mathbf{x}_h^{\text{new}} = \tilde{\mathbf{x}}_h - P_h \mathbf{e}_H. \quad (2.28)$$

The above procedure is presented in the following algorithm:

Algorithm 1. Two-grid algorithm $\mathbf{x}_h^{m+1} = \text{TGA}(\mathbf{x}_h^m, \mathbf{b}_h)$

1. $\tilde{\mathbf{x}}_h = S^{\nu_1}(\mathbf{x}_h^m, \mathbf{b}_h)$: ν_1 smoothing iterations
2. $\mathbf{d}_h = A_h \tilde{\mathbf{x}}_h - \mathbf{b}_h$: Calculation of the defect
3. $\mathbf{d}_H = R_h \mathbf{d}_h$: Restriction of the defect to the coarse grid H
4. $\mathbf{e}_H = A_H^{-1} \mathbf{d}_H$: Solution on the coarse grid H
5. $\mathbf{x}_h = \tilde{\mathbf{x}}_h - P_h \mathbf{e}_H$: Correction of $\tilde{\mathbf{x}}_h$
6. $\mathbf{x}_h^{\text{new}} = S^{\nu_2}(\mathbf{x}_h, \mathbf{b}_h)$: ν_2 smoothing iterations
7. $\text{TGA}(\mathbf{x}_h^m, \mathbf{b}_h) = \mathbf{x}_h^{m+1}$

In step 6 above, ν_2 steps of smoothing iterations are introduced to the two-grid algorithm. In this case, the steps 1 and 6 are called *pre-* and *post-* smoothing, respectively. The iteration matrix M_h^{TGI} of the two-grid iteration is

$$M_h^{\text{TGI}}(\nu_1, \nu_2) = S_h^{\nu_2} (I - P_h A_H^{-1} R_h A_h) S_h^{\nu_1}. \quad (2.29)$$

It can be seen that M_h^{TGI} is the product of the iteration matrix M_h^{CGC} of the coarse grid correction

$$M_h^{\text{CGC}} = I - P_h A_H^{-1} R_h A_h, \quad (2.30)$$

the pre-smoothing iteration matrix $S_h^{\nu_1}$ and the post-smoothing iteration matrix $S_h^{\nu_2}$.

The convergence of the two-grid iteration depends on the combination of the approximation property of the coarse grid correction and the smoothing property of the pre- and post- smoothing iterations.

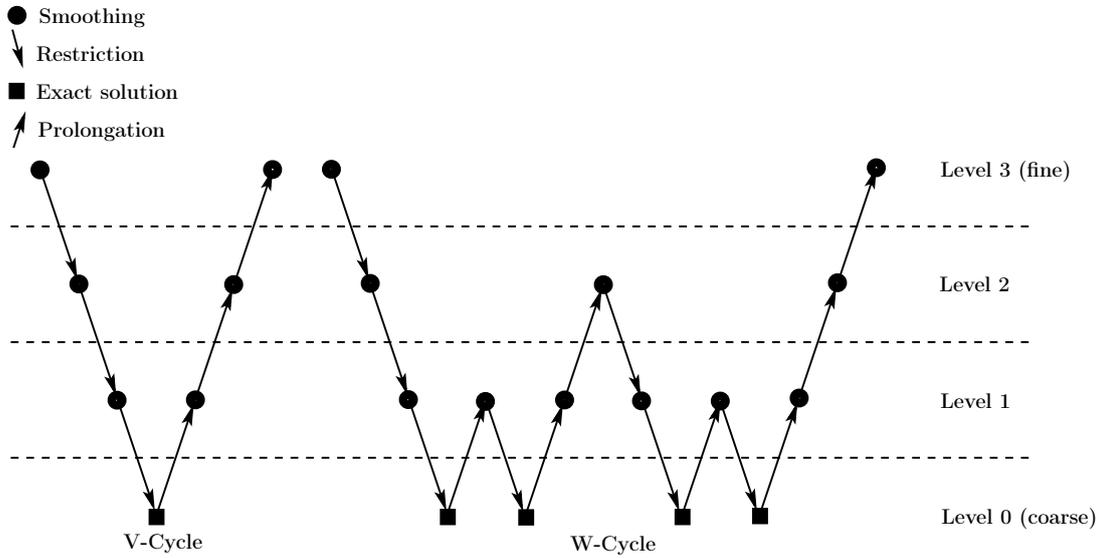


Figure 2.2 MG V-cycle and W-cycle for $l_{\max} = 3$.

2.4 Multigrid method

In step 4 of the algorithm 1, the defect equation $A_H \mathbf{e}_H = \mathbf{d}_H$ has to be exactly solved. This step can be replaced by another two-grid algorithm; this operation can be done recursively to generate a recursive algorithm called the MG iteration.

Based on a hierarchy of grids, the associated system matrices for each level of the MG method A_l ($l = 0, 1, 2, \dots, l_{\max}$) have to be constructed, where l is the level number of grids. The finest and coarsest grids are denoted as $l = l_{\max}$ and $l = 0$, respectively. The MG algorithm can be described as:

Algorithm 2. Multigrid algorithm $\mathbf{x}^{m+1} = \text{MGA}_l(\mathbf{x}_l^m, \mathbf{b}_l)$

1. if $l = 0$ then $A_0 \mathbf{x}_0^{m+1} = \mathbf{b}_0$ is exactly solved else
2. $\tilde{\mathbf{x}}_l = S^{\nu_1}(\mathbf{x}_l^m, \mathbf{b}_l)$
3. $\mathbf{d}_{l-1} = R_l(A_l \tilde{\mathbf{x}}_l - \mathbf{b}_l)$
4. $\mathbf{e}_{l-1}^{(0)} = \mathbf{0}$
5. for $i = 1$ to γ do $\mathbf{e}_{l-1}^{(i)} = \text{MGA}_{l-1}(\mathbf{e}_{l-1}^{(i-1)}, \mathbf{d}_{l-1})$

2.5 Preconditioned conjugate gradient method

6. $\mathbf{x}_l = \tilde{\mathbf{x}}_l - P_l \mathbf{e}_{l-1}^{(\gamma)}$
7. $\mathbf{x}_l^{m+1} = S^{\nu_2}(\mathbf{x}_l, \mathbf{b}_l)$
8. $\text{MGA}_l(\mathbf{x}_l^m, \mathbf{b}_l) = \mathbf{x}_l^{m+1}$

In step 5 of the above algorithm, γ steps of MG iterations are applied to solve the system of equations based on coarser grids. Only cases $\gamma = 1$ and $\gamma = 2$ are of practical interest. The MG iteration with $\gamma = 1$ is called *V-cycle*, whereas the iteration with $\gamma = 2$ has the name *W-cycle*. The MG iterations for four MG levels ($l_{\max} = 3$) with $\gamma = 1$ and $\gamma = 2$ are illustrated in Fig. 2.2.

The iteration matrix M_l^{MGI} of the MG iteration is

$$\begin{aligned} M_0^{\text{MGI}}(v_1, v_2) &= 0, & M_1^{\text{MGI}} &= M_1^{\text{TGI}}(v_1, v_2), \\ M_l^{\text{MGI}}(v_1, v_2) &= M_l^{\text{TGI}}(v_1, v_2) + S_l^{\nu_2} P_l (M_{l-1}^{\text{MGI}})^\gamma (v_1, v_2) A_{l-1}^{-1} R_l S_l^{\nu_1} \text{ for } l \geq 1 \end{aligned} \quad (2.31)$$

and the iteration matrix of the coarse grid correction is

$$M_l^{\text{CGC}} = I - P_l [I - (M_{l-1}^{\text{MGI}})^\gamma] A_{l-1}^{-1} R_l A_l. \quad (2.32)$$

The solution of linear equations on the coarsest grid $l = 0$ has to be carried out. In this case, a direct solver needs to be used. Since the coarsest grid has the smallest number of grid points, the solution should not lead to practical difficulties.

Although the MG method is very efficient, in some cases especially nonlinear problems it can be improved by using the *nested iteration*, also called *full MG*. The system of equations on the coarsest grid is exactly solved and the solution is used as an initial guess for the approximation on a finer grid (see Fig. 2.3). The disadvantage of the W-cycle MG iteration and the full MG iteration is that they are more expensive than the V-cycle MG iteration, therefore, in this work only the V-cycle MG iteration is applied to solve the linear system of equations.

2.5 Preconditioned conjugate gradient method

Another important iterative method is the *conjugate gradient* (CG) method whose purpose is to approximate the solution of the equation (2.1) by means of a *search direction* $\mathbf{p}^m \in \mathbb{R}^n$. It is known that two vectors \mathbf{p} and \mathbf{q} are conjugate if $\mathbf{p}^T \mathbf{A} \mathbf{q} = 0$ which is denoted by $\mathbf{A} \mathbf{q} \perp \mathbf{p}$. The optimal search direction is guaranteed if

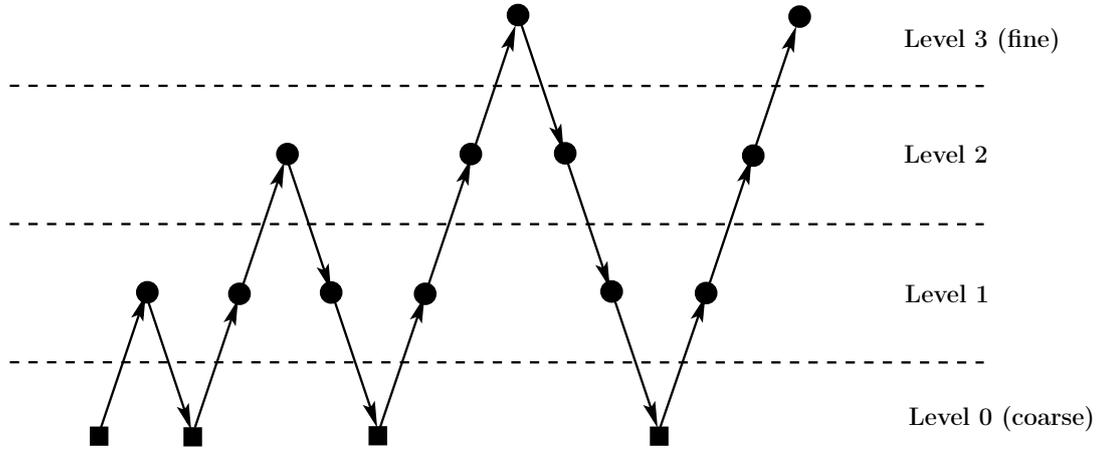


Figure 2.3 Full MG for $l_{\max} = 3$.

$\mathbf{p}^m \neq \mathbf{0}$ and \mathbf{p}^m is conjugate to all preceding directions \mathbf{p}^l ($l < m$). Assuming exact arithmetic the CG method would terminate after n iterations with the exact solution [36].

The CG method can be improved when it is applied to a symmetric iteration

$$\Phi : \quad \mathbf{x}^{m+1} = \mathbf{x}^m - W^{-1}(A\mathbf{x}^m - \mathbf{b}) \quad (2.33)$$

Substituting A and \mathbf{b} in (2.33) by $\check{A} = W^{-1/2}AW^{-1/2}$ and $\check{\mathbf{b}} = W^{-1/2}\mathbf{b}$ and applying the CG method to the associated system of equations $\check{A}\check{\mathbf{x}} = \check{\mathbf{b}}$ leads to the following *preconditioned conjugate gradient* (PCG) algorithm:

Algorithm 3. Preconditioned conjugate gradient method $\text{PCG}(\mathbf{x}^0, \mathbf{b}, A, W)$

1. $\mathbf{r}^0 = \mathbf{b} - A\mathbf{x}^0$
2. $\mathbf{p}^0 = W^{-1}\mathbf{r}^0$
3. $\rho^0 = \langle \mathbf{p}^0, \mathbf{r}^0 \rangle \quad m = 0$
4. do while ($\|\mathbf{r}\| \geq \epsilon$)
5. $\mathbf{a}^m = A\mathbf{p}^m$
6. $\lambda_{\text{opt}} = \rho^m / \langle \mathbf{a}^m, \mathbf{p}^m \rangle$
7. $\mathbf{x}^{m+1} = \mathbf{x}^m + \lambda_{\text{opt}}\mathbf{p}^m$

2.5 Preconditioned conjugate gradient method

8. $\mathbf{r}^{m+1} = \mathbf{r}^m - \lambda_{\text{opt}} \mathbf{p}_m$
9. $\mathbf{q}^{m+1} = W^{-1} \mathbf{r}^{m+1}$
10. $\rho^{m+1} = \langle \mathbf{q}^{m+1}, \mathbf{r}^{m+1} \rangle$
11. $\mathbf{p}^{m+1} = \mathbf{q}^{m+1} + \frac{\rho^{m+1}}{\rho^m} \mathbf{p}^m$
12. $m = m + 1$
13. end do

where \mathbf{r} denotes the *residual* of (2.1) defined as $\mathbf{r} = \mathbf{b} - A\mathbf{x}$ and $\langle \cdot, \cdot \rangle$ denotes the *Euclidean scalar product*.

The CG method preconditioned by the incomplete Cholesky factorization (ICCG) is often used as an iterative solver [37]. In this work, the SGS iteration preconditioned CG method (SGSCG) will be used as the smoother of the MG method. Additionally, the MG method as the preconditioner of the CG method (MGCG) [12] will be investigated and its performance for solving linear systems of equations will be compared to that of the ICCG method.

Multigrid for finite elements

3.1 $H^1(\Omega)$ and $\mathbf{H}(\text{curl}, \Omega)$ spaces

This section gives a brief introduction to the *Sobolev spaces* [40] used in this work. These spaces are built on the function space $L^2(\Omega)$ which is defined as follows:

In a given domain Ω with the boundary Γ which is composed of the boundary Γ_D with Dirichlet boundary conditions and the boundary Γ_N with Neumann boundary conditions, $L^2(\Omega)$ consists of all the square-integrable functions u in Ω :

$$\int_{\Omega} |u(\mathbf{r})|^2 d\Omega < \infty. \quad (3.1)$$

$L^2(\Omega)$ is a *Hilbert space* with the scalar product

$$(u, v)_{L^2(\Omega)} = \int_{\Omega} u \cdot v d\Omega \quad (3.2)$$

and the norm

$$\|u\|_{L^2(\Omega)} = \sqrt{(u, u)_{L^2(\Omega)}}. \quad (3.3)$$

In following sections of this chapter, the scalar product $(\cdot, \cdot)_{L^2(\Omega)}$ and the norm $\|\cdot\|_{L^2(\Omega)}$ are written as (\cdot, \cdot) and $\|\cdot\|$ for the sake of brevity, respectively.

The other two Hilbert spaces used are

$$H^1(\Omega) = \{v \in L^2(\Omega); \text{grad}v \in (L^2(\Omega))^3\}, \quad (3.4)$$

$$\mathbf{H}(\text{curl}; \Omega) = \{\boldsymbol{\eta} \in L^2(\Omega)^3; \text{curl}\boldsymbol{\eta} \in (L^2(\Omega))^3\} \quad (3.5)$$

as introduced in [41].

The corresponding scalar products and norms of these two function spaces are:

$$(u, v)_{H^1(\Omega)} = (u, v) + (\operatorname{grad}u, \operatorname{grad}v), \quad (3.6)$$

$$\|u\|_{H^1(\Omega)}^2 = \|u\|^2 + \|\operatorname{grad}u\|^2, \quad (3.7)$$

and

$$(\boldsymbol{\xi}, \boldsymbol{\eta})_{\mathbf{H}(\operatorname{curl}, \Omega)} = (\boldsymbol{\xi}, \boldsymbol{\eta}) + (\operatorname{curl}\boldsymbol{\xi}, \operatorname{curl}\boldsymbol{\eta}), \quad (3.8)$$

$$\|\boldsymbol{\eta}\|_{\mathbf{H}(\operatorname{curl}, \Omega)}^2 = \|\boldsymbol{\eta}\|^2 + \|\operatorname{curl}\boldsymbol{\eta}\|^2. \quad (3.9)$$

When Dirichlet boundary conditions are taken into account, these spaces can be defined as

$$H_0^1(\Omega; \Gamma_D) = \{v \in L^2(\Omega); \operatorname{grad}v \in (L^2(\Omega))^3; v = 0 \text{ on } \Gamma_D\}, \quad (3.10)$$

$$\mathbf{H}_0(\operatorname{curl}; \Omega; \Gamma_D) = \{\boldsymbol{\eta} \in L^2(\Omega)^3; \operatorname{curl}\boldsymbol{\eta} \in (L^2(\Omega))^3; \boldsymbol{\eta} \times \mathbf{n} = \mathbf{0} \text{ on } \Gamma_D\} \quad (3.11)$$

where \mathbf{n} is the normal vector to Γ_D .

It is well-known that a vector function whose curl is zero can be represented as the gradient of a scalar function. For an arbitrary scalar function $\varphi \in H_0^1(\Omega; \Gamma_D)$, it satisfies

$$\operatorname{grad}\varphi = \boldsymbol{\eta} \in \mathbf{H}_0^0(\operatorname{curl}; \Omega; \Gamma_D) \quad (3.12)$$

where the space $\mathbf{H}_0^0(\operatorname{curl}; \Omega; \Gamma_D)$ is the kernel of the curl operator and defined as in [41]

$$\mathbf{H}_0^0(\operatorname{curl}; \Omega; \Gamma_D) = \{\boldsymbol{\eta} \in \mathbf{H}_0(\operatorname{curl}; \Omega; \Gamma_D); \operatorname{curl}\boldsymbol{\eta} = \mathbf{0}\}. \quad (3.13)$$

For a vector function $\mathbf{w} \in (L^2(\Omega))^3$, there exist one scalar function $\varphi \in H_0^1(\Omega; \Gamma_D)$ and one vector function $\mathbf{u} \in \mathbf{H}_0(\operatorname{curl}; \Omega; \Gamma_D)$ so that

$$\mathbf{w} = \mathbf{u} + \operatorname{grad}\varphi \quad (3.14)$$

and

$$(\operatorname{grad}\varphi, \operatorname{grad}\psi) = (\mathbf{w}, \operatorname{grad}\psi) \quad \forall \psi \in H_0^1(\Omega; \Gamma_D) \quad (3.15)$$

with $\operatorname{div}\mathbf{u} = 0$ and $\mathbf{u} \cdot \mathbf{n} = 0$ on Γ_N where \mathbf{n} is the unit normal vector to Γ_N satisfied.

The space that consists of functions \mathbf{u} satisfies

$$\mathbf{H}_0^\perp(\operatorname{curl}; \Omega; \Gamma_D) = \{\mathbf{u} \in \mathbf{H}_0(\operatorname{curl}; \Omega; \Gamma_D); \operatorname{div}\mathbf{u} = 0\}. \quad (3.16)$$

This is the space of the vector functions in $\mathbf{H}_0(\text{curl}; \Omega; \Gamma_D)$ with their divergence equal to zero.

Derived from equation (3.14) called the *Helmholtz decomposition* of a vector function, the Helmholtz decomposition for the space $\mathbf{H}_0(\text{curl}; \Omega; \Gamma_D)$ can be written as

$$\mathbf{H}_0(\text{curl}; \Omega; \Gamma_D) = \mathbf{H}_0^o(\text{curl}; \Omega; \Gamma_D) \oplus \mathbf{H}_0^\perp(\text{curl}; \Omega; \Gamma_D) \quad (3.17)$$

where \oplus denotes the sum of two function spaces.

3.2 Finite elements

3.2.1 Introduction to finite elements

A *finite element* defined in [42] is a triple (T, P, Σ) , where

1. T , a finite element, is a closed subset of \mathbb{R}^3 with non-empty interior and the Lipschitz-continuous boundary.
2. P is a space that contains polynomials on T with the dimension k .
3. Σ is a set of linear functionals on P . These linear functionals are called degrees of freedom of the finite element.

The most characteristic aspect of the finite element method is that the triangulation \mathcal{T}_h is established over the domain $\bar{\Omega}$, i.e., the set $\bar{\Omega}$ is subdivided into a finite number of subsets T such that the following conditions are satisfied [42]:

1. $\bar{\Omega} = \cup_{T \in \mathcal{T}_h} T$.
2. For each $T \in \mathcal{T}_h$, the set T is closed and the interior $\overset{o}{T}$ is non-empty.
3. For each distinct $T_1, T_2 \in \mathcal{T}_h$, one has $\overset{o}{T}_1 \cap \overset{o}{T}_2 = 0$.
4. For each $T \in \mathcal{T}_h$, the boundary ∂T is Lipschitz-continuous.

Over the domain $\bar{\Omega}$, the finite element space of nodal elements and edge elements presented in [42, 43] are introduced in the following sections, respectively.

3.2.2 Nodal elements

Scalar functions can be discretized by nodal elements. The space spanned by the k th-order finite elements is denoted by $\mathcal{S}_k(\mathcal{T}_h)$. This subspace of $H^1(\Omega)$ consists of piecewise continuous polynomials of degree k . The interpolation operator that projects $H^1(\Omega)$ onto $\mathcal{S}_k(\mathcal{T}_h)$ is denoted by $I_k^h : H^1(\Omega) \mapsto \mathcal{S}_k(\mathcal{T}_h)$. The degrees of freedom $\alpha_n \in \Sigma$ at the node n are written as $\alpha_n(u)$ with $u \in \mathcal{S}_k(\mathcal{T}_h)$.

With homogeneous Dirichlet boundary conditions on Γ_D considered, the space $\mathcal{S}_k(\mathcal{T}_h; \Gamma_D)$, a subspace of $H_0^1(\Omega; \Gamma_D)$, is called the finite element space of nodal elements with Dirichlet boundary conditions on Γ_D :

$$\mathcal{S}_k(\mathcal{T}_h; \Gamma_D) = \{\varphi_h \in \mathcal{S}_k(\mathcal{T}_h); \varphi_h = 0 \text{ on } \Gamma_D\}. \quad (3.18)$$

For brevity, $\mathcal{S}_k(\mathcal{T}_h)$ and $\mathcal{S}_k(\mathcal{T}_h; \Gamma_D)$ are written as \mathcal{S}_h^k and $\mathcal{S}_{h,D}^k$, respectively.

In this work, second-order ($k = 2$) hexahedral finite elements with 20 nodes shown in Fig. 3.1(a) are used. The finite element has local coordinates ξ , η , ζ which all range from -1 to 1 . The global coordinate system and the local coordinate system are related by the following coordinate transformation:

$$\begin{aligned} x(\xi, \eta, \zeta) &= \sum_{i=1}^{20} N_i(\xi, \eta, \zeta) x_i, \\ y(\xi, \eta, \zeta) &= \sum_{i=1}^{20} N_i(\xi, \eta, \zeta) y_i, \\ z(\xi, \eta, \zeta) &= \sum_{i=1}^{20} N_i(\xi, \eta, \zeta) z_i \end{aligned} \quad (3.19)$$

where (x_i, y_i, z_i) are the coordinates of the i -th node, and $N_i \in \mathcal{S}_{h,D}^k$ is the basis function for the i -th node given by

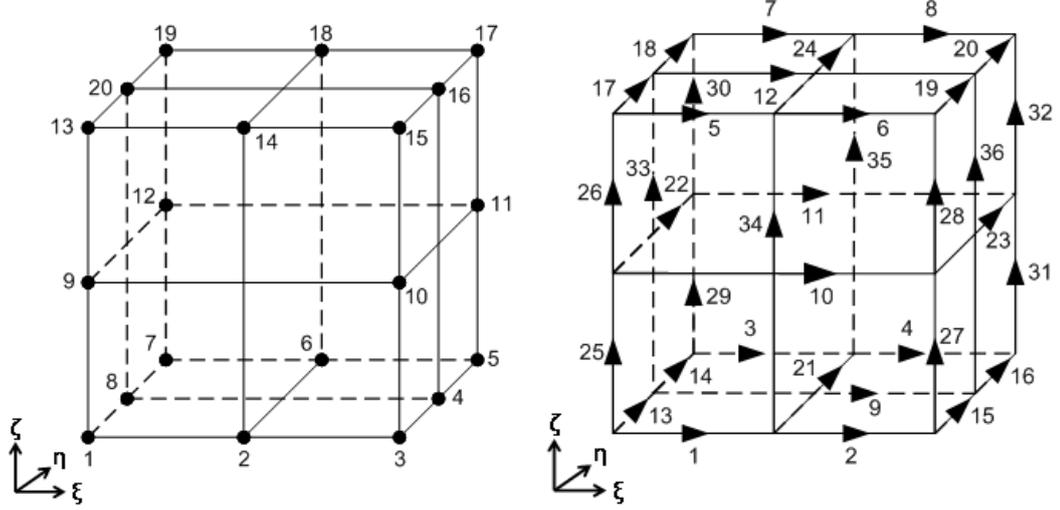
$$N_i = \frac{1}{8}(1 + \xi\xi_i)(1 + \eta\eta_i)(1 + \zeta\zeta_i)(\xi\xi_i + \eta\eta_i + \zeta\zeta_i - 2) \quad (3.20)$$

for the corner nodes and

$$N_i = \frac{1}{4}(1 - \xi^2)(1 + \eta\eta_i)(1 + \zeta\zeta_i) \quad (3.21)$$

for the midside nodes $i = 4, 8, 16, 20$ and

$$N_i = \frac{1}{4}(1 - \eta^2)(1 + \xi\xi_i)(1 + \zeta\zeta_i) \quad (3.22)$$



(a) Second order hexahedral finite elements with 20 nodes (b) Second order hexahedral finite elements with 36 edges

Figure 3.1 Second order hexahedral finite elements

for the midside nodes $i = 2, 6, 14, 18$ and

$$N_i = \frac{1}{4}(1 - \zeta^2)(1 + \xi\xi_i)(1 + \eta\eta_i) \quad (3.23)$$

for the midside nodes $i = 9, 10, 11, 12$. Here (ξ_i, η_i, ζ_i) are the local coordinates of the i -th node.

3.2.3 Edge elements

Edge elements, in contrast to nodal elements with scalar basis functions, have vector basis functions with amplitudes and directions. In the three-dimensional case, with vector functions in \mathbb{R}^3 , P is a subspace of $(C^\infty(\bar{T}))^3$. For any $\mathbf{u} \in (C^\infty(\bar{T}))^3$, one can define a unique interpolation $\Pi\mathbf{u}$ so that for the degrees of freedom $\alpha_i(\cdot) \in \Sigma$, one can obtain

$$\alpha_i(\mathbf{u} - \Pi\mathbf{u}) = 0, \forall \alpha_i(\cdot) \in \Sigma; \Pi\mathbf{u} \in P \quad (3.24)$$

as defined in [43].

Let T_1 and T_2 be two elements with a common face f and $\mathbf{u} \in (C^\infty(\overline{T_1 \cup T_2}))^3$ so that the functions \mathbf{v} defined by $\mathbf{v} = \Pi_1 \mathbf{u}$ on T_1 , $\mathbf{v} = \Pi_2 \mathbf{u}$ on T_2 , belong to $\mathbf{H}(\text{curl}; T_1 \cup T_2)$. A finite element is *conforming* in $\mathbf{H}(\text{curl}; \Omega)$ if and only if

1. The tangential components of $\Pi_1 \mathbf{u}$ and $\Pi_2 \mathbf{u}$ are the same on the boundary face f .
2. For any $\alpha_i(\cdot) \in \Sigma$ defined only on the face f , $\alpha_i(\mathbf{p}) = 0$ with $\mathbf{p} \in P$, the tangential component of \mathbf{p} on the face f equals zero, i.e.

$$\mathbf{n} \times \mathbf{p} = \mathbf{0} \quad \text{on } f \quad (3.25)$$

where \mathbf{n} is the unit normal vector to the face f .

Finite elements, which follow the above properties, are called *edge elements* or *Nédélec's elements* [43].

The edge elements used in this work are conforming hexahedral edge elements in $\mathbf{H}(\text{curl}; \Omega)$ defined as:

Definition 6. For each index k , a conforming hexahedral edge element $(T, \mathcal{N}(T), \Sigma)$ is defined as follows:

$$\mathcal{N}_k(T) = \{\mathbf{u}; u_1 \in Q_{k-1,k,k}; u_2 \in Q_{k,k-1,k}; u_3 \in Q_{k,k,k-1}\} \quad (3.26)$$

where Q_{k_1,k_2,k_3} is the space of polynomials of degree $k \leq k_i$ in the i th coordinate direction, $i = 1, 2, 3$.

The discretization $\mathcal{T}_h := \{T_i\}_i$ of Ω consists of the hexahedral elements T_i so that the global finite element space of order k is defined as

$$\mathcal{N}_k(\mathcal{T}_h) = \{\boldsymbol{\eta}_h \in \mathbf{H}(\text{curl}; \Omega); \boldsymbol{\eta}_{h,T} \in \mathcal{N}_k(T) \forall T \in \mathcal{T}_h\}. \quad (3.27)$$

Vector functions in $\mathcal{N}_k(\mathcal{T}_h)$ have the property that the continuity of their tangential components at the interface of the finite elements is enforced, but not that of their normal components, which means that the normal components are free to jump at the interface.

Definition 7. The degree of freedom $\alpha_e \in \Sigma$ is defined by the functional

$$\alpha_e(\mathbf{u}) = \int_e \mathbf{u} \cdot d\mathbf{l} \quad (3.28)$$

where e denotes edges of finite elements. Based on the degrees of freedom, the interpolation operators Π_k^h onto $\mathcal{N}_k(\mathcal{T}_h)$ is defined in [15].

With homogeneous Dirichlet boundary conditions on Γ_D one can define

$$\mathcal{N}_{k,0}(\mathcal{T}_h; \Gamma_D) = \{\boldsymbol{\eta}_h \in \mathcal{N}_k(\mathcal{T}_h); \boldsymbol{\eta}_h \times \boldsymbol{n} = \mathbf{0} \text{ on } \Gamma_D\} \quad (3.29)$$

where \boldsymbol{n} is the normal vector to Γ_D . In following sections, $\mathcal{N}_k(\mathcal{T}_h)$ and $\mathcal{N}_{k,0}(\mathcal{T}_h; \Gamma_D)$ are written as \mathcal{N}_h^k and $\mathcal{N}_{h,D}^k$ for the sake of brevity, respectively.

In this work, second-order ($k = 2$) hexahedral finite elements with 36 edges [44] as shown in Fig. 3.1(b) are used. Among the 12 edges along each of ξ , η and ζ directions, 8 edges lie along the edges of the element and 4 edges are on the surface of the element. For edges in the ξ direction, the shape functions $N_i \in \mathcal{N}_{h,D}^k$ are expressed as

$$N_i = \frac{1}{8}(1 + \xi\xi_i)(1 + \eta\eta_i)(1 + \zeta\zeta_i)(\xi\xi_i + \eta\eta_i + \zeta\zeta_i - 2)\text{grad}\xi \quad (3.30)$$

for edges along the edges of the element and

$$N_i = \frac{1}{4}(1 + \zeta\zeta_i)(1 + \eta\eta_i)(1 - \xi^2)\text{grad}\xi \quad (3.31)$$

for edges on the surface of the element. The shape functions for edges in the η - and ζ - directions can be obtained by the permutations of ξ , η and ζ .

3.2.4 Relation between the nodal element space and the edge element space

The relation between the nodal element space and the edge element space can be described by the following diagram [15]:

Definition 8. For all $k \geq 1$:

$$\begin{array}{ccc} \mathcal{D}(I_k^h) \subset H^1(\Omega) & \xrightarrow{\text{grad}} & \mathcal{D}(\Pi_k^h) \subset \mathbf{H}(\text{curl}; \Omega) \\ \downarrow I_k^h & & \downarrow \Pi_k^h \\ \mathcal{S}_h^k & \xrightarrow{\text{grad}} & \mathcal{N}_h^k \end{array} \quad (3.32)$$

where $\mathcal{D}(X)$ represents the domain of definition of the operator X . From the diagram it can be seen that for all $\varphi \in \mathcal{D}(I_k^h)$

$$\text{grad}I_k^h\varphi = \Pi_k^h\text{grad}\varphi. \quad (3.33)$$

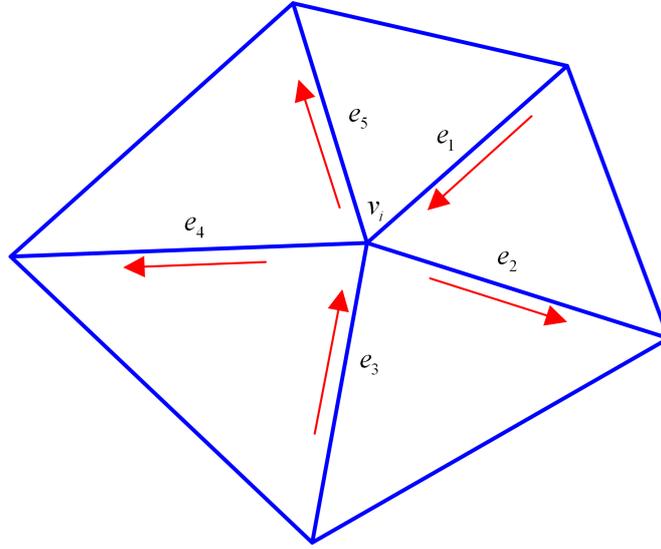


Figure 3.2 First-order triangular elements containing node v_i . The gradient of the scalar basis function associated with the node v_i is the linear combination of vector basis functions of all edges containing the node v_i .

Applying the relation $\text{grad}\varphi = \boldsymbol{\eta} \in \ker(\text{curl})$ in (3.12) to (3.33) and substituting $\text{grad}I_k^h\varphi$ by \mathcal{S}_h^k lead to

$$\text{grad}\mathcal{S}_h^k = \boldsymbol{\eta}_h \in \mathcal{N}_h^{k,0} \quad (3.34)$$

where $\mathcal{N}_h^{k,0}$ is defined as

$$\mathcal{N}_h^{k,0} = \{\boldsymbol{\eta}_h \in \mathcal{N}_h^k; \text{curl}\boldsymbol{\eta}_h = \mathbf{0}\}. \quad (3.35)$$

This indicates that the kernel of the curl operator in the finite element space can be written as the gradient of nodal basis functions. Thus, the following relation is satisfied:

$$\text{grad}N_i = \sum_{j=1}^{n_e} g_{ij}\mathbf{N}_j, \quad i = 1, 2, \dots, n_n \quad (3.36)$$

where N_i and \mathbf{N}_j are the nodal and edge basis functions (see 3.2.2 and 3.2.3), respectively, n_n , n_e are numbers of nodes and edges of the finite element grid, respectively, and the weighting values g_{ij} are -1 for edges j with the starting node i , 1 for edges j with the ending node i and elsewhere 0. The n_n by n_e matrix

with entries g_{ij} is the incidence matrix of the finite element mesh multiplied by -1 and is denoted by G .

For instance, as shown in Fig. 3.2, for a scalar basis function $\psi_h(\mathbf{r}) \in \mathcal{S}_h^1$ associated with the node v_i , the gradient can be easily calculated: according to (3.36), the linear combination of vector basis functions of all edges emanating from the node v_i is equal to $\text{grad}\psi_h(\mathbf{r})$.

The discrete form of the space $\mathbf{H}_0^\perp(\text{curl}; \Omega; \Gamma_D)$ can be written as in [41],

$$\mathcal{N}_{h,D}^{k,\perp} = \{ \boldsymbol{\eta}_h \in \mathcal{N}_{h,D}^k; (\boldsymbol{\eta}_h, \text{grad}\psi_h) = \mathbf{0} \forall \psi \in \mathcal{S}_{h,D}^k \} \quad (3.37)$$

so that the Helmholtz decomposition of the discrete form of the function space is described as follows [41]:

Theorem 2. For each function $\mathbf{w}_h \in \mathcal{N}_{h,D}^k$, there exist one function $\mathbf{u}_h \in \mathcal{N}_{h,D}^{k,\perp}$ and $\varphi_h \in \mathcal{S}_{h,D}^k$ so that

$$\mathbf{w}_h = \mathbf{u}_h + \text{grad}\varphi_h. \quad (3.38)$$

Together with (3.37) one can find that the function $\text{grad}\varphi_h$ satisfies

$$(\text{grad}\varphi_h, \text{grad}\psi_h) = (\mathbf{w}_h, \text{grad}\psi_h) \forall \psi \in \mathcal{S}_{h,D}^k. \quad (3.39)$$

3.3 Hierarchy of grids

For the geometric MG algorithm, a hierarchy of finite element spaces must be constructed. Let $\mathcal{X}(\mathcal{T}_{h_0})$ be a finite dimensional finite element subspace based on the finite element grid \mathcal{T}_{h_0} which is regarded as the coarse grid with h_0 defined as the discretization parameter, i.e., the minimum characteristic size of the finite elements. Each element in the coarse grid can be subdivided in order to obtain the finite element space $\mathcal{X}(\mathcal{T}_{h_1})$ based on the grid \mathcal{T}_{h_1} which is considered as the fine grid.

To simplify the notations, $\mathcal{X}(\mathcal{T}_{h_l})$ for $0 \leq l \leq l_{\max}$ is written as \mathcal{X}_l :

$$\mathcal{X}_0 \subset \mathcal{X}_1 \subset \cdots \mathcal{X}_{l_{\max}} \quad (3.40)$$

where l is the level number of the grids. The systems of equations corresponding to \mathcal{X}_l are

$$A_l \mathbf{x}_l = \mathbf{b}_l \quad (l = 0, 1, 2, \dots, l_{\max}) \quad (3.41)$$

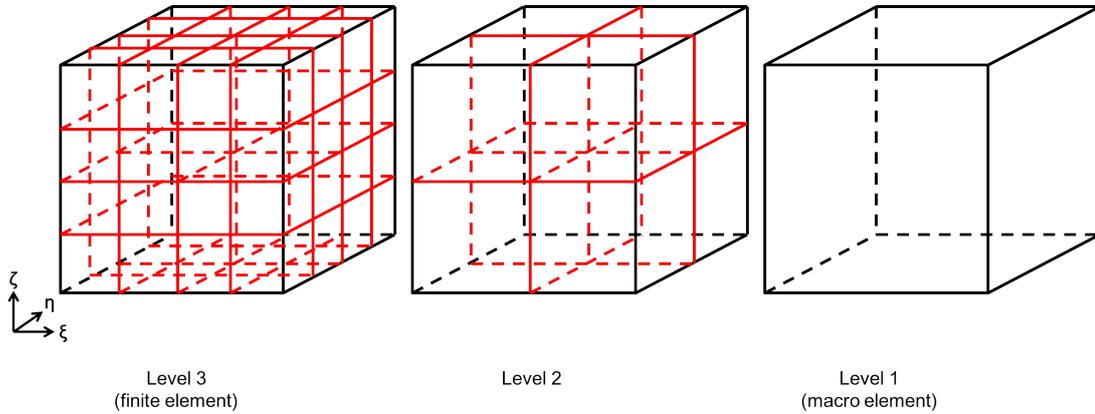


Figure 3.3 Hierarchy of multigrid grids with three levels. The macro element is indicated by black, and the lines representing subdivisions are indicated by red.

with the dimension of equations n_l . The system matrices A_l and right hand side vectors \mathbf{b}_l are calculated during the pre-processing.

There are two main approaches to construct the hierarchy of grids. In the bottom up approach, only the coarsest grid \mathcal{T}_0 is available. The elements in the coarse grid are subdivided to obtain fine grids \mathcal{T}_l , for example, $h_{l+1} = \frac{h_l}{2}$, which results in a nested hierarchy of finite element spaces $\mathcal{X}_0 \subset \mathcal{X}_1 \subset \dots \mathcal{X}_{l_{\max}}$. In the other approach called top down approach, the finest grid $\mathcal{T}_{l_{\max}}$ is available. Finite elements $T_{l_{\max}-1} \in \mathcal{T}_{l_{\max}-1}$ can be obtained by combining the elements $T_{l_{\max}} \in \mathcal{T}_{l_{\max}}$, for example, $T_{l_{\max}-1} = T'_{l_{\max}} \cup T''_{l_{\max}}$ where $T'_{l_{\max}}, T''_{l_{\max}} \in \mathcal{T}_{l_{\max}}$, which also results in $\mathcal{X}_{l-1} \subset \mathcal{X}_l$.

In this work a method which combines both approaches is used to construct the hierarchy of grids. For a problem discretized by the finite element method, its geometry is described by elements called *macro elements* T_{mac} as shown on the right hand side of Fig. 3.3. The grid composed of the macro element $T_{\text{mac}} = T_0$ is considered as the coarse grid \mathcal{T}_0 .

The macro elements are subdivided individually to generate finite elements to form finer grids. For this purpose, macro elements are subdivided in all three local directions ξ, η, ζ with subdivisions denoted by u_i ($i = \xi, \eta, \zeta$). For brevity, u_i ($i = \xi, \eta, \zeta$) is written as u_l . When a finest grid as shown in the graph on the left side of Fig. 3.3 with the subdivision $u_{l_{\max}}$ is available and the subdivision for the coarsest grid u_0 is set to 1, a sequence of subdivisions $\{u_1, u_2, \dots, u_{l_{\max}-1}\}$

can be calculated by

$$u_l = \text{Round}(u_{l_{\max}}^p), \quad p = l/l_{\max}, \quad (3.42)$$

and the corresponding grids with level $l = 1, 2, \dots, l_{\max} - 1$ can be produced. For the case shown in Fig. 3.3, the hierarchy of grids is composed of three levels, and their corresponding subdivisions are 1, 2, and 4 from the coarsest grid to the finest grid.

For cases with multiple macro elements, Table 3.1 gives an example on how to calculate the number of subdivisions u_l of the coarser grids from the subdivision $u_{l_{\max}}$ of the finest grid. The example consists of five macro elements, and the corresponding subdivisions $u_{l_{\max}}$ for the finest grid are 2, 3, 6, 9, 11 listed in the first row of the table, respectively. In the corresponding rows below, the subdivisions u_2 and u_3 for grids on level two and level one are listed.

Table 3.1 Subdivisions u_l for different finest subdivision $u_{l_{\max}}$ with $l_{\max} = 3$

Level	No. of subdivisions u_l				
3	2	3	6	9	11
2	2	2	3	4	5
1	1	1	2	2	2
0	1	1	1	1	1

3.4 Restriction and prolongation

The restriction R is an injection that maps functions on the fine grid h onto functions on the coarse grid H . The restriction for nodal elements can be generated in the following way:

If $\mathcal{S}_H \subset \mathcal{S}_h$ holds, there is an $n_H \times n_h$ matrix R^v with

$$N_i^H = \sum_{j=1}^{n_h} r_{ij}^v N_j^h \quad (3.43)$$

where $i = 1, 2, \dots, n_H$ and $j = 1, 2, \dots, n_h$, r_{ij}^v are entries of the matrix R^v , and N_i^H and N_j^h are the basis functions of the coarse grid and the fine grid (see 3.2.2), respectively.

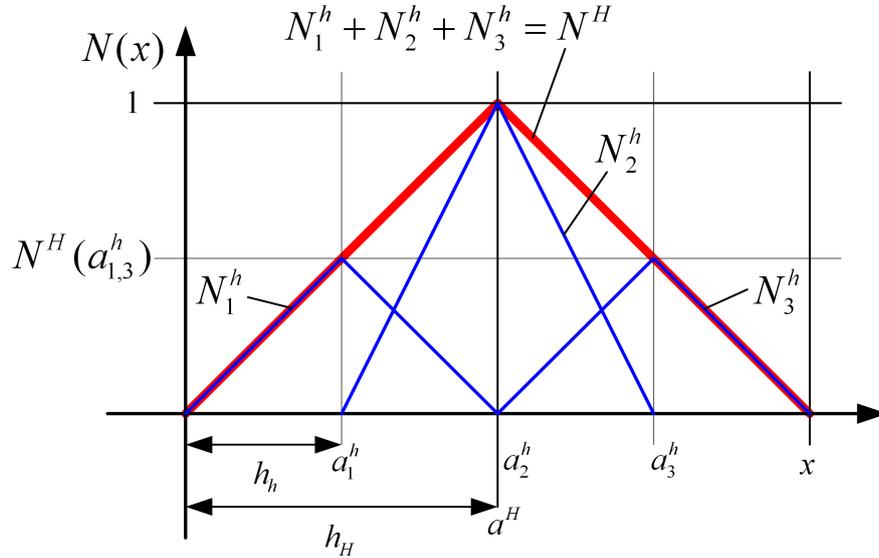


Figure 3.4 The nodal basis function N^H of the coarse grid and the nodal basis functions N_i^h of the fine grid. Entries of the restriction matrix are values of the coarse grid basis function at nodes of the fine grid.

For each $v \in \mathcal{S}_h$, one can find

$$v = \sum_{j=1}^{n_h} v(a_j^h) N_j^h \quad (3.44)$$

where $v(a_j^h)$ are values at the nodes a_j^h of the fine grid. It is known that $N_i^H \in \mathcal{S}_H$ can be expressed as

$$N_i^H = \sum_{j=1}^{n_h} N_i^H(a_j^h) N_j^h. \quad (3.45)$$

Compared to (3.43), it can be easily seen that

$$r_{ij}^v = N_i^H(a_j^h), \quad (3.46)$$

which means that entries of the restriction matrix are values of the coarse grid basis function at nodes of the fine grid. The construction of the restriction matrix R^v in a one-dimensional problem is illustrated in Fig. 3.4.

The transpose of R^v is used as the prolongation matrix P^v , i.e.:

$$P^v = (R^v)^T. \quad (3.47)$$

3.4 Restriction and prolongation

For edge elements, a vector function $\mathbf{A} \in \mathbf{H}(\text{curl}; \Omega)$ on the coarse grid H can be approximated by

$$\mathbf{A} \approx \mathbf{A}_H = \sum_{j=1}^{n_H} a_{H,j} \mathbf{N}_{H,j} \quad (3.48)$$

where $\mathbf{N}_{H,j} \in \mathcal{N}_{H,D}^k$ is the vector basis function associated with the j -th edge on the coarse grid (see 3.2.3). In order to obtain $a_{h,i}$ associated with the i -th edge on the fine grid h , the integration of the vector potential \mathbf{A}_H along edges on the fine grid needs to be done:

$$a_{h,i} = \int_{e_{h,i}} \mathbf{A}_H \cdot d\mathbf{l} \quad (3.49)$$

where $a_{h,i}$ denotes the i -th edge on the fine grid. Substituting equation (3.48) into (3.49) and interchanging the order of the summation and the integration, the degrees of freedom on the fine grid can be represented as a linear combination of integrals of the vector basis functions of the coarse grid along edges of the fine grid:

$$a_{h,i} = \sum_{j=1}^{n_H} \int_{e_{h,i}} a_{H,j} \mathbf{N}_{H,j} \cdot d\mathbf{l}. \quad (3.50)$$

This leads to the matrix P^e [17]:

$$p_{ij}^e = \int_{e_{h,i}} \mathbf{N}_{H,j} \cdot d\mathbf{l}. \quad (3.51)$$

where $1 \leq i \leq n_h$ and $1 \leq j \leq n_H$

To obtain the prolongation matrix, the integrals of the basis functions of the coarse grid have to be calculated on each edge of the fine grid. The restriction matrix R^e is the transpose of the prolongation matrix

$$R^e = (P^e)^T. \quad (3.52)$$

The calculation of the prolongation matrix for a one-dimensional problem is illustrated in Fig. 3.5.

For problems discretized by both nodal and edge elements, the prolongation matrix can be written as [27]

$$P = \begin{bmatrix} P^e & 0 \\ 0 & P^n \end{bmatrix}. \quad (3.53)$$

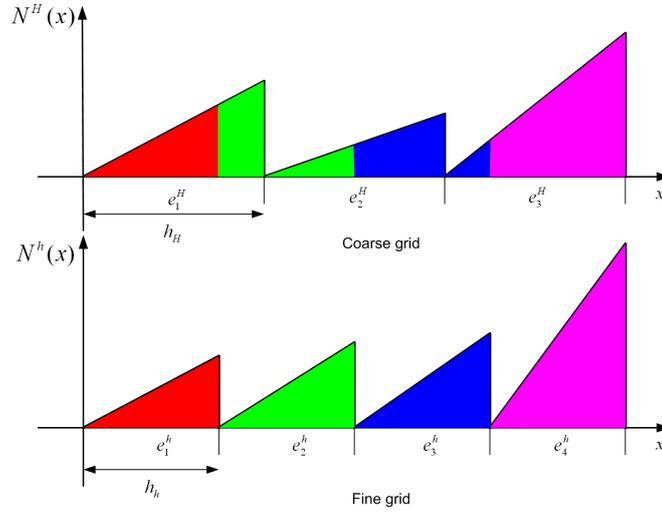


Figure 3.5 The edge basis function N^H of the coarse grid and edge basis functions N_i^h of the fine grid. The prolongation matrix is calculated by the integrals of basis functions of the coarse grid on each edge of the fine grid.

The restriction matrix R is the transpose of the prolongation matrix

$$R = (P)^T. \tag{3.54}$$

3.5 Smoother

As introduced in section 2.2, the task of the smoother $\mathcal{S}_l^\nu(\mathbf{x}_l, \mathbf{b}_l)$ is to damp the high frequency components of the error \mathbf{e}_l^m . After ν steps of smoothing iterations, a better smoothing measure $\|A_l \mathcal{S}_l^\nu \mathbf{e}_l\|_2$ can be obtained [42]. For a good smoother, the number ν should be small.

For both nodal and edge elements, the SGS iteration is often used as the smoother. A better smoothing effect can be obtained by using the SGSCG iteration instead of the SGS iteration as the smoother [21, 28]. Since both SGS and SGSCG iterations solve the linear system of equations with respect to components of points or edges, they are called *pointwise* smoothers when they are applied to the MG method.

Another type of smoothers is called *block* smoother or *blockwise* smoother based on block iterations where the system of equations is solved with respect to blocks

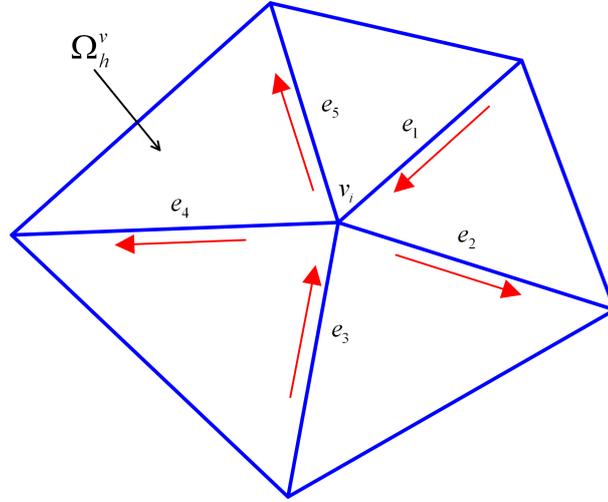


Figure 3.6 AFW domain decomposition for first-order triangular elements. In the subdomain Ω_h^v corresponding to the node v_i , the function space \mathcal{N}_h^v is the space spanned by the vector basis function \mathbf{N}_e of edges e_1 to e_5 .

consisting of groups of points or edges [36]. The block iteration is a *domain decomposition* method whereby the system of equations $A\mathbf{x} = \mathbf{b}$ representing the discretization of a complete boundary value problem in the domain Ω is split into smaller systems of equations corresponding to boundary value problems in subdomains $\Omega' \subset \Omega$. Let $\mathbf{X} = \mathbb{R}^n$ be the linear space containing the solution \mathbf{x} .

In this work, the *multiplicative Schwarz* iteration (MSI) (also called block GS iteration) is applied to the MG method as the smoother [45].

3.5.1 Multiplicative Schwarz iteration

The space \mathbb{R}^n is subdivided into \hat{n} smaller subspaces with the dimension n_i ($i = 1, 2, \dots, \hat{n}$). The solutions corresponding to the lower-dimensional subproblems are represented by $\mathbf{x}_i \in \mathbf{X}_i = \mathbb{R}^{n_i}$. For the linear mapping from \mathbf{X}_i to \mathbf{X} , a prolongation operator denoted by $P_i^s : \mathbf{X}_i \rightarrow \mathbf{X}$ is used and represented by an $n \times n_i$ rectangular matrix. The transpose of the matrix P_i^s is a mapping from \mathbf{X} to \mathbf{X}_i , i.e. a restriction matrix denoted by $R_i^s : \mathbf{X} \rightarrow \mathbf{X}_i$.

For each subproblem, an $n_i \times n_i$ square matrix is defined by

$$A_i := R_i^s A P_i^s, \quad (3.55)$$

and one step of the MSI for updating the unknowns corresponding to the i -th subproblem is written as

$$\mathbf{x}_i^m = \mathbf{x}_i^{m-1} - P_i^s A_i^{-1} R_i^s ((D - U)\mathbf{x}^{m-1} - L\mathbf{x}^m - \mathbf{b}) \quad i = 1, 2, \dots, \hat{n}. \quad (3.56)$$

The iteration matrix of the MSI is

$$M^{\text{MSI}} = (I - P_{\hat{n}})(I - P_{\hat{n}-1}) \dots (I - P_1) \quad (3.57)$$

where

$$P_i = P_i^s A_i^{-1} R_i^s A. \quad (3.58)$$

Similarly to the SGS iteration, the symmetric MSI also consists of one step of forward iteration (3.56) and one step of backward iteration. Therefore, it can be described by the following algorithm:

Algorithm 4. Symmetric Multiplicative Schwarz Iteration

1. $\mathbf{x}_0^{m-1} = \mathbf{x}^{m-1}$
2. $\mathbf{x}_i^m = \mathbf{x}_i^{m-1} - P_i^s A_i^{-1} R_i^s ((D - U)\mathbf{x}^{m-1} - L\mathbf{x}^m - \mathbf{b}) \quad i = 1, 2, \dots, \hat{n}$
3. $\mathbf{x}_i^m = \mathbf{x}_i^{m-1} - P_i^s A_i^{-1} R_i^s ((D - L)\mathbf{x}^{m-1} - U\mathbf{x}^m - \mathbf{b}) \quad i = \hat{n}, \hat{n} - 1, \dots, 1$
4. $\mathbf{x}^m = \mathbf{x}_{2\hat{n}}^m$

3.5.2 Arnold, Falk and Winther smoother

A domain decomposition method of the finite element space for tetrahedral elements has been introduced by Arnold, Falk and Winther (AFW) in [16].

Definition 9 (AFW Block). Let \mathcal{T}_h be the finite element grid in the domain Ω_h . This domain can be subdivided into subdomains Ω_h^v formed by elements sharing a node v

$$\Omega_h^v = \text{interior} \{T \in \mathcal{T}_h | v \in T\}. \quad (3.59)$$

Using this definition, the decomposition of the finite element space can be described as

$$\mathcal{N}_h = \sum_v \mathcal{N}_h^v \quad (3.60)$$

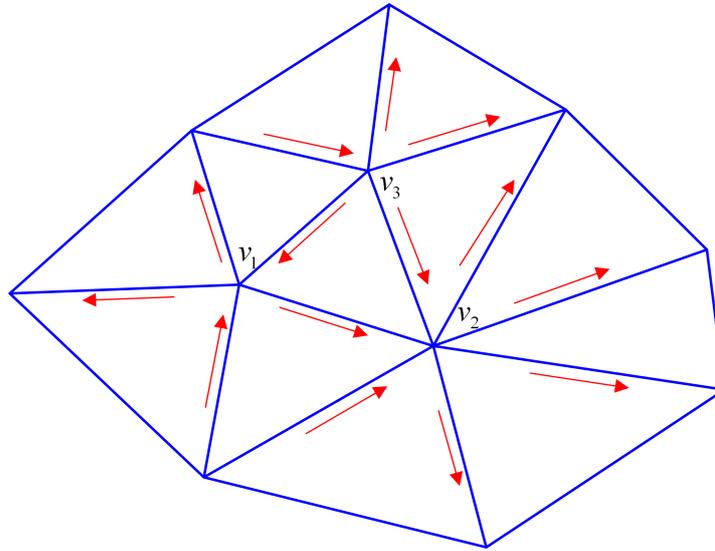


Figure 3.7 A large AFW block is formed by combining multiple AFW blocks.

where $\mathcal{N}_h^v = \{N \in \mathcal{N}_h : \text{supp}N \subset \overline{\Omega_h^v}\}$ with $\text{supp}N$ defined as the closure in \mathcal{N}_h of the set $\{x \in \mathcal{N}_h : N(x) \neq \mathbf{0}\}$. For first-order triangular elements, the subspace \mathcal{N}_h^v is the space spanned by vector basis functions corresponding to edges emanating from the vertex v as shown in Fig. 3.6. Multiple AFW blocks can be combined together and form a larger block as shown in Fig. 3.7. All vector basis functions whose support lies in $\overline{\Omega_h^v}$ and of edges in those finite elements contained in the larger block form the corresponding finite element space.

For second-order hexahedral elements in a regular mesh, on the one hand, the subdomain Ω_h^v for a corner node consists of all eight elements containing the node, the schematic illustration of the AFW domain decomposition for second order hexahedral elements is depicted in Fig. 3.8. From the picture it can be seen that an AFW block for second-order hexahedral elements consists of edges containing the corner node denoted by red, as well as edges emanating from mid-side nodes denoted by green so that the patch for the corner node is formed by 36 edges. On the other hand, for non-regular meshes, the block is formed by edges containing the corner node and can be extended to all edges of elements sharing the corner nodes [18].

Since edges containing mid-side nodes are already included in the block, only eight corner nodes have been used to generate the patches so that this decomposition results in the reduction of the number of blocks by about 75% compared

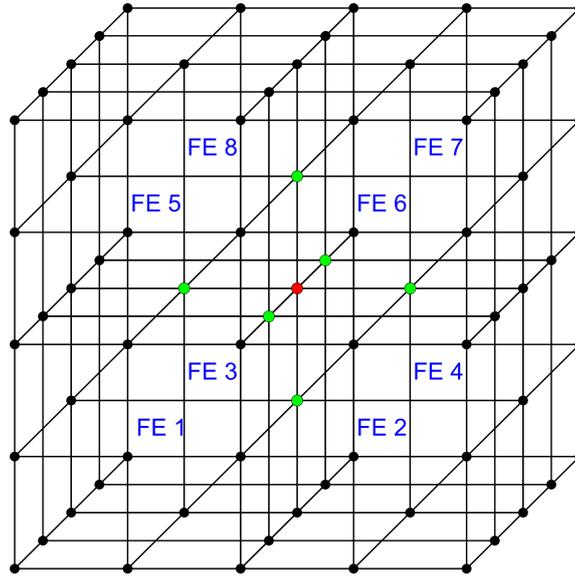


Figure 3.8 The AFW block for second-order hexahedral elements. All edges which contain the corner node denoted by red and mid-side nodes denoted by green are included in one block.

to the original one [18].

The assembly and inversion of block matrices based on the AFW block is done in the pre-processing and the corresponding submatrix equation is solved by the MSI algorithm.

3.5.3 Plane smoother

One factor that influences the smoothing capability of the smoother is the quality of the finite element mesh measured by the maximum aspect ratio $\delta_{\max} = h_{\max}/h_{\min}$ which is defined as the ratio of the length of the longest edge h_{\max} of the element to the length h_{\min} of its shortest edge.

For problems discretized by elements with small δ_{\max} as shown in Fig. 3.9(a), a pointwise smoother such as the SGS method or the SGSCG method can result in a highly efficient MG algorithm. However, for problems discretized by thin elements with large values of δ_{\max} as shown in Fig. 3.9(b) where the width of the element is illustrated not to scale, the SGS or the SGSCG iteration smooths the error slowly or even fails. For example, applying the GS iteration in (2.19) to

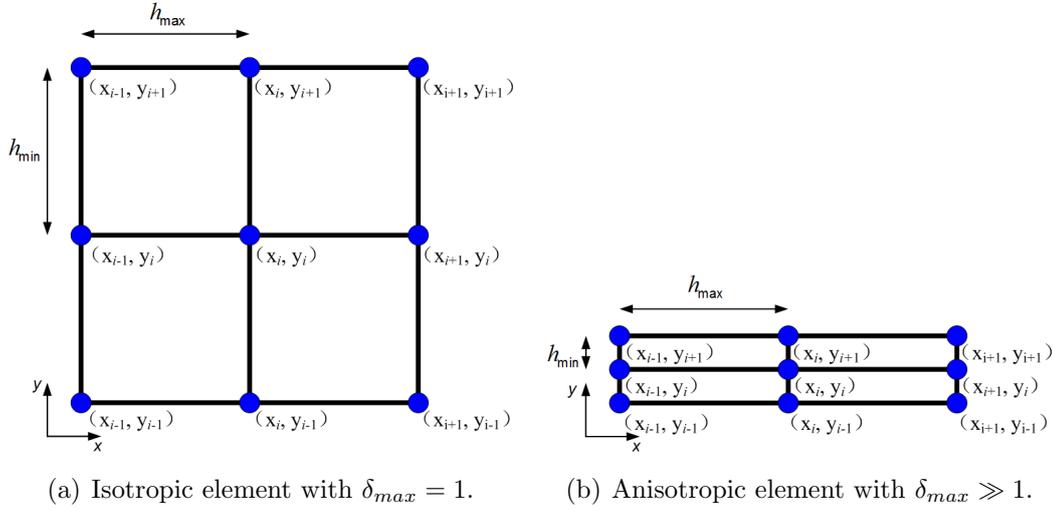


Figure 3.9 Rectangular elements with different values of δ_{\max} .

the 2-D Poisson equation discretized by nodal elements, one can find the error relation [12]:

$$e_h^{m+1}(x_i, y_j) = \frac{1}{2(1 + \frac{1}{\delta_{max}^2})} \left[\frac{1}{\delta_{max}^2} e_h^{m+1}(x_{i-1}, y_j) + \frac{1}{\delta_{max}^2} e_h^m(x_{i+1}, y_j) \right. \\ \left. + e_h^{m+1}(x_i, y_{j-1}) + e_h^m(x_i, y_{j+1}) \right], \quad (3.61)$$

which for $\delta_{max} \rightarrow \infty$ becomes

$$e_h^{m+1}(x_i, y_j) = \frac{1}{2} [e_h^{m+1}(x_i, y_{j-1}) + e_h^m(x_i, y_{j+1})]. \quad (3.62)$$

where x_i and y_j are the coordinates of nodes in the 2D domain as illustrated in Fig. 3.9(b).

It is obvious that there is no averaging effect with respect to the x -direction, which means that no smoothing with respect to this direction is achieved so that the direction x , thus will be called the anisotropic direction of the grid. (For $\delta_{max} = h_y/h_x \gg 1$, it is the other way around.) Such errors which cannot be smoothed by the pointwise smoother can no longer be efficiently reduced by means of the coarse grid correction.

The smoothing effect of the pointwise smoother is very poor with respect to the anisotropic direction of the grid. Therefore, the smoother applied to the MG

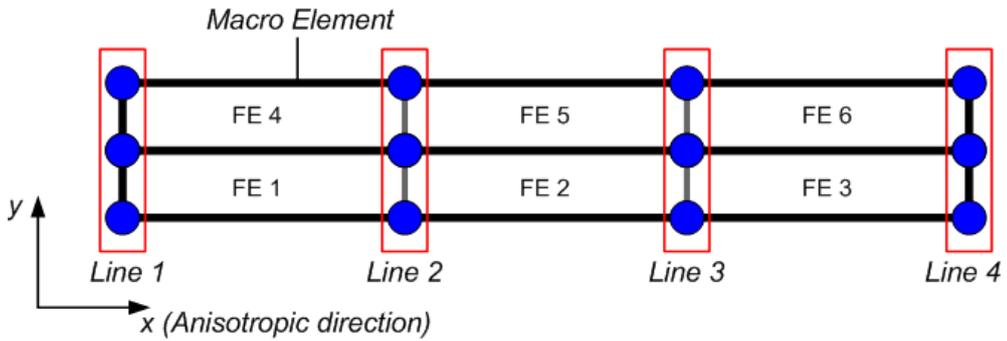


Figure 3.10 The line smoothing block for first-order rectangular nodal elements. All nodes on a line along the y direction are included in one block and four line smoothing blocks are generated from the macro element.

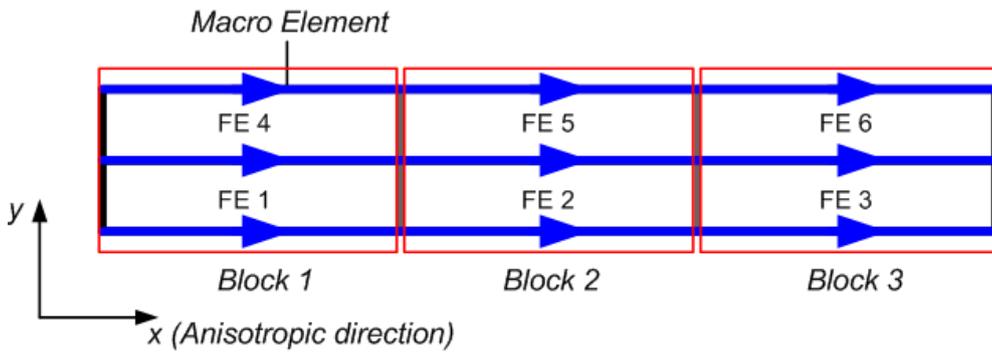


Figure 3.11 The line smoothing block for first-order rectangular edge elements. Three blocks are generated from the macro element.

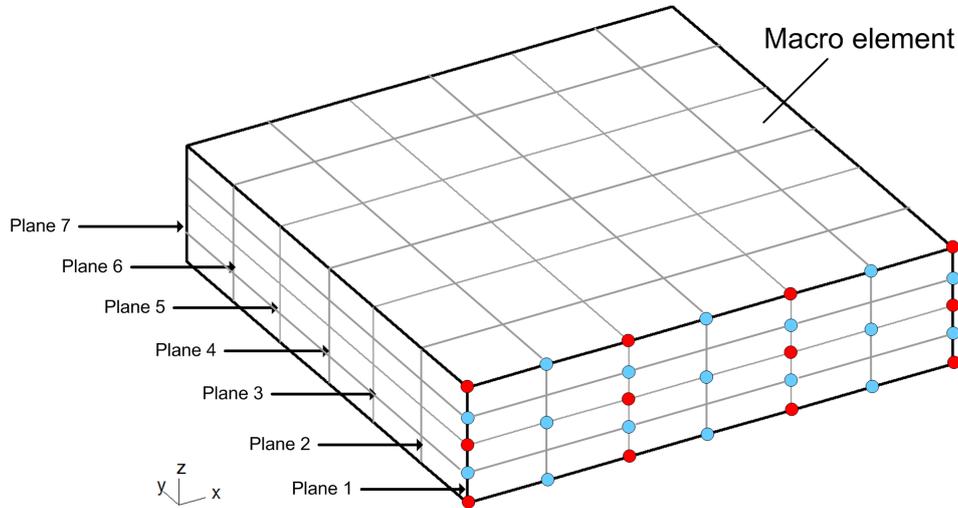


Figure 3.12 The plane smoothing block for second-order hexahedral nodal elements. All nodes on the plane of the macro element are included in one block.

method has to be adapted and tuned to problems discretized by elements with large values of δ_{max} . An alternative approach is to change the smoothing procedure from pointwise smoothing to *line smoothing* [12]. Thereby all unknowns of points which are strongly coupled on a line of the finite element grid, e.g., points on a line along the y axis in Fig. 3.9(b), are included in one block, and the errors of unknowns in the block are smoothed collectively by a block iteration such as the symmetric MSI.

As introduced in section 3.3, the geometry of the problem is described by macro elements T_{mac} which are subdivided into finite elements afterwards. The selection of nodes for the generation of line smoothing blocks is introduced by the example shown in Fig. 3.10. One macro element is subdivided into three finite elements, and subdivisions in x and y directions are 3 and 2, respectively. All nodes on the line along the y direction are contained in one block so that the corresponding submatrix is a tridiagonal matrix. The number of blocks is equal to the number of lines along the y direction in the grid.

Different from the line smoother for nodal elements, for the structure discretized by 2-D first-order edge elements, the edges for one block are selected in the manner

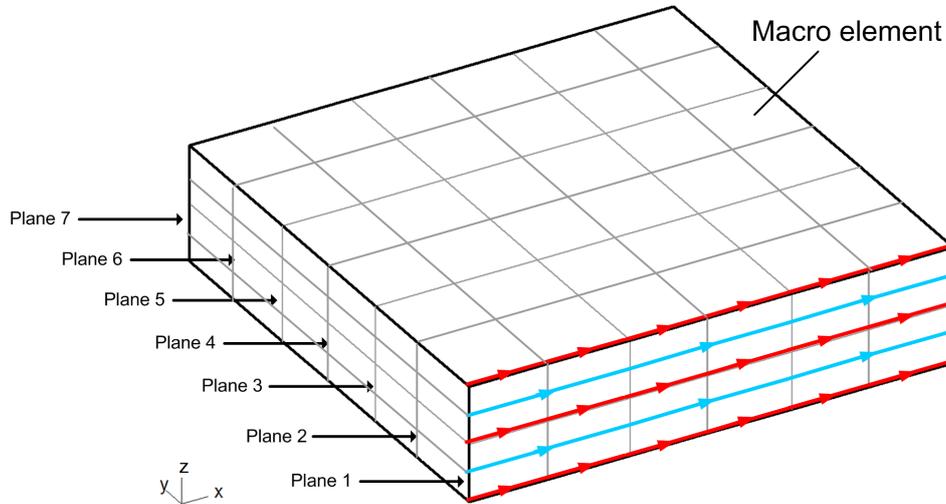


Figure 3.13 The plane smoothing block for second-order hexahedral edge elements. All edges directed along the x direction on the plane of the macro element are included in one block.

illustrated in Fig. 3.11 so that a tridiagonal submatrix associated with the block can be obtained [33]. For example, the block 1 is obtained by including edges parallel to the x axis in elements FE1 and FE4.

For 3-D problems discretized by second-order hexahedral finite elements, instead of the line smoother, the *plane smoother* which is applied to the plane of the macro element needs to be implemented. As shown in Fig. 3.12, one macro element is subdivided into finite elements with subdivisions 3, 3 and 2 in x , y and z -directions, respectively. On one of the x - z planes of the macro element for instance, all nodes on lines of the plane along the y direction are grouped in one block. Roughly speaking, on the one hand, the block contains all nodes on the x - z plane as shown in Fig. 3.12. On the other hand, blocks for edge elements contain all edges along the x -direction as shown in Fig. 3.13. In addition to x - z planes of the macro element, the y - z planes also need to be considered by plane smoothing blocks.

The submatrices corresponding to line or plane smoothing blocks need to be assembled and their inverses need to be calculated in the pre-processing. In this work the MSI based on plane smoothing blocks is used as the smoother of the

MG method.

3.6 Multigrid preconditioned conjugate gradient method

The MG iteration is symmetric if the number of pre- and post-smoothing steps are equal:

$$\nu_1 = \nu_2 = \nu > 0, \quad (3.63)$$

and the iteration matrices M_{pre} and M_{post} of the pre- and post-smoothing iterations are adjoint in the energy inner product:

$$\langle M_{\text{pre}}\mathbf{u}, \mathbf{v} \rangle_A = \langle \mathbf{u}, M_{\text{post}}\mathbf{v} \rangle_A \quad \forall \mathbf{u}, \mathbf{v} \in \mathbb{R}^n. \quad (3.64)$$

which can be also expressed as

$$\langle \mathbf{v}^T A M_{\text{pre}} \mathbf{u} \rangle = \langle \mathbf{v} M_{\text{pre}}^T A \mathbf{u} \rangle \quad \forall \mathbf{u}, \mathbf{v} \in \mathbb{R}^n \quad \text{or} \quad M_{\text{pre}} = M_{\text{post}}^* \quad (3.65)$$

Derived from the above equation, it can be seen that

$$A M_{\text{pre}} = M_{\text{post}}^T A \quad (3.66)$$

needs to be satisfied.

In the case of the SGS iteration as the smoother of the MG method, substituting M_{pre} and M_{post} by the iteration matrices in (2.14) and (2.16) leads to

$$A M_{\text{FGS}} = M_{\text{BGS}}^T A, \quad (3.67)$$

which indicates that (2.14) and (2.16) are adjoint. Therefore, the MG method with the SGS smoother can be used as the preconditioner of the CG method. However, since (3.66) doesn't hold for the SGSCG iteration, the MG method with the SGSCG smoother cannot be the preconditioner of the CG method.

As introduced in (2.5), in the PCG algorithm, the system of equations $\check{A}\check{\mathbf{x}} = \check{\mathbf{b}}$ with $\check{A} = W^{-1/2} A W^{-1/2}$ and $\check{\mathbf{b}} = W^{-1/2} \mathbf{b}$ is solved by the CG method. The matrix W of the symmetric MG iteration is expressed as

$$W_l^{\text{MGI}}(\nu) = (I - M_l^{\text{MGI}}(\nu)) A^{-1} \quad (3.68)$$

Chapter 3 Multigrid for finite elements

where $M_l^{\text{MGI}}(\nu)$ is the iteration matrix of the symmetric MG iteration with ν steps of pre and post-smoothing iterations.

The use of the MGCG method is of practical interest with respect to *robustness* especially cases where the MG method fails to exhibit high efficiency [12]. As problems to be solved become more and more complicated, and anisotropies, nonlinearities or both occur simultaneously, the fundamental idea of the MG method to reduce the high frequency components of the error by smoothing iterations and to take care of the low frequency error components by the coarse grid correction, does not work optimally. Certain error components may remain large after the MG iterations. These specific errors are responsible for the poor MG convergence. In such situations, the combination of the MG method with the CG method may have the potential of a substantial acceleration.

In the next two chapters, the MGCG solver with the AFW smoother as well as the plane smoother will be applied to solve the matrix equations resulting from finite element formulations for magnetostatic problems and time harmonic eddy current problems.

The magnetostatic problem

There are two ways to describe the static magnetic field either by the *magnetic scalar potential* discretized by nodal elements [46] or by the *magnetic vector potential* discretized by edge elements [47].

4.1 Differential equations and boundary conditions

Maxwell's equations for the static magnetic field are

$$\operatorname{curl}\mathbf{H} = \mathbf{J}_0, \quad (4.1)$$

$$\operatorname{div}\mathbf{B} = 0 \quad (4.2)$$

where \mathbf{H} is the magnetic field intensity, \mathbf{B} is the magnetic flux density and \mathbf{J}_0 is the given current density of the current excitation in coils. \mathbf{B} and \mathbf{H} satisfy the following constitutive equation

$$\mathbf{B} = \mu\mathbf{H} \quad \text{or} \quad \mathbf{H} = \nu\mathbf{B} \quad (4.3)$$

with the reluctivity ν and the permeability $\mu = \mu_0\mu_r = 1/\nu$ where μ_0 is the vacuum permeability and μ_r is the relative permeability.

The given current density \mathbf{J}_0 of the current excitation is assumed to be described by an impressed field quantity \mathbf{T}_0 satisfying [47]

$$\operatorname{curl}\mathbf{T}_0 = \mathbf{J}_0. \quad (4.4)$$

Chapter 4 The magnetostatic problem

Instead of using its exact value, \mathbf{T}_0 can be approximated by edge basis functions \mathbf{N}_j (see 3.2.3),

$$\mathbf{T}_0 = \sum_{j=1}^{n_e} t_j \mathbf{N}_j \quad (4.5)$$

where t_j are the line integrals of \mathbf{T}_0 along the edges.

In a domain Ω , the boundary conditions can be given at artificial far boundaries or along symmetry planes. The homogeneous boundary conditions for the magnetic field and flux density are given as follows:

$$\mathbf{H} \times \mathbf{n} = \mathbf{0} \quad \text{on } \Gamma_H, \quad (4.6)$$

$$\mathbf{B} \cdot \mathbf{n} = 0 \quad \text{on } \Gamma_B. \quad (4.7)$$

The boundary conditions (4.6) and (4.7) can be replaced by

$$\mathbf{H} \times \mathbf{n} = \mathbf{T}_0 \times \mathbf{n} \quad \text{on } \Gamma_H, \quad (4.8)$$

$$\mathbf{B} \cdot \mathbf{n} = \mu \mathbf{T}_0 \cdot \mathbf{n} \quad \text{on } \Gamma_B \quad (4.9)$$

which in some cases can simplify the boundary value problems resulting from the potential formulations.

The boundary conditions (4.6) and (4.8) specify the tangential components of the magnetic field and the current density on the boundary surface Γ_H . By (4.7) and (4.9) the normal component of the magnetic flux density on Γ_B is given.

The potential formulations introduced in following sections can be used to describe the static magnetic field either in a magnetostatic problem or in the nonconducting eddy current free region of an eddy current problem.

4.1.1 The $\bar{\Phi}$ -formulation

The magnetic scalar potential $\bar{\Phi}$ is defined by

$$\mathbf{H} = \mathbf{T}_0 - \text{grad} \bar{\Phi} \quad (4.10)$$

where \mathbf{T}_0 is given by (4.4).

The assumption (4.10) exactly satisfies (4.1), and substituting \mathbf{B} in (4.2) by the first relation in (4.3) leads to the differential equation

$$\text{div}(\mu \text{grad} \bar{\Phi}) = \text{div}(\mu \mathbf{T}_0) \quad \text{in } \Omega. \quad (4.11)$$

4.1 Differential equations and boundary conditions

Using the boundary conditions in (4.7) and (4.8) results in the Dirichlet boundary conditions:

$$\Phi = 0 \quad \text{on } \Gamma_H \quad (4.12)$$

and the Neumann boundary conditions:

$$\mu \text{grad} \Phi \cdot \mathbf{n} = \mu \mathbf{T}_0 \cdot \mathbf{n} \quad \text{on } \Gamma_B \quad (4.13)$$

on the scalar potential.

The scalar potential $\Phi \in H_0^1(\Omega; \Gamma_H)$ is approximated by nodal basis functions $N_j \in \mathcal{S}_0(\mathcal{T}_h; \Gamma_H)$ (see 3.2.2) as

$$\Phi \approx \Phi_h = \Phi_D + \sum_{j=1}^{n_n - n_{n_d}} \Phi_j N_j, \quad \Phi_D = \sum_{j=1}^{n_{n_d}} \Phi_j N_j \quad (4.14)$$

where Φ_j are the nodal values of Φ_h , Φ_D represents the magnetic scalar potential of nodes on the Dirichlet boundary, n_n is the total number of nodes, and n_{n_d} is the number of nodes on Γ_H .

Applying *Galerkin's techniques* to (4.11) one can obtain the following algebraic equations:

$$-\int_{\Omega} \text{grad} N_i \cdot \mu \text{grad} \Phi_h d\Omega = -\int_{\Omega} \text{grad} N_i \cdot \mu \mathbf{T}_0 d\Omega, \quad i = 1, 2, \dots, n_n - n_{n_d}. \quad (4.15)$$

The equations (4.15) can be written in a matrix form

$$\mathbf{A} \mathbf{x} = \mathbf{b} \quad (4.16)$$

with entries of \mathbf{A} and \mathbf{b} expressed as

$$a_{ij} = (\text{grad} N_i, \mu \text{grad} N_j), \quad i, j = 1, 2, \dots, n_n - n_{n_d}, \quad (4.17)$$

$$b_i = (\text{grad} N_i, \mu \mathbf{T}_0), \quad i = 1, 2, \dots, n_n - n_{n_d}. \quad (4.18)$$

4.1.2 The \mathbf{A} -formulation

Since the divergence of the magnetic flux density is zero, one can define the magnetic vector potential \mathbf{A} as

$$\mathbf{B} = \text{curl} \mathbf{A}. \quad (4.19)$$

Chapter 4 The magnetostatic problem

Writing (4.1) with the second relation in (4.3) leads to the differential equation

$$\operatorname{curl}(\nu \operatorname{curl} \mathbf{A}) = \operatorname{curl} \mathbf{T}_0 \quad \text{in } \Omega. \quad (4.20)$$

Using the boundary conditions in (4.7) or (4.8) results in the following Dirichlet boundary conditions on the vector potential:

$$\mathbf{A} \times \mathbf{n} = \mathbf{0} \quad \text{on } \Gamma_B \quad (4.21)$$

or the Neumann boundary conditions:

$$\nu \operatorname{curl} \mathbf{A} \times \mathbf{n} = \mathbf{T}_0 \times \mathbf{n} \quad \text{on } \Gamma_H. \quad (4.22)$$

The vector potential $\mathbf{A} \in \mathbf{H}_0(\operatorname{curl}; \Omega; \Gamma_B)$ can be approximated by the edge basis function $\mathbf{N}_i \in \mathcal{N}_0(\mathcal{T}_h; \Gamma_B)$ (see 3.2.3) as

$$\mathbf{A} \approx \mathbf{A}_h = \mathbf{A}_D + \sum_{j=1}^{n_e - n_{e_d}} a_j \mathbf{N}_j, \quad \mathbf{A}_D = \sum_{j=1}^{n_{e_d}} a_j \mathbf{N}_j \quad (4.23)$$

where a_j is the line integral of \mathbf{A} along the edge e_j , \mathbf{A}_D represents the magnetic vector potential of edges on the Dirichlet boundary, n_e is the total number of edges, and n_{e_d} is the number of edges on Γ_B .

Setting (4.23) into (4.20) as well as into the Neumann boundary conditions in (4.22), the following *Galerkin equations* are obtained:

$$\int_{\Omega} \operatorname{curl} \mathbf{N}_i \cdot \nu \operatorname{curl} \mathbf{A}_h \, d\Omega = \int_{\Omega} \operatorname{curl} \mathbf{N}_i \cdot \mathbf{T}_0 \, d\Omega, \quad i = 1, 2, \dots, n_e - n_{e_d}. \quad (4.24)$$

These equations can be written in a matrix form:

$$\mathbf{A} \mathbf{x} = \mathbf{b} \quad (4.25)$$

where

$$a_{ij} = (\operatorname{curl} \mathbf{N}_i \cdot \nu \operatorname{curl} \mathbf{N}_j), \quad i, j = 1, 2, \dots, n_e - n_{e_d}, \quad (4.26)$$

$$b_i = (\operatorname{curl} \mathbf{N}_i \cdot \mathbf{T}_0), \quad i = 1, 2, \dots, n_e - n_{e_d}. \quad (4.27)$$

The right hand side of (4.25) is consistent since the same linear interdependence among its elements is present as among the rows of the left-hand side.

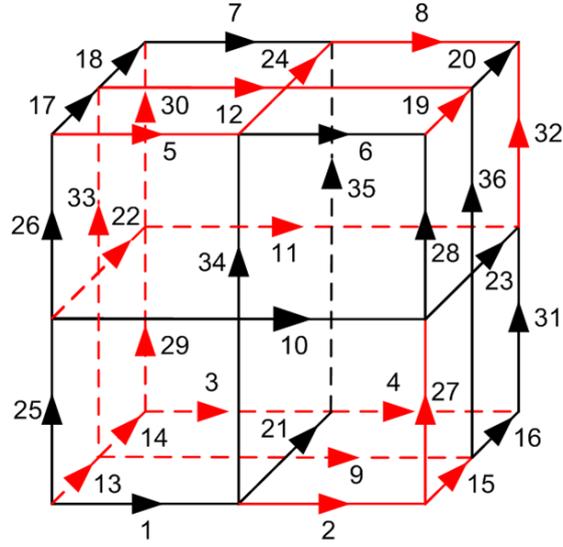


Figure 4.1 A tree-cotree graph of the second-order hexahedral edge elements with tree edges denoted by red and cotree edges denoted by black.

4.2 Tree and Cotree

The sum of all nodal basis functions is obviously 1 (see 3.2.2):

$$\sum_{i=1}^{n_n} N_i = 1. \quad (4.28)$$

Taking the gradient of (4.28) leads to

$$\sum_{i=1}^{n_n} \text{grad} N_i = 0. \quad (4.29)$$

This indicates that the maximal number of linearly independent gradients of the nodal basis functions is $n_n - 1$, i.e. the number of tree edges of the finite element in the graph as shown in Fig. 4.1.

As pointed out in section 3.2.4, since the gradients of the nodal basis functions are in the function space spanned by the edge basis functions (see (3.36)), the following $n_n - 1$ linearly independent relations can be obtained:

$$\text{grad} N_i = \sum_{j=1}^{n_e} g_{ij} \mathbf{N}_j, \quad i = 1, 2, \dots, n_n - 1 \quad (4.30)$$

Chapter 4 The magnetostatic problem

where

$$\sum_{j=1}^{n_e} g_{ij}^2 > 0, \quad i = 1, 2, \dots, n_n - 1. \quad (4.31)$$

Taking the curl of each of the equations in (4.30) results in

$$\sum_{j=1}^{n_e} g_{ij} \text{curl} \mathbf{N}_i = \mathbf{0} \quad i = 1, 2, \dots, n_n - 1. \quad (4.32)$$

which indicates that the number of linear interdependences between the functions $\text{curl} \mathbf{N}_i$ is also $n_n - 1$, i.e. the number of tree edges.

The approximation of the vector potential in (4.23) can be rewritten as

$$\mathbf{A} \approx \mathbf{A}_h = \mathbf{A}_h^{\text{tree}} + \mathbf{A}_h^{\text{cotree}} \quad (4.33)$$

where $\mathbf{A}_h^{\text{tree}}$ and $\mathbf{A}_h^{\text{cotree}}$ are the approximations of the vector potential on the tree edges and on the cotree edges, respectively. Since $\mathbf{A}_h^{\text{tree}}$ is a gradient function, there exists a scalar function u so that

$$\mathbf{A}_h^{\text{tree}} = \text{grad} u. \quad (4.34)$$

The scalar function u can be constructed as $\text{grad} u = \sum_{j=1}^{n_n} u_j \mathbf{N}_j$ by setting the nodal values u_j and the value of the vector potential of the j -th edge as $a_j = u_{j_2} - u_{j_1}$ for all edge $j \in \text{Tree}$ pointing from node j_1 to j_2 .

Setting $\mathbf{A}_h^{\text{tree}}$, i.e. all degrees of freedom corresponding to the tree edges to zero changes the vector potential by a gradient function and does not change $\mathbf{B} = \text{curl} \mathbf{A}$. This also eliminates the linear interdependences between the functions $\text{curl} \mathbf{N}_i$. Meanwhile, the functions $\text{grad} \mathbf{N}_i$ can then no more be expressed as linear combinations of the functions \mathbf{N}_j as pointed out in (4.30), since at least one tree edge is always incident with each node due to the properties of a tree (connects all nodes without a loop).

This means that the vector potential can be approximated as

$$\mathbf{A} \approx \mathbf{A}_h = \mathbf{A}_h^{\text{cotree}}. \quad (4.35)$$

If the co-tree edges are numbered first, then

$$\mathbf{A}_h^{\text{cotree}} = \sum_{j=1}^{n_e - (n_n - 1)} a_j \mathbf{N}_j. \quad (4.36)$$

4.3 Smoothers for magnetostatic problems

The Dirichlet boundary conditions can be taken into account by considering that the a_j values corresponding to the edges on Γ_B are known. If these edges are numbered first:

$$\mathbf{A}_h^{\text{cotree}} = \mathbf{A}_D^{\text{cotree}} + \sum_{j=1}^{n'_{\text{cotree}}} a_j \mathbf{N}_j \quad (4.37)$$

where $\mathbf{A}_D^{\text{cotree}}$ satisfies the Dirichlet boundary conditions and n'_{cotree} is the number of cotree edges outside Γ_B .

The solution of cotree edges can be obtained from solving the following non-singular matrix equation by a direct solver:

$$A_{cc} \mathbf{x}_c = \mathbf{b}_c \quad (4.38)$$

where A_{cc} , \mathbf{x}_c , and \mathbf{b}_c correspond to the cotree edges outside Γ_B .

In this work, on the coarsest level of the MG method, a parallel direct solver (PARDISO) [48] is used for solving the linear system of equations. When the system matrix is singular, the above tree-cotree gauge has to be applied to the PARDISO solver.

4.3 Smoothers for magnetostatic problems

For solving the systems of equations arising from magnetostatic problems, the smoothers applied to the MG method have to be carefully chosen.

In the case of the Φ formulation, the scalar Poisson equation (4.11) is discretized with nodal elements. For the resulting system of equations (4.16), the smoothing iterations are usually carried out by GS type iteration such as the SGS iteration, the smoothing efficiency for the MG method of which can be improved by using the SGSCG iteration [21].

4.3.1 The smoother for the \mathbf{A} formulation

In the case of the \mathbf{A} formulation, the discretized form of the differential equation (4.20) can be described in finite element spaces with $\mathbf{A}_h \in \mathcal{N}_{h,D}$:

$$(\text{curl} \mathbf{A}_h, \nu \text{curl} \boldsymbol{\xi}_h) = f(\boldsymbol{\xi}_h), \quad \forall \boldsymbol{\xi}_h \in \mathcal{N}_{h,D}. \quad (4.39)$$

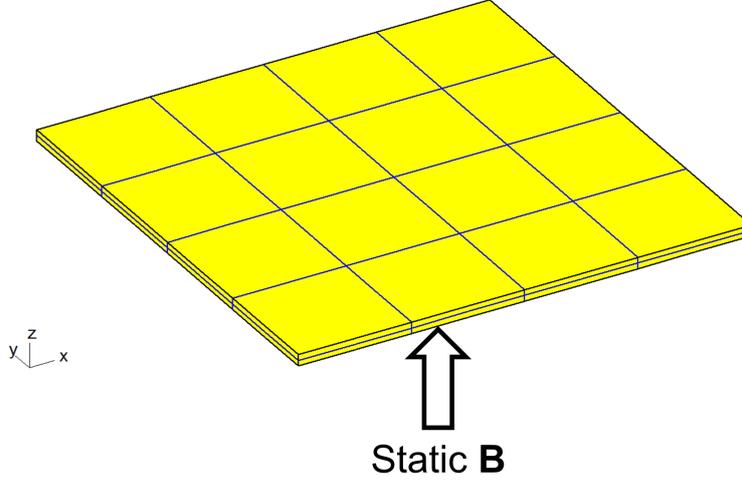


Figure 4.2 Magnetic thin plate exposed to a static magnetic flux density.

According to Theorem 2 (see 3.2.4), \mathbf{A}_h and $\boldsymbol{\xi}_h$ can be decomposed as:

$$\mathbf{A}_h = \mathbf{A}_h^\perp + \mathbf{A}_h^\circ \quad (4.40)$$

$$\boldsymbol{\xi}_h = \boldsymbol{\xi}_h^\perp + \boldsymbol{\xi}_h^\circ \quad (4.41)$$

where $\mathbf{A}_h^\perp, \boldsymbol{\xi}_h^\perp \in \mathcal{N}_{h,D}^\perp$, $\mathbf{A}_h^\circ = \text{grad}\varphi_h \in \mathcal{N}_{h,D}^\circ$ with $\varphi_h \in \mathcal{S}_{h,D}$, and $\boldsymbol{\xi}_h^\circ = \text{grad}\varphi_h \in \mathcal{N}_{h,D}^\circ$ with $\varphi_h \in \mathcal{S}_{h,D}$.

Substituting (4.40) and (4.41) in (4.39) leads to the following equations:

$$(\text{curl}\mathbf{A}_h^\perp, \nu\text{curl}\boldsymbol{\xi}_h^\perp) = f(\boldsymbol{\xi}_h^\perp) \quad (4.42)$$

$$(\text{curl}\mathbf{A}_h^\circ, \nu\text{curl}\boldsymbol{\xi}_h^\circ) = f(\boldsymbol{\xi}_h^\circ) \quad (4.43)$$

where $f(\boldsymbol{\xi}_h) = (\text{curl}\boldsymbol{\xi}_h, \mathbf{T}_0)$ in (4.20). This indicates that the solutions of both $\mathbf{A} \in \mathbf{H}_0^\circ(\text{curl}; \Omega; \Gamma_D)$ and $\mathbf{A} \in \mathbf{H}_0^\perp(\text{curl}; \Omega; \Gamma_D)$ can be solved when an iterative method is applied to (4.39). Therefore, it is not necessary to use special block smoothers but SGS or SGSCG smoothing iterations adapted to the \mathbf{A} formulation for the MG solver are sufficient.

4.3.2 The plane smoother for thin elements

As pointed out in section 3.5.3, the efficiency of pointwise smoothers deteriorates when thin plates of magnetic materials need to be modeled by finite elements.

4.3 Smoothers for magnetostatic problems

In this case, the pointwise smoother has to be replaced by the plane smoother for three-dimensional magnetostatic problems, e.g. the symmetric MSI based on plane smoothing blocks adapted to the Φ and \mathbf{A} formulations, respectively.

The efficiency of the plane smoother will be investigated by the test problem shown in Fig. 4.2. The problem consists of a magnetic thin plate located in air with its geometry described by a macro element which is subdivided into finite elements. The relative permeability of the plate is $\mu_r = 1 \times 10^3$ and the imposed magnetic field in z direction is produced by a current-fed coil around the plate. The maximum aspect ratio δ_{\max} of the finite elements varies from 1 to 1000 by simply varying the thickness of the plate.

The problem can be described by either the Φ formulation or the \mathbf{A} formulation. On the one hand, in case of the Φ formulation, the symmetric MSI based on plane smoothing blocks for nodal elements is used as the smoother. On the other hand, for the \mathbf{A} formulation, the symmetric MSI associated with plane smoothing blocks for edge elements needs to be applied to the MG algorithm. The resulting system of equations is solved by either the MG solver or the MGCG solver with different smoothers, the performance of which will be compared in the following. For the stopping criterion of the iterative solver, $(\|A_h \mathbf{x}_h - \mathbf{b}_h\|)/(\|\mathbf{b}_h\|) \leq 10^{-6}$ is used in this work. The computations are performed based on the platform of the Intel Fortran compiler installed on a computer with Intel core 2Quad Processor 2.66GHz, 8-GByte RAM.

The number of iterations of the MG solver with 6 steps of SGSCG smoothing iterations, the MG solver with two steps of plane smoothing iterations and the MGCG solver with one step of the plane smoothing iteration for the Φ and \mathbf{A} formulations is compared in Table 4.1 and Table 4.2, respectively. It can be seen that the MG solver with the SGSCG smoother has the same or even better efficiency than the MG solver with the plane smoother for δ_{\max} less than 10. However, its number of iterations increases significantly for δ_{\max} larger than 100. In contrast, the MG solver with the plane smoother does not deteriorate noticeably with the increase of δ_{\max} indicating that the plane smoother is able to smooth errors along the anisotropic direction of the grid as discussed in section (3.5.3). From the data of the MGCG solver with the plane smoother, it can be noted that the combination of the MG method with the plane smoother and the CG method gives a satisfactory solution. The solver retains its good efficiency even for large values of the maximum aspect ratio δ_{\max} .

Table 4.1 Number of iterations versus δ_{max} for the problem described by the Φ formulation

Maximum aspect raio $\delta_{max} = 1$	MG with 6 SGSCG	MG with two plane smoothing	MGCG with one plane smoothing
1	5	5	4
10	7	7	5
100	19	6	4
500	40	5	3
1000	89	5	4

Table 4.2 Number of iterations versus δ_{max} for the problem described by the \mathbf{A} formulation

Maximum aspect raio $\delta_{max} = 1$	MG with 6 SGSCG	MG with two plane smoothing	MGCG with one plane smoothing
1	4	5	4
10	5	8	6
100	24	12	8
500	60	13	9
1000	93	13	10

4.3 Smoothers for magnetostatic problems

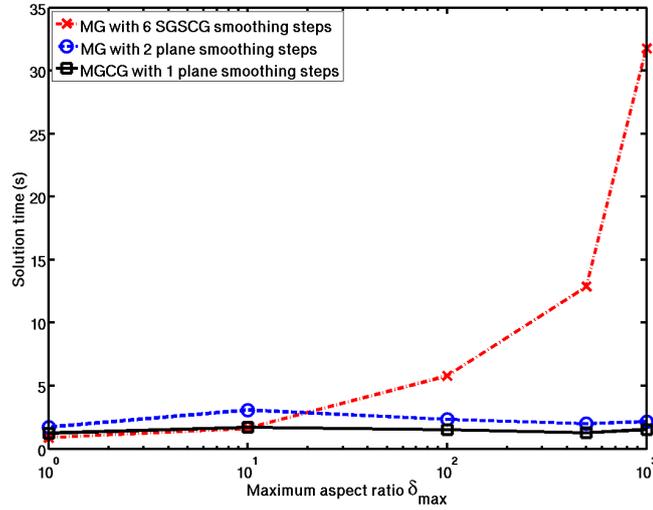


Figure 4.3 Solution times versus the maximum aspect ratio δ_{\max} for the problem described by the Φ formulation

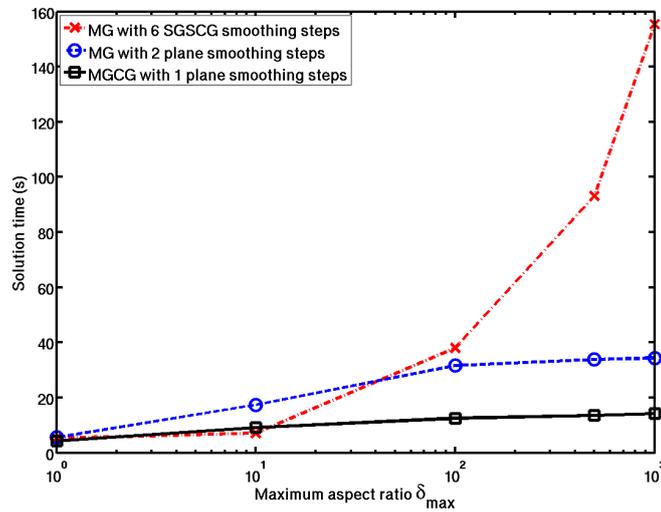


Figure 4.4 Solution time versus the maximum aspect ratio δ_{\max} for the problem described by the \mathbf{A} formulation

The solution time is a more important factor for evaluating the efficiency of the solver than the number of iterations. Fig. 4.3 compares the solution times of all three solvers applied to the Φ formulation. For $\delta = 100$, the solution times of the MG and MGCG solvers with the plane smoother are 40% and 26% of that of the MG solver with the SGSCG smoother, respectively, and for $\delta = 1000$, the solution times of the MG and MGCG solvers with the plane smoother are 6.8% and 4.9% of that of the MG solver with the SGSCG smoother, respectively.

Fig. 4.4 compares solution times of the solvers applied to the \mathbf{A} formulation. Similar results to the case of the Φ formulation are obtained. For example, for thin elements with $\delta = 1000$, solution times of the MG and MGCG solvers with the plane smoother are 22% and 9.1% of that of the MG solver with the SGSCG smoother, respectively.

From the comparison, it can be concluded that two steps of plane smoothing iterations are sufficient for an efficient reduction of errors associated with unknowns of nodal and edge elements and the MG solver with the plane smoother has a stable convergence with the increase of the maximum aspect ratio. In addition, the MGCG solver with the plane smoother results in further improvement of the efficiency.

4.4 Numerical examples

In this section, a practical problem shown in Fig. 4.5 consisting of a magnetic box shield and a coil above the shield is discretized by the finite element method using both the Φ formulation and the \mathbf{A} formulation. The relative permeability of the box is $\mu_r = 1 \times 10^3$. The finite element discretization is shown in Fig. 4.6. As the thickness of the shield plate is 0.01mm and the plate is divided into two layers of finite elements, the maximum aspect ratio δ_{\max} is 2000. Since the mesh is composed of only cuboids and generated orthogonally, δ_{\max} can reach nearly 10000 at the far boundary of the computational domain.

In Table 4.3, the solution data of the MG solver with 6 steps of SGSCG iterations, the MGCG solver with one step of the plane smoothing iteration and the commonly used ICCG solver for solving the system of equations resulting from the Φ formulation are compared. The computations are performed based on the platform of the Intel Fortran compiler installed on a computer with Intel core 2Quad Processor 2.66GHz, 8-GByte RAM. Although the preprocessing time for

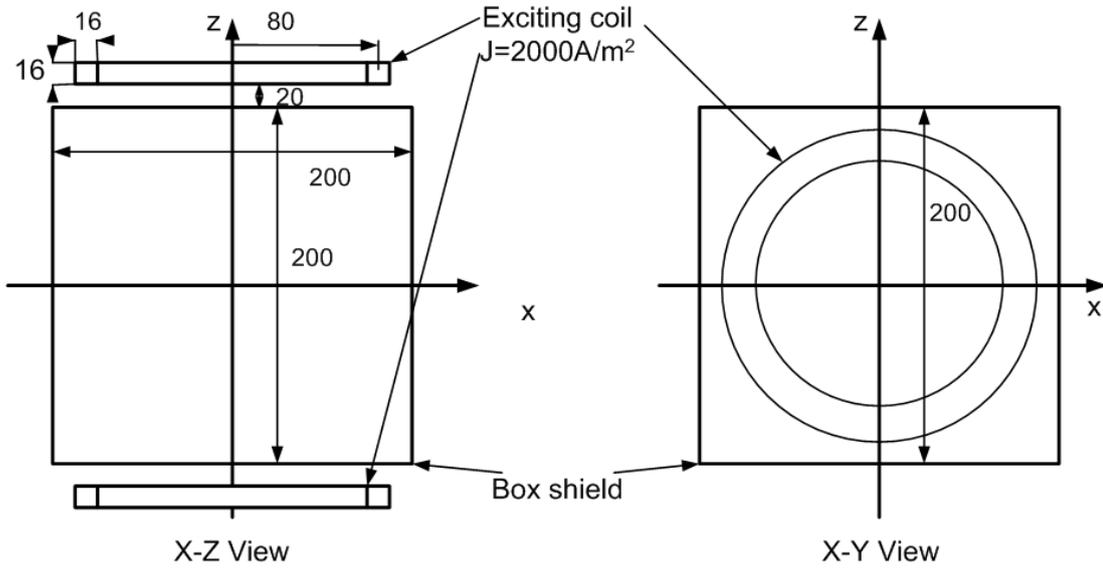


Figure 4.5 The geometry of the box and the coil

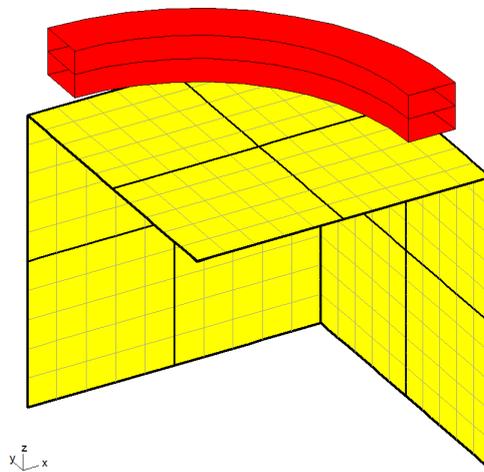


Figure 4.6 The finite element discretization of one eighth of the box and the coil due to the symmetry

Table 4.3 Solution data for the problem described by the Φ formulation

	ICCG	MG with 6 SGSCG	MGCG with one plane smoothing
Number of equations	92568	92568	92568
Number of blocks	0	0	4879
Preprocessing time (s)	1.1	11.6	32.7
Iterations	7894	298	13
Solution time (s)	351.1	374.0	32.4

Table 4.4 Solution data for the problem described by the \mathbf{A} formulation

	ICCG	MG with 6 SGSCG	MGCG with one plane smoothing
Number of equations	254072	254072	254072
Number of blocks	0	0	9086
Preprocessing time (s)	4.3	27.8	60.6
Iterations	12422	566	15
Solution time (s)	895.1	2167.6	68.4

the plane smoother is higher than that of the SGSCG smoother since the inverse of the submatrices has to be calculated, the solution time has been reduced by more than 90% using the MGCG with the plane smoother. Compared to the ICCG solver, the MG solver with the SGSCG smoother loses its advantage but the MGCG solver with the plane smoother achieves approximate 90.8% of decrease of the solution time, which means that the MG method with the plane smoother behaves as a better preconditioner than the incomplete Cholesky factorization in this case.

Similar results as listed in Table 4.4 have been obtained for the problem described by the \mathbf{A} formulation. The number of iterations, the preprocessing and solution times of the ICCG solver, the MG solver with 6 steps of SGSCG iterations and the MGCG solver with one step of the plane smoothing iteration have

4.4 Numerical examples

been compared. It can be seen that the solution time has been reduced by 92.4% and 95% compared to the ICCG solver and the MG method with the SGSCG smoother by the MGCG solver with the plane smoother, respectively.

The analyses above indicate that the MGCG solver with the plane smoother acts as an efficient solver for magnetostatic problems discretized by either nodal elements or edge elements. The next chapter will present that it can be also applied to finite element formulations for eddy current problems.

The time-harmonic eddy current problem

An eddy current problem arises if a time varying magnetic field is excited within a conductor. As shown in Fig. 5.1, an eddy current problem involves two regions: a conducting region Ω_c constituted by eddy current carrying conductors and a nonconducting region Ω_n including nonconductors and coils with given current densities. In Ω_c , the current density distribution is unknown, an eddy current field is present here.

5.1 Differential equations and boundary conditions

The time variation of fields is assumed to be sinusoidal with an angular frequency of ω and the displacement current density to be negligible, hence, the time-harmonic Maxwell equations for both regions are the following:

$$\text{curl}\mathbf{H} = \mathbf{J}_0 \quad \text{in } \Omega_n, \quad (5.1)$$

$$\text{div}\mathbf{B} = 0 \quad \text{in } \Omega_n, \quad (5.2)$$

$$\text{curl}\mathbf{H} = \mathbf{J} \quad \text{in } \Omega_c, \quad (5.3)$$

$$\text{curl}\mathbf{E} = -j\omega\mathbf{B} \quad \text{in } \Omega_c, \quad (5.4)$$

$$\text{div}\mathbf{J} = 0 \quad \text{in } \Omega_c, \quad (5.5)$$

$$\text{div}\mathbf{B} = 0 \quad \text{in } \Omega_c. \quad (5.6)$$

where \mathbf{E} is the electric field intensity.

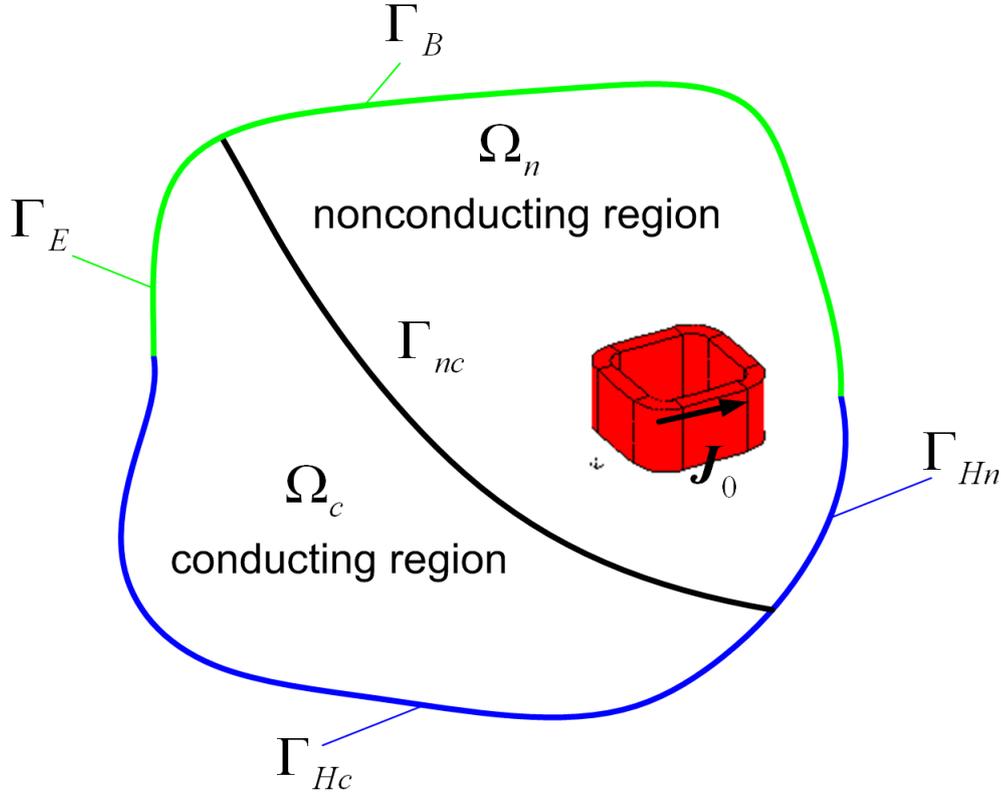


Figure 5.1 The domain of the eddy current problems.

The material equations are

$$\mathbf{B} = \mu \mathbf{H}, \quad \mathbf{H} = \nu \mathbf{B} \quad \text{in } \Omega_c \text{ and } \Omega_n, \quad (5.7)$$

$$\mathbf{J} = \sigma \mathbf{E}, \quad \mathbf{E} = \rho \mathbf{J} \quad \text{in } \Omega_c \quad (5.8)$$

where σ is the conductivity and ρ the resistivity.

The boundary conditions assumed on the boundary of Ω_c are

$$\mathbf{H} \times \mathbf{n} = \mathbf{T}_0 \times \mathbf{n} \quad \text{on } \Gamma_{Hc}, \quad (5.9)$$

$$\mathbf{E} \times \mathbf{n} = \mathbf{0} \quad \text{on } \Gamma_E \quad (5.10)$$

where \mathbf{T}_0 is defined by (4.4).

The boundary conditions assumed on the boundary of Γ_{Hn} and Γ_B are the same as the boundary conditions in (4.6)-(4.9) assumed on the boundary of Γ_H and Γ_B in magnetostatic problems, respectively.

5.1 Differential equations and boundary conditions

The continuity conditions on the interface Γ_{nc} between the conducting and nonconducting regions are

$$\mathbf{H} \times \mathbf{n} \text{ and } \mathbf{B} \cdot \mathbf{n} \text{ are continuous on } \Gamma_{nc}. \quad (5.11)$$

In the nonconducting region Ω_n , the potential formulations introduced in sections 4.1.1 and 4.1.2 can be used. For the conducting region Ω_c , two following formulations: the \mathbf{A}, V formulation and the \mathbf{T}, Φ formulation are presented [26].

5.1.1 The \mathbf{A}, V formulation

The magnetic vector potential \mathbf{A} and the electric scalar potential V are defined as

$$\mathbf{B} = \text{curl} \mathbf{A}, \quad (5.12)$$

$$\mathbf{E} = -j\omega \mathbf{A} - j\omega \text{grad} V. \quad (5.13)$$

Faraday's law (5.4) is exactly satisfied and using the first relationship in (5.7) and the second in (5.8), Ampere's law (5.3) results in the differential equation

$$\text{curl}(\nu \text{curl} \mathbf{A}) + j\omega \sigma \mathbf{A} + j\omega \sigma \text{grad} V = \mathbf{0} \quad \text{in } \Omega_c, \quad (5.14)$$

and the divergence free property of the current density (5.5) leads to

$$-\text{div}(j\omega \sigma \mathbf{A} + j\omega \sigma \text{grad} V) = \mathbf{0} \quad \text{in } \Omega_c. \quad (5.15)$$

Using the boundary conditions (5.10) and (5.9), the Dirichlet and Neumann boundary conditions for \mathbf{A} are

$$\mathbf{A} \times \mathbf{n} = \mathbf{0} \quad \text{on } \Gamma_E, \quad (5.16)$$

$$\nu \text{curl} \mathbf{A} \times \mathbf{n} = \mathbf{T}_0 \times \mathbf{n} \quad \text{on } \Gamma_{Hc}, \quad (5.17)$$

and for V they are

$$V = V_0 = \text{constant} \quad \text{on } \Gamma_E, \quad (5.18)$$

$$\mathbf{n} \cdot (-j\omega \sigma \mathbf{A} - j\omega \sigma \text{grad} V) = 0 \quad \text{on } \Gamma_{Hc}. \quad (5.19)$$

The vector potential $\mathbf{A} \in \mathbf{H}_0(\text{curl}; \Omega; \Gamma_E)$ is approximated by edge basis functions $\mathbf{N}_j \in \mathcal{N}_0(\mathcal{T}_h; \Gamma_E)$ (see 3.2.3) as

$$\mathbf{A} \approx \mathbf{A}_h = \mathbf{A}_D + \sum_{j=1}^{n_e - n_{e_d}} a_j \mathbf{N}_j, \quad \mathbf{A}_D = \sum_{j=1}^{n_{e_d}} a_j \mathbf{N}_j \quad (5.20)$$

Chapter 5 The time-harmonic eddy current problem

where a_j are the line integrals of \mathbf{A} along the edges, \mathbf{A}_D represents the magnetic vector potential of edges on the Dirichlet boundary, n_e is the total number of edges, and n_{e_d} is the number of edges on Γ_E .

The electric scalar potential $V \in H_0^1(\Omega; \Gamma_E)$ is approximated by nodal basis functions $N_j \in \mathcal{S}_0(\mathcal{T}_h; \Gamma_E)$ (see 3.2.2) as

$$V \approx V_h = V_D + \sum_{j=1}^{n_n - n_{n_d}} V_j N_j, \quad V_D = \sum_{j=1}^{n_{n_d}} V_j N_j \quad (5.21)$$

where V_j are the nodal values of V_h , n_n is the total number of nodes, V_D represents the electric scalar potential of nodes on the Dirichlet boundary, and n_{n_d} is the number of nodes on Γ_E .

Setting (5.20) and (5.21) into (5.14) and (5.15) as well as into the Neumann boundary conditions in (5.17) and (5.19) and using the basis functions \mathbf{N}_i and N_i as weighting functions, the following Galerkin equations are obtained:

$$\begin{aligned} & \int_{\Omega_c} \text{curl} \mathbf{N}_i \cdot \nu \text{curl} \mathbf{A}_h d\Omega + \int_{\Omega_c} j\omega\sigma \mathbf{N}_i \cdot \mathbf{A}_h d\Omega \\ & + \int_{\Omega_c} j\omega\sigma \mathbf{N}_i \cdot \text{grad} V_h d\Omega = \int_{\Omega_c} \text{curl} \mathbf{N}_i \cdot \mathbf{T}_0 d\Omega, \quad i = 1, 2, \dots, n_e - n_{e_d}, \end{aligned} \quad (5.22)$$

$$\int_{\Omega_c} j\omega\sigma \text{grad} N_i \cdot \mathbf{A}_h d\Omega + \int_{\Omega_c} j\omega\sigma \text{grad} N_i \cdot \text{grad} V_h d\Omega = \mathbf{0}, \quad i = 1, 2, \dots, n_n - n_{n_d} \quad (5.23)$$

where $\text{curl} \mathbf{T}_0 = \mathbf{J}_0$ is assumed in the nonconducting region.

These equations (5.22) and (5.23) can be written in the matrix form

$$\begin{bmatrix} K & C \\ C^T & B \end{bmatrix} \begin{Bmatrix} \mathbf{a} \\ \mathbf{v} \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \end{Bmatrix} \quad (5.24)$$

where the entries of the matrices K , C and B are

$$k_{ij} = \int_{\Omega_c} \text{curl} \mathbf{N}_i \cdot \nu \text{curl} \mathbf{N}_j + \int_{\Omega_c} j\omega\sigma \mathbf{N}_i \cdot \mathbf{N}_j d\Omega, \quad (5.25)$$

$$c_{ij} = \int_{\Omega_c} j\omega\sigma \mathbf{N}_i \cdot \text{grad} N_j d\Omega, \quad (5.26)$$

$$b_{ij} = \int_{\Omega_c} j\omega\sigma \text{grad} N_i \cdot \text{grad} N_j d\Omega \quad (5.27)$$

5.1 Differential equations and boundary conditions

and the entries of \mathbf{f} are

$$f_{1i} = \int_{\Omega_c} \text{curl} \mathbf{N}_i \cdot \mathbf{T}_0 d\Omega, \quad (5.28)$$

$$f_{2i} = 0. \quad (5.29)$$

Since the linear combinations of the equations in (5.22) with the coefficients g_{ij} defined in (4.30) result in equations in (5.23), the system matrix in (5.24) is singular. This means that if the matrix equation is solved by a direct solver, a tree in the graph of the finite element mesh has to be selected and the values of the unknowns on the tree edges must be set to zero (see section 4.2).

5.1.2 The \mathbf{T}, Φ formulation

The field quantities are derived from the current vector potential \mathbf{T} and the magnetic scalar potential Φ as

$$\mathbf{H} = \mathbf{T}_0 + \mathbf{T} - \text{grad}\Phi, \quad (5.30)$$

$$\mathbf{J} = \text{curl}\mathbf{T}. \quad (5.31)$$

Ampere's law (5.3) follows from (5.30) and (5.31), and using the first relationship in (5.7) and the second in (5.8), Faraday's law (5.4) results in the differential equation

$$\text{curl}(\rho \text{curl}\mathbf{T}) + j\omega\mu\mathbf{T} - j\omega\mu\text{grad}\Phi = -j\omega\mu\mathbf{T}_0 \quad \text{in } \Omega_c, \quad (5.32)$$

and the divergence free property of the magnetic flux density (5.6) leads to

$$j\omega\text{div}(\mu\mathbf{T} - \mu\text{grad}\Phi) = -j\omega\text{div}(\mu\mathbf{T}_0) \quad \text{in } \Omega_c. \quad (5.33)$$

Using the boundary conditions (5.10) and (5.9), the following Dirichlet and Neumann boundary conditions on the vector potential are:

$$\mathbf{T} \times \mathbf{n} = \mathbf{0} \quad \text{on } \Gamma_{Hc} \quad (5.34)$$

$$\nu \text{curl}\mathbf{T} \times \mathbf{n} = -\rho\mathbf{T}_0 \times \mathbf{n} \quad \text{on } \Gamma_E, \quad (5.35)$$

and for V they are

$$V = V_0 = \text{constant} \quad \text{on } \Gamma_{Hc} \quad (5.36)$$

$$\mathbf{n} \cdot (-j\omega\sigma\mathbf{A} - j\omega\sigma\text{grad}V) = 0 \quad \text{on } \Gamma_E. \quad (5.37)$$

The current vector potential $\mathbf{T} \in \mathbf{H}_0(\text{curl}; \Omega; \Gamma_{Hc})$ is approximated by edge basis functions $\mathbf{N}_j \in \mathcal{N}_0(\mathcal{T}_h; \Gamma_{Hc})$ as (see 3.2.3)

$$\mathbf{T} \approx \mathbf{T}_h = \mathbf{T}_D + \sum_{j=1}^{n_e - n_{e_d}} t_j \mathbf{N}_j, \quad \mathbf{T}_D = \sum_{j=1}^{n_{e_d}} t_j \mathbf{N}_j \quad (5.38)$$

where t_j are the line integrals of \mathbf{T}_h along the edges, \mathbf{T}_D represents the current vector potential of edges on the Dirichlet boundary, n_e is the total number of edges, n_{e_d} is the number of edges on Γ_{Hc} .

The magnetic scalar potential $\Phi \in H_0^1(\Omega; \Gamma_{Hc})$ is approximated by nodal basis functions $N_j \in \mathcal{S}_0(\mathcal{T}_h; \Gamma_{Hc})$ as (see 3.2.2)

$$\Phi \approx \Phi_h = \Phi_D + \sum_{j=1}^{n_n - n_{n_d}} \Phi_j N_j, \quad \Phi_D = \sum_{j=1}^{n_{n_d}} \Phi_j N_j \quad (5.39)$$

where Φ_j are the nodal values of Φ , Φ_D represents the magnetic scalar potential of nodes on the Dirichlet boundary, n_n is the total number of nodes, and n_{n_d} is the number of nodes on Γ_{Hc} .

Setting (5.38) and (5.39) into (5.32) and (5.33) as well as into the Neumann boundary conditions in (5.35) and (5.37) and using the basis functions \mathbf{N}_i and N_i as weighting functions, the following Galerkin equations are obtained:

$$\begin{aligned} & \int_{\Omega_c} \text{curl} \mathbf{N}_i \cdot \rho \text{curl} \mathbf{T}_h d\Omega + \int_{\Omega_c} j\omega\mu \mathbf{N}_i \cdot \mathbf{T}_h d\Omega \\ & - \int_{\Omega_c} j\omega\mu \mathbf{N}_i \cdot \text{grad} \Phi_h d\Omega = - \int_{\Omega_c} j\omega\mu \mathbf{N}_i \cdot \mathbf{T}_0 d\Omega \quad i = 1, 2, \dots, n_e - n_{e_d}, \end{aligned} \quad (5.40)$$

$$\begin{aligned} & - \int_{\Omega_c} j\omega\mu \text{grad} N_i \cdot \mathbf{T}_h d\Omega + \int_{\Omega_c} j\omega\mu \text{grad} N_i \cdot \text{grad} \Phi_h d\Omega \\ & = - \int_{\Omega_c} j\omega\mu \text{grad} N_i \cdot \mathbf{T}_0 d\Omega, \quad i = 1, 2, \dots, n_n - n_{n_d}. \end{aligned} \quad (5.41)$$

The matrix form of equations (5.40) and (5.41) are

$$\begin{bmatrix} K & C \\ C^T & B \end{bmatrix} \begin{Bmatrix} \mathbf{t} \\ \boldsymbol{\varphi} \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \end{Bmatrix} \quad (5.42)$$

5.1 Differential equations and boundary conditions

where the entries of the matrices K , C and B are

$$k_{ij} = \int_{\Omega_c} \text{curl} \mathbf{N}_i \cdot \rho \text{curl} \mathbf{N}_j + \int_{\Omega_c} j\omega\mu \mathbf{N}_i \cdot \mathbf{N}_j d\Omega, \quad (5.43)$$

$$c_{ij} = \int_{\Omega_c} -j\omega\mu \mathbf{N}_i \cdot \text{grad} N_j d\Omega \quad (5.44)$$

$$b_{ij} = \int_{\Omega_c} j\omega\mu \text{grad} N_i \cdot \text{grad} N_j d\Omega \quad (5.45)$$

and the entries of \mathbf{f}_1 and \mathbf{f}_2 are

$$f_{1i} = - \int_{\Omega_c} j\omega\mu \mathbf{N}_i \cdot \mathbf{T}_0 d\Omega \quad (5.46)$$

$$f_{2i} = \int_{\Omega_c} j\omega \text{grad} N_i \cdot \mathbf{T}_0 d\Omega. \quad (5.47)$$

Similarly to (5.24), the system matrix of equations (5.42) is also singular since a linear combination of (5.40) is (5.41).

5.1.3 Coupling of the potential formulations

The potential formulations in Ω_n and Ω_c presented in sections 4.1 and 5.1 can be coupled by satisfying the interface conditions (5.11). The most commonly used coupled formulations are the \mathbf{A} , V - \mathbf{A} formulation and the \mathbf{T} , Φ - Φ formulation [26].

In the case of the \mathbf{A} , V - \mathbf{A} formulation, the continuity of the tangential component of the vector potential \mathbf{A} on Γ_{nc} is explicitly satisfied by the use of the edge basis function. This enforces the continuity of normal component of the magnetic flux density \mathbf{B} . The continuity of $\mathbf{H} \times \mathbf{n}$ is considered by the term

$$\int_{\Gamma_{nc}} \mathbf{N}_i \cdot (\mathbf{H} \times \mathbf{n}) d\Gamma. \quad (5.48)$$

Equation (5.48) arises in the Galerkin equations in both regions and is cancelled. Therefore, the equations of the \mathbf{A} , V - \mathbf{A} formulation are (5.22), (5.23) and (4.24).

For the \mathbf{T} , Φ - Φ formulation, the continuity of $\mathbf{H} \times \mathbf{n}$ is enforced by the continuity of the magnetic scalar potential and by setting the tangential component of the current vector potential to zero on the interface:

$$\mathbf{T} \times \mathbf{n} = \mathbf{0} \quad \text{on } \Gamma_{nc}. \quad (5.49)$$

The normal component of the flux density is taken into account by

$$\int_{\Gamma_{nc}} N_i \mathbf{B} \cdot \mathbf{n} d\Gamma \quad (5.50)$$

arising in Galerkin equations in both regions and cancelling. So the Galerkin equations of the \mathbf{T} , Φ - Φ formulation are (5.40), (5.41) and (4.15).

5.2 Multigrid for nonlinear problems

In ferromagnetic materials the relationship between the magnetic flux density \mathbf{B} and the magnetic field intensity \mathbf{H} is nonlinear. In general cases the function $\mathbf{B} = f(\mathbf{H})$ or $\mathbf{H} = f(\mathbf{B})$ is described by hysteresis curves shown in Fig. 5.2. In this work, it is assumed that the material is isotropic and that no hysteresis effects arise. Hence, the nonlinearity can be described by the reluctivity ν or the permeability μ :

$$\mathbf{H} = \nu(|\mathbf{B}|)\mathbf{B} \quad \text{or} \quad \mathbf{B} = \mu(|\mathbf{H}|)\mathbf{H} \quad (5.51)$$

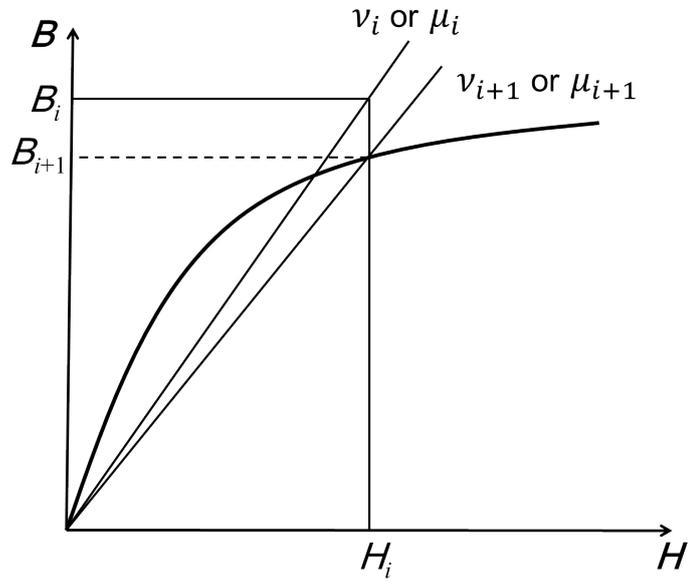
In the case of nonlinear problems with a time-harmonic excitation, only the fundamental harmonic is taken into account in the complex formulations in this work. To solve the nonlinear equations, a modified Picard-Banach method has been used to guarantee the stable choice of the new reluctivity ν_{i+1} or the new permeability μ_{i+1} [49]. This means that a series of linear system of equations

$$A(\nu_i)\mathbf{x}_i = \mathbf{b}_i \quad \text{or} \quad A(\mu_i)\mathbf{x}_i = \mathbf{b}_i \quad i = 1, 2, \dots \quad (5.52)$$

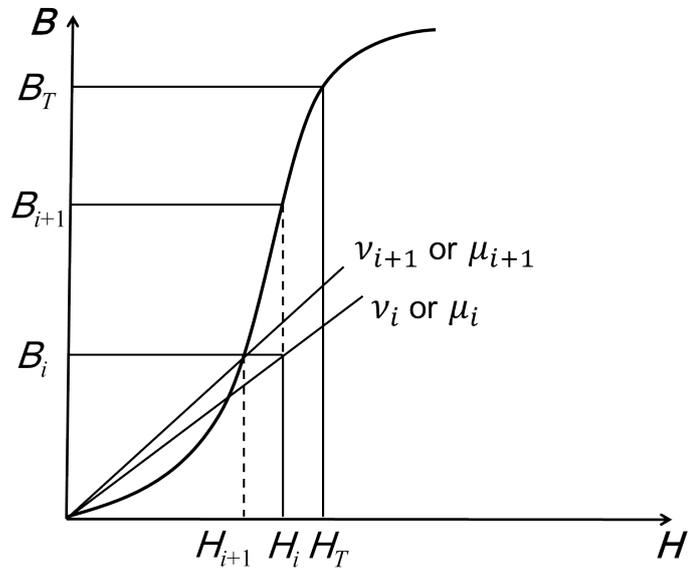
needs to be solved.

The new value of ν_{i+1} or μ_{i+1} can be calculated as follows:

In the case of the \mathbf{A} , V - \mathbf{A} formulation, if the magnetizing curve is monotonous and convex (i.e. $dB/dH > 0$ and $d^2B/dH^2 < 0$) as shown in Fig. 5.2(a), from the solution of \mathbf{x}_i (5.52) one can obtain the value of B_i . The magnetic field intensity H_i is calculated by multiplying B_i with ν_i obtained from the previous step. B_{i+1} is obtained from the B - H curve and the new reluctivity ν_{i+1} is derived as the secant H_i/B_{i+1} . But if the magnetizing curve is concave ($d^2B/dH^2 > 0$) for low field values as in the domain $B < B_T$, $H < H_T$ in the curve shown in Fig. 5.2(b), the method is changed as follows: the value of B_i is obtained from the solution \mathbf{x}_i and H_{i+1} from the B - H curve to yield the new reluctivity ν_{i+1} as H_{i+1}/B_i . By



(a) The nonlinear iteration of the modified Picard-Banach method for convex magnetizing curves



(b) The nonlinear iteration of the modified Picard-Banach method for concave magnetizing curves

Figure 5.2 Modified Picard-Banach method for nonlinear iterations.

the above procedure the new value ν_{i+1} can be calculated in each Gaussian point of the finite elements.

In the case of the \mathbf{T} , Φ - Φ formulation, for the curve shown in Fig. 5.2(a), the value of H_i can be obtained from the solution. Afterwards, B_i is calculated by $B_i = \mu_i H_i$. Finally, from the B - H curve, the new H_{i+1} and the new permeability μ_{i+1} are obtained. A similar process is carried out in the case of the concave curve as presented for the \mathbf{A} , V - \mathbf{A} formulation. The value of H_i is obtained from the solution \mathbf{x}_i , and then B_i is calculated with the aid of the old value μ_i and H_{i+1} is afterwards obtained from the B - H curve. Thereupon, the value μ_{i+1} is derived as $B_i = \mu_i H_{i+1}$.

Since the system matrices $A_l(\nu)$ or $A_l(\mu)$ depend on ν or μ , for different MG levels $l = 0, 1, \dots, l_{max}$ they have to be assembled in every nonlinear iteration step. For the MG method, the solution \mathbf{x}_l for calculating ν_{i+1} or μ_{i+1} is the solution on the finest grid, which is also used to compute their values on the coarser grids. This results in a proper hierarchy of systems of equations which can be solved by the following algorithms:

Algorithm 5. Nonlinear iteration for the \mathbf{A} , V - \mathbf{A} formulation

1. $i = 0$
2. ν_0 and the B - H curve $\mu_{BH}(\mathbf{H})$ are given
3. Assemble the matrices $A_l(\nu_i)$, $l = 0, 1, 2, \dots, l_{max}$
4. Solve $A_{l_{max}}(\nu_i)\mathbf{x}_i = \mathbf{b}_{l_{max},i}$
5. Calculate \mathbf{B}_i from \mathbf{x}_i
6. $\mathbf{H}_i = \nu_i \mathbf{B}_i$
7. $\nu_{i+1} = 1/\mu_{BH}(\mathbf{H}_i)$
8. If $\left\| \frac{\nu_{i+1} - \nu_i}{\nu_{i+1}} \right\|_{\max} < \varepsilon_1$ or $\left\| \frac{\nu_{i+1} - \nu_i}{\nu_{i+1}} \right\|_{\text{mean}} < \varepsilon_2$ then $A_l(\nu)\mathbf{x}_l = \mathbf{b}_l$ is solved else
9. $i = i + 1$
10. back to step 3

Algorithm 6. Nonlinear iteration for the T , Φ - Φ formulation

1. $i = 0$
2. μ_0 and the B - H curve $\mu_{BH}(\mathbf{H})$ are given
3. Assemble the matrices $A_l(\mu_i)$, $l = 0, 1, 2, \dots, l_{max}$
4. Solve $A_{l_{max}}(\mu_i)\mathbf{x}_i = \mathbf{b}_{l_{max},i}$
5. Calculate \mathbf{H}_i from \mathbf{x}_i
6. $\mathbf{B}_i = \mu_i\mathbf{H}_i$
7. $\mu_{i+1} = \mu_{BH}(\mathbf{B}_i)$
8. If $\left\| \frac{\mu_{i+1} - \mu_i}{\mu_{i+1}} \right\|_{\max} < \varepsilon_1$ or $\left\| \frac{\mu_{i+1} - \mu_i}{\mu_{i+1}} \right\|_{\text{mean}} < \varepsilon_2$ then $A_l(\nu)\mathbf{x}_l = \mathbf{b}_l$ is solved else
9. $i = i + 1$
10. back to step 3

5.3 Smoothers for eddy current problems

5.3.1 Gauss-Seidel smoother for the A, V formulation

In the \mathbf{A}, V formulation, the magnetic vector potential \mathbf{A} and the electric scalar potential V can be approximated by $\mathbf{A}_h \in \mathcal{N}_{h,D}$ and $V_h \in \mathcal{S}_{h,D}$, respectively, hence, with these approximations, (5.14) and (5.15) can be described in finite element spaces as follows:

$$(\nu \text{curl} \mathbf{A}_h, \text{curl} \boldsymbol{\xi}_h) + j\omega(\sigma \mathbf{A}_h, \boldsymbol{\xi}_h) + j\omega(\sigma \text{grad} V_h, \boldsymbol{\xi}_h) = f_h(\boldsymbol{\xi}_h) \quad (5.53)$$

$$j\omega(\sigma \mathbf{A}_h, \text{grad} \eta_h) + j\omega(\sigma \text{grad} V_h, \text{grad} \eta_h) = 0 \quad (5.54)$$

for all $\boldsymbol{\xi}_h \in \mathcal{N}_{h,D}$ and $\eta_h \in \mathcal{S}_{h,D}$.

Equation (5.54) can be rewritten as

$$j\omega(\sigma \mathbf{A}_h + \sigma \text{grad} V_h, \text{grad} \eta_h) = 0, \quad \forall \eta_h \in \mathcal{S}_{h,D}. \quad (5.55)$$

This weak form is equivalent to the Laplace equation $\operatorname{div}(\sigma \operatorname{grad} \hat{V}) = 0$ with

$$\operatorname{grad} \hat{V} = j\omega(\mathbf{A}_h + \operatorname{grad} V_h). \quad (5.56)$$

From the above equation, one can obtain

$$(\operatorname{grad} \hat{V}_h - j\omega \operatorname{grad} V_h) = j\omega \mathbf{A}_h \in \mathcal{N}_{h,D}^o. \quad (5.57)$$

This means that in the scalar products $j\omega(\sigma \operatorname{grad} V_h, \boldsymbol{\xi}_h)$ and $j\omega(\sigma \mathbf{A}_h, \operatorname{grad} \eta_h)$ in (5.53) and (5.54) which describe the coupling of \mathbf{A}_h and V_h , only $\mathbf{A}_h \in \mathcal{N}_{h,D}^o$ is taken into account, whereas for vector functions in $\mathbf{A}_h \in \mathcal{N}_{h,D}^\perp$, these scalar products are zero due to the definition in (3.37).

The vector potential $\mathbf{A}_h \in \mathcal{N}_{h,D}$ can be decomposed as

$$\mathbf{A}_h = \mathbf{A}_h^\perp + \mathbf{A}_h^o \quad (5.58)$$

with $\mathbf{A}_h^o \in \mathcal{N}_{h,D}^o$ and $\mathbf{A}_h^\perp \in \mathcal{N}_{h,D}^\perp$.

The test function $\boldsymbol{\xi}_h$ can be also decomposed as

$$\boldsymbol{\xi}_h = \boldsymbol{\xi}_h^\perp + \boldsymbol{\xi}_h^o \quad (5.59)$$

with $\boldsymbol{\xi}_h^o \in \mathcal{N}_{h,D}^o$ and $\boldsymbol{\xi}_h^\perp \in \mathcal{N}_{h,D}^\perp$.

Substituting $\boldsymbol{\xi}_h$ in (5.53) by $\boldsymbol{\xi}_h^o = \operatorname{grad} \phi_h$ leads to the equation

$$j\omega(\sigma(\mathbf{A}_h + \operatorname{grad} V_h), \operatorname{grad} \phi_h) = f_h(\operatorname{grad} \phi_h) \quad \forall \phi_h \in \mathcal{S}_h, D. \quad (5.60)$$

which is already solved by (5.54) for $\mathbf{A}_h^o \in \mathcal{N}_{h,D}^o$. It can be seen that the consistency of the equation is ensured with $f_h(\operatorname{grad} \phi_h) = 0$.

Inserting $\boldsymbol{\xi}_h^\perp$ into (5.54), one can obtain:

$$(\sigma \operatorname{curl} \mathbf{A}_h^\perp, \operatorname{curl} \boldsymbol{\xi}_h^\perp) + j\omega(\sigma \mathbf{A}_h^\perp, \boldsymbol{\xi}_h^\perp) = f_h(\boldsymbol{\xi}_h^\perp) \quad \forall \boldsymbol{\xi}_h^\perp \in \mathcal{N}_{h,D}^\perp \quad (5.61)$$

where only the function in $\mathcal{N}_{h,D}^\perp$ is solved, which indicates that no error in the kernel of the curl operator has to be taken care of by the smoothing iteration so that the MG method with the GS type smoother can be used as the solver.

It has been shown that the equations (5.54) are equivalent to Laplacian equations which can be also solved by the GS smoother. By solving this equation, not only the error of the scalar potential V_h is smoothed, but also the error of the magnetic vector potential in the kernel of the curl operator. Therefore, the

pointwise SGS or SGSCG iteration can be a proper smoother of the MG method applied to (5.24).

Since the system matrix of (5.42) resulting from the \mathbf{T}, Φ formulation is similar to that of (5.24) resulting from the \mathbf{A}, V formulation, the GS iteration can also be the smoother of the MG method applied to the matrix equation resulting from \mathbf{T}, Φ formulation.

5.3.2 Block smoothers for eddy current problems

The modified AFW smoother for the \mathbf{A}, V - \mathbf{A} formulation

In addition to the GS iteration adapted to the \mathbf{A}, V formulation, another choice is to use a block smoother based on the domain decomposition method. As introduced in section 3.5.2, the property of the AFW blocks is that they consist of at least one function of the kernel functions of the curl operator. Therefore, in the case of the \mathbf{A}, V formulation, by applying the symmetric MSI based on the AFW block to (5.14), the error of the magnetic vector potential \mathbf{A}_h in the kernel space of the curl operator can be eliminated. Adding one step of the SGS or SGSCG iteration to the smoother applied to (5.15), the error of the scalar potential V_h can be also smoothed. The iteration which combines the AFW block iteration with the SGS or SGSCG iteration used as the smoother in the MG algorithm is called the modified AFW (MAFW) smoother for the \mathbf{A}, V formulation [31]. In addition, the MG method with the MAFW smoother can be also used as the preconditioner of the CG solver for solving eddy current problems [31, 50, 51].

The performance of the MG solver with the SGSCG smoother, as well as the MG and MGCG solvers with the MAFW smoother for the \mathbf{A}, V - \mathbf{A} formulation will be investigated with the aid of the following test problem. It is worth mentioning that when the AFW smoother is applied to the \mathbf{A} formulation used in the nonconducting region, the resulting submatrices associated with the AFW blocks are singular. Hence the corresponding unknowns cannot be directly solved with the aid of the symmetric MSI. As shown in section 4.2, eliminating tree edges in the finite element grid can result in nonsingular matrices. For AFW blocks based on second order hexahedral elements shown in Fig. 3.8, one emanating edge can be eliminated [18] for every interior node, i.e., a tree can be generated by all edges emanating from the center node plus one arbitrary edge as illustrated in Fig. 5.3. As a result, the nonsingular submatrix corresponding to unknowns of

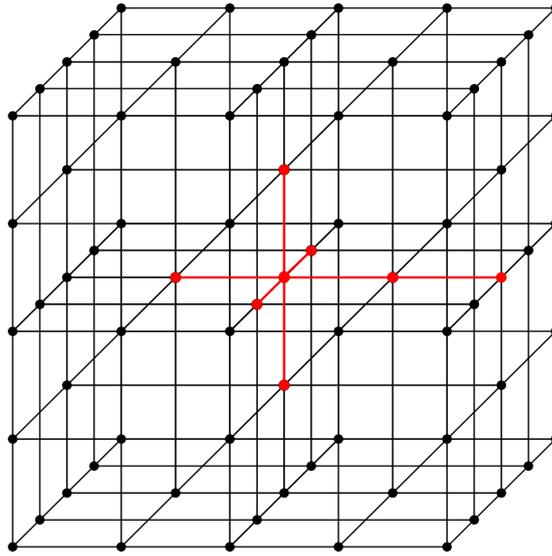


Figure 5.3 Tree edges of a AFW block for second order hexahedral finite elements.

cotree edges can be applied to the symmetric MSI.

The model of the test problem in Fig. 5.4 consists of a cubic core surrounded by a cylindrical coil fed with a current. The relative permeability and conductivity of the core are $\mu_r = 2.3 \times 10^3$ and $\sigma = 4 \times 10^6 \text{S/m}$, respectively, and the frequency of the current excitation is 50Hz. To accurately simulate the eddy currents in the conductor, a key factor to be taken into account is the penetration depth defined as $\delta_p = \sqrt{2/\sigma\mu\omega}$. Since for most cases ferromagnetic materials have high values of conductivity and relative permeability, the penetration depth is much smaller than the scale of the object. Therefore, for this problem, since δ_p and the length of the longest edge belonging to the elements in the cube are $7.4 \times 10^{-4} \text{m}$ and $5 \times 10^{-2} \text{m}$, respectively, finite elements with $\delta_{max} = 50$ have to be used in the mesh. The computations are performed based on the platform of the Intel Fortran compiler installed on a computer with Intel core 2Quad Processor 2.66GHz, 8-GByte RAM.

The convergence of the MG solver with the SGSCG smoother, the MG solver and the MGCG solver with the MAFW smoother is compared by the residual plot shown in Fig. 5.5. As suggested in [28], 6 SGSCG smoothing steps result in an optimal solution time for the MG method. Since the MAFW iteration is about three times as expensive as the SGSCG iteration, two MAFW smoothing

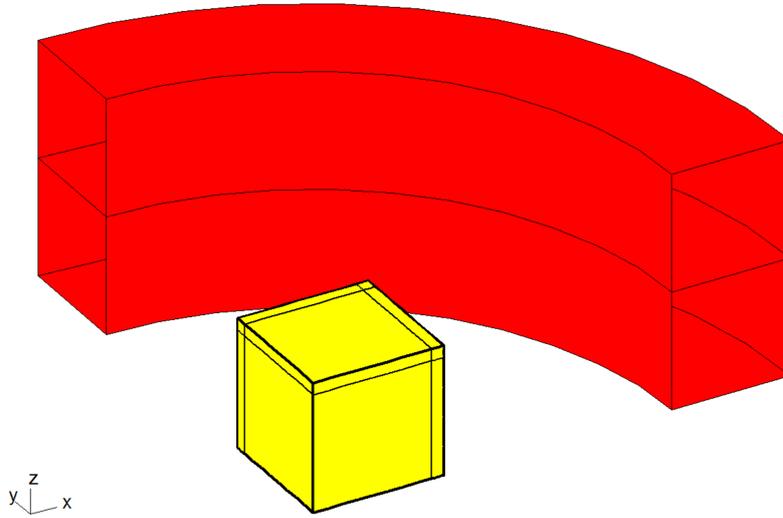


Figure 5.4 Geometry of the current-fed coil with a core.

steps are used in the MG algorithm. The number of iterations of the three solvers for solving this problem is 30, 8 and 4, respectively, which means that the MAFW smoother has better efficiency than the SGSCG smoother for eddy current problems discretized by anisotropic meshes.

Table 5.1 compares the computational data of all three solvers. Since the inverse of the submatrices associated with the AFW blocks has to be calculated, the preprocessing time of the MAFW smoother is more than that of the SGSCG smoother. However, the MG and MGCG solvers with the MAFW smoother achieve a decrease by 65% and 81% of the solution time compared to that of the MG solver with the SGSCG smoother.

The plane smoother

As presented in section 3.5.3, the plane smoother applied to the \mathbf{A} and Φ formulations has a good smoothing ability for thin elements used in magnetostatic problems. In this section, the plane smoother for both the \mathbf{A}, V and \mathbf{T}, Φ formulations will be presented for thin plates of conductors discretized by thin finite elements [32, 33].

For finite element formulations discretized by both vector and scalar potentials, both edge and node based plane smoothing blocks associated with their

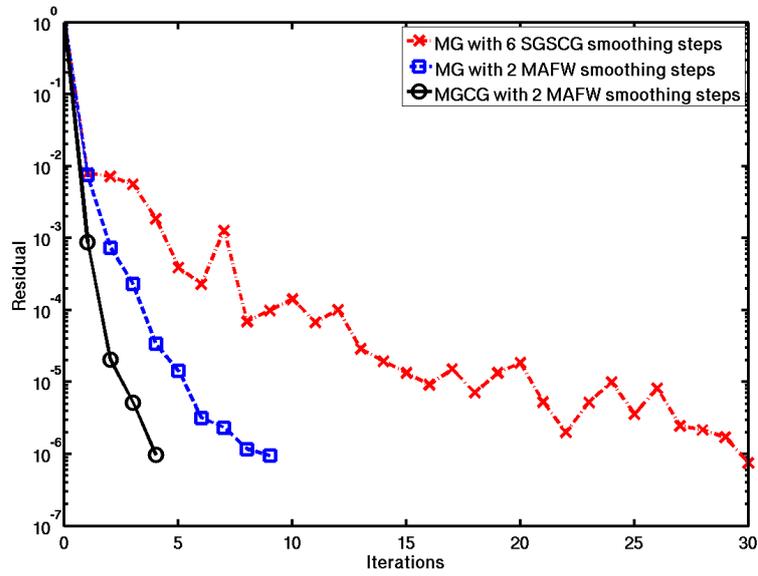


Figure 5.5 Residual versus iterations.

Table 5.1 Solution data for the test eddy current problem with 19336 unknowns.

	MG with 6 SGSCG	MG with 2 MAFW	MGCG with 2 MAFW
Number of blocks	0	1331	1331
Preprocessing time (s)	1.8	5.3	5.3
Solution time (s)	275.1	96.8	52.2

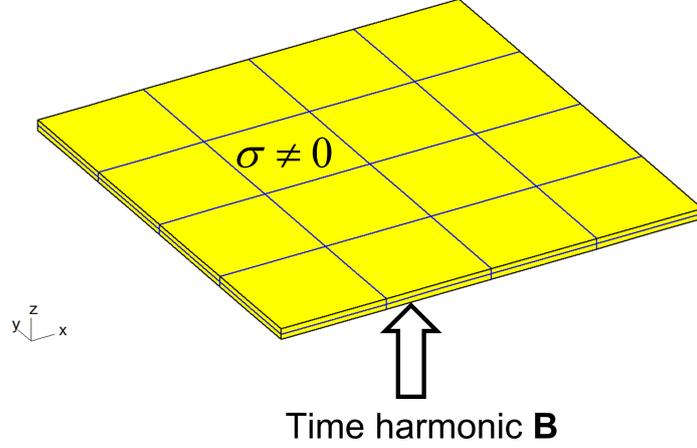


Figure 5.6 One thin conductor exposed to a time-harmonic magnetic flux density.

submatrices have to be constructed. For a plane indexed by i ($i = 1, 2, \dots, \hat{n}$), the symmetric MSI iteration instead of the pointwise SGS or SGSCG iteration can be applied to (5.24) or (5.42), and its forward iteration can be written as

$$\mathbf{x}_{i,e}^m = \mathbf{x}_{i,e}^{m-1} + P_i^K K_i^{-1} R_i^K (\mathbf{f}_1 - (K_D - K_U) \mathbf{x}_e^{m-1} - (-K_L) \mathbf{x}_e^m - C \mathbf{x}_n^{m-1}), \quad (5.62)$$

$$\mathbf{x}_{i,n}^m = \mathbf{x}_{i,n}^{m-1} + P_i^B B_i^{-1} R_i^B (\mathbf{f}_2 - C^T \mathbf{x}_e^m - (B_D - B_U) \mathbf{x}_n^{m-1} - (-B_L) \mathbf{x}_n^m) \quad (5.63)$$

where \mathbf{x}_e and \mathbf{x}_n represent vector potentials and scalar potentials, respectively, K_D , K_U and K_L are the diagonal matrix, strictly upper and lower triangular matrices of K , respectively, B_D , B_U and B_L are the diagonal matrix, strictly upper and lower triangular matrices of B , respectively, K_i and B_i are the submatrices of K and B , respectively, and P_i^K and P_i^B are the corresponding mapping operators of K_i and B_i , respectively.

The convergence of the MG method with the proposed plane smoother will be investigated by a test problem which consists of a thin plate shown in Fig. 5.6. The relative permeability of the plate is $\mu_r = 1 \times 10^3$ and the imposed magnetic field in z direction is produced by a current-fed coil around the plate. The conductivity of the plate is assumed to be $2.3 \times 10^3 \text{S/m}$. δ_{max} can be adjusted from 1 to 1×10^3 by simply varying the thickness of the plate.

Figure 5.7 compares the solution times of the MG solver with the SGSCG smoother, and the MG solver, as well as the MGCG solver with the proposed

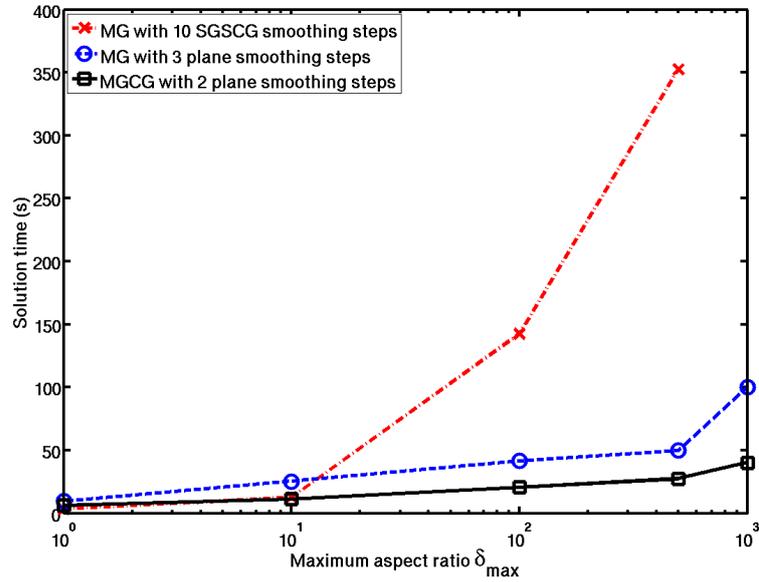


Figure 5.7 Solution time versus the maximum aspect ratio for the problem described by the $\mathbf{A}, V\text{-}\mathbf{A}$ formulation.

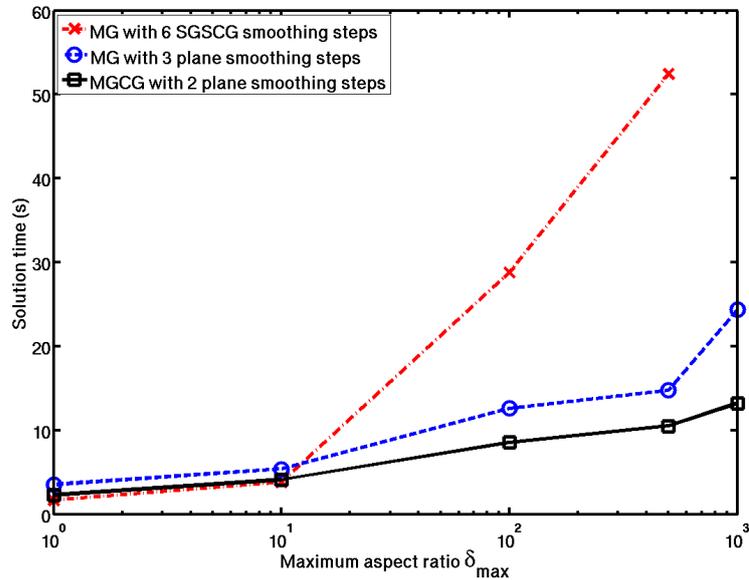


Figure 5.8 Solution time versus the maximum aspect ratio for the problem described by the $\mathbf{T}, \Phi\text{-}\Phi$ formulation.

plane smoother for solving the resulting system of equations arising from the $\mathbf{A}, V\text{-}\mathbf{A}$ formulation. It can be seen that the solution time of the MG method with 10 steps of SGSCG smoothing iterations is less than those of the other two solvers for $\delta_{max} = 1$ but increases significantly as δ_{max} becomes larger than 10. In contrast to that, the MG method with 3 plane smoothing iterations is capable of efficiently reducing the errors and has a stable convergence with the increase of δ_{max} . Additionally, the MGCG solver with two plane smoothing iterations results in further improvement of the efficiency. For instance, the MGCG solver achieves 83% of the decrease of the solution time compared to the MG solver with the SGSCG smoother for the case of $\delta_{max} = 500$. For the case of $\delta_{max} = 1000$, the MG solver with 10 steps of SGSCG smoothing iterations fails to converge to the solution and the increase of solution times of the MG and MGCG solvers with plane smoothing iterations are noticeable but still acceptable.

Fig. 5.8 shows similar results obtained from the MG and MGCG solvers for the $\mathbf{T}, \Phi\text{-}\Phi$ formulation to those for the $\mathbf{A}, V\text{-}\mathbf{A}$ formulation. For the case of $\delta_{max} = 500$, the MG and MGCG solvers with two plane smoothing iterations need only approximate 27% and 19% of the solution time of the MG solver with the SGSCG smoother. In the case of $\delta_{max} = 1000$, for the MGCG solver, the increase of its solution time is only 13% of that of the case with $\delta_{max} = 500$, which means that the MGCG solver with the plane smoother maintains its high efficiency for extremely thin elements.

5.4 Numerical examples

Two practical examples will be presented in this section to demonstrate the efficiency of the MGCG solver with the MAFW smoother and the plane smoother for nonlinear eddy current problems, and their performance will be compared with that of the MG solver with the SGSCG and the ICCG solvers.

5.4.1 Example 1

A single-phase power transformer consisting of two coils wound around a core in a tank is described by the $\mathbf{A}, V\text{-}\mathbf{A}$ formulation and the corresponding MG method is a four-level V-cycle algorithm detailed in Table 5.2. The geometry and discretization of the finite elements for the power transformer are displayed in

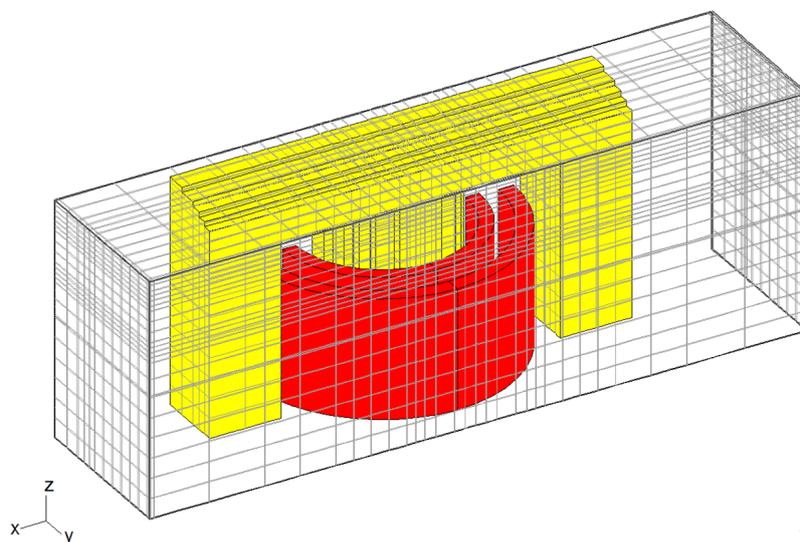
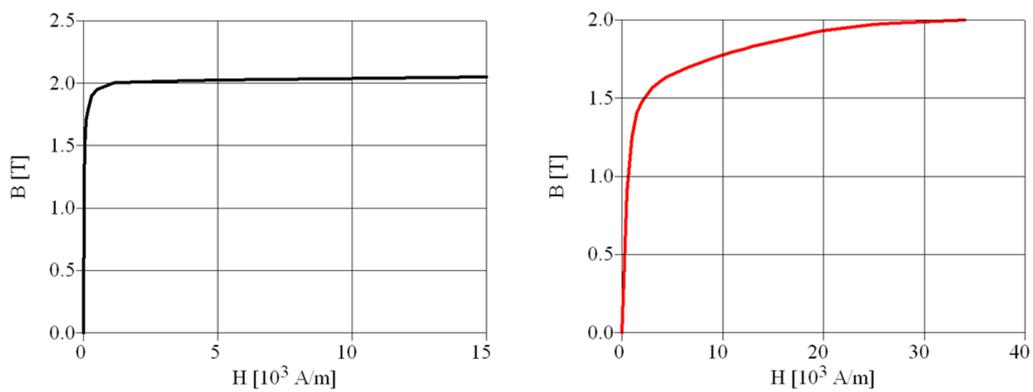


Figure 5.9 Finite element discretization of the single-phase power transformer 1 in a metallic tank



(a) The B-H curve of the tank

(b) The B-H curve of the core

Figure 5.10 The magnetization curves of (a) the tank and (b) the core

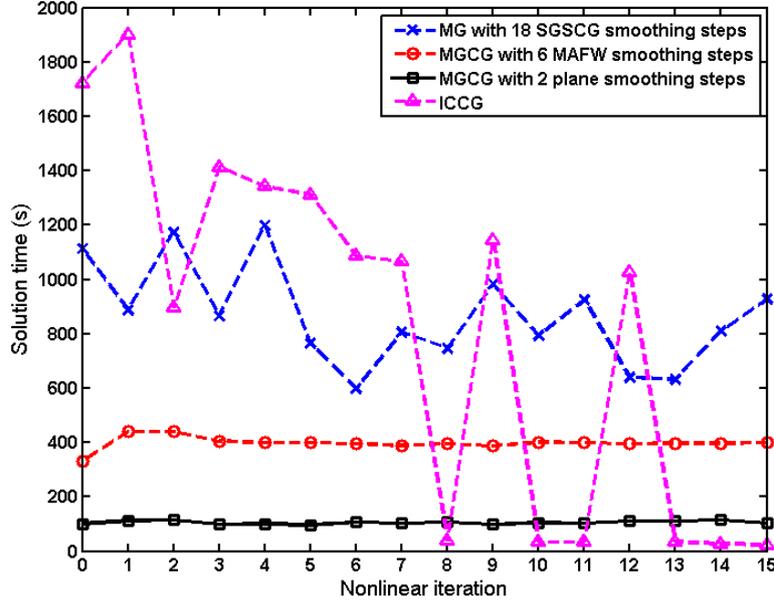


Figure 5.11 Solution time of different methods versus the nonlinear iteration

Fig. 5.9. Due to the symmetry, it is only necessary to simulate one quarter of the power transformer by applying proper boundary conditions. The B - H curves of the tank in Ω_c and the core in Ω_n are depicted in Fig. 5.10(a) and Fig. 5.10(b). The conductivity of the tank is $\sigma = 4.5 \times 10^6 \text{S/m}$. The current density of the excitation in coils is $2.16 \times 10^6 \text{A/m}^2$ and $1.95 \times 10^6 \text{A/m}^2$, and the frequency of the current is 60Hz. δ_{max} of the FEM mesh modelling the tank is about 115.

According to algorithm 5 (see section 5.2), the linear solution of this problem is calculated with a given relative reluctivity of the core and tank equal to 4×10^{-5} and 1×10^{-3} , respectively. The number of iterations and solution times of the MGCG solver with the MAFW smoother and the plane smoother compared to those of the MG solver with the SGSCG smoother and the ICCG solver for the linear solution are listed in Table 5.3.

It is found that the convergence of the solution cannot be obtained until the number of the smoothing iterations is increased to 18 in the MG method, which means that the smoothing ability of the SGSCG smoother is poor for this problem. In contrary to that, the number of block smoothing iterations applied to the MGCG method is relatively small. In particular, only 2 steps of plane smoothing

Table 5.2 Discretization of the finite elements for example 1 at each MG level.

Level	No. of equations	No. of elements
1	307847	24244
2	180396	14040
3	81286	6384
4	45985	4950

Table 5.3 Linear solution data of different solvers for example 1.

	ICCG	MG with 18 SGSCG	MGCG with 6 MAFW	MGCG with two plane smoothing
Preprocessing time(s)	24.697	58.343	376.274	270.766
Iterations	3602	6	5	8
Solution time (s)	1361.78	880.647	331.43	101.96

iterations are applied to the MGCG solver. The linear solution times of the MGCG with 6 MAFW smoothing steps and two plane smoothing steps are about 20% and 6% of that of the ICCG solver, respectively.

Using the linear solution as the initial guess, following the nonlinear algorithm 5 introduced in section 5.2, it applies 15 iterations to obtain the nonlinear solution for all four methods. Fig. 5.11 compares the solution times of all solvers at each nonlinear iteration step. The solution time of the MGCG remains nearly constant and is not influenced by the variation of the nonlinear reluctivity, while those of the MG and the ICCG solvers vary in a wide range. Moreover, compared to the ICCG solver, the MGCG solver is much faster in the first few steps but slower in the following steps. This is because at certain steps an initial guess which is close to the exact solution can more efficiently reduce the number of iterations for the ICCG than for the MGCG. Nevertheless, the overall solution time of the MGCG with the MAFW smoother and the plane smoother is about 59.1% and 36.9% of that of the MG solver with the SGSCG smoother, and 62.7% and 39.2% of that of the ICCG, respectively.

5.4.2 Example 2

The geometry of the problem and the discretization of the coarse grid of the second example are displayed in Fig 5.12. Due to the symmetry, it is only necessary to simulate one half of the power transformer by applying proper boundary conditions. The \mathbf{T}, Φ - Φ formulation is used to describe the problem.

The numerical computation is carried out by a four-level V-cycle MG characterized in Table 5.4. The linear solution of this problem is calculated by assuming the values of the relative permeability of the core and tank constants equal to 2.5×10^4 and 1×10^3 , respectively. The B - H curves of the tank and the core are the same as those of the first example. The δ_{max} of the FEM mesh modelling the tank is larger than 150. The convergence behavior of different methods is compared in Fig 5.13.

The most time-consuming step for solving this problem by the MG method is that of solving the system of equations with 202039 unknowns for the coarsest level by the direct solver. To improve this, the MGCG solver with four plane smoothing iterations converges to the solution in four iterations, which results in the direct solver being called only four times during the MG iterations. Conversely, the MG with the SGSCG smoother fail to converge to the normalized

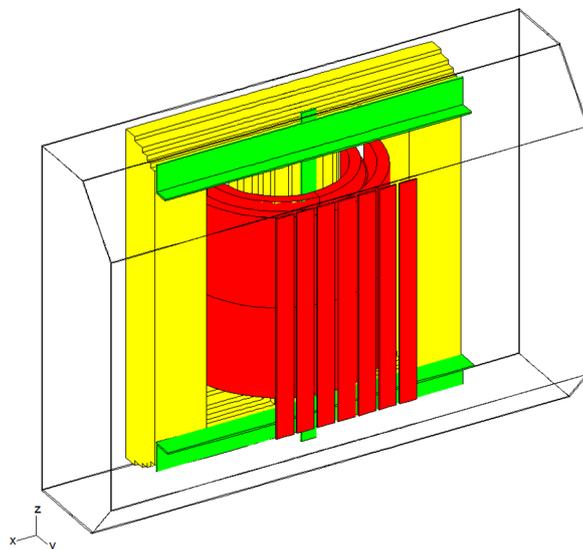


Figure 5.12 Geometry of example 2 in a metallic tank.

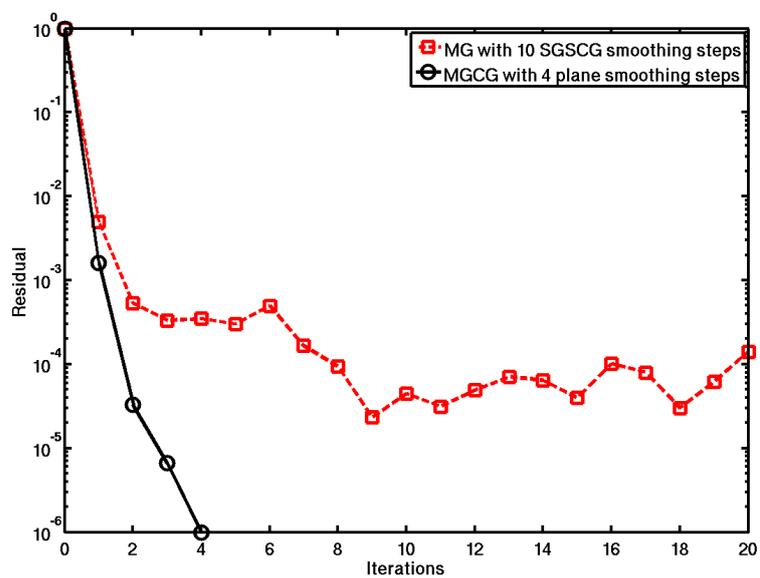


Figure 5.13 Number of iterations for the power transformer.

Table 5.4 Discretization for example 2

Level	No. of equations	No. of elements
1	1511021	238134
2	560256	90720
3	202039	75600

Table 5.5 Solution data for example 2

	ICCG	MGCG with 4 plane smoothing
Preprocessing time (s)	133.240	2269.159
Linear solution time (s)	9576.327	2141.725
Nonlinear solution time (s)	28419.91	18811.34

residual of 10^{-6} in twenty iterations.

The solution times of the ICCG and MGCG solvers are compared in Table 5.5. For the linear and nonlinear solutions, the solution times of the MGCG solver are 22.3% and 66.2% of those of the ICCG solver.

Conclusion

The multigrid method for solving the systems of equations resulting from the $\mathbf{A}, V\text{-}\mathbf{A}$ formulation and the $\mathbf{T}, \Phi\text{-}\Phi$ formulation is introduced in this work. Different smoothers are developed to improve the multigrid performance for problems with nonlinearities and anisotropies.

It has been shown that block iterations have better smoothing ability than pointwise iterations for dealing with the anisotropy in the finite element mesh. In particular, plane smoothers for nodal elements applied to the Φ formulation and those for edge elements applied to the \mathbf{A} formulation are cheap and efficient for the static magnetic analysis with thin magnetic plates discretized by anisotropic elements. For eddy current problems, the multigrid with the plane smoother adapted to the \mathbf{A}, V formulation and the \mathbf{T}, Φ formulation retains its high efficiency even for elements with extremely high aspect ratios. In addition, the convergence of the solver can be further improved by using the multigrid method as the preconditioner of the conjugate gradient method.

For practical applications, numerical results show that the multigrid preconditioned conjugate gradient iteration with the proposed block smoothers can substantially decrease the solution time in both linear and nonlinear problems.

The major contributions of this work can be summarized as follows:

- Various smoothers adapted to $\mathbf{A}, V\text{-}\mathbf{A}$ formulation and the $\mathbf{T}, \Phi\text{-}\Phi$ formulation are successfully applied to the multigrid method for solving magnetostatic and eddy current problems.
- The efficiency of the multigrid method for problems with thin elements has been enhanced by different block iterations especially the plane smoothing

Chapter 6 Conclusion

iteration.

- The performance of the solver has been further improved by using the multi-grid method as the preconditioner of the conjugate gradient method.
- Numerical results of practical applications show the advantages of the multi-grid preconditioned conjugate gradient method with the proposed block smoothers over the incomplete Cholesky preconditioned conjugate gradient method.

Bibliography

- [1] K. Stüben and U. Trottenberg, *Multigrid Methods: Fundamental Algorithm, Model Problem Analysis and Algorithms*. In *Multigrid Methods*, Lecture Notes in Mathematics, pp.1-176, Springer-Verlag, Berlin, 1982.
- [2] R. P. Fedorenko, “A relaxation method for solving elliptic difference equations,” *U.S.S.R. Comput. Math. and Math. Phys.*, vol. 1, pp. 1092–1096, 1962.
- [3] —, “The speed of convergence of one iterative process,” *U.S.S.R. Comput. Math. and Math. Phys.*, vol. 4, pp. 227–335, 1964.
- [4] N. S. Bakhvalov, “On the convergence of a relaxation method under natural constraints on an elliptic operator,” *Z. Vycisl. Mat. i. Mat. Fiz.*, vol. 6, pp. 861–883, 1964.
- [5] A. Brandt, “Multi-level adaptive solutions to boundary value problems,” *Math. Comp.*, vol. 31, pp. 333–390, 1977.
- [6] V. G. Korneev, *Finite element schemes of higher order of accuracy*. Leningrad University Press, 1977.
- [7] H. Deconinck and C. Hirsch, *A multigrid finite element method for the transonic potential equation*. In *Multigrid Methods*, Lecture Notes in Mathematics, pp. 387-409, Springer-Verlag, Berlin, 1982.

Bibliography

- [8] A. Brandt, *Guide to multigrid development*. In *Multigrid Methods*, Lecture Notes in Mathematics, pp. 220-312, Springer-Verlag, Berlin, 1982.
- [9] W. Hackbusch, *Multigrid methods and applications*. Berlin: Springer-Verlag, 1985.
- [10] P. Wesseling, *An introduction to multigrid methods*. Reprint of the 1992 edition, <http://www.MGNet.org>, Cos Cob, 2001.
- [11] K. Stüben, *An introduction to algebraic multigrid*. In *Multigrid, Appendix A*, Academic Press, London, 2000.
- [12] U. Trottenberg and A. Schüller, *Multigrid*. London: Academic Press, 2001.
- [13] C. S. Koh, K. Choi, S. Hahn, and H. K. Jung, “An adaptive finite element scheme using multi-grid method for magnetostatic problems,” *IEEE Trans. on Magn.*, vol. 25, no. 4, pp. 2959–2961, 1989.
- [14] H. Igarashi, “On the property of the curl-curl matrix in finite element analysis with edge elements,” *IEEE Trans. on Magn.*, vol. 37, no. 5, pp. 3129–3132, 2001.
- [15] R. Hiptmair, “Multigrid method for maxwell’s equations,” *SIAM J. Numer. Anal.*, vol. 36, no. 1, pp. 204–225, 1998.
- [16] D. N. Arnold, R. S. Falk, and R. Winther, “Multigrid in $h(\text{div})$ and $\mathbf{H}(\text{curl})$,” *Numer. Math.*, vol. 85, no. 2, pp. 197–217, 2000.
- [17] M. Schinnerl, J. Schöberl, and M. Kaltenbacher, “Nested multigrid methods for the fast numerical computation of 3d magnetic fields,” *IEEE Trans. on Magn.*, vol. 36, no. 4, pp. 1557–1560, 2000.
- [18] B. Weiß and O. Bíró, “Smoothing operators for edge element multigrid,” *IEEE Trans. on Magn.*, vol. 38, no. 2, pp. 357–364, 2002.
- [19] H. D. Gersem and K. Hameyer, “Full multigrid for magnetostatics using unstructured and non-nested meshes,” *IEEE Trans. on Magn.*, vol. 37, no. 5, pp. 3460–3464, 2001.
- [20] B. Weiß and O. Bíró, “Edge element multigrid solution of nonlinear magnetostatic problems,” *Compel*, vol. 20, no. 2, pp. 357–364, 2001.

- [21] V. Spasov and H. Yamashita, “Fast 3-d edge element analysis by the geometric multigrid method using an accelerated symmetric gauss-seidel smoother,” *IEEE Trans. on Magn.*, vol. 39, no. 3, pp. 1685–1688, 2003.
- [22] T. Iwashita, T. Mifune, and M. Shimasaki, “Similarities between implicit correction multigrid method and a-phi formulation in electromagnetic field analysis,” *IEEE Trans. on Magn.*, vol. 44, no. 6, pp. 946–949, 2009.
- [23] M. Kaltenbacher and S. Reitzinger, *Algebraic multigrid for solving electromechanical problems*. In *Multigrid Methods*, Lecture Notes in Mathematics, pp. 129-135, Springer-Verlag, Berlin, 2000.
- [24] S. Reitzinger and J. Schöberl, “An algebraic multigrid methods for finite element discretizations with edge elements,” *Numer. Linear Algebra Applicat.*, vol. 31, no. 3, pp. 223–238, 2002.
- [25] O. Bíró and K. Preis, “On the use of the magnetic vector potential in the finite element analysis of three-dimensional eddy currents,” *IEEE Trans. on Magn.*, vol. 25, pp. 3145–3159, 1989.
- [26] O. Bíró, “Edge element formulations of eddy current problems,” *Computer Methods in Applied Mechanics and Engineering*, vol. 169, pp. 391–405, 1999.
- [27] B. Weiß and O. Bíró, “Multigrid for transient 3d eddy current analysis,” *Compel*, vol. 22, no. 3, pp. 779–788, 2003.
- [28] —, “Multigrid for time-harmonic 3-d eddy-current analysis with edge elements,” *IEEE Trans. on Magn.*, vol. 41, no. 5, pp. 1712–1715, 2005.
- [29] M. Hofer, M. Kaltenbacher, and S. Reitzinger, “Algebraic multigrid preconditioner for harmonic eddy current problems in 3-d,” *IEEE Trans. on Magn.*, vol. 40, no. 2, pp. 1342–1345, 2004.
- [30] T. Boonen, J. L. Van, D. H. Gersem, J. Driesen, and S. Vandewalle, “Algebraic multigrid for implicit runge kutta discretizations of the eddy current problem,” *IEEE Trans. on Magn.*, vol. 43, no. 4, pp. 1265–1268, 2006.
- [31] C. Chen and O. Bíró, “Edge element multigrid solution of time-harmonic 3d nonlinear eddy current problems,” in *Proc. of the Compumag2009, Florianopolis, Brazil*, 2009, pp. 9–12.

Bibliography

- [32] Chen and O. Bíró, “Geometric multigrid with plane smoothing for thin elements in 3-d magnetic fields calculation,” in *Proc. of the Compumag2011, Sydney, Australia*, 2011, pp. 173–174.
- [33] C. Chen and O. Bíró, “Geometric multigrid with plane smoothing for thin elements in 3-d magnetic fields calculation,” *IEEE Trans. on Magn.*, vol. 48, no. 2, pp. 443–446, 2012.
- [34] D. Braess, *On the combination of the multigrid method and conjugate gradients*. In *Multigrid Methods*, Lecture Notes in Mathematics, pp. 52-64, Springer-Verlag, Berlin, 1986.
- [35] T. Mifune, T. Iwashita, and M. Shimasaki, “New algebraic multigrid preconditioning for iterative solvers in electromagnetic finite edge-element analyses,” *IEEE Trans. on Magn.*, vol. 39, no. 3, pp. 1677–1680, 2003.
- [36] W. Hackbusch, *Iterative Solution of Large Sparse Systems of Equations*. Berlin: Springer-Verlag, 1993.
- [37] Y. Saad, *Iterative Methods for sparse linear systems*. Boston, USA: PWS Publishing Company, 1996.
- [38] D. J. Paddon and H. Holstein, *Multigrid Methods for Integral and Differential Equations*. Oxford: Clarendon Press, 1985.
- [39] S. Reitzinger, *Algebraic Multigrid Methods for Large Scale Finite element Equations*. PhD thesis, Linz, Austria: Johannes Kelper University, 2001.
- [40] J. T. Oden and J. Reddy, *An Introduction to the Mathematical Theory of Finite Elements*. New York: John Wiley and Sons, 1976.
- [41] V. Girault and P. A. Raviart, *Finite Elements Methods for the Navier-Stokes Equations*. New York: Springer-Verlag, 1986.
- [42] P. G. Ciarlet, *The Finite Element Method for Elliptic Problems*. Amsterdam, New York, Oxford: North-Holland Publishing Company, 1986.
- [43] J. C. Nédélec, “Mixed finite elements in \mathbb{R}^3 ,” *Numer. Math*, vol. 35, pp. 315–341, 1980.

- [44] A. Kameari, “Calculation of transient 3d eddy current using edge elements,” *IEEE Trans. on Magn.*, vol. 26, pp. 466–469, 1990.
- [45] J. H. Bramble, J. E. Pasciak, J. Wang, and J. Xu, “Convergence estimates for product iterative methods with applications to domain decompositions and multigrid,” *Math. Comp.*, vol. 57, no. 195, pp. 1–21, 1991.
- [46] O. Bíró, K. Preis, and K. R. Richter, “On the use of the magnetic vector potential in the nodal and edge finite element analysis of 3d magnetostatic fields,” *IEEE Trans. on Magn.*, vol. 25, no. 4, pp. 651–654, 1996.
- [47] O. Bíró, K. Preis, G. Vrisk, K. R. Richter, and I. Tícar, “Computation of 3-d magnetostatic fields using a reduced scalar potential,” *IEEE Trans. on Magn.*, vol. 29, no. 2, pp. 1329–1332, 1993.
- [48] O. Schenk and K. Gärtner, “Solving unsymmetric sparse systems of linear equations with pardiso,” *Journal of Future Generation Computer Systems*, vol. 20, no. 3, pp. 475–487, 2004.
- [49] O. Bíró and K. Preis, *Time-periodic 3D eddy currents*. In Honma, T. (Ed.), *Advanced Computational Electromagnetics*, IOS Press, Amsterdam, 1995.
- [50] C. Chen and Bíró, “3d time-harmonic eddy current problems solved by the geometric multigrid preconditioned conjugate gradient method,” in *Proc. of the CEM2011, Wroclaw, Poland*, 2011, pp. 18–19.
- [51] C. Chen and O. Bíró, “Three-dimensional time-harmonic eddy current problems solved by the geometric multigrid preconditioned conjugate gradient method,” *IET Sci. Meas. Technol.*, vol. 6, no. 5, pp. 319–323, 2012.