FINITE ELEMENT ANALYSIS WITH EMBEDDED GLOBAL OPTIMIZATION METHOD FOR OPTIMAL DESIGN OF ELECTRIC DEVICES

CHEN NINGNING

Ph.D

The Hong Kong Polytechnic University

2012

The Hong Kong Polytechnic University

Department of Electrical Engineering

Finite Element Analysis with Embedded Global Optimization Method for Optimal Design of Electric Devices

CHEN Ningning

A thesis submitted in partial fulfillment of the requirements for
the degree of Doctor of Philosophy

October 2011

Certificate of Originality

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

Name: Chen Ningning

Abstract of thesis entitled

"Finite Element Analysis Embedded Global Optimization Method for Optimal Design of Electric Devices"

Submitted by

Chen Ningning

for the degree of Doctor of Philosophy

at The Hong Kong Polytechnic University

in October 2011

In this project, three-dimensional (3D) Finite element method (FEM) embedded global optimization method is applied to obtain optimal design electric devices. FEM with nodal basis function and edge basis function are used to analysis magnetic field and the eddy current problem in electric devices. Time step FEM with slave master technique, circuit coupling technique is applied to simulate the performance of electric motor. Global optimization method including Genetic algorithm (GA), simulated annealing (SA), taboo search (TA) and particle swarm optimization (PSO) are coupled with FEM program to find the parameters of the optimal design. Parameter extraction technique is applied to extract the mass parameters of electric motor to accelerate the computation speed of optimization process. Moving least square (MLS) based surface response model is applied to reduce the optimization time. The coupled FEM with optimization method is applied to optimize the performance of surface mounted PM motor, magnetic gear and an axial flux magnetic motor.

In the thesis, the following work has been done:

(1) 2D and 3D FEM program for eddy current have been implemented and couple with optimization method.

(2) A two-grid FEM has been studied for reducing the 3-D FEM computation time to reduce the computation time of nonlinear problems. TEAM Workshop problem 13 is used to test the two-grid method and the results obtained using the two-grid algorithm are compared with those using conventional methods. Since the two-grid method requires less computing time, it can be effectively applied to study large-scale nonlinear problems.

(3) Particle swam optimization (PSO) optimal algorithm and genetic algorithm (GA) have been implemented for optimization computation. A surrogate based on moving least squares (MLS) method has been developed to approximate the expected fitness evaluations, which would replace many times of FEA computation and would save much computation time. FEM with embedded GA has been applied to optimize the shape design of permanent magnet (PM) in motors to reduce the cogging torque.

(4) Moving mesh FEM has been proposed for coupling FEM with optimization method.

(5) The state of art programming method with object oriented programming technique for FEM is introduce to faculae the complexity of programming.

The major contributions of the thesis are reflected in the following aspects:

It proposes a novel moving mesh method to handle the mesh re-generation in the optimization step; Two-grid method is introduced to reduce the computational time of the nonlinear Maxwell system with nonlinear materials, and an effective interpolation of the FE solution from coarse grid to non-nested fine grid is proposed. 3D nodal and edge FEMs are implemented to analyze the eddy current problem involved in electric devices, and coupled with some global optimization process to design optimal electric devices. The FEM with global optimization method is applied to optimize the performance of PM motors and magnetic gears. Slave master technique is extended for 3D FEM computations to handle the non-conforming meshed on the interface between the moving and static parts of the devices. Time step FEM along with the slave master technique and circuit coupling technique is used to simulate the performance of the electric motors.

List of Publications

Journal papers:

[1] Ningning Chen, S. L. Ho, and W. N. Fu, “Optimization of permanent magnet surface shapes of electric motors for minimization of cogging torque using FEM”, *IEEE Trans. Magn*., vol. 46, no. 6, pp.2478-2481, Jun. 2010.

[2] Ningning Chen, S. L. Ho, and W. N. Fu, “An efficient two-grid finite-element method of 3-D nonlinear magnetic-field computation”, *IEEE Trans. Magn*., vol. 45, no. 10, pp.4797-4800, Oct. 2009.

[3] S. L. Ho, Ningning Chen, and W. N. Fu, “A moving mesh embedded algorithm in finite element method for optimal design of electromagnetic devices,” *IEEE Trans. Magn*., vol. 47, no. 10, to be published, Oct. 2011.

[4] S. L. Ho, Ningning Chen, and W. N. Fu, “An optimal design method for the minimization of cogging torques of a permanent magnet motor using FEM and genetic algorithm”, *IEEE Trans. Applied Superconductivity,* vol. 20, no. 3, pp.861-864, Jun. 2010.

[5] W. N. Fu, S. L. Ho, and Ningning Chen, “Application of shell element method to 3-D finite-element computation of the force on one body in contact with others”, *IEEE Trans. Magn*., vol. 46, no. 11, pp.3893-3898, Nov. 2010.

Conference papers:

[1] Ningning Chen, Shuangxia Niu, S. L. Ho, W. N. Fu and Jianguo Zhu, “State-of-the-art programming techniques of finite element methods for electromagnetic field computation” *18th International Conference on the Computation of Electromagnetic Fields*, Sydney, Australia, July, 2011, Paper No 273.

[2] Ningning Chen, S. L. Ho, W. N. Fu and Jianguo Zhu, “A mapping technique in finite element method of magnetic field computation for reduction of optimization

computation time,” *18th International Conference on the Computation of Electromagnetic Fields*, Sydney, Australia, July, 2011, Paper No 573.

[3] Ningning Chen, S. L. Ho and W. N. Fu , “A novel mapping Technique between the FEM solution on two mesh for reduction of the computation time of optimization,” *IEEE International Magnetics Conference*, Taipei, Taiwan, April, 2011, Paper No.

[4] S. L. Ho, Ningning Chen, W. N. Fu and Jianguo Zhu, “An efficient FEM and FVM coupled algorithm for eddy current analysis in electromagnetic devices with high-speed moving conductors,” *18th International Conference on the Computation of Electromagnetic Fields*, Sydney, Australia, July, 2011, Paper No 356.

[5] S. L. Ho, Ningning Chen and W. N. Fu, “A moving mesh algorithm for electromagnetic devices optimization using finite element method,” *18th International Conference on the Computation of Electromagnetic Fields*, Sydney, Australia, July, 2011, Paper No 574.

[6] S. L. Ho, Ningning Chen and W. N. Fu, “Moving least-square approximation based interface element with variable nodes for modeling sliding-interface in electric machines,” *IEEE Conference on Electromagntic Field Computation 2011*, Chicago, Illinois, USA, May, 2010, Paper No. 1314.

[7] S. L. Ho, Ningning Chen and W. N. Fu, “An efficient mesh morphing algorithm for optimizing the shapes of electromagnetic devices using finite element method,” *IEEE Conference on Electromagntic Field Computation 2011*, Chicago, Illinois, USA, May, 2010, Paper No. 1457.

[8] S. L. Ho, Ningning Chen and W. N. Fu, “Automatic Design of High-speed Permanent Magnet Motors Based on 3-D Finite Element Methods, Artificial Neural Network and Genetic Algorithm,” *21st International Conference on Magnet Technology,* Hefei, Anhui, China, October, 2009.

[9] S. L. Ho, N. N. Chen, and W. N. Fu, “A Transient 3-D Finite Element Analysis of Magnetizing Process in Permanent Magnets,” *21st International Conference on*

Magnet Technology, Hefei, Anhui, China, October, 2009.

[10] Shuangxia Niu, Ningning Chen, S. L. Ho, W. N. Fu and Jianguo Zhu, “Design optimization of magnetic gears using a PSO based mesh adjustable finite element algorithm,” *18th International Conference on the Computation of Electromagnetic Fields*, Sydney, Australia, July, 2011, Paper No 357.

ACKNOLEDGEMENTS

I would like to express my sincere gratitude to my chief supervisor, Prof. S. L. Ho for his invaluable guidance, kind support and generous encourage in the past years. It is my great honor to study and work with Prof. Ho. I have learn a lot from my supervisor-his attitude towards life, devotion to research and endeavor in his career and this experience will benefit me for the whole life.

I would also like to thank my co-supervisor, Dr. W. N. Fu. I have learnt lots of knowledge from him. Thanks him for his valuable suggestion and encouragement in my research.

I also wish to express my appreciation to all the group members in our Computational Elctromagnetics Lab, especially Wang Junhua, Zhang Xiu, Dr. Niu Shuangxia.

I am grateful for the constant support and encouragement from my parents.

This work was supported by the Research Studentship of the Hong Kong Polytechnic University.

Contents

List of Figures

1. INTRODUCTION

1.1. APPLICATION OF FEM IN ELECTROMAGNETISM

Finite element method (FEM) was introduced to and applied in computational electromagnetism since 1950. With the development and wide application of FEM in electrical engineering, it has become a powerful tool to simulate the magnetic field and eddy current in electric devices, especially in electric motor. In the past, ***A***-formulation, ***A***-*φ* formulation and ***T***-*Ω* formulation are applied to solve Maxwell Equations.

When FEM is applied to simulate the motion of electric motor, the mesh on the rotor rotates with the rotor while the mesh on the stator remains static, and thus non-conforming mesh on the interface between the stator and the rotor is usually encountered. Slave master method is usually applied to deal with such problem. In the thesis, such method is extended to 3D problem for simulation of the dynamic performance of the electric motor.

Edge element is the newly constructed vector based element for solving Maxwell equations. The advantage of edge element is that it ensures the continuity of the tangential component of the vector variable and does not impose the continuity on the normal component. The vector potential ***A*** is approximated by edge element, and thus the jump of the normal component of ***A*** will provide the jump of flux density ***B*** on the iron-air interface.

The type of vector elements was first described by Whitney [1] in 1957. In 1980, Nedelec constructed edge elements on tetrahedral element for 3D computation [2]. M. L. Barton and Z. J. Cenders introduced the new vector basis functions for 3D magnetic field conputation [3]. In application, Bosavit applied edge element method to 3D eddy current problems [4]. Biro analyzed the characteristic of the matrix of ***A***-method, ***A***-*φ* method by using edge element [5].

There are many ways to implement FEM programming with computer language on computer. State-of-the-art programming techniques for implement of FEM by object oriented programming language for electromagnetic field computation are presented in the thesis. It covers program structure, data structure and algebraic matrix equation.

In the FEM programming, linear solver such as ICCG, BICG-stable and Pardiso solver are used. The FEM computation efficiency is determined mostly by the efficiency of linear equation solver. The stiff matrix of Maxwell Equations obtained by FEM are not positive definite and usually are singular. However, if the right hand side conforms to the curl-free, the linear equation will converge when iteration method are used. Pardiso solver is a high-performance serial and parallel sparse linear solver and it is used in the program to accelerate the computation speed.

1.2. OPTIMIZATION METHOD FOR ELECTROMAGNETICS

In electric engineering, the performance of the electric devices as a whole is considered and the objective is to improve its performance. The design of devices is represented by several parameters, which are the optimization variables in optimization method. Global optimization method is usually used to find the optimal solution of design to improve its performance.

Global optimization methods are more suitable for the inverse problem in electromagnetic problems because of its ability to find the global optimal solution. Several global optimization methods including genetic algorithm (GA), taboo search (TA), simulated annealing (SA), particle swarm optimization (PSO) method, ant colony algorithm and are artificial neural network (ANN) invented through the analysis and simulation of the natural phenomena, society activity and physical progress. Such optimization methods are good at finding the global solution, which are more suitable in the optimization electromagnetic devices. The objective function of the optimization of electromagnetic devices is usually constructed to maximize the performance or to minimize the loss of energy of the devices.

1.3. COUPLING FEM WITH OPTIMIZATION METHOD

In the thesis, global optimization method and FEM are coupled to obtain the optimal design of electric devices. In the coupling algorithm, FEM is applied to compute the electric and magnetic field. Then the result of FEM is used to calculate the objective function of the optimization method.

There are several difficulties in coupling FEM program with optimization method.

(1) The slow convergence of global optimization method;

(2) The computation time for FEM is too long, especially for 3D problems;

(3) The mesh regeneration for FEM is not convenient for coupling the FEM with optimization method;

In the thesis, moving mesh method is proposed to facilitate the complexity of regenerating mesh in the optimization step.

It is difficult to couple commercial FEM software with optimization method. Most commercial FEM software does not provide variable interface open to the user. The aim of most prevail FEM software is to analysis the fixed model of the devices.

(1) It is not convenient to get the result to do post process in commercial software;

(2) It is not convenient to embed the execute program of commercial software to the optimization algorithm;

(3) New algorithms are not convenient to be implemented with the commercial software

In order to accelerate the convergence and reduce the computation time, MLS based surface response model (SRM) is constructed to approximate the objective function. Several hundreds times results of FEM computation are used to construction the SRM and then the objective function is calculated by the SRM which will then reduce the computation time.

1.4. RESEARCH OBJECTIVES

The research objectives in the project are:

(1) FEM programming for 2D and 3D electromagnetic problem is implemented for simulating electromagnetic field. The objective oriented C++ programming language is used to implement the FEM program. The implemented program can be used to analysis the forward problem in electromagnetic problem.

(2) Optimization methods are investigated and applied to obtain optimal design of electromagnetic devices. The coupled optimization method with FEM program is applied to several application of electromagnetic devices design to improve its performance.

(3) The proposed method is applied to optimize the design of an axial flux magnetic motor.

1.5. THESIS OUTLINE

The scope of the dissertation covers the basic literature review of Maxwell equations and the potential formulation. Finite element method with nodal and edge basis functions are discussed and the programming structure for implement FEM are illustrated. Field - circuit coupling for simulation the dynamic characteristic of electric devices is analyzed. Several global optimization methods are discussed and the surface response model (SRM) based on MLS is applied to an application example. The structure of coupled FEM with optimization method is illustrated.

There are seven chapters in this thesis. In chapter one, the main point of the thesis and the objective of the project are introduced. In chapter two, the basic literature on computational electromagnetism and optimization methods are reviewed. FEM programming technique involving eddy current problems is proposed in charter three. In chapter four, two grid techniques for reduce computation time are investigated. In chapter five, optimization methods are introduced and the moving mesh method for coupling optimization method with FEM is proposed. In chapter six, three applications of coupled FEM with optimization method in electric motor are introduced. Conclusion and recommendation are discussed in chapter 7.

2. LITERATURE REVIEW ON EDDY CURRENT PROBLEM AND OPTIMIZATION METHOD

2.1. MAXWELL EQUATIONS

Maxwell equations describe the whole basic dieseline that electric field and magnetic field observe. The differential form of Maxwell equations shows as:

$$\nabla \times \boldsymbol{H} = \boldsymbol{J} + \frac{\partial \boldsymbol{D}}{\partial t} \tag{2.1}$$

$$\nabla \times \boldsymbol{E} = -\frac{\partial \boldsymbol{B}}{\partial t} \tag{2.2}$$

$$\nabla \cdot \boldsymbol{B} = 0 \tag{2.3}$$

$$\nabla \cdot \boldsymbol{D} = \rho \tag{2.4}$$

The variables are: ***E*** electro field intensity (volt/meter), **H** magnetic field intensity (ampere/meter), ***D*** electric flux density (coulomb/meter^3), ***B*** magnetic flux density (tesla), ***J*** electric current density (ampere/meter^2), and ρ is electric charge density (coulomb/meter^3). These variables may be functions of coordinate (x, y, z) and time t. The continuity equation which derived from (2.1) and (2.4) is:

$$\nabla \cdot \boldsymbol{J} + \frac{\partial \rho}{\partial t} = 0 \tag{2.5}$$

The following constitution relationships in Maxwell's equations describing the material property of a medium are:

$$\boldsymbol{B} = \mu \boldsymbol{H} \tag{2.6}$$

$$\boldsymbol{D} = \varepsilon \boldsymbol{E} \tag{2.7}$$

$$\boldsymbol{J} = \sigma \boldsymbol{E} \tag{2.8}$$

In the research scope of the project, the applications are focused on low frequency electromagnetic field. In low frequency problem, the frequency of the source varies slowly with time such that the wavelength is immensely longer than the size of geometry in the considering domain. Thus the field varies instantaneously with the change of the source other than lagging behind the change of the source. Many electric devices in engineering application belong to the low frequency problems. For such problem, the displace current density $\frac{\partial \boldsymbol{D}}{\partial t}$ in (2.1) is so small that it can be ignored when compared with conducting current density ***J***. The problems in electromagnetic field without considering the displace current density are called eddy current problem.

2.2. POTENTIALS FOR EDDY CURRENT PROBLEM

Usually the domain of eddy current problem is divided into two parts for further research, one part V_1 is eddy current region which includes the conduct media but not includes the source current and the other part V_2 is source region which includes the current source but not includes the conduct media [6]. Maxwell equations (2.1) - (2.4) for eddy current problems are shown as:

In eddy current region V_1 :

$$\nabla \times \boldsymbol{H} = \sigma \boldsymbol{E} \tag{2.9}$$

$$\nabla \times \boldsymbol{E} + \frac{\partial \boldsymbol{B}}{\partial t} = 0 \tag{2.10}$$

$$\nabla \cdot \boldsymbol{B} = 0 \tag{2.11}$$

In non eddy current region V_2 :

$$\nabla \times \boldsymbol{H} = \boldsymbol{J}_s \tag{2.12}$$

$$\nabla \cdot \boldsymbol{B} = 0 \tag{2.13}$$

The boundary conditions are:

$$\boldsymbol{B}\cdot\boldsymbol{n} = \mathbf{0} \tag{2.14}$$

$$\boldsymbol{H}\times\boldsymbol{n} = \mathbf{0} \tag{2.15}$$

On the interface of V_1 and V_2, the boundary conditions are:

$$\boldsymbol{B}_1\cdot\boldsymbol{n}_{12} = \boldsymbol{B}_2\cdot\boldsymbol{n}_{12} \tag{2.16}$$

$$\boldsymbol{H}_1\times\boldsymbol{n}_{12} = \boldsymbol{H}_2\times\boldsymbol{n}_{12} \tag{2.17}$$

Where, $\boldsymbol{J}_s$ represents for source current density, $\boldsymbol{n}$ for the unit normal direction on surface, $\boldsymbol{n}_{12}$ for the unit normal direction from region V_1 to V_2.

In the numerical method used for solving eddy current equations, several formulations are used. There are three kind of potentials applied typically which are scalar potential φ formulation, vector potential $\boldsymbol{A}$-φ formulation and vector formulation $\boldsymbol{T}$-Ω formulation. In the scope of the thesis, vector potential $\boldsymbol{A}$-φ formulation is applied.

2.2.1. *A-Φ* FORMULATION

$\boldsymbol{A}$-φ formulation is usually applied to solve equation (2.9) - (2.13). The computation domain is divided into eddy current region and non-eddy current region. In eddy current region, vector potential $\boldsymbol{A}$ for magnetic field and scalar potential φ formulation for electric field is applied; in non-eddy current region, only vector potential for magnetic field is applied.

From (2.11) and (2.13), there exists vector function $\boldsymbol{A}$, which satisfies:

$$\boldsymbol{B} = \nabla\times\boldsymbol{A} \tag{2.18}$$

Then substitute (2.18) to (2.10), we get:

$$\nabla\times\left(\boldsymbol{E}+\frac{\partial\boldsymbol{A}}{\partial t}\right)=0 \tag{2.19}$$

From (2.19), there exists a scalar φ which satisfies:

$$\boldsymbol{E}+\frac{\partial\boldsymbol{A}}{\partial t}=-\nabla\varphi \tag{2.20}$$

$$\boldsymbol{E} = -\frac{\partial \boldsymbol{A}}{\partial t} - \nabla \varphi \tag{2.21}$$

Substitute (2.18) - (2.19) to eddy current equations (2.9) - (2.13), we get the equations based on the vector potential ***A*** and scalar potential *φ*:

In region V_1:

$$\nabla \times (\nu \nabla \times \boldsymbol{A}) = -\sigma \frac{\partial \boldsymbol{A}}{\partial t} - \sigma \nabla \varphi \tag{2.22}$$

In region V_2:

$$\nabla \times (\nu \nabla \times \boldsymbol{A}) = \boldsymbol{J}_s \tag{2.23}$$

$$\boldsymbol{n} \cdot \nabla \times \boldsymbol{A} = \boldsymbol{0} \tag{2.24}$$

$$\nu \nabla \times \boldsymbol{A} \times \boldsymbol{n} = \boldsymbol{0} \tag{2.25}$$

$$\boldsymbol{n}_{12} \cdot \nabla \times \boldsymbol{A}_1 = \boldsymbol{n}_{12} \cdot \nabla \times \boldsymbol{A}_2 \tag{2.26}$$

$$(\nu_1 \nabla \times \boldsymbol{A}_1) \times \boldsymbol{n}_{12} = (\nu_2 \nabla \times \boldsymbol{A}_2) \times \boldsymbol{n}_{12} \tag{2.27}$$

Where, $v=1/\mu$ is the magnetic reluctance. Due to the definition of vector potential ***A*** in (2.18) ensures the divergence free of flux density ***B***, $\nabla \cdot \boldsymbol{B} = 0$ in (2.11) and (2.13) satisfies automatically.

However, (2.21) - (2.25) does not ensures the uniqueness of vector potential ***A***. Coulomb gauge need to be imposed in order to ensures the uniqueness of vector potential ***A*** [7]. The divergence free of A is imposed by Coulomb gauge.

$$\nabla \cdot \boldsymbol{A} = 0 \tag{2.28}$$

The Coulomb gauge is combined into equation (2.20) and (2.21) using penalty method, which lead to:

In V_1 region:

$$\nabla \times (\nu \nabla \times \boldsymbol{A}) - \nabla(\lambda \nabla \cdot \boldsymbol{A}) + \sigma \frac{\partial \boldsymbol{A}}{\partial t} + \sigma \nabla \varphi = 0 \tag{2.29}$$

In V_2 region:

$$\nabla \times (\nu \nabla \times \boldsymbol{A}) - \nabla(\lambda \nabla \cdot \boldsymbol{A}) = \boldsymbol{J}_s \tag{2.30}$$

Where λ is the penalty coefficient and it can be chosen to be same as *v*.

In V_1 region, $\nabla \cdot \boldsymbol{J}_e = 0$ implies that $\nabla \cdot \sigma \boldsymbol{E} = 0$ and the following equation should be satisfied:

$$\nabla \cdot \left(-\sigma \frac{\partial \boldsymbol{A}}{\partial t} - \sigma \nabla \varphi \right) = 0 \tag{2.31}$$

Finally the $\boldsymbol{A}$-φ formulation for eddy current equations (2.9) - (2.17) can be synthetically expressed as [6]:

In V_1 region:

$$\nabla \times (\nu \nabla \times \boldsymbol{A}) - \nabla(\lambda \nabla \cdot \boldsymbol{A}) + \sigma \frac{\partial \boldsymbol{A}}{\partial t} + \sigma \nabla \varphi = 0 \tag{2.32}$$

$$\nabla \cdot \left(-\sigma \frac{\partial \boldsymbol{A}}{\partial t} - \sigma \nabla \varphi \right) = 0 \tag{2.33}$$

In V_2 region:

$$\nabla \times (\nu \nabla \times \boldsymbol{A}) - \nabla(\lambda \nabla \cdot \boldsymbol{A}) = \boldsymbol{J}_s \tag{2.34}$$

On the boundary of computation domain:

$$\boldsymbol{n} \times \boldsymbol{A} = \boldsymbol{0} \tag{2.35}$$

$$\boldsymbol{n} \cdot \nabla \times \boldsymbol{A} = \boldsymbol{0} \tag{2.36}$$

$$\boldsymbol{n} \cdot \boldsymbol{A} = 0 \tag{2.37}$$

$$\nu \nabla \times \boldsymbol{A} \times \boldsymbol{n} = \boldsymbol{0} \tag{2.38}$$

On the interface of two different materials:

$$\boldsymbol{A_1}=\boldsymbol{A_2} \tag{2.39}$$

$$\boldsymbol{n}_{12} \cdot \nabla \times \boldsymbol{A}_1 = \boldsymbol{n}_{12} \cdot \nabla \times \boldsymbol{A}_2 \tag{2.40}$$

$$(\nu_1 \nabla \times \boldsymbol{A}_1) \times \boldsymbol{n}_{12} = (\nu_2 \nabla \times \boldsymbol{A}_2) \times \boldsymbol{n}_{12} \tag{2.41}$$

$$\boldsymbol{n} \cdot \left(-\sigma \frac{\partial \boldsymbol{A}}{\partial t} - \sigma \nabla \varphi \right) = 0 \tag{2.42}$$

The advantages of $\boldsymbol{A}$-φ formulation are:

(1) Due to the uniqueness of $\boldsymbol{A}$-φ, the stable numeric solution for any eddy current problem can be obtained;

(2) The boundary condition on internal interface are normal boundary condition which enable the such boundary condition will be automatically satisfied when FEM is applied;

(3) Such method is valid for multiply connected conductor domain;

(4) It is very convenient to deal with the current source;

(5) High accuracy solutions will be obtained by using ***A***-*φ* formulation.

2.3. Finite Element Method for Eddy Current Problem

The weak form of eddy current equation (2.32) - (2.42) and the Galerkin finite element method for solving it will be introduced. Because stiff matrix calculation on element is a crucial step for establish the linear equation for FEM, the detailed stiff matrix computation based on nodal element and edge element will be deduced.

2.3.1. Galerkin finite element method

Partial differential equations are usually solved in Soblev space and the equations are satisfied in weak sense [8]. The weak form of equation (2.32) - (2.42) is deduced below. For the function (*δ**A***, *δφ*) in test function space, the weak form of equation (2.32) - (2.34) becomes:

$$\int_V \left(\nabla \times (\nu \nabla \times \boldsymbol{A}) - \nabla(\lambda \nabla \cdot \boldsymbol{A}) + \sigma \frac{\partial \boldsymbol{A}}{\partial t} + \sigma \nabla \varphi \right) \delta \boldsymbol{A} dV = \int_V \boldsymbol{J}_s \delta \boldsymbol{A} dV \qquad (2.43)$$

$$\int_V \nabla \cdot \left(-\sigma \frac{\partial \boldsymbol{A}}{\partial t} - \sigma \nabla \varphi \right) \delta \varphi dV = 0 \qquad (2.44)$$

Equation (2.32) in region V_1 and (2.34) in region V_2 are written into (2.44) and in region V_2 , σ =0; in region V_1, $\boldsymbol{J}_s$=0.

From the basic formulation of vector function:

$$\nabla\cdot(\boldsymbol{a}\times\boldsymbol{b})=\boldsymbol{b}\cdot(\nabla\times\boldsymbol{a})-\boldsymbol{a}\cdot(\nabla\times\boldsymbol{b}) \tag{2.45}$$

$$\begin{aligned}\int_V\big(\nabla\times(\nu\nabla\times\boldsymbol{A})\big)\delta\boldsymbol{A}dV &= \int_V(\nu\nabla\times\boldsymbol{A})\nabla\times\delta\boldsymbol{A}dV+\int_V\nabla\cdot\big((\nu\nabla\times\boldsymbol{A})\times\delta\boldsymbol{A}\big)dV\\ &=\int_V(\nu\nabla\times\boldsymbol{A})\nabla\times\delta\boldsymbol{A}dV+\int_{\partial V}(\nu\nabla\times\boldsymbol{A})\times\delta\boldsymbol{A}\cdot\boldsymbol{n}ds\\ &=\int_V(\nu\nabla\times\boldsymbol{A})\nabla\times\delta\boldsymbol{A}dV+\int_{\partial V}\big((\nu\nabla\times\boldsymbol{A})\times\boldsymbol{n}\big)\cdot\delta\boldsymbol{A}ds\end{aligned} \tag{2.46}$$

From the boundary condition (2.38), we have $\int_{\partial V}\big((\nu\nabla\times\boldsymbol{A})\times\boldsymbol{n}\big)\cdot\delta\boldsymbol{A}ds=0$.

Therefore, (2.45) can be written as:

$$\int_V\left(\sigma\frac{\partial\boldsymbol{A}}{\partial t}\delta\boldsymbol{A}+(\nu\nabla\times\boldsymbol{A})\nabla\times\delta\boldsymbol{A}+\lambda\nabla\cdot\boldsymbol{A}\nabla\cdot\delta\boldsymbol{A}+\sigma\nabla\varphi\delta\boldsymbol{A}\right)dV=\int_V\boldsymbol{J}_s\delta\boldsymbol{A}dV \tag{2.47}$$

From the basic formulation of vector function,

$$\nabla\cdot(\alpha\boldsymbol{a})=\alpha\nabla\cdot\boldsymbol{a}+\boldsymbol{a}\cdot\nabla\alpha \tag{2.48}$$

The left term of (2.44) can be written as:

$$\begin{aligned}&\int_V\nabla\cdot\left(-\sigma\frac{\partial\boldsymbol{A}}{\partial t}-\sigma\nabla\varphi\right)\delta\varphi dV\\ &=\int_V\nabla\cdot\left(\left(-\sigma\frac{\partial\boldsymbol{A}}{\partial t}-\sigma\nabla\varphi\right)\delta\varphi\right)-\int_V\left(-\sigma\frac{\partial\boldsymbol{A}}{\partial t}-\sigma\nabla\varphi\right)\nabla\delta\varphi dV\\ &=\int_{\partial V}\left(-\sigma\frac{\partial\boldsymbol{A}}{\partial t}-\sigma\nabla\varphi\right)\delta\varphi\cdot\boldsymbol{n}ds+\int_V\left(\sigma\frac{\partial\boldsymbol{A}}{\partial t}+\sigma\nabla\varphi\right)\nabla\delta\varphi dV\end{aligned} \tag{2.49}$$

Substitute the boundary condition in (2.42) to (2.49), we obtain:

$$\int_V\left(\sigma\frac{\partial\boldsymbol{A}}{\partial t}\nabla\delta\varphi+\sigma\nabla\varphi\nabla\delta\varphi\right)dV=0 \tag{2.50}$$

Equation (2.47) and (2.50) are the weak form of equation (2.32) - (2.42).

When applying Galerking FEM, the computation domain is firstly divided into a set of nonintersecting elements. Then a set of basis functions defined locally on the nodes of elements $\{\boldsymbol{A}_1, \boldsymbol{A}_2, ..., \boldsymbol{A}_{3N}\}$ and $\{\varphi_1, \varphi_2, ..., \varphi_N\}$ are constructed. The unknown variable $\boldsymbol{A}$ and φ is express as the linear combination of the basis functions.

$$\boldsymbol{A} = \sum_{i=1}^{3N} a_i \boldsymbol{A}_i \tag{2.51}$$

$$\varphi = \sum_{i=1}^{N} b_i \varphi_i \tag{2.52}$$

Where N is the number of the nodes on the division of the computation domain; a_i and b_i are defined on nodes and represent the value of unknown ($\boldsymbol{A}$, φ) on the i^{th} node.

The test function ($\delta\boldsymbol{A}$, $\delta\varphi$) can be chosen as ($\boldsymbol{A}_i$, φ_j), $i=1, .., 3N, j=1, .., N$.

$$\delta\boldsymbol{A} = \begin{pmatrix} N_{xi} \\ 0 \\ 0 \end{pmatrix} \qquad \delta\boldsymbol{A} = \begin{pmatrix} 0 \\ N_{yi} \\ 0 \end{pmatrix} \qquad \delta\boldsymbol{A} = \begin{pmatrix} 0 \\ 0 \\ N_{zi} \end{pmatrix}$$

Substitute (2.51) and (2.52) and the test function into the weak form (2.47) and (2.50), one can obtain a linear equation of the unknowns (a_i ,b_i). After solving the linear equation and the unknowns (a_i ,b_i) will be obtained and then the solution of eddy current equation will be obtain by (2.51) and (2.52).

2.3.2. NODAL ELEMENT

In the section, stiff matrix calculation on element by using nodal element will be introduced. Firstly, a general method for stiff matrix calculation will be illustrated by using hexahedral element. Secondly the commonly used analysis basis functions on tetrahedral will be introduced.

Take the hexahedral element for example. A transformation from original coordinate (x, y, z) to reference coordinate (ξ, η, ζ) will be found to transform it to the standard cube as shown in Fig. 2.1. The transformation is defined as:

$$x = \sum_{i=1}^{8} x_i N_i(\xi,\eta,\zeta) \qquad y = \sum_{i=1}^{8} y_i N_i(\xi,\eta,\zeta) \qquad z = \sum_{i=1}^{8} z_i N_i(\xi,\eta,\zeta) \tag{2.53}$$

Where

$$N_i(\xi,\eta,\zeta) = (1+\xi_i\xi)(1+\eta_i\eta)(1+\zeta_i\zeta)/8 \tag{2.54}$$

is the linear interpolation basis functions on reference coordinate.

Where (ξ_i, η_i, ζ_i) are the reference coordinate of i^{th} vertex; for example, the reference coordinate of vertex 1 is (*-1, -1, -1*).

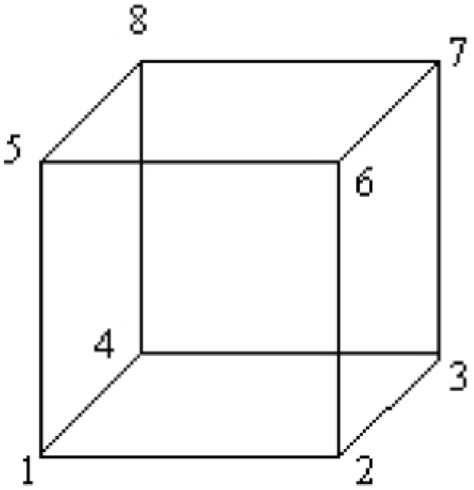


Fig. 2.1 The standard cube

The derivate of the basis function to the reference coordinate are:

$$N_i^{(\xi)}(\xi,\eta,\zeta) = \xi_i(1+\eta_i\eta)(1+\zeta_i\zeta)/8$$

$$N_i^{(\eta)}(\xi,\eta,\zeta) = (1+\xi_i\xi)\eta_i(1+\zeta_i\zeta)/8$$

$$N_i^{(\zeta)}(\xi,\eta,\zeta) = (1+\xi_i\xi)(1+\eta_i\eta)\zeta_i/8$$

The computation of Jacobian matrix and the derivate of the basis function to the original coordinate are shown below.

According to the chain derivation law,

$$\frac{\partial N_i}{\partial \xi} = \frac{\partial N_i}{\partial x}\frac{\partial x}{\partial \xi} + \frac{\partial N_i}{\partial y}\frac{\partial y}{\partial \xi} + \frac{\partial N_i}{\partial z}\frac{\partial z}{\partial \xi}$$

$$\frac{\partial N_i}{\partial \eta} = \frac{\partial N_i}{\partial x}\frac{\partial x}{\partial \eta} + \frac{\partial N_i}{\partial y}\frac{\partial y}{\partial \eta} + \frac{\partial N_i}{\partial z}\frac{\partial z}{\partial \eta}$$

$$\frac{\partial N_i}{\partial \zeta} = \frac{\partial N_i}{\partial x}\frac{\partial x}{\partial \zeta} + \frac{\partial N_i}{\partial y}\frac{\partial y}{\partial \zeta} + \frac{\partial N_i}{\partial z}\frac{\partial z}{\partial \zeta}$$

Therefore, we can write:

$$\begin{bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \\ \frac{\partial N_i}{\partial \zeta} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{bmatrix} \begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial z} \end{bmatrix} = \boldsymbol{J} \begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial z} \end{bmatrix} \tag{2.55}$$

From the transformation defined in (2.53), we obtain

$$\frac{\partial x}{\partial \xi} = \sum_{i=1}^{8} x_i \frac{\partial N_i}{\partial \xi} \qquad \frac{\partial y}{\partial \xi} = \sum_{i=1}^{8} y_i \frac{\partial N_i}{\partial \xi} \qquad \frac{\partial z}{\partial \xi} = \sum_{i=1}^{8} z_i \frac{\partial N_i}{\partial \xi}$$

$$\frac{\partial x}{\partial \eta} = \sum_{i=1}^{8} x_i \frac{\partial N_i}{\partial \eta} \qquad \frac{\partial y}{\partial \eta} = \sum_{i=1}^{8} y_i \frac{\partial N_i}{\partial \eta} \qquad \frac{\partial z}{\partial \eta} = \sum_{i=1}^{8} z_i \frac{\partial N_i}{\partial \eta}$$

$$\frac{\partial x}{\partial \zeta} = \sum_{i=1}^{8} x_i \frac{\partial N_i}{\partial \zeta} \qquad \frac{\partial y}{\partial \zeta} = \sum_{i=1}^{8} y_i \frac{\partial N_i}{\partial \zeta} \qquad \frac{\partial z}{\partial \zeta} = \sum_{i=1}^{8} z_i \frac{\partial N_i}{\partial \zeta}$$

Therefore, the Jacobian matrix ***J*** defined as:

$$J = \begin{pmatrix} x_\xi & y_\xi & z_\xi \\ x_\eta & y_\eta & z_\eta \\ x_\zeta & y_\zeta & z_\zeta \end{pmatrix} \tag{2.56}$$

can be obtained by the coordinate of the nodes and the derivate of the basis functions to reference coordinate. Thus, the inverse of Jacobian matrix $\boldsymbol{J}^1$ can be obtained through ***J***.

$$J^{-1} = \begin{pmatrix} \xi_x & \eta_x & \zeta_x \\ \xi_y & \eta_y & \zeta_y \\ \xi_z & \eta_z & \zeta_z \end{pmatrix} \tag{2.57}$$

It is noted that the Jacobian matrix and the inverse of Jacobian matrix in defined on the Guass integration point.

After the inverse of Jacobian matrix is obtain, the derivate of basis function to original coordinates can be obtained by

$$\begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial z} \end{bmatrix} = \boldsymbol{J}^{-1} \begin{bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \\ \frac{\partial N_i}{\partial \zeta} \end{bmatrix} \tag{2.58}$$

For tetrahedral element as shown in Fig. 2.2, the linear basis function can be expressed as the function of original coordinate directly.

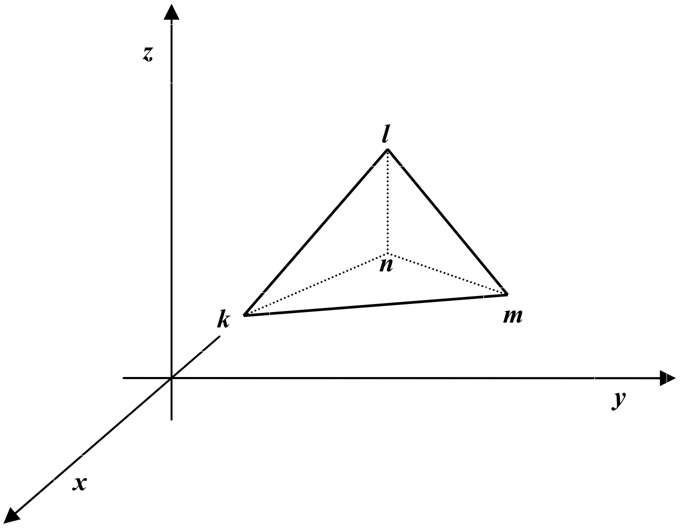


Fig. 2.2 Tetrahedral element

The basis functions on tetrahedral element are:

$$N_i = \frac{1}{6V}\left(p_i + q_i x + r_i y + s_i z\right) \quad i = k, m, n, l \tag{2.59}$$

Where V is the volume of the tetrahedral element,

$$p_k = \begin{vmatrix} x_m & y_m & z_m \\ x_n & y_n & z_n \\ x_l & y_l & z_l \end{vmatrix} \quad p_m = -\begin{vmatrix} x_k & y_k & z_k \\ x_n & y_n & z_n \\ x_l & y_l & z_l \end{vmatrix} \quad p_n = \begin{vmatrix} x_k & y_k & z_k \\ x_m & y_m & z_m \\ x_l & y_l & z_l \end{vmatrix} \quad p_l = -\begin{vmatrix} x_k & y_k & z_k \\ x_m & y_m & z_m \\ x_n & y_n & z_n \end{vmatrix}$$

$$q_k = -\begin{vmatrix} 1 & y_m & z_m \\ 1 & y_n & z_n \\ 1 & y_l & z_l \end{vmatrix} \quad q_m = \begin{vmatrix} 1 & y_k & z_k \\ 1 & y_n & z_n \\ 1 & y_l & z_l \end{vmatrix} \quad q_n = -\begin{vmatrix} 1 & y_k & z_k \\ 1 & y_m & z_m \\ 1 & y_l & z_l \end{vmatrix} \quad q_l = \begin{vmatrix} 1 & y_k & z_k \\ 1 & y_m & z_m \\ 1 & y_n & z_n \end{vmatrix}$$

$$r_k = \begin{vmatrix} x_m & 1 & z_m \\ x_n & 1 & z_n \\ x_l & 1 & z_l \end{vmatrix} \quad r_m = -\begin{vmatrix} x_k & 1 & z_k \\ x_n & 1 & z_n \\ x_l & 1 & z_l \end{vmatrix} \quad r_n = \begin{vmatrix} x_k & 1 & z_k \\ x_m & 1 & z_m \\ x_l & 1 & z_l \end{vmatrix} \quad r_l = -\begin{vmatrix} x_k & 1 & z_k \\ x_m & 1 & z_m \\ x_n & 1 & z_n \end{vmatrix}$$

$$s_k = -\begin{vmatrix} x_m & y_m & 1 \\ x_n & y_n & 1 \\ x_l & y_l & 1 \end{vmatrix} \quad s_m = \begin{vmatrix} x_k & y_k & 1 \\ x_n & y_n & 1 \\ x_l & y_l & 1 \end{vmatrix} \quad s_n = -\begin{vmatrix} x_k & y_k & 1 \\ x_m & y_m & 1 \\ x_l & y_l & 1 \end{vmatrix} \quad s_l = \begin{vmatrix} x_k & y_k & 1 \\ x_m & y_m & 1 \\ x_n & y_n & 1 \end{vmatrix}$$

2.3.3. EDGE ELEMENT

The advantage of edge element in electromagnetic is analyzed in [5]. When using nodal element method, not only the tangential component but also the normal component of the vector potential ***A*** are assumed to be continuous on all the surfaces. However, the continuous of normal component of the vector potential ***A*** on any surfaces does not follow any physical consideration. On the iron-air interface, the tangential component of flux density can be written as

$$\boldsymbol{B}_{t1} = \frac{\partial \boldsymbol{A}_n}{\partial t_2} - \frac{\partial \boldsymbol{A}_{t2}}{\partial n} \tag{2.60}$$

where t_1 and t_2 are two tangential coordinates and n stands for the normal direction. if $\boldsymbol{A}_n$ is continuous on the iron-air interface, the first term $\frac{\partial \boldsymbol{A}_n}{\partial t_2}$ is continuous and only the second term $\frac{\partial \boldsymbol{A}_{t2}}{\partial n}$ can represent the necessary jump in B_{t1}. Since ***A*** is unique, it may turns out that in many case, the continuous term is much larger than the discontinuous one and thus the approximation of the discontinuity is numerically difficult.

If the continuity of A_n is not imposed, the jump in the tangential component of the flux density can be easy to satisfy.

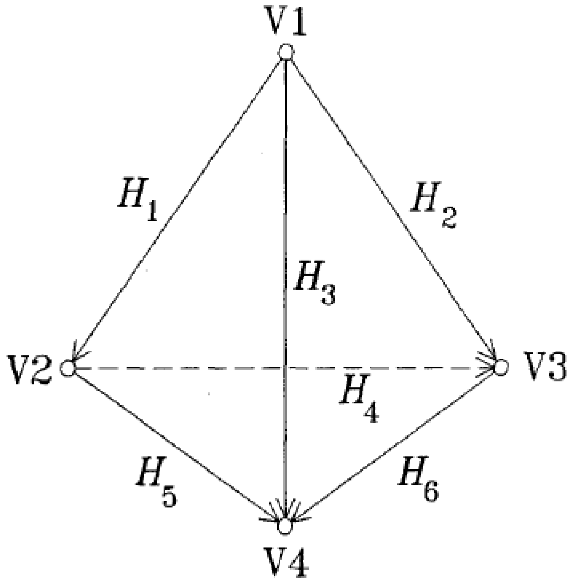


Fig. 2.3 The order of edge in tetrahedral element

The basis functions of edge element can be derived from nodal element basis functions (2.59).

$$\boldsymbol{N}_{ij} = (N_i \nabla N_j - N_j \nabla N_i) l_{ij} \tag{2.61}$$

where $\boldsymbol{N}_{ij}$ is the basis function on edge *i-j* and $\boldsymbol{N}_i$ is the basis function of i^{th} node; l_{ij} is the length of edge *i-j*.

The feature of the edge element is analyzed below [9].

Firstly, it is easy to see that the basis function $\boldsymbol{N}_{ij}$ is divergence free $\nabla \cdot \left(\nabla N_i\right) = 0$ because N_i is linear interpolation basis functions.

$$\frac{\nabla \cdot \boldsymbol{N}_{ij}}{l_{ij}} = N_i \nabla \cdot \left(\nabla N_j\right) + \nabla N_i \nabla N_j - N_j \nabla \cdot \left(\nabla N_i\right) - \nabla N_j \nabla N_i = 0$$

Secondly, $\boldsymbol{N}_{ij}$ has a constant tangential component along edge *i-j* and at the same time has no tangential component along the other edges. Let $\boldsymbol{e}$ denotes the unit vector pointing from node *i* to node *j*. Since N_i is a linear function that varies from 1 at node *i* to zero at node *j* and N_j is a linear function that varies from 1 at node *j* to zero at node *i*, $\boldsymbol{e}\cdot\nabla N_i = -\frac{1}{l_{ij}}$ and $\boldsymbol{e}\cdot\nabla N_j = \frac{1}{l_{ij}}$. Therefore,

$$\boldsymbol{e}\cdot\boldsymbol{N}_{ij} = \boldsymbol{e}\cdot(N_i\nabla N_j - N_j\nabla N_i)l_{ij} = N_i + N_j = 1 \tag{2.62}$$

Further, from the definition of N_i and N_j, it is easy to see that N_{ij} has no tangential component along the other edges.

From the definition $N_i = \frac{1}{6V_e}\left(p_i + q_i x + r_i y + s_i z\right), i = 1,2,3,4$, the basis function of edge element can be expressed as:

$$\frac{\boldsymbol{N}_{12}}{l_{12}} = N_1\nabla N_2 - N_2\nabla N_1 = N_1\frac{1}{6V_e}\begin{pmatrix} q_2 \\ r_2 \\ s_2 \end{pmatrix} - N_2\frac{1}{6V_e}\begin{pmatrix} q_1 \\ r_1 \\ s_1 \end{pmatrix}$$

$$\frac{\boldsymbol{N}_{13}}{l_{13}} = N_1\nabla N_3 - N_3\nabla N_1 = N_1\frac{1}{6V_e}\begin{pmatrix} q_3 \\ r_3 \\ s_3 \end{pmatrix} - N_3\frac{1}{6V_e}\begin{pmatrix} q_1 \\ r_1 \\ s_1 \end{pmatrix}$$

$$\frac{\boldsymbol{N}_{14}}{l_{14}} = N_1\nabla N_4 - N_4\nabla N_1 = N_1\frac{1}{6V_e}\begin{pmatrix} q_4 \\ r_4 \\ s_4 \end{pmatrix} - N_4\frac{1}{6V_e}\begin{pmatrix} q_1 \\ r_1 \\ s_1 \end{pmatrix}$$

$$\frac{\boldsymbol{N}_{23}}{l_{23}} = N_2\nabla N_3 - N_3\nabla N_2 = N_2\frac{1}{6V_e}\begin{pmatrix} q_3 \\ r_3 \\ s_3 \end{pmatrix} - N_3\frac{1}{6V_e}\begin{pmatrix} q_2 \\ r_2 \\ s_2 \end{pmatrix}$$

$$\frac{\boldsymbol{N}_{24}}{l_{24}} = N_2\nabla N_4 - N_4\nabla N_2 = N_2\frac{1}{6V_e}\begin{pmatrix} q_4 \\ r_4 \\ s_4 \end{pmatrix} - N_4\frac{1}{6V_e}\begin{pmatrix} q_2 \\ r_2 \\ s_2 \end{pmatrix}$$

$$\frac{\boldsymbol{N}_{34}}{l_{34}} = N_3\nabla N_4 - N_4\nabla N_3 = N_3\frac{1}{6V_e}\begin{pmatrix} q_4 \\ r_4 \\ s_4 \end{pmatrix} - N_4\frac{1}{6V_e}\begin{pmatrix} q_3 \\ r_3 \\ s_3 \end{pmatrix}$$

According to the weak form of eddy current equation (2.63)

$$\begin{cases} \sigma\dfrac{1}{\delta t}\left(A,\delta A\right) + \nu(\nabla\times A,\nabla\times\delta A) + \sigma(\nabla\varphi,\delta A) = \sigma\dfrac{1}{\delta t}\left(A^{n-1},\delta A\right) \\ \sigma(A,\nabla\delta\varphi) + \delta t\sigma(\nabla\varphi,\nabla\delta\varphi) = \sigma(A^{n-1},\nabla\delta\varphi) \end{cases} \tag{2.63}$$

The element matrix can be obtained from (2.64),

$$\begin{aligned} (A,\delta A)_{MN} &= \int_{V^e} N_{ij}N_{kl} = l_{ij}l_{kl}\int_{V^e}\left(N_i\nabla N_j - N_j\nabla N_i\right)\left(N_k\nabla N_l - N_l\nabla N_k\right)dv \\ &= l_{ij}l_{kl}\int_{\Delta}\left(N_iN_k\nabla N_j\nabla N_l - N_iN_l\nabla N_j\nabla N_k\right)dv \\ &+ l_{ij}l_{kl}\int_{\Delta}\left(-N_jN_k\nabla N_i\nabla N_l + N_jN_l\nabla N_i\nabla N_k\right)dv \\ &= l_{ij}l_{kl}\left(\frac{q_jq_l + r_jr_l + s_js_l}{36V_e^2}\frac{\delta_{ik}}{20}V_e - \frac{q_jq_k + r_jr_k + s_js_k}{36V_e^2}\frac{\delta_{il}}{20}V_e\right) \\ &+ l_{ij}l_{kl}\left(-\frac{q_iq_l + r_ir_l + s_is_l}{36V_e^2}\frac{\delta_{jk}}{20}V_e + \frac{q_iq_k + r_ir_k + s_is_k}{36V_e^2}\frac{\delta_{jl}}{20}V_e\right) \end{aligned} \tag{2.64}$$

According to the following equations,

$$\int_{V^e} L_1^{m1}L_2^{m2}L_3^{m3}L_4^{m4}dV = \frac{m1!m2!m3!m4!}{(m1+m2+m3+m4+3)}6V^e$$

$$\nabla\times N_{ij} = \nabla\times(N_i\nabla N_j - N_j\nabla N_i)l_{ij} = 2\nabla N_i\times\nabla N_j l_{ij}$$

$$= 2\frac{1}{36V_e^2}\begin{pmatrix} q_i \\ r_i \\ s_i \end{pmatrix}\times\begin{pmatrix} q_j \\ r_j \\ s_j \end{pmatrix}l_{ij} = \frac{1}{18V_e^2}l_{ij}\begin{pmatrix} r_is_j - s_ir_j \\ s_iq_j - q_is_j \\ q_ir_j - r_iq_j \end{pmatrix}$$

The integration of shape function of edge element and that of nodal element can be written as (2.65):

$$
\begin{aligned}
(\nabla\varphi,\delta A)_{1j} &= l_{12}\int_{V^e}\nabla N_j\cdot N_{12} = l_{12}\int_{V^e} N_1\nabla N_j\cdot\nabla N_2 - N_2\nabla N_j\cdot\nabla N_1 \\
&= l_{12}\left(\frac{q_jq_2+r_jr_2+s_js_2}{36V_e^{\,2}}\int_{V^e}N_1-\frac{q_jq_1+r_jr_1+s_js_1}{36V_e^{\,2}}\int_{V^e}N_2\right) \\
&= l_{12}\,\frac{q_jq_2+r_jr_2+s_js_2-\left(q_jq_1+r_jr_1+s_js_1\right)}{144V_e}
\end{aligned}
\tag{2.65}
$$

By this way, the stiff matrix on each element can be calculated out.

3. TECHNIQUES FOR FEM APPLICATION

Several techniques for FEM application in electromagnetic are proposed in the chapter. Electromagnetic applications have its own features and need special techniques when FEM is applied to. Slave master method is usually applied to treat with the problems with rotation part [10]. Circuit coupling with field computation is always used to simulate the whole performance of the electric devices [11]. Parameter extraction is the method used to extract the mass circuit parameter of the devices [12]. Such technique was used to 2D FEM computation in reference. In the thesis, slave master technique is extended to 3D FEM computation. Circuit coupling and parameter extraction method are used in coupled 3D FEM and optimization method. In last part of the chapter, the state of art programming technique is proposed to illustrate the implement FEM by object oriented programming method.

3.1. TECHNIQUES FOR MOVING OBJECT

3.1.1. SLAVE MASTER TECHNIQUE

For problem with moving object on computation domain, in traditional FEM the moving part are studied in Euler coordinate which lets the coordinates of the moving part remain stable and at the same time will result in an additional convection term $(v\cdot\nabla)\boldsymbol{A}$ in the equation. The convection term $(v\cdot\nabla)\boldsymbol{A}$ will bring with numeric oscillation when the speed v is large and so it should be avoid bringing into the Maxwell

Equations. Another method is to study the moving part in Lagrange coordinate and the coordinates of the moving part will change according to its speed. Using such method, the mesh on the interface of moving part and static part always will not conform. Slave master method is an efficient method for dealing with the non-conform mesh. In the thesis, slave master method is extended to 3D FEM computation for solving axial flux PM motor.

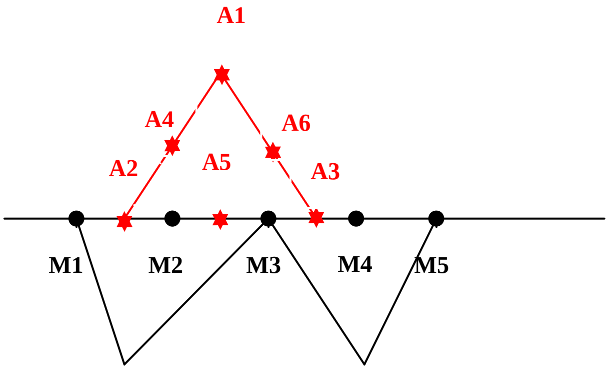


Fig. 3.1 Slave master nodes on the interface

The slave master method is illustrated by 2D example by triangular element as shown in Fig. 3.1. The nodes in the red triangular and the black triangular are not accord with the requirement of traditional FEM. Therefore, slave master method is applied. The nodes A_2, A_5, A_3 marked with star in the red triangular are defined as slave and nodes M_1, M_2, M_3, M_4, M_5 marked with dot in the black triangular are defined as master. The value A on the slave node can be expressed by the value on the master nodes as:

$$A_2 = \lambda_{21}M_1 + \lambda_{22}M_2 + \lambda_{23}M_3$$

$$A_5 = \lambda_{51}M_1 + \lambda_{52}M_2 + \lambda_{53}M_3$$

$$A_3 = \lambda_{31}M_3 + \lambda_{32}M_4 + \lambda_{33}M_5$$

There is a two order interpolation function M on line M_1M_3. The interpolation function can be expressed as $M = M_1N_1+M_2N_2+M_3N_3$, where N_1, N_2, N_3 are defined as: $N_i(M_i) = 1$ and $N_i(M_j) = 0$ when $i \neq j$. Therefore, λ_{ij} can be obtained by the coordinates of node *i* and the interpolation function *M*. The freedom in the element becomes $A'=(A_1, M_1, M_2, M_3, A_4, M_4, M_5,A_6)^T$ and the transform between the original freedom $A= (A_1, A_2, A_3, A_4, A_5,A_6)^T$ and the new freedom is:

$$\begin{bmatrix} A_1 \\ A_2 \\ A_3 \\ A_4 \\ A_5 \\ A_6 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \lambda_{21} & \lambda_{22} & \lambda_{23} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda_{31} & 0 & \lambda_{32} & \lambda_{33} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & \lambda_{51} & \lambda_{52} & \lambda_{53} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} A_1 \\ M_1 \\ M_2 \\ M_3 \\ A_4 \\ M_4 \\ M_5 \\ A_6 \end{bmatrix} \tag{3.1}$$

We abbreviate (3.1) as

$$A = T A' \tag{3.2}$$

When calculating the stiff matrix on element which contains slave nodes, we get the original form of stiff matrix and the load:

$$\begin{bmatrix} a_{11} & & & & & a16 \\ \cdots & a_{22} & & & & \\ \cdots & & a_{33} & & & \\ \cdots & & & a_{44} & & \\ \cdots & & & & a_{55} & \\ a61 & & & & & a_{66} \end{bmatrix} \begin{bmatrix} A_1 \\ A_2 \\ A_3 \\ A_4 \\ A_5 \\ A_6 \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ F_6 \end{bmatrix} \tag{3.3}$$

We abbreviate (3.3) as

$$S A = F \tag{3.4}$$

Substitute (3.2) to (3.4),

$$T^T S T A'=T^TF \tag{3.5}$$

The new stiff matrix and new load on the element becomes to be $T^T S T$ and T^TF respectively. Then the stiff matrix and load can be added on the total stiff matrix automatically.

3.1.2. FEM AND FVM COUPLED TECHNIQUE

For problem with moving part on computation domain, in traditional FEM such moving part are studied in Euler coordinate which lets the coordinates of the moving part remain stable and at the same time will result in an additional convection term $(v \cdot \nabla)\boldsymbol{A}$ in the equation. For eddy current problems, an additional induction electrical field $\boldsymbol{E}_v = \boldsymbol{v} \times \boldsymbol{B}$ will arise. The eddy current analysis in electromagnetic devices with high speed moving conductors therefore becomes a convection-diffusion problem with high Peclet number [13]. The governing equations of ***A***-*φ* method for such problem are:

$$\nabla \times \frac{1}{\mu} \nabla \times \boldsymbol{A} - \sigma[\boldsymbol{v} \times \nabla \times \boldsymbol{A} - j\omega \boldsymbol{A} - \nabla \varphi] = \boldsymbol{J}_s \tag{3.6}$$

$$\nabla \cdot \sigma[\boldsymbol{v} \times \nabla \times \boldsymbol{A} - j\omega \boldsymbol{A} - \nabla \varphi] = 0 \tag{3.7}$$

where; ***A*** and *φ* are magnetic vector potential and electrical scalar potential, respectively; *μ* and *σ* are permeability and conductivity, respectively; ***v*** is the velocity of the media relative to the motion.

Equation (3.6) is a typical convection-diffusion problem. Because the solutions of (3.6) - (3.7) are exponential functions, spurious oscillations will arise when dealing with the convection process using linear or polynomial basis function in standard FEM [14]. Finite difference method (FDM) with upwind scheme is usually used to deal with such convection-diffusion problems [15]. However, it is difficult to use FDM to deal with geometry models with complex boundaries and arbitrary shape triangles or tetrahedron elements. Streamline upwind Petrov–Galerkin FEM (SUPG FEM) is proposed to overcome the oscillations due to the convection term [16]. However, the optimal stabilization parameters in SUPG method are difficult to choose. A finite analytic element method (FAEM) has therefore been proposed to analyze this problem [13]. Nonetheless, FAEM is too complex for implementation in practical applications because the method requires finding of the exact solutions of the characteristic equation [17].

FVM with upwind scheme is used to solving convection dominated equations. FVM possess the merits of having eliminated the numerical oscillations in the solutions. Diffusion problem leads to second order differential equation and it may contain complex boundary conditions. However, it is difficult to deal with the complex boundary condition using FVM. Moreover, FEM is more widely used to solving diffusion problems compared with FVM [18].

Convection-diffusion problems are viewed as coupling of two different physical processes: convection and diffusion. The two physical processes result in different characteristic stiff matrixes mathematically and therefore need to be discretized by a suitable method for each other. FVM is well suited for dealing with convection dominant problems having oscillations in their numerical solutions, whereas FEM is widely used to solve diffusion problems on complex geometry shapes. In this paper, a conformal discretization scheme for FEM and FVM on the same mesh is proposed to allow the designer to combine the two methods when solving convection-diffusion problems.

For linear conductive medium, with Lorentz gauge$\nabla \cdot \boldsymbol{A} = -\mu\sigma\varphi$, the equivalent form of (3.6) - (3.7) can be written as follows:

$$\nabla\left(\frac{1}{\mu}\nabla \cdot \boldsymbol{A}\right) + \sigma(\boldsymbol{v} \times \nabla \times \boldsymbol{A} - j\omega\boldsymbol{A}) = -\boldsymbol{J}_s \tag{3.8}$$

$$\nabla^2\varphi - j\omega\mu\sigma\varphi = 0 \tag{3.9}$$

For simplification, the proposed coupled FEM and FVM algorithms will be derived in two-dimension (2D). Equation (3.8) in 2D becomes:

$$\nabla\left(\frac{1}{\mu}\nabla \cdot \boldsymbol{A}\right) + \sigma(\boldsymbol{v} \cdot \nabla A) + k^2 A = 0 \tag{3.10}$$

where k^2 is defined as k^2=-$j\omega\mu\sigma$.

Equation (3.10) includes diffusion terms $\nabla\left(\frac{1}{\mu}\nabla \cdot \boldsymbol{A}\right)$, k^2A and convection term $\sigma(\boldsymbol{v} \cdot \nabla A)$. The diffusion terms with the boundary conditions are discretized by standard

FEM with linear basis functions (or second order functions) and the convection term $\sigma(\boldsymbol{v} \cdot \nabla A)$ is discretized by one order upwind scheme FVM (or two order upwind schemes FVM). The detail discretization of upwind FVM is illustrated below.

On the FEM mesh (triangle element or quadrilateral element) in the computation domain, the dual mesh is defined virtually for the FVM discretization analysis. Taking the triangle for example, as shown in Fig. 1, for each node P_i in FEM mesh, there is a set of affected elements which contain it. The circumcenter M_i (or orthocenter) of all the affected elements can be found and all the circumcenter of the affected elements are then linked. The included area (the area surrounded by the dotted line in Fig. 3.2 is called the control volume of P_i. The control volumes of the entire vertexes in FEM mesh are called the dual mesh of FVM which covers the whole computation domain.

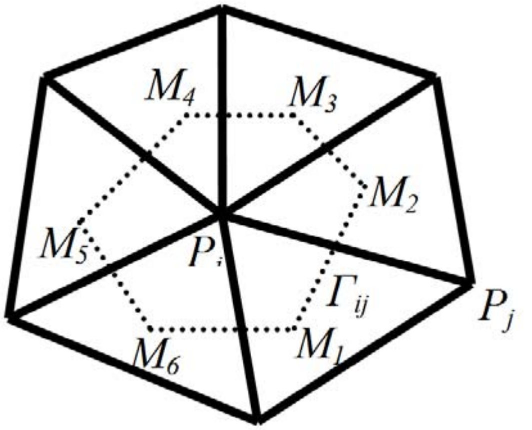


Fig. 3.2 The control volume of vertex P_i.

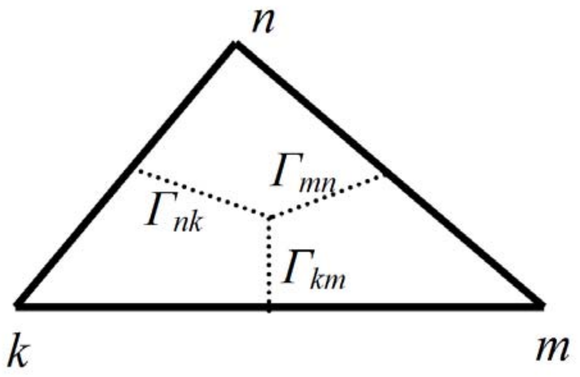


Fig. 3.3 Triangle vertex *K, M, N*.

The discretization of the convection term $(\boldsymbol{v}\cdot\nabla A)$ is.

$$\int_{\Omega}(\boldsymbol{v}\cdot\nabla A)dxdy=\sum_{i=1}^{N}\int_{\Omega i}(\boldsymbol{v}\cdot\nabla)Adxdy \tag{3.11}$$

where *N* is the number of the control volume and Ω_i is the i-th control volume.

$$\int_{\Omega i}(\boldsymbol{v}\cdot\nabla)Adxdy=\int_{\Omega i}\nabla\cdot(\boldsymbol{v}A)dxdy-\int_{\Omega i}A\nabla\cdot\boldsymbol{v}dxdy \tag{3.12}$$

The first part of the right hand side of (3.12):

$$\int_{\Omega i}\nabla\cdot(\boldsymbol{v}A)dxdy=\oint_{\partial\Omega i}(\boldsymbol{v}\cdot\boldsymbol{n})Ads=\sum_{j=1}^{M}\int_{\overline{MjMj+1}}(\boldsymbol{v}\cdot\boldsymbol{n})Ads \tag{3.13}$$

Using first order upwind scheme to represent of *A* on line M_jM_{j+1},

$$A_{\overline{MjMj+1}}=r_{ij}A_i+(1-r_{ij})A_j \tag{3.14}$$

where r_{ij} is the upwind parameter which is defined by Samarskii scheme:

$$r_{ij}=1-\frac{1}{z_{ij}}\left(1-\frac{z_{ij}}{e^{z_{ij}}-1}\right)$$

where $Z_{ij}=\int_{\Gamma ij}(\boldsymbol{v}\cdot\boldsymbol{n})ds$, which means the flux on the edge Γ_{ij}.

The second part of the right hand side of (3.12) is:

$$\int_{\Omega i} A\nabla\cdot\boldsymbol{v}dxdy = A_i\int_{\Omega i}\nabla\cdot\boldsymbol{v}dxdy = A_i\oint_{\partial\Omega i}\boldsymbol{v}\cdot\boldsymbol{n}ds = A_i\sum_{j=1}^{M}\int_{\overline{MjMj+1}}(\boldsymbol{v}\cdot\boldsymbol{n})ds \tag{3.15}$$

Equation (3.12) can be written as:

$$\begin{aligned}\int_{\Omega i}\boldsymbol{v}\cdot\nabla Adxdy &= \sum_{j}^{M}\left(r_{ij}A_i+(1-r_{ij})A_j\right)\int_{\overline{MjMj+1}}(\boldsymbol{v}\cdot\boldsymbol{n})ds - A_i\sum_{j=1}^{M}\int_{\overline{MjMj+1}}(\boldsymbol{v}\cdot\boldsymbol{n})ds\\ &= \sum_{j}^{M}(1-r_{ij})(A_j-A_i)\int_{\overline{MjMj+1}}(\boldsymbol{v}\cdot\boldsymbol{n})ds\end{aligned} \tag{3.16}$$

In each triangle on the finite element mesh based on (3.16) the FVM discretization matrix of convection term $(\boldsymbol{v}\cdot\nabla A)$ is:

$$(1\text{-}r_{km})(A_m\text{-}A_k)FLUX_{km} + (1\text{-}r_{kn})(A_n\text{-}A_k)FLUX_{kn} \tag{3.17}$$

$$(1\text{-}r_{nk})(A_k\text{-}A_n)FLUX_{nk} + (1\text{-}r_{nm})(A_m\text{-}A_n)FLUX_{nm} \tag{3.18}$$

$$(1\text{-}r_{mk})(A_k\text{-}A_m)FLUX_{mk}+(1\text{-}r_{mn})(A_n\text{-}A_m)FLUX_{mn} \tag{3.19}$$

where A_i is the unknown variable on node *i*, $FLUX_{ij}$ is the flux $Z_{ij}=\int_{\Gamma ij}(\boldsymbol{v}\cdot\boldsymbol{n})ds$ on edge Γ_{ij}.

Equations (3.17), (3.18) and (3.19) are the discretization of the convection term $\sigma(\boldsymbol{v}\cdot\nabla A)$ of FVM on nodes *k, m* and *n,* respectively, as shown in Fig. 3.3, which have the same form of stiff matrices of the FEM on the element. Therefore, the discretization form of FVM can be added to the stiff matrix of the FEM stiff matrix directly to form a total stiff matrix.

The FEM and FVM coupled method is firstly verified by a classical one-dimensional convection-diffusion equation:

$$\begin{cases} u''-2Pu'+k^2u=0 & in\,(0,1)\\ u\,|_{x=0}=0, u\,|_{x=1}=1\end{cases} \tag{13}$$

where; $P = v\sigma\mu/2$; $P_e = Ph = v\sigma\mu h/2$ is the Peclet number; *h* is nodes scale; σ and μ are conductivity and permeability, respectively.

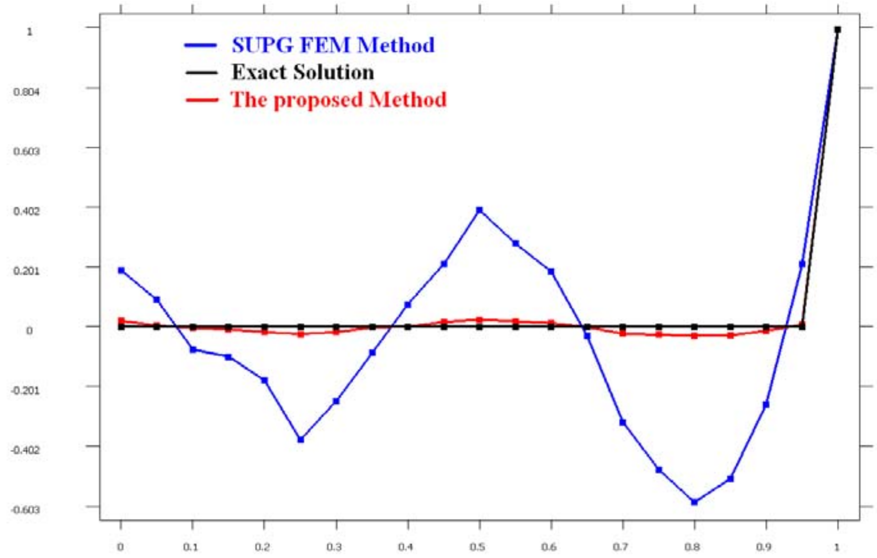


Fig. 3.4 Comparison of the SUPG FEM and the proposed method.

There are 21 nodes distributed in the solving domain: 19 nodes for interior and 2 nodes for boundary. Peclet number $P_e=3$. The numerical solutions of SUPG FEM and the proposed algorithm are compared with the analytical solution in Fig. 3. It can be seen that the proposed algorithm obtains essentially the analytical solution without suffering the oscillations encountered using SUPG FEM.

3.1.3. MLS Approximation Based Interface Element

Modeling the sliding-surface between stator and rotor in motor on which the mesh is not conforming has been studied for decades. Many researches are focused on it and many techniques have been proposed: the moving band technique, the locked-step approach, the use of Lagrange multipliers, the nodal interpolation method [19], the mortar-element approach, to name but a few. Each of this method presents advantages and drawbacks and many adaptations have been proposed to improve them.

Recently, a moving least-square (MLS) approximation based interface element method (IEM) is introduced to deal with non-conformed meshes [20]. IEM shows distinguish advantage on other methods. At the non-conform mesh interface, the continuity is satisfied automatically while Lagrange multipliers based methods guarantee it only in variation sense. Secondly, the construction of new shape functions on element with variation nodes allow IEM to use the frame of conventional finite element method (FEM), therefore the stiff matrix remains symmetric and positive defined. In this section, new IEM shape functions on triangular element are constructed and applied to deal with the sliding-surface in modeling the rotation of electric machines. Shape function of variable-node elements will be introduced in the following.

Fig. 3.5 shows that when the rotor rotates, the nodes between the rotor and stator will always not coincide with each other.

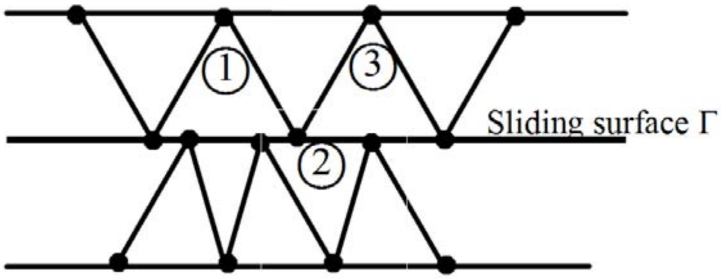


Fig. 3.5 The sliding surface between stator and rotor.

The nodes on the sliding interface Γ are not conforming and so the shape functions of general FEM become incompatible on the sliding interface. New shape functions need to be constructed to ensure continuity on the sliding surface. The procedural of construction of the new shape functions is as follows.

Firstly, map the triangular element to the standard element in (ξ, η) plane according to the vertex of the element, as shown in Fig. 3.6.

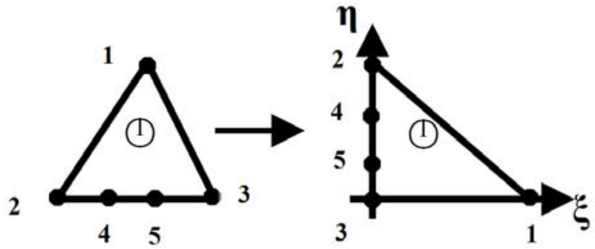


Fig. 3.6 Mapping triangle element with variable nodes to (ξ, η) plane.

On the element with 5-node as shown in Fig. 3.6, the new shape functions are expressed as follows, where N_i denotes the shape function associated with node *i*.

$N_1 = W(\xi, 1, 0)$, $N_2 = W(\eta, 1, \eta_4)$, $N_3 = W(1\text{-}\xi\text{-}\eta, 1, \eta_5)$,

$N_4 = W(\eta, \eta_4, min\,(|\eta_4 - \eta_5|, |\eta_4 - \eta_2|)\,)$,

$N_5 = W(\eta, \eta_5, min\,(|\eta_5|, |\eta_5 - \eta_4|)$,

where,

$$W(x, x_0, x_1) = \begin{cases} (1 - 6\bar{x}^2 - 8\bar{x}^3 - 3\bar{x}^4) & for\ \bar{x} = \left| \dfrac{x\text{-}x_0}{x_1 - x_0} \right| \leq 1 \\ 0 & for\ \bar{x} = \left| \dfrac{x\text{-}x_0}{x_1 - x_0} \right| \geq 1 \end{cases} \tag{3.20}$$

The quadratic spline in (3.20) is the popular weight functions in MLS-based methods. It is easy to check that $N_i(x_i) = 1$ and $N_i(x_j) = 0$ when $j \neq i$ and it will be proved in full paper that the shape functions are compatible on the sliding surface.

If the triangular element is constituted by (m+3) nodes, the uniform expression of shape functions can be constructured in the same way.

3.2. STATE-OF-THE-ART PROGRAMMING TECHNIQUES OF FEM

State-of-the-art programming techniques of implement of FEM by object oriented programming language for electromagnetic field computation are presented in the chapter. It covers program structure, data structure and algebraic matrix equation. The advantages of the proposed program structure are that multi-developers are empowered to work on different solvers and share common algorithms. The beauty of the proposed data structure is that two-dimensional FEM, multi-slice FEM and three-dimensional FEM can share the same data structure. It allows quick access to all data and is efficient for organizing FEM programs and convenient for mesh generation. It can also deal with motion problems readily. The merits of the matrix equation solver are that it can automatically deal with Dirichlet boundary conditions, master-salve boundary conditions and all other constraints in sparse matrix equations. The sub-matrix operation technique allows the electric circuit equations to be coupled easily with electromagnetic field equations.

3.2.1. INTRODUCTION

Finite element method is a proven powerful numerical method for electromagnetic field computation. Most prior art focus either on means to improve the efficiency of FEM to reduce the computation time when dealing with complex nonlinear problems or on new algorithms to reduce the numerical errors of FEM [21][22]. Many of the researches are on procedure-oriented programming technologies. However, these programs are inflexible in dealing with the data transmission among different solvers and since there is a high consistence requirement for data storage, it is inconvenient for several developers to organize, maintain and extend the programs.

FEM is usually utilized to model problems with complicated geometry and different materials, and the basic characteristics of these elements, such as the materials, coordinates, potentials have to be stored in arrays and tables [23]. For the modeling of many different electric devices, different models such as two-dimensional model, multi-slice model or three-dimensional model may be used in FEM [23]. They simulate the same problem at different levels. The multi-slice and 3-D models have higher accuracy but long computing time is needed. It is usually a good practice to use 2-D model to get a rough initial solution, then followed by multi-slice and 3-D models to consider the effects due to skewed slots and end-windings. As all of these models need to access to the same basic data of elements, it is convenient for the developers to adopt object-oriented program to solve the FEM problems [23].

In object-oriented programming, the data structure is the foundation of all algorithms [24][25]. In this chapter, a new data structure based on C++ object-oriented programming technique and a program structure are proposed. An equation solver is also presented. In the program structure, multi-developers can work on different solvers and share common algorithms. 2-D FEM, multi-slice FEM and 3-D FEM can also share the same data structure [26]. Hence it is convenient to switch among these three models when simulating the same problems. It allows quick access to all data and is efficient for organizing FEM programs, convenient for mesh generation and the proposed algorithm can also be used to address motion problems readily. The equation solver can automatically deal with Dirichlet boundary conditions, master-salve boundary conditions and all other constraints in sparse matrix equations. The sub-matrix operation technique allows the electric circuit equations to be easily coupled with electromagnetic field equations.

Many other aforementioned advantages are included in the proposed programming techniques. For example, the geometry has a nested structure so it is easy to deal with holes inside the geometry. The mesh is a property of face and volume objects. The mesh generation starts from the innermost internal object, first on faces, then on volume objects [27]. It is convenient to deal with mechanical motion. Overall, the proposed software offers a convenient platform to allow data to be accessed quickly.

3.2.2. PROGRAMMING STRUCTURES

The FEM program adopts object-oriented technology, and inheritance is one of the main concepts of the proposed software. The base class defines the interface, and the derived one provides implementations that are specific to this derived class. Virtual functions can be used to define a set of interfaces for the base class. It is convenient to share the codes and maintain as well as extend the software efficiently. Template is another technology to express the commonality. With it the programmers can regenerate the entire family of related classes [28]. With object-oriented technology, the FEM program structure allows multi-programmers to work on different solvers and share common algorithms.

The FEM program comprises of pre-processing, the solvers and post-processing modules, and the solvers have the functions of meshing, pre-assembly, assembly, matrix equation solver. In the FEM program, the base class of the solver is named the **FeSolverBase**. All other solvers are derived from it. The sizes of the included data items in the solver, such as **FeData**, **FeVertex**, **FeFace and etc.** are not known beforehand, so it is convenient to organize them in a dynamic data structure and a template class is used here.

In each base class, if necessary, it has the functions of reading data and writing data. Each solver can easily derive its own class from the base classes and the functions of reading data and writing data are always shared by different solvers. Therefore the input and output data formats can be kept unchanged. All solvers follow the same formats of input and output files to allow the FEM data files to be readily transferable.

The derived solvers are **FeSolver2d**, **FeSolverMs, FeSolver3d**, and they are developed by different programmers and can be used to solve the 2-D, multi-slice and 3-D problems conveniently.

3.2.3. DATA STRUCTURES

To develop a FEM program, a proper data structure is of paramount importance. In our program, the data of a FEM project before meshing is all defined in the class **FeData**. The **FeData** has its own methods of reading and writing, so it is easy to pass the data among the pre-processor, the solvers and the post-processor. All data files use standard and structured Extensible Markup Language (XML) so it is convenient for the FEM program to communicate with other software. Fig. 3.7 shows the main data members of **FeData**.

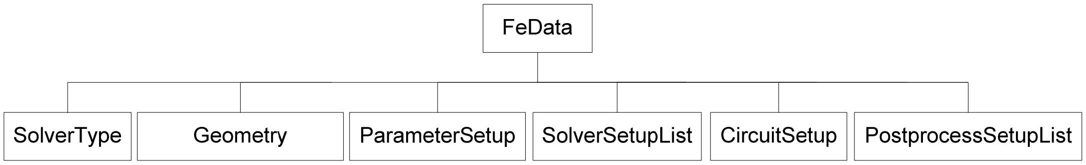


Fig. 3.7 The class **FeData.**

Besides the geometry information, the class **FeData** also contains solver setup information, motor and circuit data which are necessary in electromagnetic computation and post processing. Another advantage of this class is that it integrates the data of 2-D FEM, or multi-slice FEM or 3-D FEM into one class.

The class **Geometry** contains the geometry data as shown in Fig. 3.8. It has a nested structure. The top is the background object. It contains the face list of the external boundaries, the volume list inside the background object, etc. Each volume object in the internal volume list is also a sub-set of the class **Geometry**, so that the background object and all internal volume objects have the same data structure and nested structure.

Each face has a pointer which points to the volume object to which it belongs. So it is quick to finding the face from the volume object, or finding the volume object from the face.

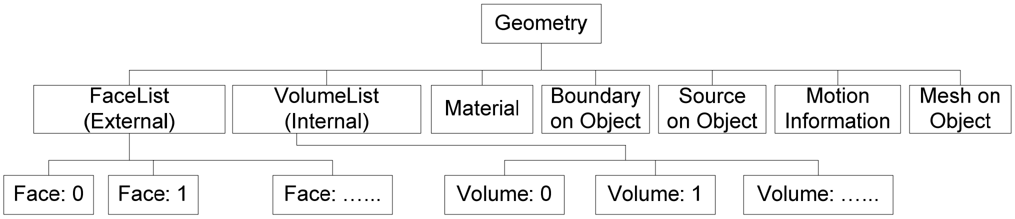


Fig. 3.8 The class **Geometry.**

The class **Face** is shown in Fig. 3.9. It contains an edge list of the external boundaries. For the 2-D FEM and multi-slice FEM, the face is not limited to a triangle, as it can be a straight line, triangle, square or polygon. For each face we define the external edges and internal faces. In the 3-D model the class **Face** may contain a list of internal faces. Each face in the face list is also a subset of the class **Face**, so that all faces have the same data structure.

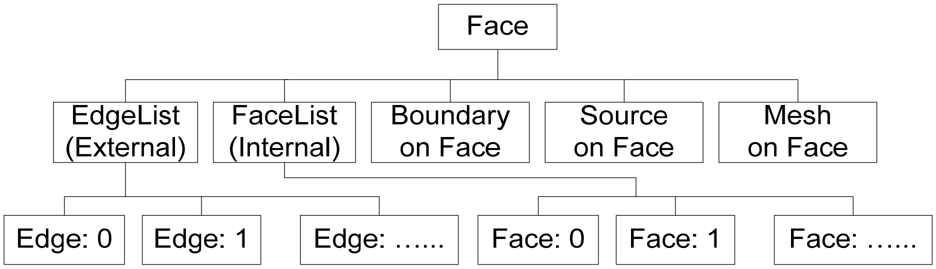


Fig. 3.9 The class **Face.**

The class **Edge** is shown in Fig. 3.10. It contains a vertex list. For the 2-D model the vertex list only has one vertex; for multi-slice model, the vertex list has two vertexes.

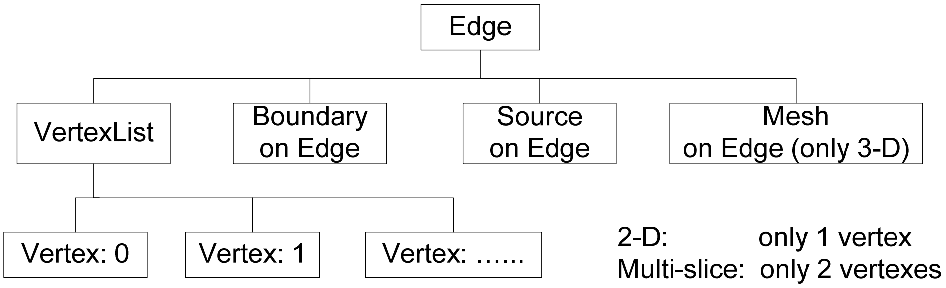


Fig. 3.10 The class **Edge.**

The relationship of the main classes is demonstrated in Fig. 3.11. The solid line means one class's dependence on another class. The dash line means the dependence and access of one class to another. In our design the material class is related with **Volume** and **Face**. Inciting Source is connected with **Volume**, **Face** and **Edge**. **Volume, Face**, **Edge** and **Vertex** all can be set as Boundary conditions.

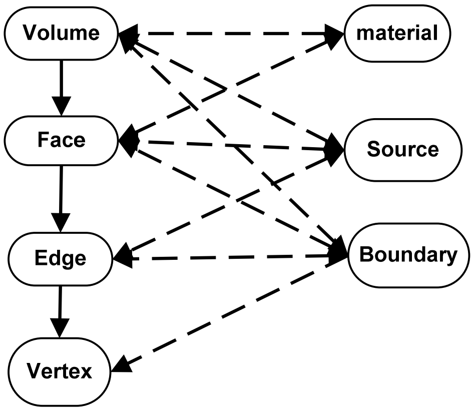


Fig. 3.11 The logic relationship of the data.

Another important class is the **FeMeshData** which is created from **FeData** or imported from mesh files. Class **FeMeshData** is general enough to take into consideration a great number of element types such as triangle or quadrilateral for 2-D programs, and as cubes or tetrahedrons for 3-D programs. It has similar structure to that of **FeData**. But it contains all mesh information which are necessary for equation solving. The mesh generation uses refining quality, optimal meshing methods and considers all possible conditions such as holes, boundaries, multi-domains.

Mesh information usually includes many volume elements, face elements, edges element, and nodes element. Class **MeshVolume**, **MeshFace**, **MeshEdge** and **MeshVertex** are used to descript it coordinately. Each class contains one pointer which point to the geometry volume or face it belongs to. Therefore the material property and boundary condition on mesh can be found from the geometry by such pointers. All the **MeshVolume** in mesh are stored in a vector called **MeshVolume** list and they are the same for class **MeshFace**, **MeshEdge** and **MeshVertex**. The class **FeMesh** contains the **MeshVolume** list, **MeshFace** list, **MeshEdge** list and **MeshVertex** list as its members. The mesh on the computation domain may be divided into several parts according to its different usage, so the mesh on each part is assigned to one instance of **FeMesh**. Several instances of **FeMesh** form one vector of **FeMesh**, which is a member of **FeMeshData. FeMeshData** actually manages all the meshes on computation domain. The structure of class **FeMeshData** is shown in Fig. 3.12.

The structure of **FeMeshData** is convenient for dealing with mesh moving problems, because the meshes on the static part and moving part are separated to form two instances of **FeMesh**. It is also easy to implement for dual-mesh method, because two sets of meshes are constructed to solve the same problem and two sets of meshes are stored in two instances of **FeMesh**.

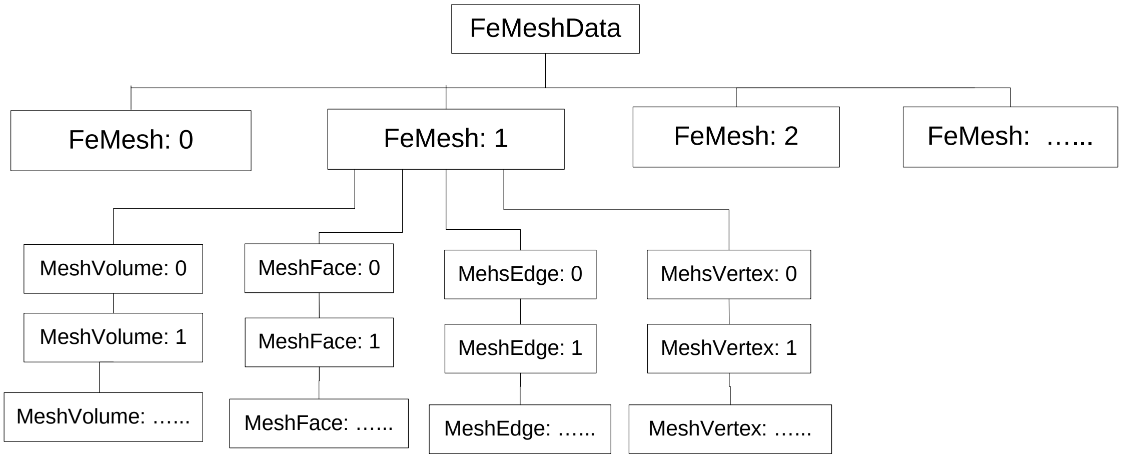


Fig. 3.12 The class **FeMeshData.**

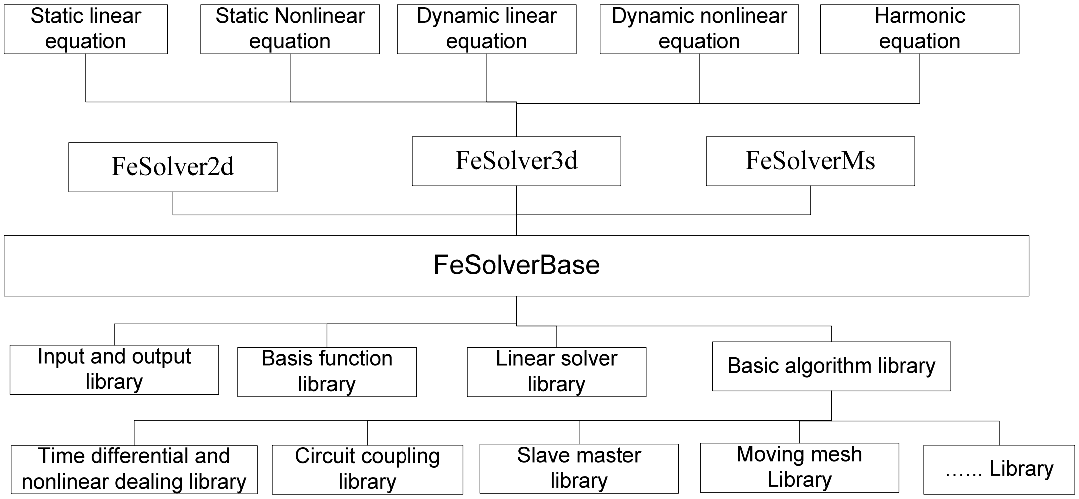


Fig. 3.13 The class **FeSolverBase.**

3.2.4. EQUATION SOLVERS

The top class in the proposed programming techniques is the **FeSolverBase**, which contains **FeData** and **FeMeshData** as data members, as show in Fig. 3.13. The classes of the 2-D **FeSolver2d**, multi-slice **FeSolverMs,** and 3-D solvers **FeSolver3d** are derived from the base-class **FeSolverBase,** so different solvers can be developed independently but they share the same data structure. The **FeSolverBase** contains not only the basic common data for the solver, such as **FeData** and **FeMeshData**, sparse matrix equation and so on. It also contains several common libraries, including the input and output library, basis function library, linear equation solver library and basic algorithm library. The input and output library is mainly used as read data to setup class **FeData** and **FeMeshData** and write out the results after solving the equations. Basis function library includes all the one-order and two-order polynomial basis functions of triangular, quadratic, tetrahedron, prism and hexahedra elements. It also includes the basis functions of edge element method. Basic algorithm library includes common algorithms used in FEM, such as the library of discretization on the time dependent variables,as well as the library to deal with nonlinear equations, circuit-field coupling, slave-master boundary conditions, moving objects and so on. Based on the class **FeSolverBase**, several classes are derived to solve different types of equations, such as static linear equations, static nonlinear equations, dynamic linear equations, dynamic nonlinear equations and harmonic equations.

3.2.5. SPARSE MATRIX

The equation solver can solve symmetric, asymmetric, and complex-valued sparse-matrix equations. It can automatically deal with Dirichlet boundary conditions, master-salve boundary conditions and all other constraints in sparse matrix equations. The sparse matrix storage technique allows the contributions to be added to the matrix at any time. It does not need to pre-determine the positions of non-zero elements. It supports many matrix operations. Sub-matrixes can be added into the main coefficient matrix conveniently which allows that electric circuit equations to be easily coupled with the electromagnetic field equations. The organization of the data structure as as described below.

Firstly, a row data structure in sparse matrix, namely class **MatrixSparseRow**, is defined as a map in standard template library (STL). The first key is the column number of sparse matrix and the datum entry is the type of the element in the matrix, which can be real number or complex number. In this way, only none-zero elements in stiff matrix are stored and their positions are marked. None-zero elements can be added to **MatrixSparseRow** anytime because its length is dynamically allocated. Secondly, the class **MatrixSparse** is defined as a vector of **MatrixSparseRow** and the vector size can also be dynamically allocated.

In FEM, the number of rows in sparse matrix is usually determined by the unknowns while the number of columns is not determined for the sparse characteristic of stiff matrix. The data structure of class **MatrixSparse** is very efficient and effective for storing the sparse FEM stiff matrix.

In traditional FEM method, the number of none zero is fixed according to the freedom on the nodes or edges before the sparse matrix is created, therefore a new sparse matrix should be allocated when dealing with circuit coupling and when new variables need to add into the equation. When using **MatrixSparse**, the stiff matrix can be calculated by FEM first. The new variables of the circuit can be added to the sparse matrix without any modification on its structure because the sparse matrix can be dynamically allocated after the matrix is created.

3.2.6. SUMMARY

The applications of this proposed FEM programming techniques have been reported in many our previous publications [21][22][26][29][30][31][32][33], but this chapter is the first documentation of the methodology in a systematic manner.

This chapter presents the state-of-the-art programming techniques of FEMs for electromagnetic field computation. A new data structure based on C++ object-oriented programming technique and a novel program structure is comprehensively described together with an equation solver. Multi-developers working on different solvers and sharing common algorithms have been proposed. The algorithm allows quick access to all data and is efficient for organizing FEM programs to look into motion problems. It is also convenient for mesh generation and the equation solver can automatically deal with Dirichlet boundary conditions, master-salve boundary conditions and all other constraints in sparse matrix equations. The sub-matrix operation technique allows the electric circuit equations to be easily coupled with electromagnetic field equations.

4. TWO GRID TECHNIQUES FOR FEM COMPUTATION

A two-grid finite element method to reduce the computing time of nonlinear magnetic problems is presented in the chapter. It is shown that the initial solution of the nonlinear iteration on the fine grid can be derived from the solutions of the coarse grid, thereby saving the computation time for iterations on the fine grid. A numerical technique based on the nonlinear functional is proposed to reduce the complexity of the discretization of the nonlinear term. The interpolation method from coarse grid to fine grid is discussed in this chapter. A method to optimize the relaxation factor of Newton-Raphson iteration is also reported. The proposed method is applied to TEAM Workshop problem 13 and the results obtained by using the proposed algorithm are compared with those using conventional methods.

4.1. INTRODUCTION

When nonlinear magnetic materials such as steel are included in the study of electromagnetics, the Maxwell's equations become nonlinear. Convergence property and iteration number are the two main factors when numerical methods are used to solve nonlinear equations. Newton-Raphson method is usually used because of its second order convergence characteristics. However it still needs several iterations before it converges. As a result, when three-dimensional (3D) nonlinear problems with a large number of elements are solved by finite element method (FEM), the nonlinear iterations are time consuming.

Two-grid finite element technique and convergence estimation are proposed in [34] to reduce the computation time of nonlinear problems. In that method, two finite element subspaces υ_H and υ_h (with mesh size $h << H$) are employed for the FEM discretization. On the coarse space υ_H , the standard FEM discretization is used to obtain a rough approximation $u_H \in \upsilon_H$ and a linearized equation based on u_H is solved to produce the corrected solution $u^h \in \upsilon_h$. With the use of that method, solving a nonlinear equation is not much more difficult than solving a linear equation, since $\dim \upsilon_H << \dim \upsilon_h$ and the effort for solving u_H is relatively insignificant.

Considering that Maxwell's equations are fully nonlinear equations, the method presented in this chapter is not to simply solve the linearized equations on the fine grid, instead it is proposed to continue the nonlinear iteration until it becomes convergent. Because iteration on the fine grid as proposed in this chapter ensures the accuracy of the solution is high, the nonlinear iteration on the coarse grid is only required to produce a reasonably good approximation solution. Consequently the space υ_H can be extremely coarse ($H^2 < h$), and the total computing time can be significantly reduced. After several iterations on the coarse grid, the solution on the coarse grid is interpolated to give the solution in the fine grid.

The number of iterations on the coarse grid affects the efficiency of the method. The initial convergence criteria on the coarse grid is set to be the same or slightly higher than that of the fine grid and the speed of the residual reduction is being watched carefully. When it is too slow the iteration on the coarse grid will stop.

Another difference between the proposed method and the method presented in [34] is that the finite element space υ_H does not need to be a subspace of υ_h . This feature allows the independent generation of the coarse and fine meshes. But new difficulties arise when the solution on the coarse mesh is interpolated to produce the solution in the fine mesh. Hence an interpolation technique designed to address this issue will be presented in this chapter.

In the chapter, the proposed two-grid FEM is applied to TEAM Workshop problem 13 and the numerical results obtained using the proposed algorithm are compared with those using conventional methods.

4.2. FORMULATIONS AND NUMERIC SCHEME

Assume the nonlinear magneto-static field equation is governed by,

$$\begin{cases} \nabla \times \boldsymbol{H} = \boldsymbol{J} \\ \nabla \cdot \boldsymbol{B} = 0 \\ \boldsymbol{H} = \nu(\boldsymbol{B})\boldsymbol{B} \end{cases} \tag{4.1}$$

where $\boldsymbol{B}$ is the flux density and $\nu(\boldsymbol{B})$ is the magnetic reluctivity varying with $\boldsymbol{B}$, $\boldsymbol{J}$ is the source current density.

The coupling of magnetic vector potential and scalar potential method is used to solve the equation [35].

The computation domain is divided into two regions: Ω_1 which comprises of nonmagnetic materials such as air and coil with permeability μ_o; Ω_2 which comprises of iron with high permeability μ or low reluctivity v.

In region Ω_1, the magnetic field can be derived from the magnetic scalar potential and is written as:

$$\boldsymbol{H} = \boldsymbol{H}_S - \nabla \phi \tag{4.2}$$

where $\boldsymbol{H}_s$ is the magnetic field due to the coils and it can be computed from Biot-Savart's Law. Hence,

$$\nabla \times \boldsymbol{H}_s = \boldsymbol{J} \tag{4.3}$$

Therefore, the magnetic scalar potential of (4.1) in region Ω_1 satisfies the differential equation:

$$-\nabla \cdot (\mu_0 \nabla \phi) = 0 \tag{4.4}$$

In region Ω_2 of iron where the current density is zero, the magnetic field can be derived from the magnetic vector potential gauged by Coulomb gauge and written as:

$$\boldsymbol{B} = \nabla \times \boldsymbol{A} \tag{4.5}$$

$$\nabla \times (\nu \nabla \times \boldsymbol{A}) - \nabla(\nu \nabla \cdot \boldsymbol{A}) = 0 \tag{4.6}$$

On the interface of the region Ω_1 and Ω_2, Γ_{12}, the continuity of the normal component of the flux density as well as the tangential component of the magnetic field intensity have to be satisfied:

$$\nu \nabla \times \boldsymbol{A} \times \boldsymbol{n} + \boldsymbol{H}_s \times \boldsymbol{n} - \nabla \phi \times \boldsymbol{n} = 0 \tag{4.7}$$

$$\mu_0 \nabla \phi \cdot \boldsymbol{n} - \boldsymbol{H}_s \cdot \boldsymbol{n} - \nabla \times \boldsymbol{A} \cdot \boldsymbol{n} = 0 \tag{4.8}$$

Consequently the Galerkin weak form of (4.4) and (4.6) is obtained as follows:

$$\begin{aligned}&\int_{\Omega_1} \mu_0 \nabla \phi \nabla N d\Omega + \int_{\Gamma_{12}} \mu_0 \nabla \phi \cdot \boldsymbol{n} N d\Gamma + \int_{\Omega_2} \nu \nabla \times \boldsymbol{A} \cdot \nabla \times \boldsymbol{N} d\Omega \\ &+ \int_{\Omega_2} \nu \nabla \cdot \boldsymbol{A} \nabla \cdot \boldsymbol{N} d\Omega - \int_{\Gamma_{12}} \nu (\nabla \times \boldsymbol{A}) \times \boldsymbol{n} \cdot \boldsymbol{N} d\Gamma = 0\end{aligned} \tag{4.9}$$

For the magnetic reluctivity $\nu(||\boldsymbol{B}||)$ which is varying with $\boldsymbol{B}$ in the iron region Ω_2, the Galerkin weak form gives rise to a nonlinear equation. A numerical technique based on nonlinear functional algorithm is proposed to reduce the complexity of the discretization of the nonlinear term. For simplification, the linear term of (4.9) is omitted and it is assumed the following nonlinear equation is to be solved.

$$\begin{aligned}&\int_{\Omega_2} \nu \nabla \times \boldsymbol{A} \cdot \nabla \times \boldsymbol{N} d\Omega + \int_{\Omega_2} \nu \nabla \cdot \boldsymbol{A} \nabla \cdot \boldsymbol{N} d\Omega \\ &- \int_{\Gamma_{12}} \nu (\nabla \times \boldsymbol{A}) \times \boldsymbol{n} \cdot \boldsymbol{N} d\Gamma = 0\end{aligned} \tag{4.10}$$

A functional $F(\boldsymbol{A})$ is introduced as,

$$\begin{aligned}F(\boldsymbol{A}) &= \int_{\Omega_2} \nu(\|\nabla \times \boldsymbol{A}\|) \nabla \times \boldsymbol{A} \cdot \nabla \times \boldsymbol{N} d\Omega \\ &+ \int_{\Omega_2} \nu(\|\nabla \times \boldsymbol{A}\|) \nabla \cdot \boldsymbol{A} \nabla \cdot \boldsymbol{N} d\Omega \\ &- \int_{\Gamma_{12}} \nu(\|\nabla \times \boldsymbol{A}\|)(\nabla \times \boldsymbol{A}) \times \boldsymbol{n} \cdot \boldsymbol{N} d\Gamma\end{aligned} \tag{4.11}$$

The nonlinear equation to be solved is,

$$F(\boldsymbol{A}) = 0 \tag{4.12}$$

Newton-Raphson method is used to linearize equation (4.12),

$$F(\boldsymbol{A}+\Delta\boldsymbol{A}) = F(\boldsymbol{A}) + F^{'}(\boldsymbol{A})\Delta\boldsymbol{A} = 0 \tag{4.13}$$

Thus,

$$F^{'}(\boldsymbol{A})\Delta\boldsymbol{A} = -F(\boldsymbol{A}) \tag{4.14}$$

Adding *F'(A)A* to both sides of (4.14) gives,

$$F^{'}(\boldsymbol{A})\Delta\boldsymbol{A} + F^{'}(\boldsymbol{A})\boldsymbol{A} = F^{'}(\boldsymbol{A})\boldsymbol{A} - F(\boldsymbol{A}) \tag{4.15}$$

Let $\boldsymbol{w} = \boldsymbol{A}+\Delta\boldsymbol{A}$, where $\boldsymbol{w}$ means the unknown variable of the current step, $\boldsymbol{A}$ is the iteration solution from the previous step. Consequently,

$$F^{'}(\boldsymbol{A})\boldsymbol{w} = F^{'}(\boldsymbol{A})\boldsymbol{A} - F(\boldsymbol{A}) \tag{4.16}$$

$$\begin{aligned}
F^{'}(\boldsymbol{A})\boldsymbol{w} &= \int_{\Omega} \nu^{'}(\|\nabla\times\boldsymbol{A}\|)\boldsymbol{w}\nabla\times\boldsymbol{A}\cdot\nabla\times\boldsymbol{N}d\Omega \\
&+ \int_{\Omega_2} \nu(\|\nabla\times\boldsymbol{A}\|)\nabla\times\boldsymbol{w}\cdot\nabla\times\boldsymbol{N}d\Omega + \int_{\Omega_2} \nu^{'}(\|\nabla\times\boldsymbol{A}\|)\boldsymbol{w}\nabla\cdot\boldsymbol{A}\nabla\cdot\boldsymbol{N}d\Omega \\
&+ \int_{\Omega_2} \nu(\|\nabla\times\boldsymbol{A}\|)\nabla\cdot\boldsymbol{w}\nabla\cdot\boldsymbol{N}d\Omega - \int_{\Gamma_{12}} \nu^{'}(\|\nabla\times\boldsymbol{A}\|)\boldsymbol{w}(\nabla\times\boldsymbol{A})\times\boldsymbol{n}\cdot\boldsymbol{N}d\Gamma \\
&- \int_{\Gamma_{12}} \nu(\|\nabla\times\boldsymbol{A}\|)(\nabla\times\boldsymbol{w})\times\boldsymbol{n}\cdot\boldsymbol{N}d\Gamma
\end{aligned} \tag{4.17}$$

The term $\nu^{'}(\|\nabla\times\boldsymbol{A}\|)\boldsymbol{w}$ is as follows.

$$\nu^{'}(\|\nabla\times\boldsymbol{A}\|)\boldsymbol{w} = \left(\frac{\partial(\nu(\|\nabla\times\boldsymbol{A}\|))}{\partial\boldsymbol{A}}\right)\boldsymbol{w}$$

$$= \left(\frac{\partial\nu(\|\nabla\times\boldsymbol{A}\|)}{\partial\|\nabla\times\boldsymbol{A}\|}\frac{\partial\|\nabla\times\boldsymbol{A}\|}{\partial\boldsymbol{A}}\right)\boldsymbol{w} = \left(\frac{\partial\nu(\|\boldsymbol{B}\|)}{\partial\|\boldsymbol{B}\|}\frac{\partial\|\nabla\times\boldsymbol{A}\|}{\partial\boldsymbol{A}}\right)\boldsymbol{w}$$

Now consider the G-derivative $\dfrac{\partial\|\nabla\times\boldsymbol{A}\|}{\partial\boldsymbol{A}}\boldsymbol{w}$.

$$\begin{aligned}
\frac{\partial\|\nabla\times\boldsymbol{A}\|}{\partial\boldsymbol{A}}\boldsymbol{w} &= \lim_{\Delta\boldsymbol{A}\to\boldsymbol{0}} \frac{\sqrt{(\nabla\times(\boldsymbol{A}+\Delta\boldsymbol{A}),\nabla\times(\boldsymbol{A}+\Delta\boldsymbol{A}))} - \sqrt{(\nabla\times\boldsymbol{A},\nabla\times\boldsymbol{A})}}{\Delta\boldsymbol{A}}\boldsymbol{w} \\
&= \lim_{\Delta\boldsymbol{A}\to\boldsymbol{0}} \frac{(\nabla\times(\boldsymbol{A}+\Delta\boldsymbol{A}),\nabla\times(\boldsymbol{A}+\Delta\boldsymbol{A})) - (\nabla\times\boldsymbol{A},\nabla\times\boldsymbol{A})}{\Delta\boldsymbol{A}(\sqrt{(\nabla\times(\boldsymbol{A}+\Delta\boldsymbol{A}),\nabla\times(\boldsymbol{A}+\Delta\boldsymbol{A}))} + \sqrt{(\nabla\times\boldsymbol{A},\nabla\times\boldsymbol{A})})}\boldsymbol{w} \\
&= \frac{(\nabla\times\boldsymbol{A},\nabla\times\boldsymbol{w})}{\sqrt{(\nabla\times\boldsymbol{A},\nabla\times\boldsymbol{A})}} = \frac{1}{\|\boldsymbol{B}\|}\nabla\times\boldsymbol{A}\cdot\nabla\times\boldsymbol{w}
\end{aligned}$$

$$
\begin{aligned}
&F^{'}(\boldsymbol{A})\boldsymbol{w} = \int_{\Omega_2} \frac{1}{\|\boldsymbol{B}\|}\frac{\partial \nu}{\partial \|\boldsymbol{B}\|} \nabla\times\boldsymbol{A}\cdot\nabla\times\boldsymbol{w}\nabla\times\boldsymbol{A}\cdot\nabla\times\boldsymbol{N}d\Omega \\
&+\int_{\Omega_2} \nu(\|\nabla\times\boldsymbol{A}\|)\nabla\times\boldsymbol{w}\cdot\nabla\times\boldsymbol{N}d\Omega + \int_{\Omega_2} \frac{1}{\|\boldsymbol{B}\|}\frac{\partial \nu}{\partial \|\boldsymbol{B}\|} \nabla\times\boldsymbol{A}\cdot\nabla\times\boldsymbol{w}\nabla\cdot\boldsymbol{A}\nabla\cdot\boldsymbol{N}d\Omega \\
&+\int_{\Omega_2} \nu(\|\nabla\times\boldsymbol{A}\|)\nabla\cdot\boldsymbol{w}\nabla\cdot\boldsymbol{N}d\Omega - \int_{\Gamma_{12}} \frac{1}{\|\boldsymbol{B}\|}\frac{\partial \nu}{\partial \|\boldsymbol{B}\|} \nabla\times\boldsymbol{A}\cdot\nabla\times\boldsymbol{w}(\nabla\times\boldsymbol{A})\times\boldsymbol{n}\cdot\boldsymbol{N}d\Gamma \\
&-\int_{\Gamma_{12}} \nu(\|\nabla\times\boldsymbol{A}\|)(\nabla\times\boldsymbol{w})\times\boldsymbol{n}\cdot\boldsymbol{N}d\Gamma
\end{aligned}
$$

$$
\begin{aligned}
&F^{'}(\boldsymbol{A})\boldsymbol{A} - F(\boldsymbol{A}) = \int_{\Omega_2} \frac{1}{\|\boldsymbol{B}\|}\frac{\partial \nu}{\partial \|\boldsymbol{B}\|} \nabla\times\boldsymbol{A}\cdot\nabla\times\boldsymbol{A}\nabla\times\boldsymbol{A}\cdot\nabla\times\boldsymbol{N}d\Omega \\
&+\int_{\Omega_2} \frac{1}{\|\boldsymbol{B}\|}\frac{\partial \nu}{\partial \|\boldsymbol{B}\|} \nabla\times\boldsymbol{A}\cdot\nabla\times\boldsymbol{A}\nabla\cdot\boldsymbol{A}\nabla\cdot\boldsymbol{N}d\Omega \\
&-\int_{\Gamma_{12}} \frac{1}{\|\boldsymbol{B}\|}\frac{\partial \nu}{\partial \|\boldsymbol{B}\|} \nabla\times\boldsymbol{A}\cdot\nabla\times\boldsymbol{A}(\nabla\times\boldsymbol{A})\times\boldsymbol{n}\cdot\vec{N}d\Gamma
\end{aligned}
$$

Hence (4.16) can be written as,

$$
\begin{aligned}
&\int_{\Omega_2} \nu(\|\nabla\times\boldsymbol{A}\|)\nabla\times\boldsymbol{w}\cdot\nabla\times\boldsymbol{N}d\Omega + \int_{\Omega_2} \nu(\|\nabla\times\boldsymbol{A}\|)\nabla\cdot\boldsymbol{w}\nabla\cdot\boldsymbol{N}d\Omega \\
&-\int_{\Gamma_{12}} \nu(\|\nabla\times\boldsymbol{A}\|)(\nabla\times\boldsymbol{w})\times\boldsymbol{n}\cdot\boldsymbol{N}d\Gamma + \int_{\Omega_2} \frac{1}{\|\boldsymbol{B}\|}\frac{\partial \nu}{\partial \|\boldsymbol{B}\|} \nabla\times\boldsymbol{A}\cdot\nabla\times\boldsymbol{w}\nabla\times\boldsymbol{A}\cdot\nabla\times\boldsymbol{N}d\Omega \\
&+\int_{\Omega_2} \frac{1}{\|\boldsymbol{B}\|}\frac{\partial \nu}{\partial \|\boldsymbol{B}\|} \nabla\times\boldsymbol{A}\cdot\nabla\times\boldsymbol{w}\nabla\cdot\boldsymbol{A}\nabla\cdot\boldsymbol{N}d\Omega - \int_{\Gamma_{12}} \frac{1}{\|\boldsymbol{B}\|}\frac{\partial \nu}{\partial \|\boldsymbol{B}\|} \nabla\times\boldsymbol{A}\cdot\nabla\times\boldsymbol{w}(\nabla\times\boldsymbol{A})\times\boldsymbol{n}\cdot\boldsymbol{N}d\Gamma \quad (4.18) \\
&=\int_{\Omega_2} \frac{1}{\|\boldsymbol{B}\|}\frac{\partial \nu}{\partial \|\boldsymbol{B}\|} \nabla\times\boldsymbol{A}\cdot\nabla\times\boldsymbol{A}\nabla\times\boldsymbol{A}\cdot\nabla\times\boldsymbol{N}d\Omega + \int_{\Omega_2} \frac{1}{\|\boldsymbol{B}\|}\frac{\partial \nu}{\partial \|\boldsymbol{B}\|} \nabla\times\boldsymbol{A}\cdot\nabla\times\boldsymbol{A}\nabla\cdot\boldsymbol{A}\nabla\cdot\boldsymbol{N}d\Omega \\
&-\int_{\Gamma_{12}} \frac{1}{\|\boldsymbol{B}\|}\frac{\partial \nu}{\partial \|\boldsymbol{B}\|} \nabla\times\boldsymbol{A}\cdot\nabla\times\boldsymbol{A}(\nabla\times\boldsymbol{A})\times\boldsymbol{n}\cdot\boldsymbol{N}d\Gamma
\end{aligned}
$$

In solving (4.9), the nonlinear iteration is based on the weak form of (4.18). It is convenient to calculate ∇x***A*** and ∇·***A*,** because ***A*** is the known solution of the previous step.

4.3. THE TWO-GRID FEM TECHNIQUE

To apply the two grid FEM technique, it is necessary to generate an auxiliary coarse mesh on the solution domain. This means there are two meshes, the coarse one and the fine one. Hence two finite element subspaces v_H and v_h (with mesh size $h << H$) are employed for the FEM discretization. The following nonlinear equation is solved first until the convergent solution $(\phi_H, \boldsymbol{A_H})$ is obtained.

$$\int_{\Omega_1} \mu_0 \nabla \phi_H \nabla N_H d\Omega + \int_{\Gamma_{12}} \mu_0 \nabla \phi_H \cdot \boldsymbol{n} N_H d\Gamma + \int_{\Omega_2} \nu \nabla \times \boldsymbol{A_H} \cdot \nabla \times \boldsymbol{N_H} d\Omega$$
$$+ \int_{\Omega_2} \nu \nabla \cdot \boldsymbol{A_H} \nabla \cdot \boldsymbol{N_H} d\Omega - \int_{\Gamma_{12}} \nu (\nabla \times \boldsymbol{A_H}) \times \boldsymbol{n} \cdot \boldsymbol{N_H} d\Gamma = 0$$

The solution $(\phi_H, \boldsymbol{A_H})$ is then interpolated to produce $(\phi_h, \boldsymbol{A_h})$ on the fine mesh. Then assume $(\phi_h, \boldsymbol{A_h})$ be the initial solution of the nonlinear iteration on the fine mesh and the following nonlinear equation would have to be solved until the convergent solution $(\phi_h, \boldsymbol{A_h})$ is obtained.

$$\int_{\Omega_1} \mu_0 \nabla \phi_h \nabla N_h d\Omega + \int_{\Gamma_{12}} \mu_0 \nabla \phi_h \cdot \boldsymbol{n} N_h d\Gamma + \int_{\Omega_2} \nu \nabla \times \boldsymbol{A_h} \cdot \nabla \times \boldsymbol{N_h} d\Omega$$
$$+ \int_{\Omega_2} \nu \nabla \cdot \boldsymbol{A_h} \nabla \cdot \boldsymbol{N_h} d\Omega - \int_{\Gamma_{12}} \nu (\nabla \times \boldsymbol{A_h}) \times \boldsymbol{n} \cdot \boldsymbol{N_h} d\Gamma = 0$$

The flow chart of the two-grid technique is shown in Fig. 4.1.

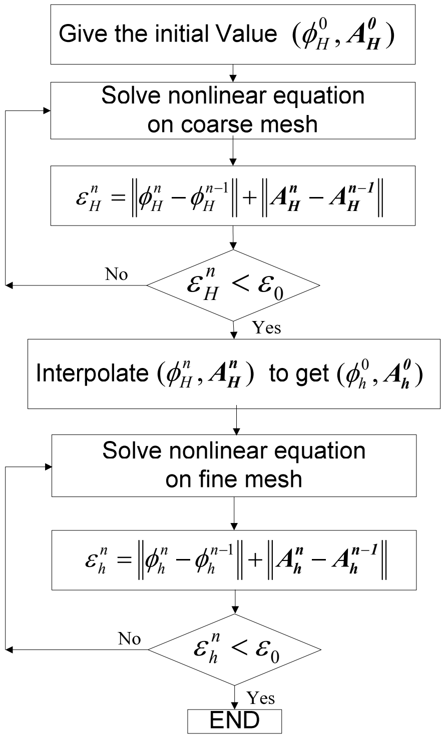

Give the initial Value $(\phi_H^0, \boldsymbol{A_H^0})$
Solve nonlinear equation on coarse mesh
$\varepsilon_H^n = \|\phi_H^n - \phi_H^{n-1}\| + \|\boldsymbol{A_H^n} - \boldsymbol{A_H^{n-1}}\|$
$\varepsilon_H^n < \varepsilon_0$
No
Yes
Interpolate $(\phi_H^n, \boldsymbol{A_H^n})$ to get $(\phi_h^0, \boldsymbol{A_h^0})$
Solve nonlinear equation on fine mesh
$\varepsilon_h^n = \|\phi_h^n - \phi_h^{n-1}\| + \|\boldsymbol{A_h^n} - \boldsymbol{A_h^{n-1}}\|$
$\varepsilon_h^n < \varepsilon_0$
No
Yes
END

Fig. 4.1 Flow chart for the two-grid FEM technique

4.4. INTERPOLATION METHOD

The accurate interpolation of the solution from coarse grid to fine grid is an important procedure of the proposed method. The nonlinear iteration solution needs to be mapped to the fine grid as accurately as possible. The proposed interpolation is based on the FEM space v_H. Given the basis function on v_H is $\{\psi_i\}_{i=1}^n$, and the solution is $\{u_i\}_{i=1}^n$, then if the node $v(x,y,z)$ is in the fine mesh, the interpolation value on node v is:

$$u(v) = \sum_{i=1}^{n} u_i \psi_i \left(\xi(x,y,z), \eta(x,y,z), \zeta(x,y,z) \right) \tag{4.19}$$

where (ξ, η, ζ) are the reference coordinates. In order to determine the reference coordinates of node v, the coarse mesh element which contains it must be found according to its global coordinates (x, y, z). There are two procedures to find the element. Firstly one must determine a set of elements that probably contain the point according to the global coordinates of the node in the element. The second step is to solve the following nonlinear algebraic transformation equation by Newton iteration method on each element in the set:

$$\begin{cases} x = f(\xi,\eta,\zeta) \\ y = g(\xi,\eta,\zeta) \\ z = h(\xi,\eta,\zeta) \end{cases} \tag{4.20}$$

since the isoparametric elements *f, g, h* are shape functions.

With this method, the reference coordinates (ξ,η,ζ) can be obtained in the set of elements. If (ξ,η,ζ) is in the defined range, the element that contains the node is found and the interpolation values on the node can be calculated by (4.19).

4.5. Newton-Raphson Method with a Relaxation Factor

Newton-Raphson method is commonly used to solve nonlinear equations. In ordinary Newton-Raphson method, the relaxation factor is set to be 1 at every iteration step. An adjustment of the relaxation factor will influence the convergence procedure and iteration number, especially when the problem is strongly nonlinear [36]. In order to guarantee the convergence on the coarse mesh, an efficient method to adjust the relaxation factor based on the inner product between the increments of the solution is proposed.

For the nonlinear iteration, the relaxation method used is:

$$u^{k+1} = u^k + \alpha^k \delta u^k \tag{4.21}$$

where *u* is the variable for the solution, *α* is the relaxation factor, δu is the increment of *u*, and the superscripts denote the iteration number. Relaxation factor *α* is adjusted by the inner product between the increments of the present solution δu^{k} and the previous solution $\delta u^{k\text{-}1}$. The cosine of the angle between δu^{k} and $\delta u^{k\text{-}1}$ will be used to determine whether to enlarge or reduce the relaxation factor; and the relaxation factor is controlled to be smaller than or equal to 1.0. A concrete implementation of the adjustment in the chapter is shown below.

```
$Step1: \quad set\ \alpha^{0} = 1.0$
$Step2: \quad for\ k = 1, n\ \ do:$
    $cc = \cos\left\langle \delta u^{k}, \delta u^{k-1} \right\rangle = \frac{\left\langle \delta u^{k}, \delta u^{k-1} \right\rangle}{\left\| \delta u^{k} \right\| \left\| \delta u^{k-1} \right\|}$
    $if\ (cc>0.8) \quad \alpha^{k} = \alpha^{k\text{-}1} \cdot 2.0$
    $if\ (cc<0.3) \quad \alpha^{k} = \alpha^{k\text{-}1} \cdot 0.5$
    $if\ (cc<-0.4)\ \alpha^{k} = \alpha^{k\text{-}1} \cdot 0.5$
    $if\ (cc<-0.8)\ \alpha^{k} = \alpha^{k\text{-}1} \cdot 0.5$
    $if\ (\alpha^{k} > 1.0) \quad \alpha^{k} = 1.0$
```

4.6. NUMERICAL EXPERIMENT

TEAM Workshop problem 13 [37] as shown in Fig. 4.2 and Fig. 4.3 is taken as a test example and the results derived from the two-grid method are compared with those obtained using a general method.

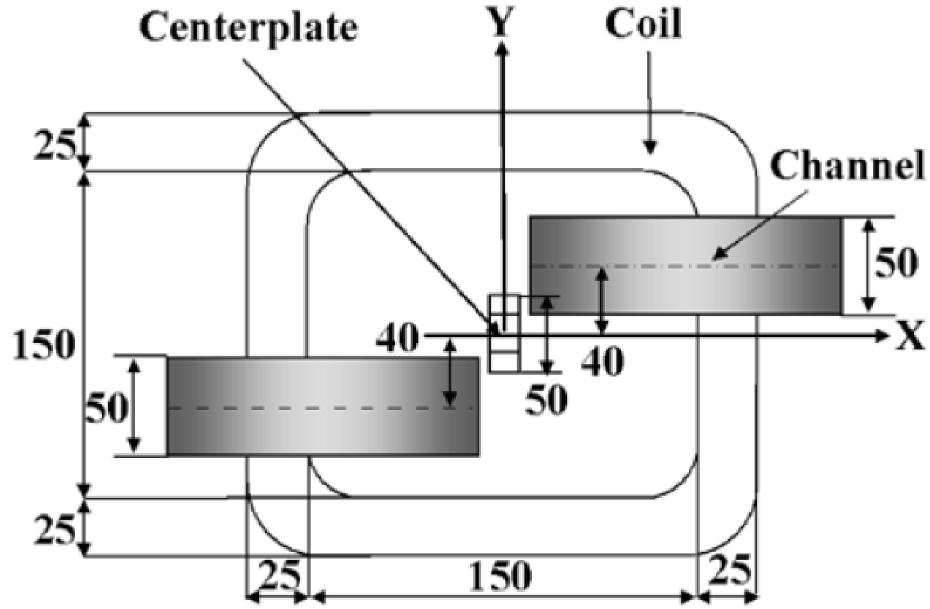


Fig. 4.2 TEAM 13 model for the two-grid method (plan view)

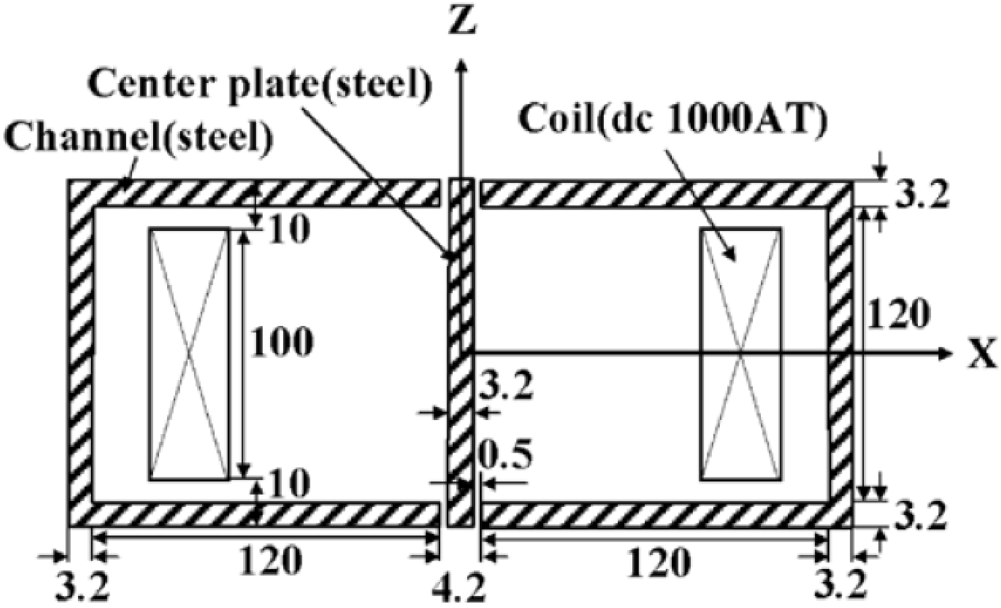


Fig. 4.3 TEAM 13 model for the two-grid method (front view)

In the FEM analysis, first-order tetrahedral elements are used. There are 4136 nodes and 17599 elements in the coarse mesh as well as 25877 nodes and 117895 elements in the fine mesh.

The common *B*-*H* curve of the steel is used and the curve of high flux density (*B*>1.8T) is approximated by the following equation:

$$\begin{aligned} B &= \mu_0 H + (aH^2 + bH + c) \qquad (1.8 \le B \le 2.22T) \\ B &= \mu_0 H + Ms \qquad\qquad\qquad (B \ge 2.22T) \end{aligned} \tag{4}$$

where μ_o is the permeability of free space and the constants *a*, *b*, *c* are $-2.23\times10^{-10}, 2.327\times10^{-5}, 1.590$, respectively. *Ms* is the saturation magnetization (2.16T) of the steel.

In the numerical experiment, the incomplete Cholesky-conjugate gradient method is used as the algebraic equation solver. Table 4-I compares the iteration number and CPU time required to obtain a converged solution. It is found that the solution time on the fine mesh is about 10 times as long as that on the coarse mesh in every iteration step. The total solution time of the two-grid FEM which includes the 66 seconds interpolation time is only about 75% of that of the general FEM. The residue of each iteration step as shown in Fig. 4.4 reveals that after iteration on the coarse grid, the residue of interpolation solution on the fine grid is 0.00030. It is the same as that of the 5th residue on the fine grid. That means, the effect of the iteration on the coarse mesh is equivalent to that of five iterations on the fine mesh. However the computation time on the coarse mesh is about 55 s whereas the computation time of the first 5 iteration steps on the fine mesh is about 300 s.

Table 4-I ITERATION NUMBER AND CPU TIME

Method		Iteration Number	One step solution time	Total Time
Two-grid method	Coarse grid	11	4-5 s	565 s
	Fine grid	6	64-71 s	

General method on fine grid	11	46-79 s	766 s

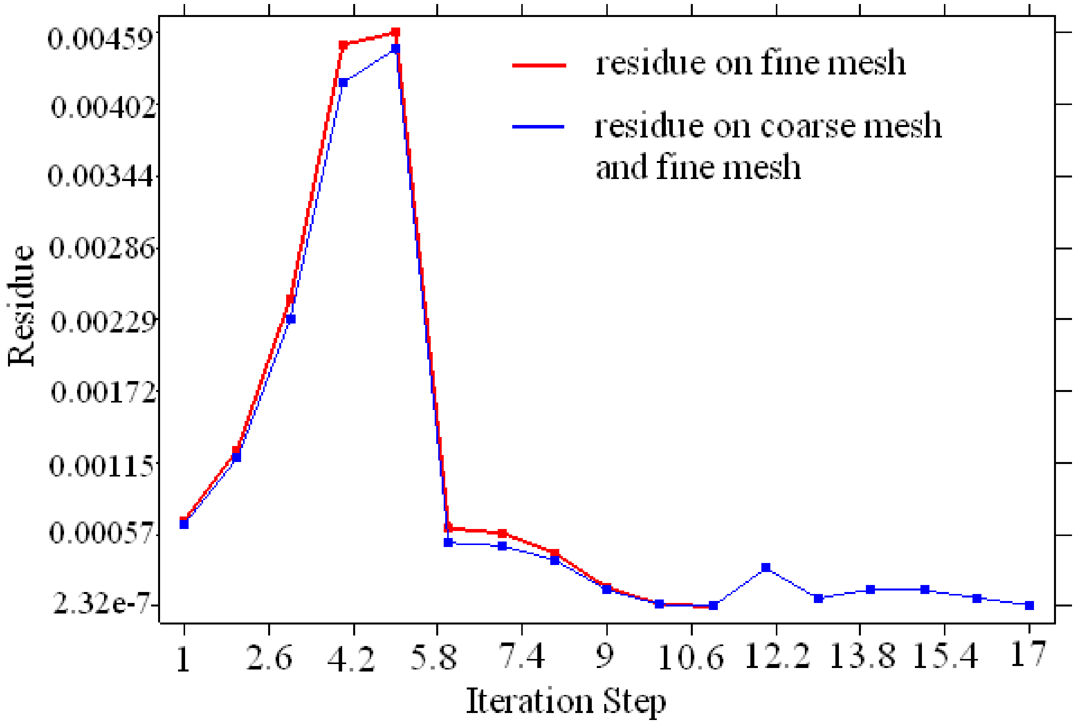


Fig. 4.4 Convergence of the two-grid method and conventional method

4.7. SUMMARY

Finite element method is an accurate numerical method for analyzing the performance of electric motors. However, the disadvantage of FEM is its long computation time, which has become a main obstacle for optimal design of motors. In the chapter, the two-grid FEM algorithm is suggested to reduce the computation time and meanwhile conserves the computation accuracy of the simulation results. TEAM Workshop problem 13 is used to test the proposed method and the results obtained by using the proposed algorithm are compared with those using conventional methods. The computing time using the proposed method is only 75% of that by using the conventional method. Since the two-grid method requires less computing time, it can be effectively applied to study large-scale nonlinear problems.

5. COUPLING FEM WITH OPTIMIZATION METHOD

Electromagnetic devices such as permanent magnet (PM) motors are widely used in industry. Properly designed PM motors can greatly reduce the energy losses during energy conversion from electric power to mechanical power. The design of PM motors is time consuming as each product needs to meet special specifications or requirements. In order to design a PM motor satisfying the demand of applications, many parameters of the motors should be determined. For each design of the PM motors, a numerical model is needed to precisely evaluate its performance. Numerical techniques can approach the practical complicated models under fewest assumptions. They enable the designer to solve the problems which are difficult to consider by using the analytical approach. Many empirical factors then become not necessary when using numerical methods. The actual flux distribution is then found from the given current or voltage waveforms. Although the development of numerical computation in the study of electromagnetic fields dates back only about 20 years ago, it has been maturing into an independent subject with high commercial potential. Among the various numerical methods, the finite element method (FEM) has a dominant position because it has the merits of versatility, strong interchangeability and ready incorporation into standard programs.

Many researchers have studied the performances of electric machines using FEM. The most complete model at present is to use a time stepping FEM coupled with circuit and mechanical equations. Time stepping method can deal with transient problem, and can also be used to calculate steady eddy-current fields when current and magnetic waveforms are non-sinusoidal. When circuit and mechanical equations are coupled with FEM, the currents and torque can be obtained readily without any iteration. The correct saturation of different parts, the eddy-current effect, the distribution of high-order harmonics and mechanical motion are readily accounted for.

Due to the advance of time stepping FEM in recent years, many assumptions, which are essential when using equivalent circuit model, static FEM model and complex FEM model, are not necessary. The time stepping FEM is becoming popular in the analysis of induction machines.

In order to find the best design and reduce the complexity of the motor design, an optimization method need to be used to maximize the performance of the motors and at the same time reduce the energy losses. In the chapter, the coupling FEM with optimization method will be introduced.

In the study of optimal motor design method, the motor design process often needs a comprehensive consideration of the total performance of the motor. Attention also needs to be paid to reduce core loss and reduce PM material. Therefore, the objective function of the optimization method is proposed to consider both to improve the performance and to reduce the loss for the PM motor designs.

5.1. OPTIMIZATION METHOD

In electric device design, the problems to obtain a good performance of a device by adjust its design parameters are often encountered. Such problems are called inverse problem. The performance of the devices is indicated by several performance index or objective. Therefore, inverse problem can be expressed as an optimization problem:

$$min\ f(x) \quad x \in D \ \text{ or } \quad max\ f(x) \quad x \in D \tag{2.41}$$

Where $f(\boldsymbol{x})$ is the function which indicates the performance index of the device and it is called as the objective function; $\boldsymbol{x} = (x_1,x_2,x_3, \ldots, x_n)^T$ is the vector of design parameters and also called as optimization variable; ***D*** is the range of the optimization variable ***x***.

5.2. INTRODUCTION OF COUPLING FEM WITH OPTIMIZATION METHOD

To meet an ever-increasing demand for high quality products to function in variable operating situations, the study of optimal motor design methodologies and techniques has become a new topical area in design during the last two decades [38] - [48]. The difficulty in optimal motor design is mainly that the time needed for computing the performance of the designed motors is too long [49] and the optimization method to find the global optimal solutions is too slow. So many approximate methods are proposed to reduce the evaluation time for new motor design and at the same time some new optimization methods have been proposed to reduce the time needed for finding the global optimal solutions.

The flow chart of coupling FEM with optimization method is show in Fig. 5.1. In the coupling FEM with optimization method, the data transfer from the variables of the optimization method to new meshes in FEM computation is always a challenge. In section 5.1, moving mesh method is proposed to facilitate the problem.

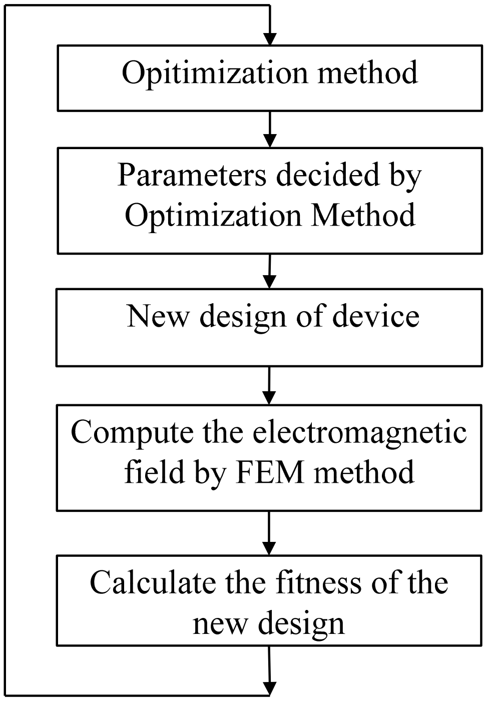

Opitimization method
Parameters decided by Optimization Method
New design of device
Compute the electromagnetic field by FEM method
Calculate the fitness of the new design

Fig. 5.1 Flow chart of coupling FEM with optimization method

5.3. MOVING MESH METHOD

An efficient moving mesh algorithm with fast solver is proposed for remeshing the parameterized computation domain so as to embed finite element methods into optimization algorithms for optimizing the shapes of electromagnetic devices. The proposed method has the merits of conserving the original mesh structure with minimal mesh deformation. The developed algorithm has been applied to TEAM workshop problem No. 25. The reported results are used to showcase the efficiency and validity of the algorithm.

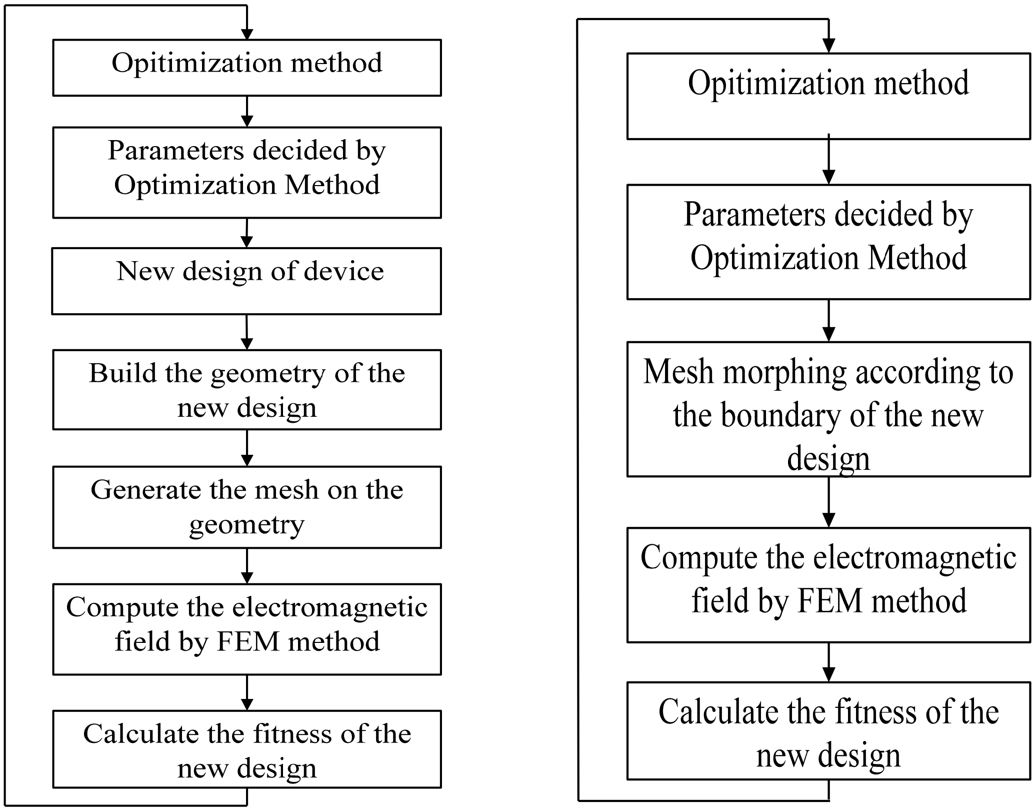
5.3.1. INTRODUCTION

Optimization of the shapes of electromagnetic devices has become an important issue in product design. In the optimization procedure, FEM is usually used to compute objective functions [50]. During the optimization process, the shapes of electromagnetic devices are inevitably changed; therefore the mesh of the devices needs to be regenerated for each FEM computation, as shown in Fig. 5.2(a). It is also well known that data transfer from the variables of the optimization method to new meshes in FEM computation is always a challenge. For example, if the geometry of the computation domain is built and the mesh on that is generated by commercial software which is separated from the FEM program, it is always difficult to rebuild the geometry and regenerate the meshes automatically. Also, the regeneration of FEM mesh is time consuming, especially in 3 dimensional (3D) problems; and the newly generated mesh generally has no relationship with the previous one, making the previous FEM solution not directly available for the new FEM computation in solving nonlinear problem.

Mesh smoothing algorithms have been extensively exploited to improve the mesh quality and, among others, Laplacian smoothing is one of the most popular methods [51]. Lately a related type of smoothing, namely Winslow smoothing, is proposed to alleviate or guard against mesh folding in the process of mesh smoothing [52]. Optimization-based smoothing methods are used to guarantee an improvement in the mesh quality by minimizing a particular mesh quality metric [53]. After an initial mesh is generated on the computation domain, these methods are used to adjust the position of the interior nodes in the meshes so as to obtain better element shapes and the positions of the boundary nodes are kept unchanged.

In this chapter, mesh smoothing methods are introduced to move the meshes in the optimization procedure in which the positions of the nodes on the boundaries are changed. An efficient moving mesh algorithm with a fast solver is proposed for moving the meshes on the parameterized computation domain while embedding FEM into the optimization method in order to optimize the shapes of electromagnetic devices. Fig. 5.2(b) gives the flowchart to illustrate how the proposed algorithm replaces the procedure of rebuilding geometry and mesh regeneration.

In the proposed algorithm, the mesh of the initial shapes of the electromagnetic devices is generated and then with a change in the shapes of the device, the initial mesh will be moved adaptively to the new design. The proposed algorithm can be applied to remesh complex two-dimensional (2D) and three-dimensional (3D) geometrical shapes and save considerable time when regenerating the meshes. The new moving mesh algorithm has the merits of conserving the original structure of the mesh with minimal mesh deformation, as the mesh smoothing methods can smooth out the incremental positions of all the mesh points in the computation domain. Another merit of the proposed method is that when solving time stepping equation or nonlinear equation, the result of the previous FEM solution can serve as a good initial solution for subsequent FEM computation, as the increment of each point is small and each point is in the same material after reconstruction of the mesh. Furthermore, in the proposed algorithm, the mesh information is stored and modified in memory, which avoids the need to retrieve the mesh information from the disk at each optimization step.


Opitimization method
Parameters decided by Optimization Method
New design of device
Build the geometry of the new design
Generate the mesh on the geometry
Compute the electromagnetic field by FEM method
Calculate the fitness of the new design
Opitimization method
Parameters decided by Optimization Method
Mesh morphing according to the boundary of the new design
Compute the electromagnetic field by FEM method
Calculate the fitness of the new design

(a) (b)

Fig. 5.2 (a) Procedure of coupling optimization method with FEM. (b) Moving mesh algorithm in the procedure of coupling FEM in the optimization algorithm.

5.3.2. MOVING MESH ALGORITHM

When optimizing several parameters of an electromagnetic device, the shapes of the device are changed at each optimization step. A novel moving mesh algorithm is proposed to obtain the new mesh according to the new shape of the device, instead of rebuilding the geometry and regenerating the mesh.

In the proposed moving mesh algorithm, an initial shape of the electromagnetic device is set up according to the range of the optimization parameters. Usually the parameters of the initial shape are the average value of its upper and lower bounds. An initial mesh is generated on the initial shape of the device. At each optimization step, changes in the parameters of the device can be considered as a change in the boundaries of the device. According to the changes in the boundaries of the computation domain, the mesh can be moved based on the initial mesh. In this chapter, the moving mesh algorithm is described in 2D, but it can be extended to 3D problems in a straightforward manner.

Laplacian Smoothing

The basic idea of the moving mesh algorithm based on Laplacian smoothing can be illustrated by a simple example in Fig. 5.3. The initial mesh of a fixed initial shape of the electromagnetic device is shown in Fig. 5.3. The shape of the boundary Γ_1 is parameterized and it needs to be optimized while the shape of Γ_2 remains unchanged.

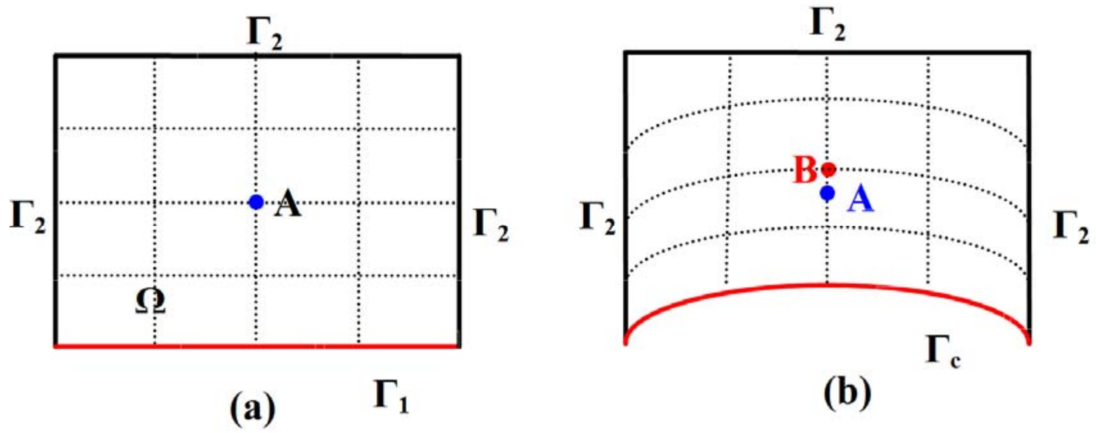


Fig. 5.3 (a) Mesh on the original geometry. (b) Mesh on the geometry of the changed shape.

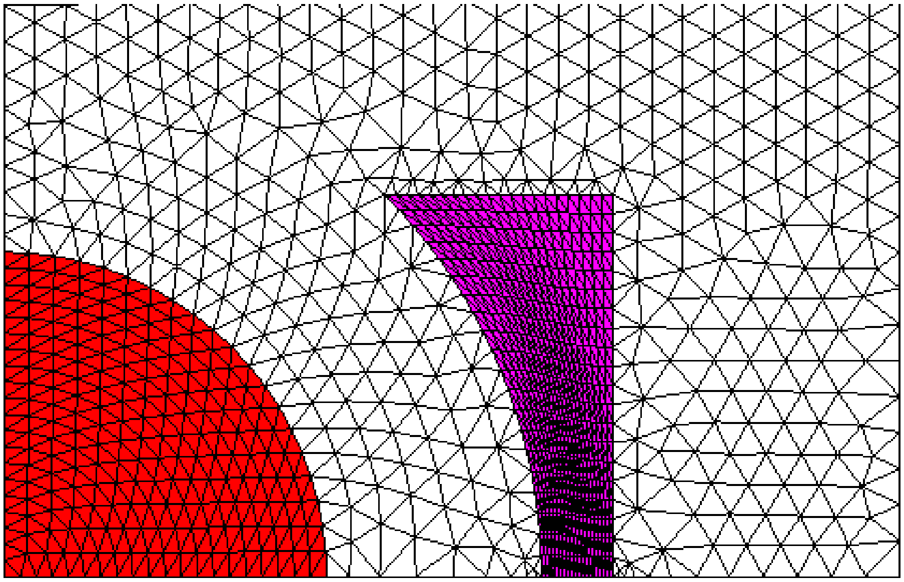


Fig. 5.4 Initial mesh.

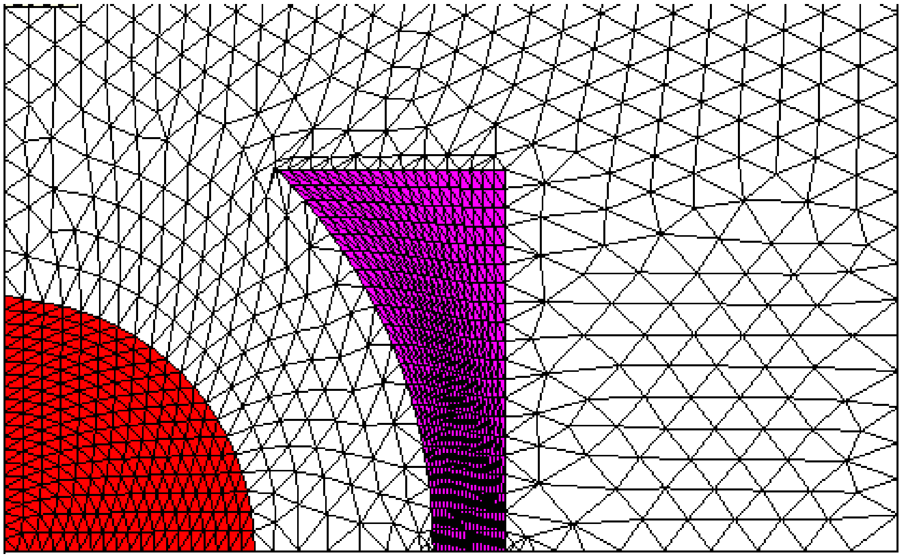


Fig. 5.5 Mesh after changes in the die molds.

When the shape of Γ_1 in Fig. 5.3(a) needs to be changed to Γ_c as shown in Fig. 5.3(b), the coordinates of some interior mesh points in Fig. 5.3(a) should be adjusted accordingly. The incremental coordinates of each interior points $\boldsymbol{u}=(\Delta x,\ \Delta y)$ are variables and satisfy the following Laplace's equation, as the Laplace's equation can smooth out the incremental coordinates of the mesh points inside the computation domain:

$$-\frac{\partial}{\partial x}\left(\frac{\partial \boldsymbol{u}}{\partial x}\right)-\frac{\partial}{\partial y}\left(\frac{\partial \boldsymbol{u}}{\partial y}\right)=0,\ \text{in } \Omega, \tag{5.1}$$

$$\boldsymbol{u}=\mathbf{0}, \qquad \text{on } \Gamma_2,$$

$$\boldsymbol{u}=\boldsymbol{u}_0(b)-\boldsymbol{u}_0(a),\ \text{on } \Gamma_1.$$

where; $\boldsymbol{u}_0(b)$, $\boldsymbol{u}_0(a)$ are the coordinates of the point on Γ_1 and Γ_c, respectively.

The mesh point coordinates in Fig. 5.3(b) can be obtained by the sum of the solution of (5.1) and their original coordinates in Fig. 5.3(a).

Moving mesh algorithm based on Laplacian smoothing is very effective in problems in which the new shape of the device is a minor modification of the initial shape [54]. As a numerical example, the initial mesh of the example mentioned is shown in Fig. 5.4 while the mesh after changing the size and positions of the die molds is shown in Fig. 5.5.

Weighted Laplacian Smoothing

Laplacian smoothing method is easy to implement and efficient. However the method is not guaranteed to work sometimes when inverting mesh elements. A weighted Laplacian smoothing method based on optimization method which is more resilient to mesh folding is introduced in [53]. The weighted Laplacian smoothing method is used with the aim to guarantee an improvement in the mesh quality by minimizing a particular mesh quality metric. Compared to other smoothing method, their main drawback is their computational expense in evaluating the weight on each edge in the mesh. However, in the moving mesh problem being studied, the initial mesh is fixed, and the computation needs to be done only once. The weights are then stored in memory for subsequent several hundreds times of computation.

The first step in the algorithm is to use nonlinear programming to generate a set of weights for each interior node that represents the relative distances of the node to each of its neighbors. The new position of the boundary nodes is then specified. Using these new positions for the boundary nodes and the sets of weights from the original mesh, a system of linear equations to determine new positions is solved for the interior nodes.

Let (x_i, y_i) denote the x- and y-coordinates respectively of the i^{th} interior node in the initial mesh and assume the x- and y-coordinates of its adjacent vertices be given by $\{(x_j, y_j) : j \in N_i\}$, where N_i denotes the set of neighbors of node i. In order to find the set of weights w_{ij}, where w_{ij} is the weight of node j on the interior node i, the log barrier function from linear programming is used to formulate the following optimization problem for each i:

$$\max_{w_{ij}, j \in N_i} \sum_{j \in N_i} log(w_{ij}) \tag{5.2}$$

Subject to:

$$w_{ij} > 0 \tag{5.3}$$

$$\sum_{j \in N_i} w_{ij} = 1 \tag{5.4}$$

$$x_i = \sum_{j \in N_i} w_{ij} x_i \tag{5.5}$$

$$y_i = \sum_{j \in N_i} w_{ij} y_i \tag{5.6}$$

The weights of all interior points w_{ij} can be obtained by solving (5.2)-(5.6) with the Projected Newton Method. This method can be applied to solve optimization problems of the following form:

$$\min_x \ f(x)$$

$$\textit{Subject to} \qquad Ax = b$$

The optimality conditions for the above problem are:

$$\nabla f(x) - A^T \lambda = 0 \tag{5.7}$$

$$b \;-\; Ax=0 \tag{5.8}$$

Here λ denotes the vector of Lagrange multipliers. Equation (5.8) is a system of nonlinear equations in x and λ; thus Newton's method can be used to solve for x and λ. If the initial point is feasible (i.e., $b - Ax_0 = 0$), then p_k must satisfy $Ap_k = 0$ where $x_{k+1} = x_k + p_k$, i.e., p_k is the Newton direction at iteration k. This implies that $p_k = Zv_k$ for an $(n - m)$-dimensional vector v_k, where Z is the basis for the null space of A. It is easy to deduce that the resulting equation for p_k is given by

$$P_k = -\left(H^{-1} - H^{-1}A^T\left[AH^{-1}A^T\right]^{-1}AH^{-1}\right)\nabla f(x_k) \tag{5.9}$$

where $H = \nabla^2 f(x_k)$ [55].

Once the weights have been generated, the new shape of the device is applied to move the boundary nodes to new locations. The final step is to use the positions of the boundary nodes and the sets of weights to simultaneously determine the final positions for the interior nodes by solving the following linear system of equations:

$$\sum_{j \in N_i} w_{ij} x_j = x_i \tag{5.10}$$

$$\sum_{j \in N_i} w_{ij} y_j = y_i \tag{5.11}$$

where x_i and y_i are the new position of the interior node. The linear systems from (5.10) and (5.11) are both $m \times m$, where m is the number of interior nodes and so the solution is unique.

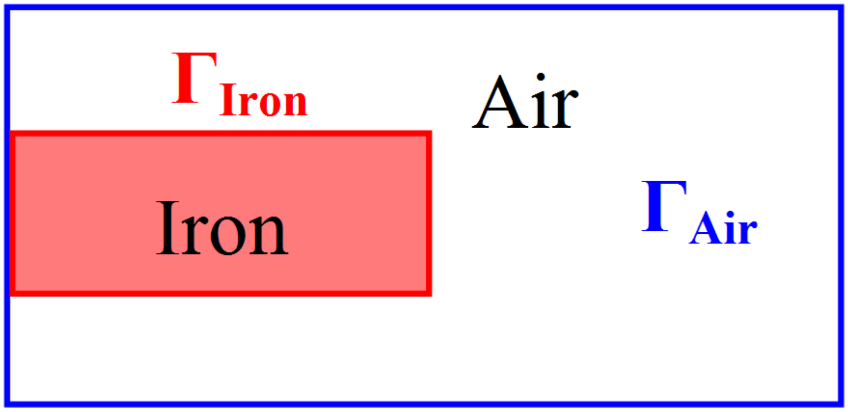


Fig. 5.6 A piece of iron surrounded by air.

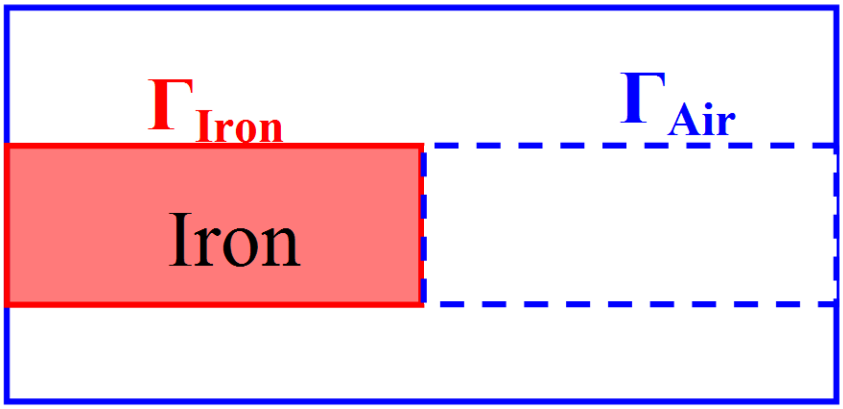


Fig. 5.7 Dividing air domain into several convex domains.

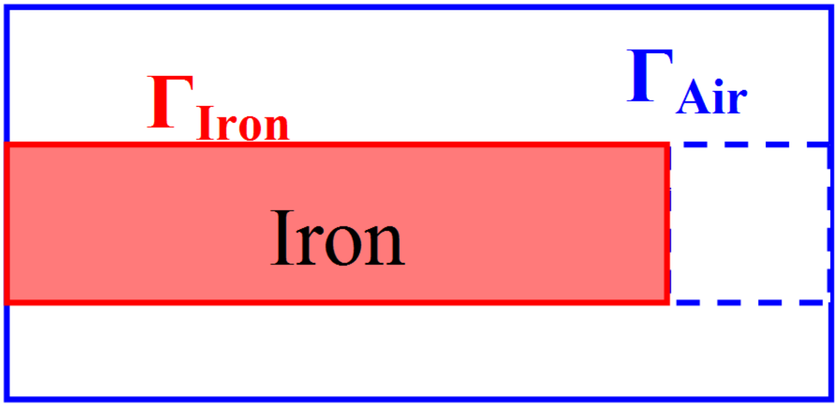


Fig. 5.8 The new shape of iron.

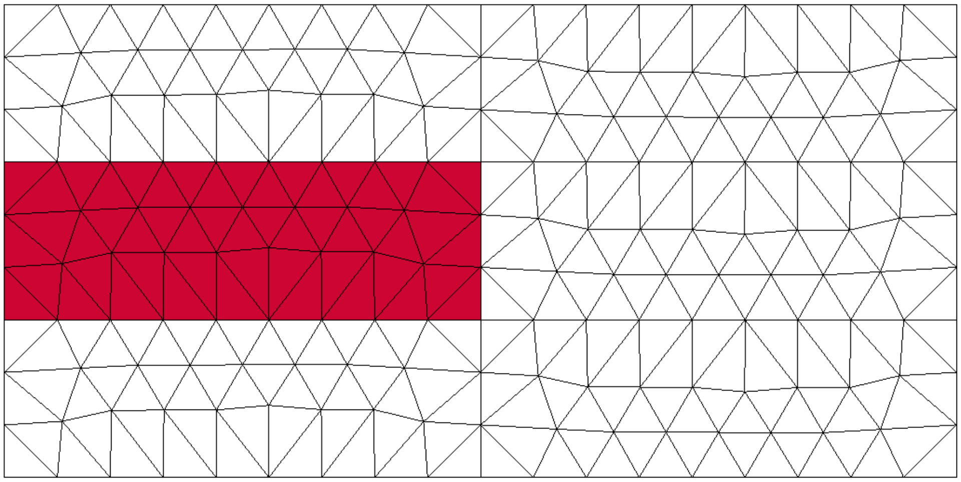


Fig. 5.9 The initial mesh.

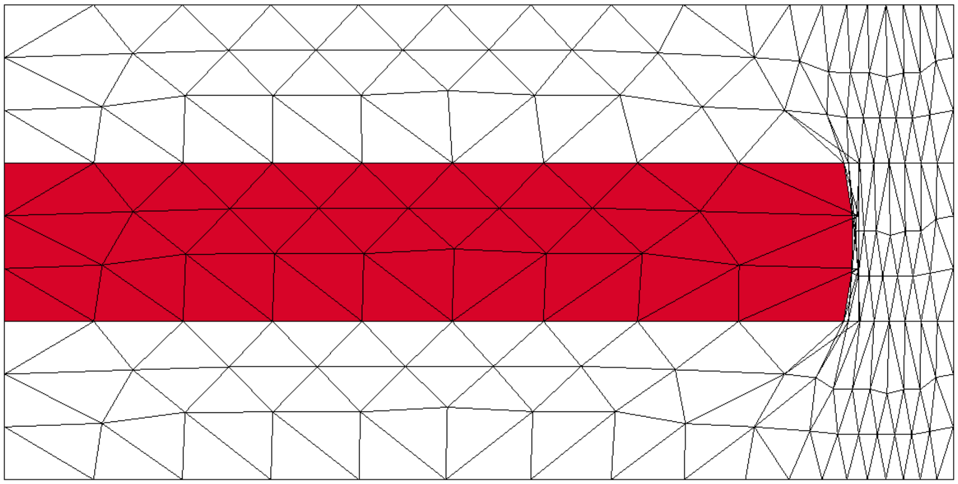


Fig. 5.10 The folding mesh on non-convex domain.

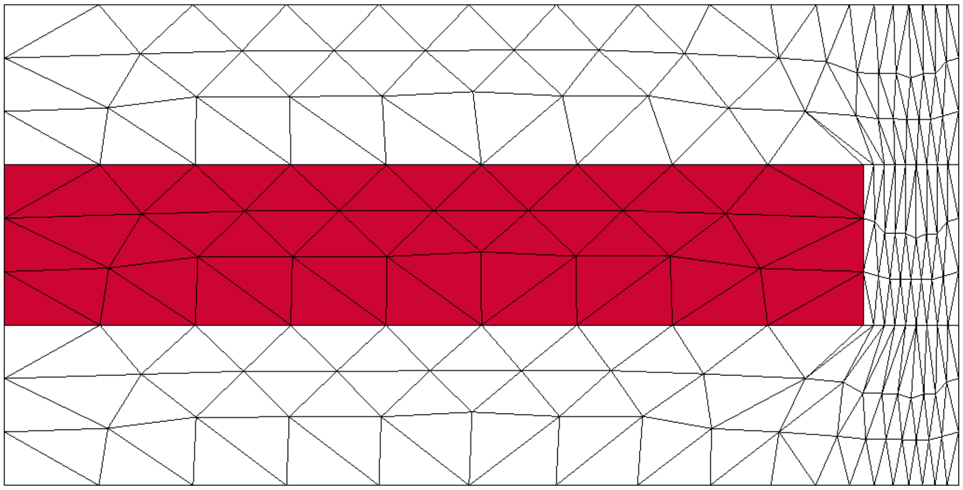


Fig. 5.11 The non-folding mesh.

Non-Convex Domain Decomposition

Laplacian smoothing over a convex domain is guaranteed to obtain a no-folding mesh [56]. But for large deformation in non-convex domain, even weighted Laplacian smoothing may result in mesh folding. In the moving mesh algorithm, in order to avoid mesh folding on non-convex domain, the domain should be decomposed to several convex domains and the smoothing method can then be applied.

In electromagnetic field computation, the computation domain includes both the device and the surrounding air. The domain of the device is often convex and the domain of the air is often non-convex. As a numerical example shown in Fig. 5.6, there is a piece of iron surrounded by air, and the length of the iron needs to be optimized. Here the node position on the shared boundary of the device and the air, namely, Γ_{Iron}, needs to be changed according to the changes in the optimization variables as shown in Fig. 5.8. It is obvious that the air domain is not convex. If the boundary of the air deforms significantly, the solution of the Laplacian equation on the non-convex domain with such Dirichlet boundary condition may become singular, therefore the moved mesh in the air domain may be folding, as shown in Fig. 5.10. So the air domain should be divided into several convex domains as shown in Fig. 5.7. At the same time the new boundary of the air domain should be applied in accordance to the changes in the optimization variable. The non-folding mesh is shown in Fig. 5.11 after the non-convex domain decomposition.

Multi-step Moving Mesh Technique

Laplacian smoothing method is efficient and easy to implement. However the method does not work sometimes when inverting mesh elements. For large deformation in non-convex domain, Laplacian smoothing may result in mesh folding, which is illustrated by the example shown in Fig. 5.10.

In the section, multi-step moving mesh algorithm is proposed to avoid mesh folding when using Laplacian smoothing. Instead of moving the boundary to the destination shape at one step, the mesh is moved gradually for several steps and finally moved to the destination shape. Therefore the following equation (5.12) needs to be solved N times to before mesh moving is completed.

$$-\frac{\partial}{\partial x}\left(\frac{\partial \boldsymbol{u}}{\partial x}\right)-\frac{\partial}{\partial y}\left(\frac{\partial \boldsymbol{u}}{\partial y}\right)=0, \quad \text{in } \Omega, \tag{5.12}$$

$$\boldsymbol{u}=0, \qquad \text{on } \Gamma_2,$$

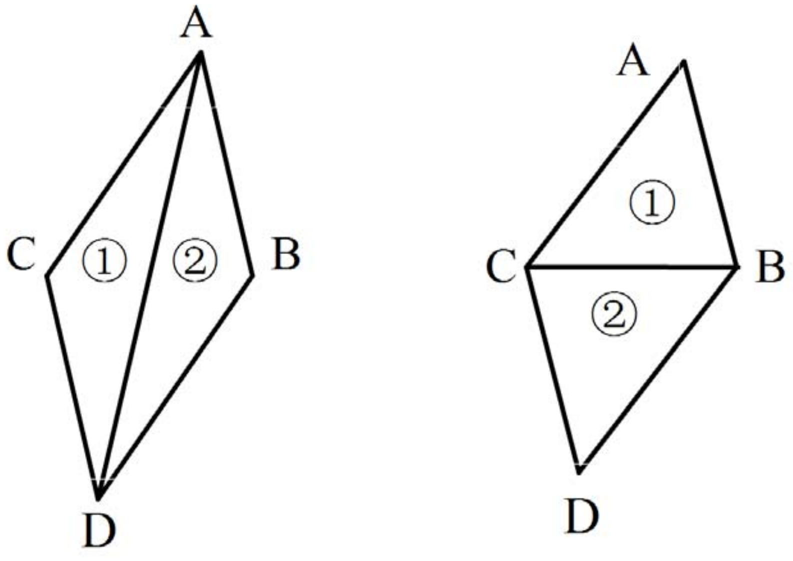
$$\boldsymbol{u} = \frac{\boldsymbol{u}_0(b) - \boldsymbol{u}_0(a)}{N}, \text{ on } \Gamma_1.$$

The boundary condition on Γ_1 in (5.12) shows that the position of boundary point moves gradually N times and the sum of moving distance of all the steps will finally reach its destination position. If no mesh operation is performed after each step movement, the mesh will still be folded because (5.12) is a linear equation and, according to the superposition principle, the solution will be the same as that of (5.1). Hence mesh reconstruction method is adopted after each step to improve the mesh quality.

The first mesh reconstruction technique is to exchange the diagonal line. For two adjacent and largely deformed triangular elements, if their shared edge faces two obtuse angles, then the diagonal line of the two elements are exchanged (e.g. a crossed edge is added and the shared edge deleted) to form two new adjacent elements. As shown in Fig. 5.12, elements 1 and 2 share the common edge AD and the angle $\angle$ACD and $\angle$ABD are obtuse angles. In other words the criteria are satisfied and a crossed edge BC is added with the edge AD deleted to form two new triangular elements 1 and 2. After exchanging the diagonal line of the two adjacent and largely deformed triangular elements, the quality of the two elements is improved.

The diagonal line exchange operation is very efficient for largely deformed triangular elements around the moving boundary. However the operation is not suitable if the two elements are of different materials, because one triangular element will then contain two type of materials that makes it overly complex to realize the exchange operation. Diagonal line exchange operation is imposed only on the largely deformed elements within a certain distance from the moving boundary and not on all elements in the computation domain.

The second mesh reconstruction technique is collapsing element. In the initial mesh, the nodes of the element are arranged in anti-clockwise order. When one point in the triangular element crosses its faced edge from one side to another side, the nodes order then become clockwise and mesh folding occurs in that element which is named the folded element. Folded element should be avoided, otherwise the area of such element calculated by FEM will become negative and such element will not satisfy the requirements of FEM. The criteria for an element to be collapsed are when there is one obtuse angle and adjacent elements are of different materials. In such case, as shown in Fig. 5.13, the largely deformed element ΔADC will be deleted and the two edges AC and DC besides the obtuse angle ∠ACD will collapse to the third edge AD in the element. A new point E is then created on edge AD which is then divided into two newly created edges AE and ED, as shown in Fig. 5.13. Point C in the initial mesh is replaced by point E in the new mesh and all elements containing point C will be changed to contain point E. The newly created point E is linked with point B in the adjacent element to form two new elements ΔABE and ΔBDE. Collapsing element operation is very efficient in avoiding folded element and is imposed only on largely deformed elements within a certain distance from the moving boundary.



(a) (b)

Fig. 5.12 (a) Large deformed triangular element. (b) Triangular element after exchanging diagonal line.

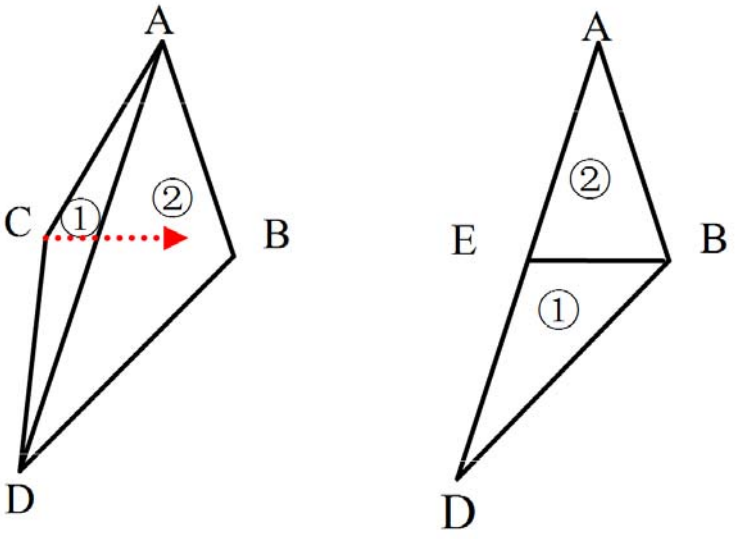


Fig. 5.13 (a) Large deformed triangular element (b) Two new created element after collapsing element.

5.3.3. BOUNDARY TRACKING MOVING MESH ALGORITHM

In the optimization of electromagnetic devices, changes of boundary may be very significant. Thus moving the boundary points in the initial mesh to its destination place may result in large deformation on these elements. Also, the distribution of nodes on the computation domain is changed significantly and the density of the node in the vicinity of the moving boundary may be not as balanced as before. In the proposed boundary tracking moving mesh technique, a set of points in the initial mesh, which are very closed to the new boundary of the devices, are found and the corresponding distances for them to move are determined. The surrounding points are moved to the new boundary by solving (5.1) to form the new mesh. The merit of the boundary tracking technique is that the moving distance of the point in the initial mesh is very slight and it is indeed the same as that for the mesh deformation. Details of boundary tracking technique are described blow.

Assume *y=f(x)* describes the destination of the new boundary shapes. Then,

Step 1: A random point (x_0, y_0) is selected on the new boundary. Search among all elements to find one element, namely E1, which contains (x_0, y_0), and then mark it with a flag. After element E1 is found, the three edges of E1 are expressed as linear functions $y = k_i x+b_i$ ($x_{i0} \leq x \leq x_{i1}$), where x_{i0} and x_{i1} are the x coordinates of two terminal points of the *i*-th edge of E1. Solving (5.13) gives:

$$\begin{cases} y - f(x) = 0 \\ y = k_i x + b_i \end{cases} \tag{5.13}$$

The solutions (x_i, y_i) which are in the range ($x_{i0} \leq x \leq x_{i1}$) are the coordinates of intersection points of the destination boundary and the three edges of element E1.

Step 2: The outward normal of the destination boundary can be determined by an arbitrary fixed internal point and two intersection points, as shown in Fig. 5.14. The nodes of E1 outside the moving boundary are specified by the outward normal, namely N1 and N2 (if two nodes are outside the moving boundary). N1 and N2 are marked as moving nodes if N1 and N2 have not been marked before.

Step 3: Determine the location at which points N1 and N2 should be moved to. If N1 and N2 have not been marked before, move N1 to one of the intersection points, namely NB1, which is closer to N1 and the same is repeated for N2. If either N1 or N2 has been marked before, then move N1 (suppose N2 is marked) to the nearest point among the two intersection points of NB1, NB2 and the middle point of these two points. Therefore, the point N1 are marked with Dirichlet boundary condition and $\Delta \boldsymbol{u} = \boldsymbol{u}\,(NB1) - \boldsymbol{u}\,(N1)$ is the moving distance on the x and y coordinates.

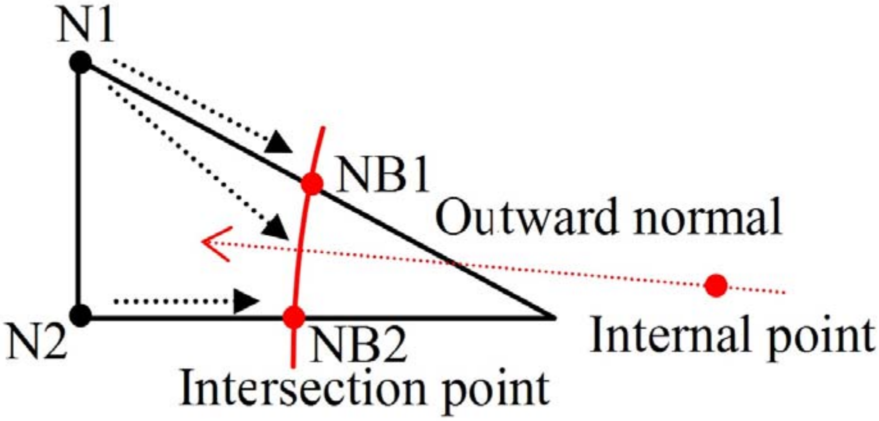


Fig. 5.14 Intersection of element and new boundary

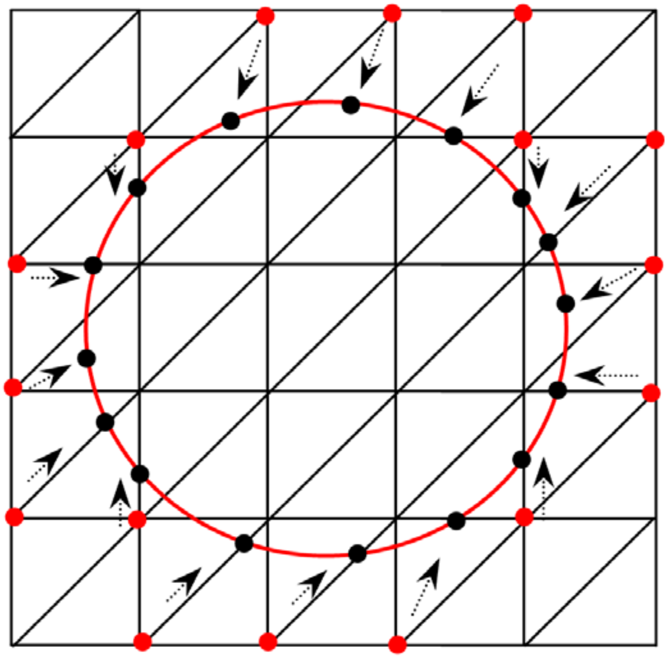


Fig. 5.15 Moving direction of the point surrounding the destination boundary

Step 4: There are three adjacent elements which share the same edge with triangular element E1, and they are E2, E3, and E4. Solve (3) to find the intersection points of E2, E3, E4 and the destination boundary. If the element intersects with the destination boundary and if it has not been marked before (supposing E2), then mark this element and repeat the steps from 1-3.

Repeat step 4 until all the elements which intersect with the destination boundary are found to form a close polygonal line linked by an outside point surrounding the destination boundary as shown in Fig. 5.15. Solve (5.1) with the boundary condition on the polygonal line to move the point on the polygonal line to the destination boundary.

The last step is to determine the material inside the destination boundary. One fixed point O_{in} inside the destination boundary is chosen for reference. For each triangular element, link its inner point with O_{in} to form a line segment which is expressed as $y = kx+b$ ($x_0 \leq x \leq x_1$). Solve (5.13) to find if there exist intersection points with the destination boundary. If it exists, the triangular element is outside. If there is no such intersection point, then the triangular element is in the destination boundary. Then assign the element with the optimized devices material property.

The proposed boundary tracking moving mesh technique is also valid for moving objects in the computation domain.

5.3.4. FAST SOLVER

In each step of optimization, moving mesh algorithm will be called to move the mesh on the initial device adaptively to a new mesh of the newly designed device. In the optimization procedure, the Laplacian smoothing equation or weighted Laplacian equation will be solved on the same initial mesh many times and the difference among the equations in each time is the boundary condition. Assume u_m to denote the unknowns on the interior nodes and u_n to denote the fixed position of the nodes on the boundary, the resulting linear system from (5.1) or (5.10) and (5.11) can be expressed as:

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} u_m \\ u_n \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} \tag{5.14}$$

u_n is known from the applied boundary condition. So the following linear equation needs to be solved:

$$A_{11} u_m = f_1 - A_{12} u_n \tag{5.15}$$

Matrix A_{11}, A_{12} and vector f_1 is the same in each step of the optimization procedure because the equation is solved on the same initial mesh. Thus the matrixes and vector can be computed only once and the solutions are then stored for subsequent computations to save the computing time for the element stiff matrix. To solve linear equation (5.15), matrix A_{11} needs to be decomposed and the decomposition result can also be stored to save the subsequent decomposition time.

Matrix A_{11} is symmetric, so two methods are usually used to solve the above equation: LDL$^{\mathrm{T}}$ decomposition method and Incomplete Cholesky Conjugate Gradient (ICCG) method.

In LDL$^{\mathrm{T}}$ decomposition method, a lower triangular matrix with diagonal element being equal to 1 and a diagonal matrix D can be found to satisfy A_{11}=LDL^T, where L^T denotes the transpose of the matrix L. From (5.15),

$$LDL^T\, u_m = f_1 - A_{12} u_n \qquad (14).$$

Equation (5.14) can be solved by a back substitution progress.

In ICCG method, the preconditioner of A_{11} is computed and stored together with A_{11} for the following steps in solving the linear equation.

In solving 2D problem, LDL$^{\mathrm{T}}$ decomposition is fast for the narrow band width of A_{11} and so LDL$^{\mathrm{T}}$ decomposition method is usually used. While in 3D problem, LDL$^{\mathrm{T}}$ decomposition is slow for the wide band width of A_{11}, so ICCG method is usually employed.

5.3.5. NUMERICAL EXPERIMENT

The proposed method is applied to TEAM workshop problem No. 25 [57]. The goal of this problem is to optimize the shape of a die mold to obtain the best performance of permanent magnets. Fig. 5.16 is the model of the die mold with the electromagnet for the orientation of the magnetic axis of the magnetic powder. The die press and electromagnet are made of steel. The die mold is described by an internal circle of radius *R*1 and by an external ellipse represented by L_2, L_3 and L_4, as shown in Fig. 5.17. The objective function *W* for the optimization problem is given by:

$$W = \sum_{i=1}^{n} \left[\left(B_{xp_i} - B_{xo_i} \right)^2 + \left(B_{yp_i} - B_{yo_i} \right)^2 \right]$$

where *n* is the number of specified points (*n*=10); B_{xpi} and B_{ypi} are computed values along the line *e-f*; B_{xi} and B_{yo} are specified as B_{xo}=1.5cos(θ), B_{yo}=1.5sin(θ) (T).

The constraints of R_1, L_2, L_3 and L_4 can be represented as follows: $5 < R_1 < 9.4$; $12.6 < L_2 < 18$; $14 < L_3 < 45$; $4 < L_4 < 19$.

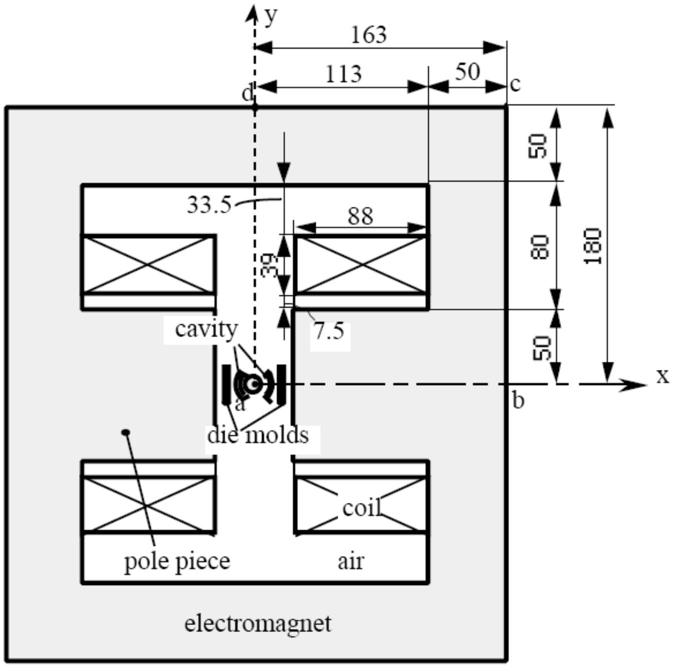


Fig. 5.16 Die mold with electromagnet.

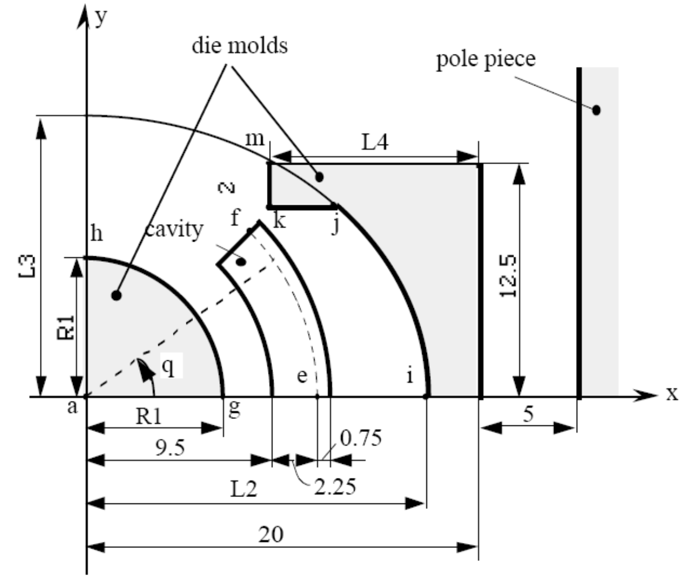


Fig. 5.17 Optimization parameter of the die mold.

Fig. 5.18 is the decomposition of the non-convex domain of air. Fig. 5.19 shows the initial shape of the die mold. The parameters that define the initial shape are: R_1= 7.2, L_2 = 15.3, L_3 = 29.5 and L_4 = 11.5.

The mesh on the optimal shape of die mold with the parameters R_1= 8.94, L_2 = 17.85, L_3 = 15.28, L_4 = 16.68 is shown in Fig. 5.20. Two cases of extreme deformation of the shape of the die mold are studied and there is no folding in the mesh, as shown in Fig. 5.21 and Fig. 5.22, respectively.

To do 100 times of remeshing, it takes about 21 seconds using moving mesh algorithm while 55 seconds are needed to regenerate mesh with commercial software. In other words, a significant of computing time can be saved using the proposed technique.

5.3.6. SUMMARY

In order to facilitate the procedure for obtaining a new mesh on the newly designed device defined by the optimized parameters and to reduce the time of regenerating mesh, a moving mesh algorithm with fast solver is presented in this chapter. TEAM Workshop problem 25 is used to test the resistance of mesh folding of the proposed algorithm and also to evaluate the computing time saved when using the proposed fast solver. The result shows that the moving mesh algorithm has successfully overcome the difficulty of data transfer when coupling FEA and optimization method.

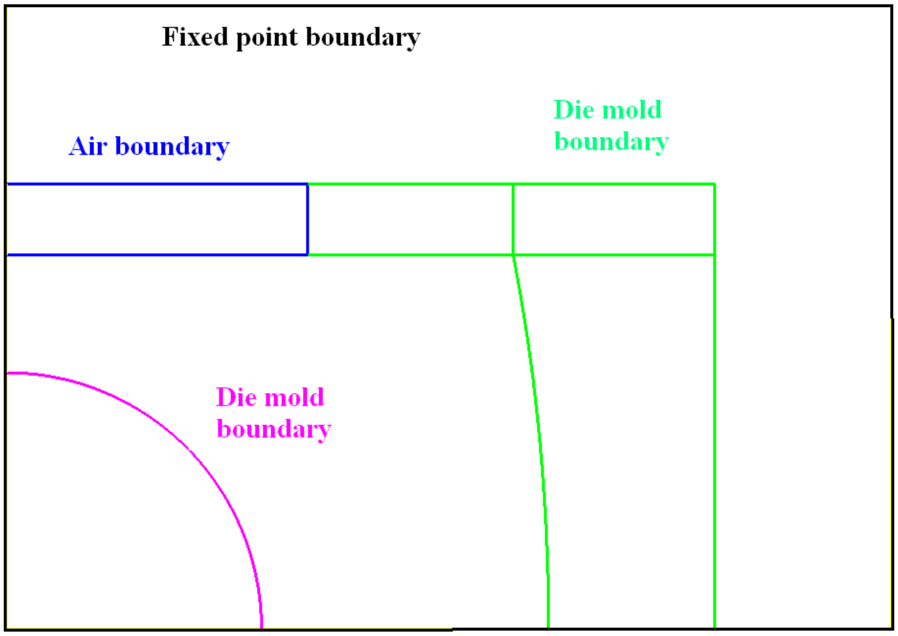


Fig. 5.18 Decomposition of the non-convex domain of air.

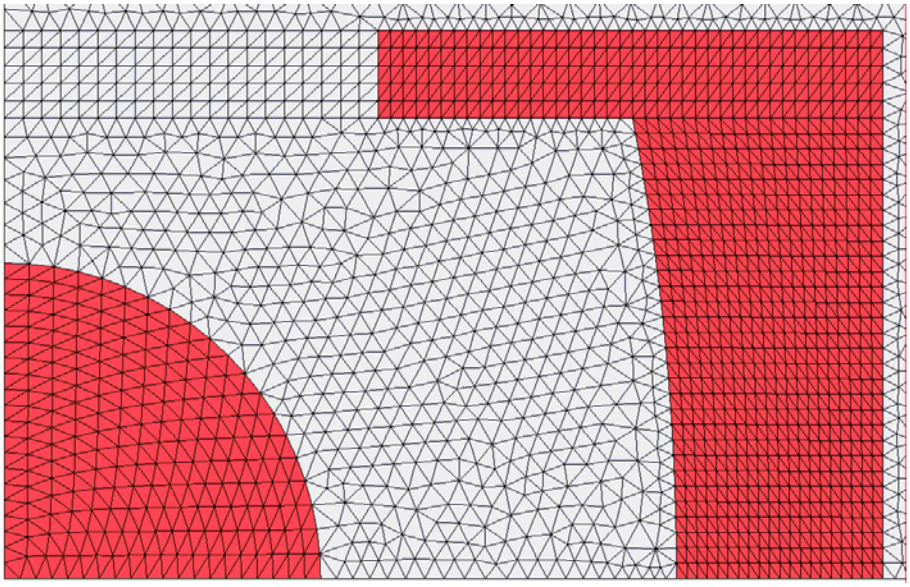


Fig. 5.19 Mesh on initial shape of die mold.

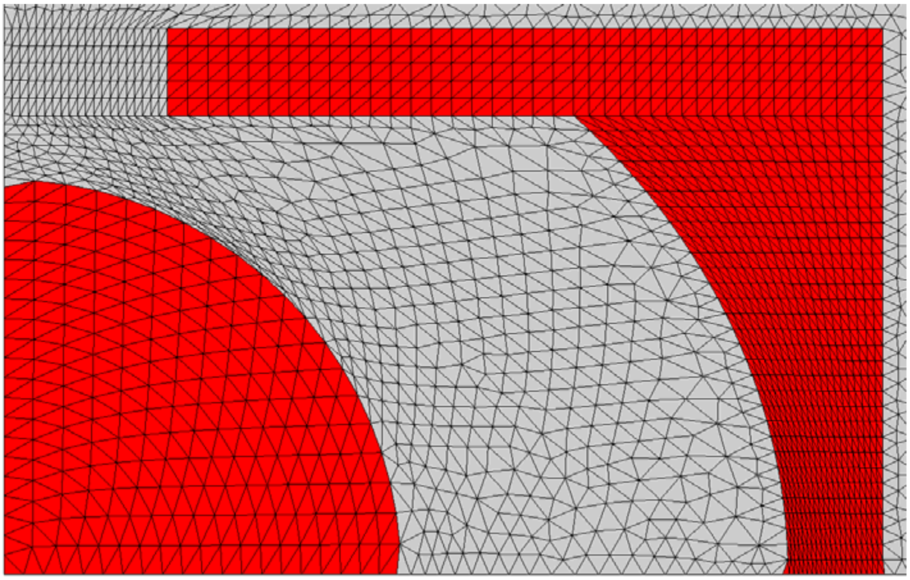


Fig. 5.20 Mesh on optimal shape of the problem.

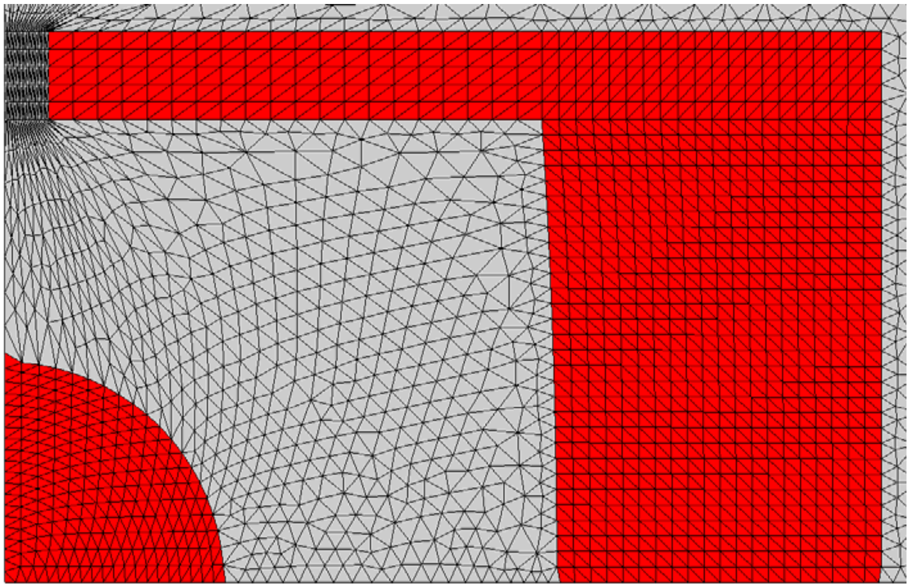


Fig. 5.21 Mesh on first case of extreme deformation of the die mold.

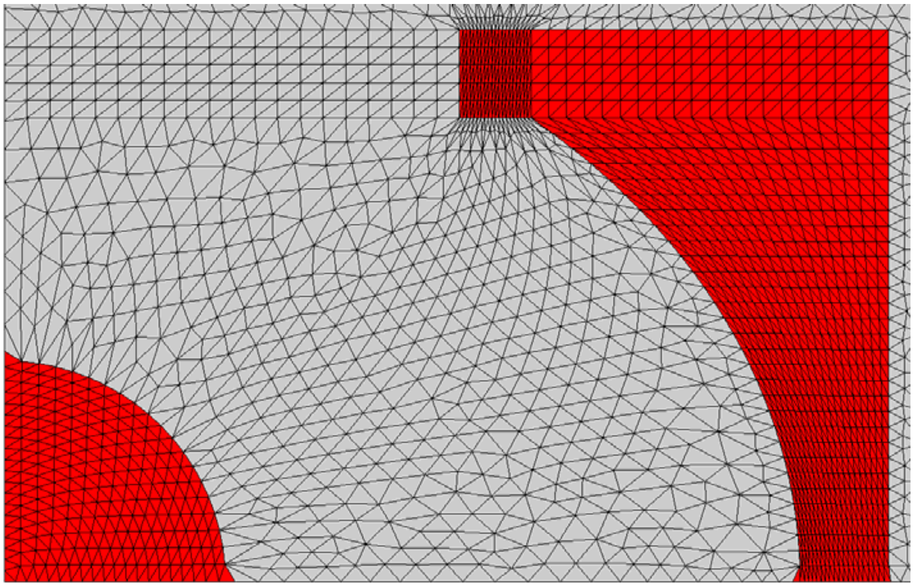


Fig. 5.22 Mesh on second case of extreme deformation of the die mold

6. APPLICATION OF THE COUPLED METHOD IN OPTIMIZATION OF ELECTROMAGNETIC MOTOR

6.1. REDUCTION OF COGGING TORQUE BY COUPLED GA AND FEM

6.1.1. INTRODUCTION

Uneven distribution of permanent magnets is an effective way to minimize the cogging torques of PM motors. A novel and efficient finite element method is proposed to calculate the cogging torque of PM motors by using coupling FEM with optimization method when the PM distribution is uneven.

Cogging torque is intrinsically an inherent feature of PM motors due to their salient geometrical structures. Because cogging torques impair the starting performance of motors, produces both noise and mechanical vibrations, it is important to reduce the cogging torque in PM motors. The use of uneven distribution of PM, as shown in Fig. 6.1, is one feasible way to reduce the cogging torques of PM motors [58] - [59]. It is however essential to determine exactly how and where each magnet should be shifted. Typically it takes one to two hours to evaluate the cogging torque of one design using general finite element method (FEM) and most optimization methods have to execute the object function many hundreds times before arriving at the optimal solution, several months are required to optimize the performance of PM motors.

To minimize the cogging torques by uneven distribution of the PM in optimization studies, the computing time to evaluate cogging torque must be reduced. In this paper an efficient FEM to compute the cogging torque is presented. The computing time of the proposed algorithm is only 0.06% of that required by general FEM. With the use of the proposed FEM, it is feasible to use genetic algorithm (GA) to optimize the PM distribution so as to minimize the cogging torque of the motor.

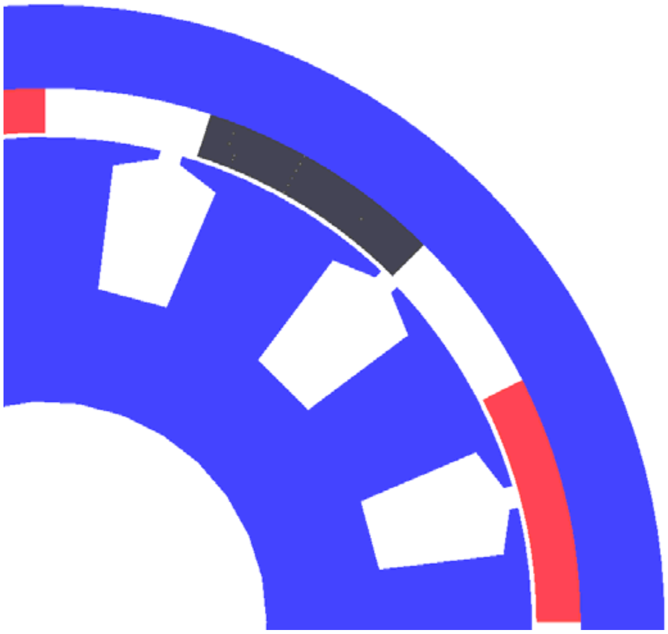


Fig. 6.1 Uneven distribution of PM in motor

6.1.2. Cogging Torque Computation by Using FEM

In the proposed method, a surface-mounted PM motor is investigated, and a 2-dimensional (2D) FEM model of the motor is used. The cogging torque can be computed according to the magnetic field computation when there are no armature currents. Maxwell's equations will give rise to the diffusion equation (6.1) as follows,

$$\left.\begin{aligned} &\nabla\times\left(\frac{1}{\mu}\nabla\times\boldsymbol{A}\right)=0 && \textit{in irons and air gap} \\ &\nabla\times\left(\frac{1}{\mu_{PM}}\nabla\times\boldsymbol{A}\right)=\nabla\times\left(\frac{1}{\mu_{PM}}\nabla\times\boldsymbol{M}\right) && \textit{in permaent magnets} \end{aligned}\right\} \tag{6.1}$$

where, $\boldsymbol{A}$ stands for the magnetic vector potential, μ is the permeability of the materials, $\boldsymbol{M}$ is the remanent flux density of the PM, μ_{PM} is the permeability of the PM. The weak form of equation (6.1):

$$\left.\begin{aligned} &\nabla\times\left(\frac{1}{\mu}\nabla\times\boldsymbol{A},\nabla\times\boldsymbol{N}\right)=0 && \textit{in irons and air gap} \\ &\nabla\times\left(\frac{1}{\mu_{PM}}\nabla\times\boldsymbol{A},\nabla\times\boldsymbol{N}\right)=\nabla\times\left(\frac{1}{\mu_{PM}}\nabla\times\boldsymbol{M},\nabla\times\boldsymbol{N}\right) && \textit{in permaent magnets} \end{aligned}\right\} \tag{6.2}$$

where $\boldsymbol{N}$ is the shape function. For cogging torque computation, the nonlinear characteristic of the permeability is negligible. Equation (6.2) can therefore be expressed as linear equations:

$SU = F$, $U = \left(A_1, A_2, \ldots\right)^T$, A_i *is the value of A on the node i*

where, S stands for the stiff matrix, U is the solution of the magnetic potential $\boldsymbol{A}$, and F is the load.

The electromagnetic torque can be computed using Maxwell stress tensor method [60]. The force on the stator can be expressed as:

$$\boldsymbol{F} = L\oint_l \frac{\boldsymbol{1}}{\mu_0}(\boldsymbol{n}\cdot\boldsymbol{B})\boldsymbol{B} - \frac{\boldsymbol{1}}{2\mu_0}B^2\boldsymbol{n}dl \tag{6.3}$$

where, $\boldsymbol{n}$ stands for the normal direction, L is the axial length of the stator and rotor cores, $\boldsymbol{B} = \nabla\times\boldsymbol{A} = \frac{\partial A_z}{\partial y}\boldsymbol{i} - \frac{\partial A_z}{\partial x}\boldsymbol{j}$ is the flux density on the air surface, l is the length of any closed path in the air gap surrounding the stator. The cogging torque $T = F_{tan}\cdot R$, where F_{tan} is the tangential component of magnetic force $\boldsymbol{F}$, R is the distance from the center of the stator to the closed path l.

In the proposed FEM method in the section, every magnet is considered as being constituted by several radically and seamlessly jointed magnet slices as shown in Fig. 6.2.

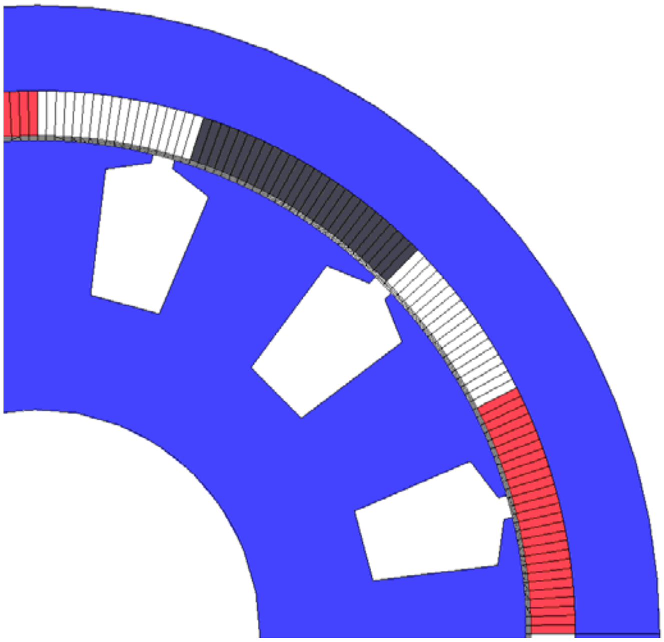


Fig. 6.2 Divide the PM to seamlessly jointed slices

Notice the two facts: 1) The cogging torque is computed when the nonlinear characteristic of the permeability is negligible, so $\mu_{iron} = const$; 2) For PMs in motor $\mu_{PM} \approx 0.9995\mu_0$, so μ_{PM} can be considered as equivalent to μ_0. That means if the PM is divided logically into several blocks, the magnetic vector potential and the flux density excited by the PM is the sum of the excitations by each PM block.

Note that $\boldsymbol{M} = \sum \boldsymbol{M}_{\boldsymbol{i}}$ and $\boldsymbol{M}_{\boldsymbol{i}}$ is the remanent flux density of each slice. So the load $F = \sum F_i$, where $F_i = \left(\dfrac{1}{\mu_{PM}} \boldsymbol{M}_i, \nabla \times \boldsymbol{N} \right)$. Notice that for the permanent magnets in the PM motor $\mu_{PM} \approx 1.0005\mu_0$, thus μ_{PM} can be approximately equal to μ_0. Hence if the magnetic vector potential $\boldsymbol{A}$ caused by two magnetic slices are required, it can be obtained by the following procedure:

(1) Calculate the magnetic vector potential $\boldsymbol{A}_1$ when slice 1 exists and the material on the position of slide 2 is air;

(2) Calculate the magnetic vector potential $\boldsymbol{A}_2$ when slice 2 exists and the material on the position of slide 1 is air;

(3) $\boldsymbol{A}=\boldsymbol{A}_1+\boldsymbol{A}_2$.

That is, the almost equivalence of the reluctance of PM and air makes the stiff matrix roughly the same when either PM or air is occupying the physical position being considered.

$$\left(\frac{1}{\mu}\nabla\times\boldsymbol{A},\nabla\times\boldsymbol{N}\right)=\left(\frac{1}{\mu_{PM}}\nabla\times\boldsymbol{A},\nabla\times\boldsymbol{N}\right), \text{ on the circle where PM is placed}$$

Equation (6.2) can be written as,

$$\left(\frac{1}{\mu}\nabla\times\boldsymbol{A},\nabla\times\boldsymbol{N}\right)=\sum_i\left(\frac{1}{\mu_{PM}}\boldsymbol{M}_i,\nabla\times\boldsymbol{N}\right) \tag{6.4}$$

Then,

$$\left(\frac{1}{\mu}\nabla\times\boldsymbol{A}_i,\nabla\times\boldsymbol{N}\right)=\left(\frac{1}{\mu_{PM}}\boldsymbol{M}_i,\nabla\times\boldsymbol{N}\right),\quad \boldsymbol{A}=\sum_i\boldsymbol{A}_i \tag{6.5}$$

$$S\sum_i A_i=\sum_i F_i,\quad \boldsymbol{B}=\nabla\times\boldsymbol{A}=\nabla\times\sum_i\boldsymbol{A}_i=\sum_i\boldsymbol{B}_i \tag{6.6}$$

That means the magnetic vector potential and the flux density excited by the PM is the sum of the excitations of each PM slice. If the basic solution excited by each slice is obtained, the principle of superposition applies. When the rotor rotates, the solution can also be obtained by a proper combination of the basic solutions; no further FEM is required. Thus only the magnetic vector potential and the flux density excited by each slice need to be computed and solved once only.

6.1.3. Implementation of the FEM Method

Implementation procedure of the proposed FEM method is:

(1) Divide the circle where the PM may be apportioned into *N* slices according to the required accuracy of the shift position of PM, so each slice occupies *360/N* degrees. The shift of the PM position in the optimization method should be an integer multiple of the width of the slice.

(2) Supposing that there is only one slice on the circle which is filled with PM and other spaces are filled with air, the magnetic vector potential and the flux density excited by this slice only is obtained by FEM. For each slice of PM on the circle, a solution of the vector potential ***A*** and the flux density ***B*** is obtained as the basis of the solution obtained by general FEM. In N equals 360, there are 360 different positions where the PM slice can be filled in, so only 360 solutions of the magnetic vector potential and their corresponding flux densities are obtained. All the solutions of the flux density for the computation of the cogging torque are then stored for subsequent computations. Fig. 6.3 shows the simulation result of the vector potential ***A*** when only one slice exists.

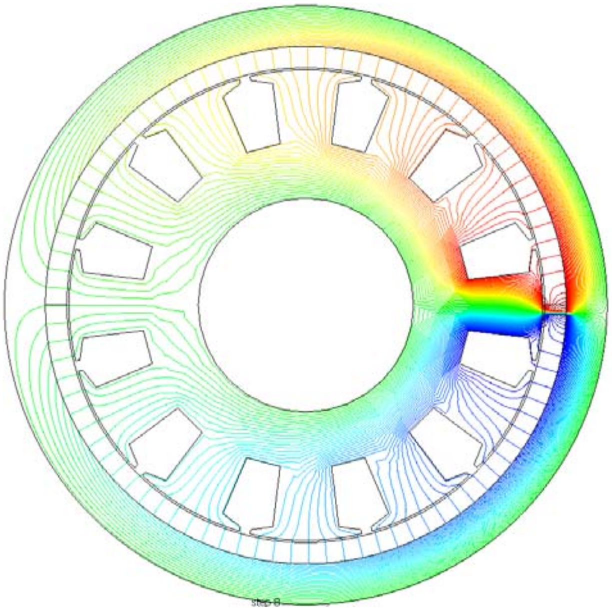


Fig. 6.3 Vector potential ***A*** when only one slice exists

(3) When the rotor rotates at a constant angular velocity *v*, the step time should be chosen as *2/N/v*, which means the rotor rotates exactly by one slice of PM width for each time step. In such case, one slice of PM will exactly fall into the position of the previous slice of PM after one time step. Hence the solution of this case can be a combination of the basis solutions.

(4) For any given distribution of the PMs, a combination of the slices can be selected to constitute the real case. When the combination of the slices for constituting the PM is known, it is not necessary to solve the linear equation $SU = F$, as the flux density $\boldsymbol{B}$ can be obtained by adding the stored result $\boldsymbol{B}_i$ according to the position of the PM (as the magnetization direction is radial, the difference between S-pole and N-pole is only on the sign). The cogging torque T can be computed according to all of the $\boldsymbol{B}_i$ values on the point of the closed path as shown by (6.3).

As a result, the computation time of cogging torque is decreased significantly using this method.

6.1.4. OPTIMIZATION PROCEDURE

Genetic algorithm (GA) is good for finding global optimal solutions [61] and it is exploited in the proposed optimization process for finding the distribution of PM so as to minimize the cogging torque of the motor.

The object function is chosen as $F = 5.0 - max(abs(T_i))$, where T_i is the cogging torque on each position when the rotor rotates through 360 degrees. The choice of the object function governs the minimum cogging torque produced.

The shift position of each PM is the variable to be optimized for the proposed algorithm.

For the 8-pole and 12-slot motor, the distribution of the PM needs to be optimized in order to minimize the cogging torque. The circle where the PM is placed is divided into 8 equivalent parts and each pole is in one part, as shown in Fig. 6.4. The position of the PM is described by polar coordinates and each PM occupies 27 degrees and air takes 18 degrees. In one part, there are 18 different positions where the PM can be filled, as shown in Fig. 6.5 and Fig. 6.6. If the PMs are getting too close to each other, the leakage of flux on one side of each PM becomes excessively high. In order to avoid such situation, only 16 different positions for each PM are allowed.

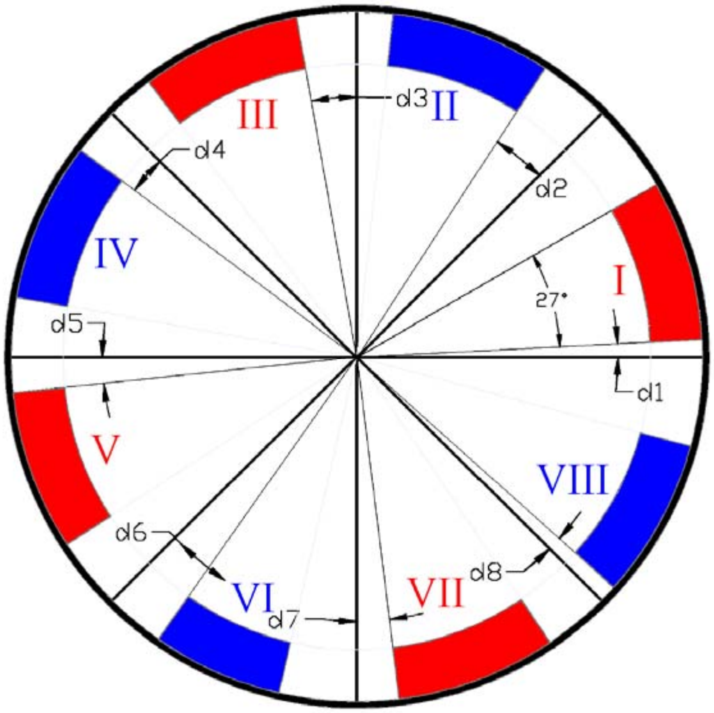


Fig. 6.4 Poles in 8 equivalent parts

The initial position of each PM is set as p_i = (i-1)·45 degrees (i=1,8), as shown in Fig. 6.4. Assume the position variable δ_i is the angle between the initial position p_i and the start position of one PM, as shown in Fig. 5, then 0 ≦ δ_1 ≦15 and δ_i is an integer because each slice of PM occupies 1 degree. Variables δ_i *(i=1,8)* are needed for the shift of 8 poles and δ_i shares the same characters of δ_1 .

Code and decode

In GA, the variable should be presented by a binary code. A 4-bit code can be used for the variable δ_i because $2^4 = 16$, which is the exactly the range of δ_i and 32 bits are needed for all the variables δ_i (i=1,8).

In the process of decoding, the position variable δ_i of the i^{th} PM is determined according to the 32 bits code being used. Firstly, the 32 bits code is divided by 8 groups and each group has 4 bits that represent the shift degree of one PM. The position variable δ_i of the i^{th} PM is the initial position p_i plus its binary code. For example, if the initial position $p_1 = 0$ and also if binary code for p_1 is 1010, then the position for the 1st PM is $\delta_1 = p_1 + 1010 = 8$ degrees.

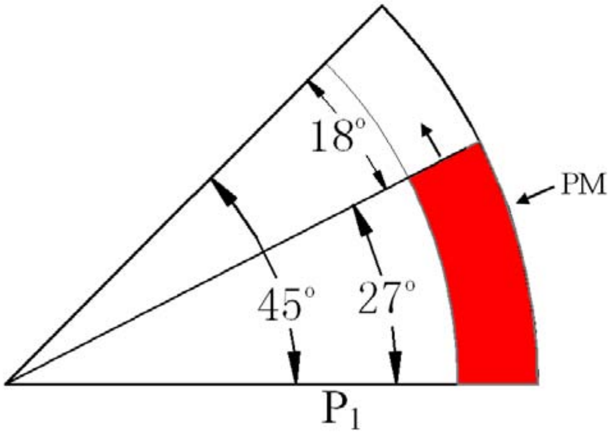


Fig. 6.5 Initial position of one PM

The main parameters to determine the general behavior of GA are as follows: the number of population is 100; the number of maximum generation is 200; the probability of crossover is 0.7 and the probability of mutation is 0.01.

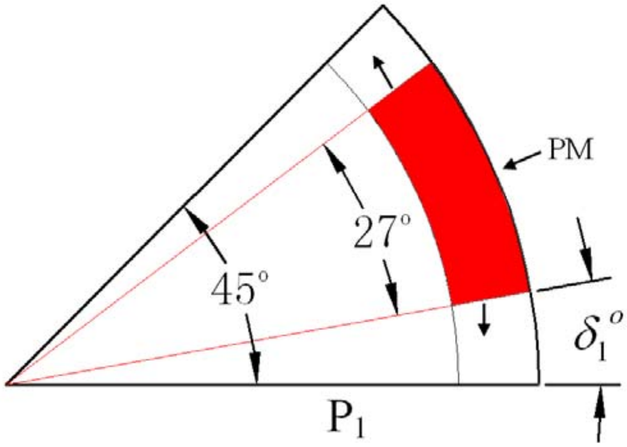


Fig. 6.6 Different position of PM

Fig. 6.7 shows the optimization procedure of GA.

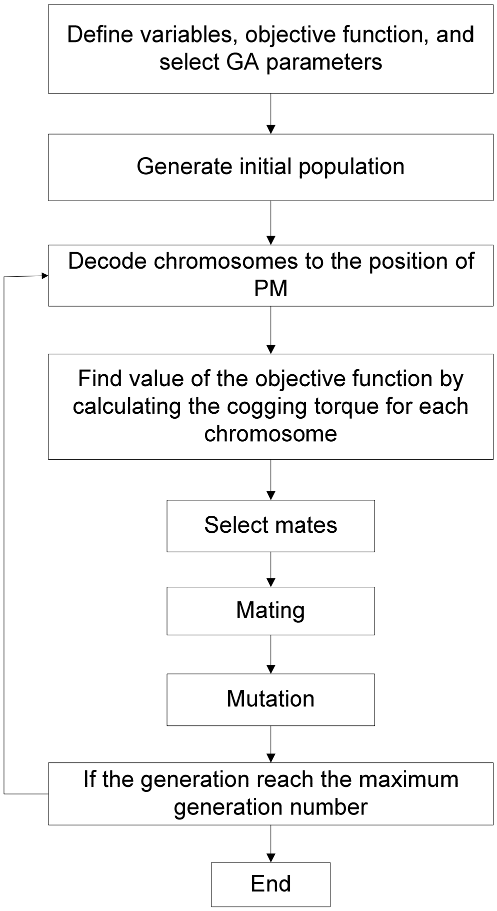

Define variables, objective function, and select GA parameters
Generate initial population
Decode chromosomes to the position of PM
Find value of the objective function by calculating the cogging torque for each chromosome
Select mates
Mating
Mutation
If the generation reach the maximum generation number
End

Fig. 6.7 Optimization procedure of GA

Fig. 6.8 shows the proposed method which has been implemented to a surface-mounted PM machine with 8 poles and 12 slots.

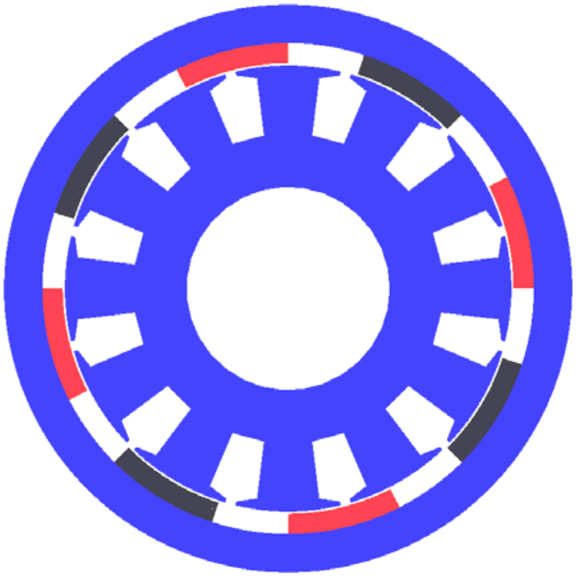


Fig. 6.8 Surface-mounted PM machine with 8 poles and 12 slots

The main parameter of the motor is listed in table I. The FEM discretization of the computation domain consists of 40559 nodes and 80006 triangle elements, as shown in Fig. 6.9 and Fig. 6.10.

TABLE 6-I MAIN PARAMETER OF MOTOR

Parameters	Value	Units
Pole number	8	
Slot number	12	
Stator outer diameter	110	mm
Stator inner diameter	50	mm
Rotor PM inner diameter	111	mm
Rotor core inner diameter	130	mm
Stack length	65	mm
Slot opening width	2.5	mm

Relative permeability of PM	0.99971	
Remnant flux density of PM	690015	A/m

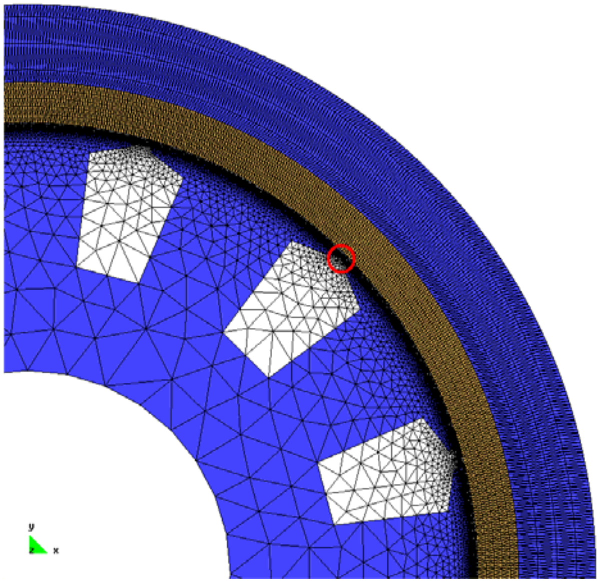


Fig. 6.9 The global view of mesh on 1/4 part of motor.

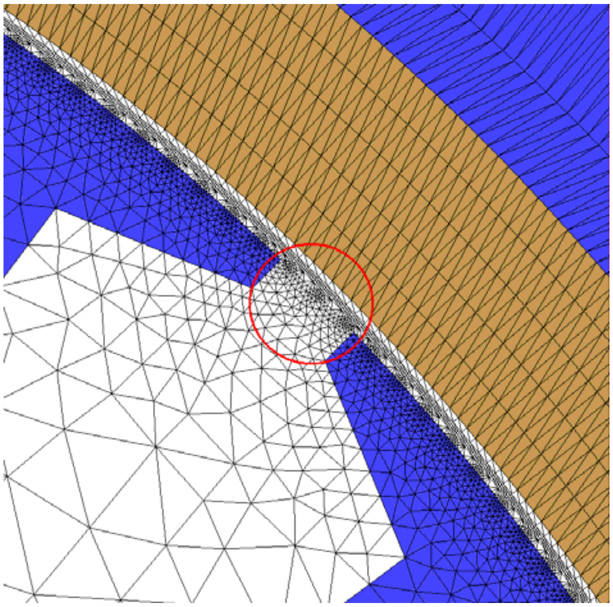


Fig. 6.10 The zoom in local view of mesh on the air gap

Because of the uneven distribution of the PM, the cogging torque T on each position as the rotor rotates through a complete cycle of 360 degrees needs to be computed. In the numerical experiment, it takes 9 seconds to obtain the torque of one time step. Therefore, it needs about 2.7 hours to compute all the time steps if general FEM is used. But it only takes 6 seconds to obtain the torque of all time steps using the proposed method. Numerical experiment also shows that the results obtained by the two methods are the same.

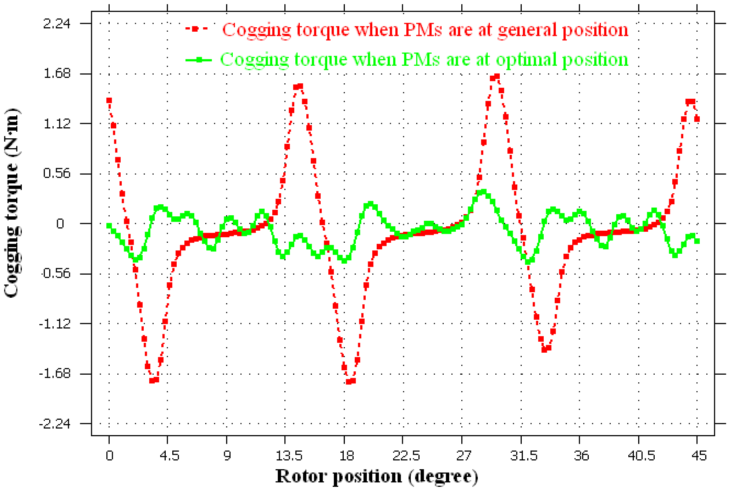


Fig. 6.11 Cogging torque when PMs are distributed at the general position and at the optimal position

The cogging torque is shown in Fig. 6.11 when the PMs are distributed at the same distance and at the optimized position. By using the FEM and GA, the peak cogging torque decreases from 1.768 N·m to 0.436 N·m. The position variable of each PM is given in Table II.

TABLE 6-II Position Variables of Each PM

PM No.	**I**	**II**	**III**	**IV**	**V**	**VI**	**VII**	**VIII**
Position variable	8	10	5	1	11	15	3	6

The position of the PM distributed at the same distance and at the optimized position is shown in Fig. 6.12.

The flux crossing the air gap is examined to check whether it will be reduced by the short distance between two PMs. Numerical experiment shows that in the process of the optimization, the average flux does not change significantly from a minimum of 1.296×10^{-3} Wb to a maximum of 1.327×10^{-3} Wb. When the PMs are at the general position and the optimal position, the average flux is 1.327 Wb and 1.324×10^{-3} Wb, respectively, indicating that the flux level is almost constant.

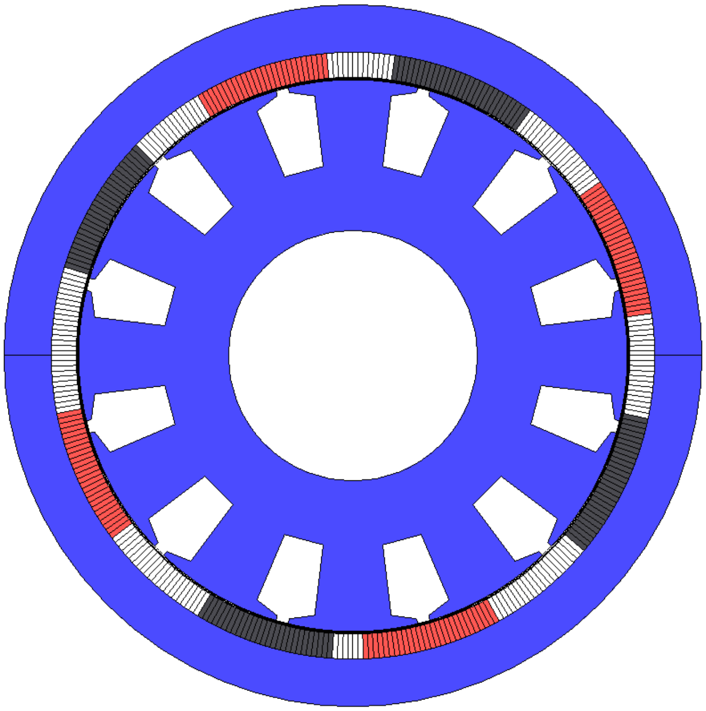


Fig. 6.12 The optimized PM positions

6.1.5. Optimization the Shape of PM to Reduce the Cogging Torque

Changes in the PM shape are reported as an efficient way to reduce the cogging torque [62]. In the section the same algorithm as proposed in section 6.1.2 is applied to minimize the cogging torque of surface-mounted PM (SPM) brushless motor by optimizing the surface profile of permanent magnet. The air gap between the stator and rotor of a surface-mounted PM motor is non-uniform in the radial direction but uniform in the axial direction, so a 2-dimensional model of the motor is needed. The cut section of the surface of the PM is not a circular arc but an arbitrary curve to be optimized. The optimal shape of the PM surface is to minimize the cogging torque of the motor. The couple FEM and optimization method is applied, which requires only 0.017% of CPU time when compared to that required by general FEM, to precisely compute the cogging torque.

In the section, the boundary of PM is approximated by fold line other than smooth curve, as shown in Fig. 6.13.

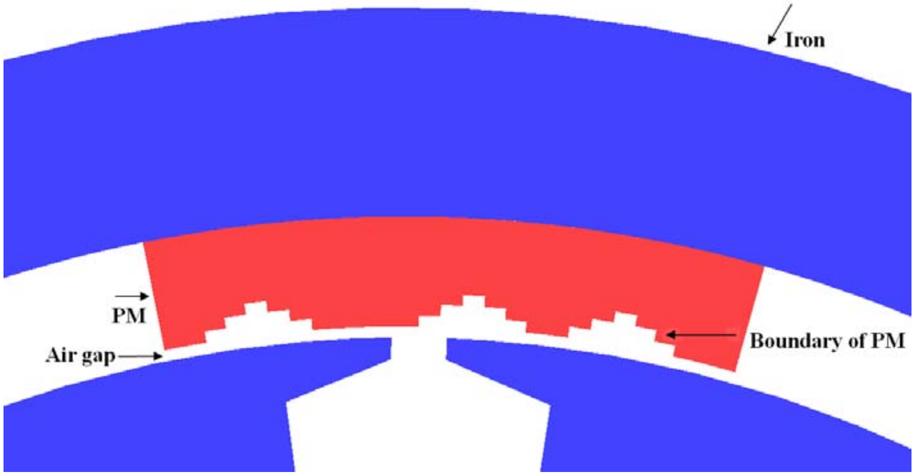


Fig. 6.13 Fold line to approximate the boundary of PM

The region where PM is fitted conventionally is divided logically into two parts at the radiate direction: a fixed PM region, and a flexible PM region, and divided logically into 60 parts at the circumferential direction, which means one part occupies 6 degree, as shown in Fig. 6.14 and Fig. 6.15. The flexible PM region is divided logically into some parts at the radiate direction further and so many small PM blocks will be obtained in the flexible region. In the PM shape design process, the fixed PM region will not be changed and some blocks in the flexible PM region will be changed into air to constitute the real shape of PM.

Hence if the magnetic vector potential $\boldsymbol{A}$ caused by two magnetic blocks are required, it can be obtained by the following procedure:

(1) Calculate the magnetic vector potential $\boldsymbol{A}_1$ when block 1 exists and the material on the position of block 2 is air;

(2) Calculate the magnetic vector potential $\boldsymbol{A}_2$ when block 2 exists and the material on the position of slide 1 is air;

(3) $\boldsymbol{A} = \boldsymbol{A}_1 + \boldsymbol{A}_2$.

That means the magnetic vector potential $\boldsymbol{A}$ and the flux density $\boldsymbol{B}$ excited by the PM are the sum of the excitations of each PM block. If the basic solution excited by each block is obtained, the principle of superposition applies. When the rotor rotates, the solution can also be obtained by a proper combination of the basic solutions; no further FEM is required. Thus only the magnetic vector potential and the flux density excited by each slice need to be computed and solved by FEM once only.

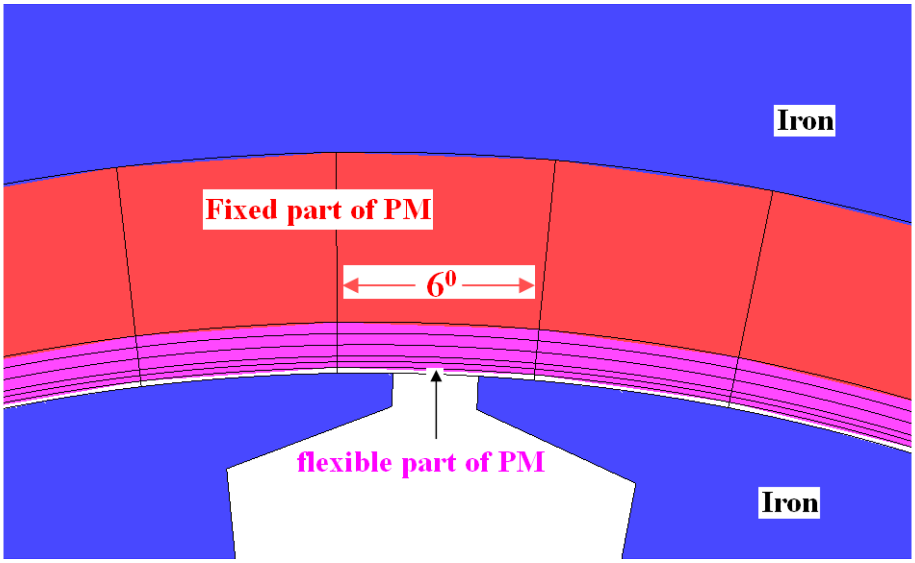


Fig. 6.14 Dividing the PM into two part logically

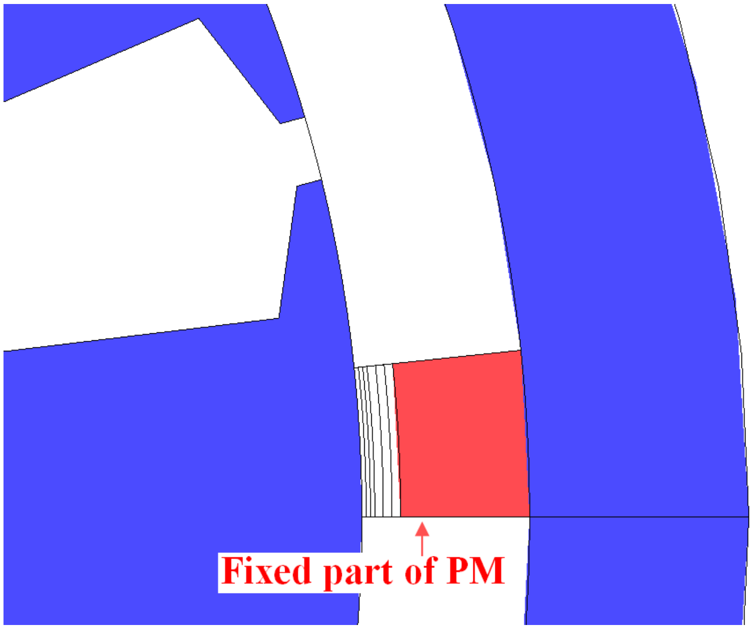


Fig. 6.15 One PM block in fixed PM region

The following steps are shown how to get the basis solutions. First, obtain four basis solutions of the magnetic vector potential $\boldsymbol{A_i}$ and the excitation flux density $\boldsymbol{B_i}$ by:

(1) one PM block in the fixed PM region, as shown in Fig. 6.16(a);

(2) one PM block in the fixed PM region and one PM block in the flexible PM region, as shown in Fig. 6.16(b);

(3) one PM block in the fixed PM region and two PM blocks in the flexible PM region, as shown in Fig. 6.16(c);

(4) one PM block in the fixed PM region and three PM blocks in the flexible PM region, as shown in Fig. 6.16(d).

By using general FEM and at the same time assuming that the other region where the PM occupies is filled with air, a solution of magnetic vector potential $\boldsymbol{A}$ of case (4) is obtained as shown in Fig. 6.17.

Secondly, let the four cases of PM blocks rotate 0.5 degree anticlockwise to the new position and then obtain another four basis solutions by FEM. By doing the same procedure, after the four cases of PM blocks rotate 360 degree, 4x720=2880 basis solutions of magnetic vector potential and the flux density are obtained.

Thirdly, find all the points on the closed path l which is the integral path to compute cogging torque, as shown in (6.3). Then pick out the flux density on those points from the basis solutions and store for further calculation.

For a given design shape of the PM, as shown in Fig. 6.13, a combination of the four cases of PM blocks, as shown in Fig. 6.16 (a)-(d) and their rotation cases can be selected to constitute the PM. When the combination of the blocks for constituting the PM is known, it is not necessary to solve the linear equation $SU = F$, as the flux density $\boldsymbol{B}$ can be obtained by adding the stored result $\boldsymbol{B}_i$ according to the position of the PM (as the magnetization direction is radial, the difference between S-pole and N-pole is only on the sign). By adding the corresponding basis solutions, the solution flux density of that design can be obtained.

When the rotor rotates 0.5 degree in one time step, the solution is the same as the that when the PM rotates 0.5 degree. Thus the solution can be obtained by adding the corresponding basis solutions also. When the rotor rotates one circle, the cogging torque T can be computed according to all of the $\boldsymbol{B}_i$ values on the point of the closed path l as shown by (6.3).

As a result, the computation time of the cogging torque with the use of the proposed methodology is decreased significantly.

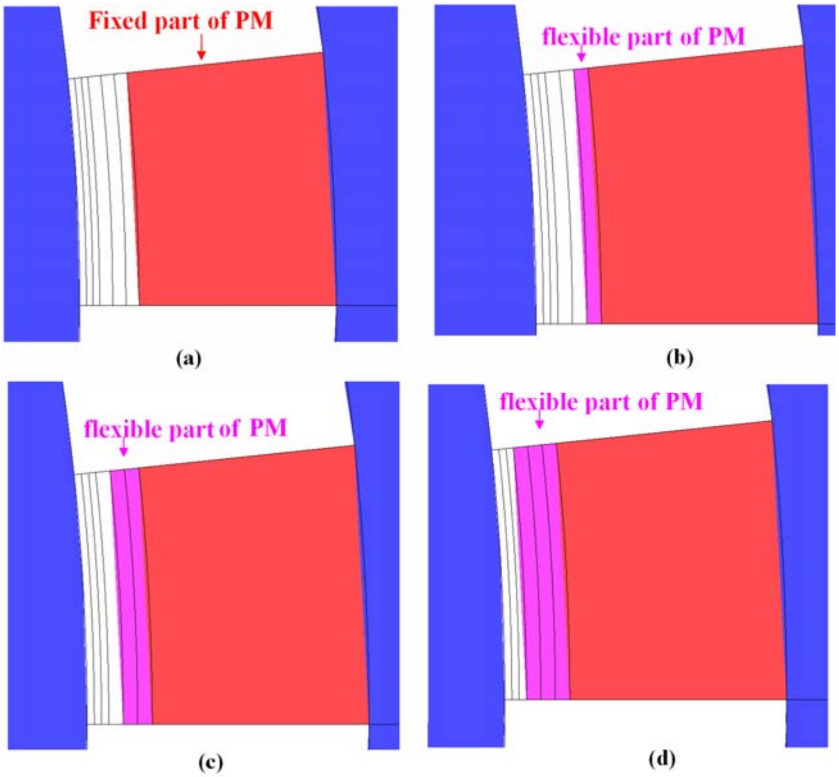


Fig. 6.16 (a) One PM block in the fixed PM region. (b) One PM block in the fixed PM region and one PM block in the flexible PM region. (c) One PM block in the fixed PM region and two PM blocks in the flexible PM region. (d) One PM block in the fixed PM region and three PM blocks in the flexible PM region.

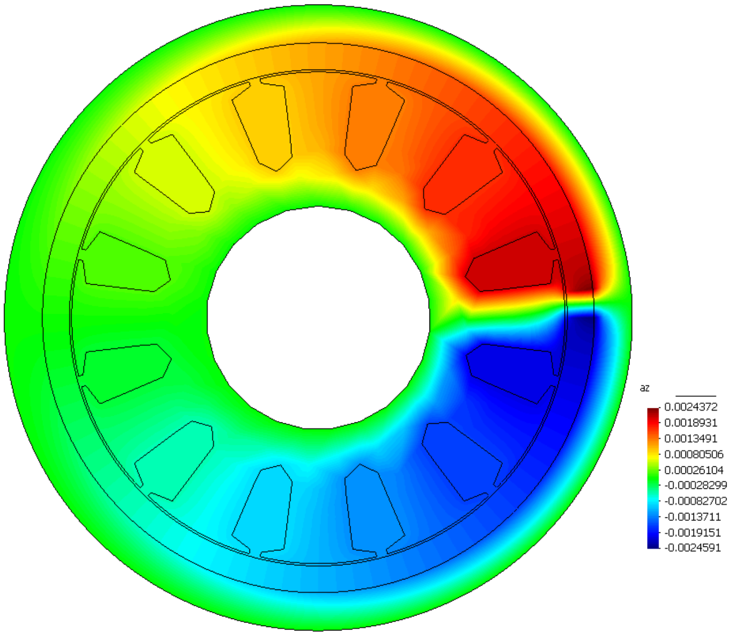


Fig. 6.17 Solution of magnetic vector potential.

Then by adding the corresponding basis solutions, the solution flux density of that design can be obtained.

GA is applied in the optimization process to find the distribution of PM to minimize the cogging torque. The objective function is chosen as $F = (flux_i/L)^8 \bullet 10^{12} \bullet nu^{0.1}/max(abs(T_i))$. T_i is the cogging torque on each position when the rotor rotates through 360 degrees, $flux_i$ is the flux crossing the air gap at the position when T_i reaches its maximum, L is the axial length of the stator and rotor cores and nu is the total number of flexible PM blocks. The flux crossing the air gap is calculated in the objective function to assure the flux is not be unduly reduced by changes in the PM shape. The total number of flexible PM blocks in the objective function ensures the number of flexible PM blocks will not be reduced too much. The choice of the objective function governs the minimum cogging torque produced.

The number of flexible block n, as shown in Fig. 6.16 in each PM block is chosen to be the variable for the optimal algorithm and n can be 0, 1, 2, 3 in each PM block. For the 4-pole motor, because each PM block occupies 6 degrees at the circumferential direction and each pole include 9 blocks, there are $4^{9\cdot4} = 2^{72}$ different kinds of PM shape. So a 72-bit binary code is needed.

The main parameters to determine the general behavior of GA are, namely, the number of population, 200; the number of maximum generation, 400; probability of crossover, 0.7 and probability of mutation, 0.05.

Fig. 6.18 shows the proposed method when the proposed algorithm is used to study the 4-pole, 12-slot SPM machine.

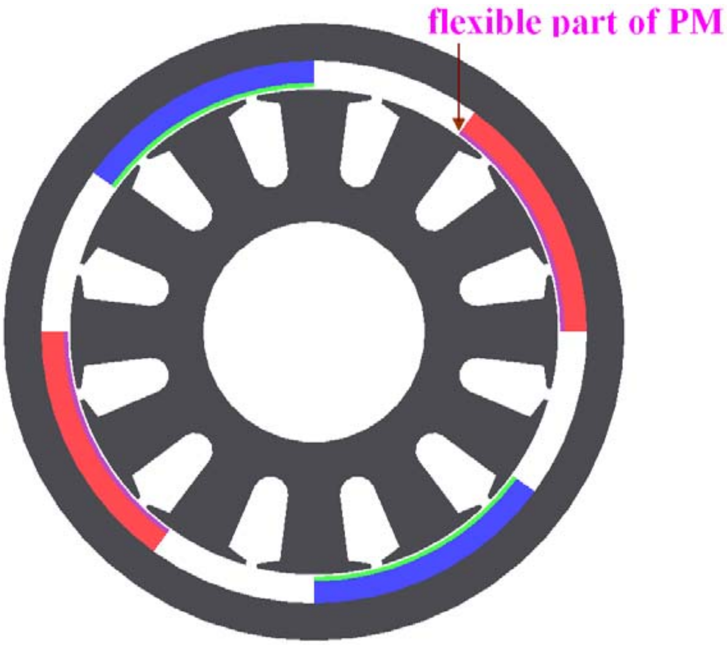


Fig. 6.18 Surface-mounted PM machine with 4 poles and 12 slots.

The main parameters of the motor are listed in TABLE 6-III.

TABLE 6-III MAIN PARAMETER OF MOTOR

Parameters	Value	Units
Pole number	4	
Slot number	12	
Stator outer diameter	110	mm
Stator inner diameter	50	mm
Fixed PM inner diameter	113	Mm

Flexible PM inner diameter	111	mm
Rotor core inner diameter	123	mm
Stack length	65	mm
Slot opening width	2.5	mm
Relative permeability of PM	0.99971	
remanent flux density of PM	690015	A/m
Pole embrace	0.6	

The FEM discretization of the computation domain consists of 21119 nodes and 41495 triangle elements, as shown in Fig. 6.19 and Fig. 6.20.

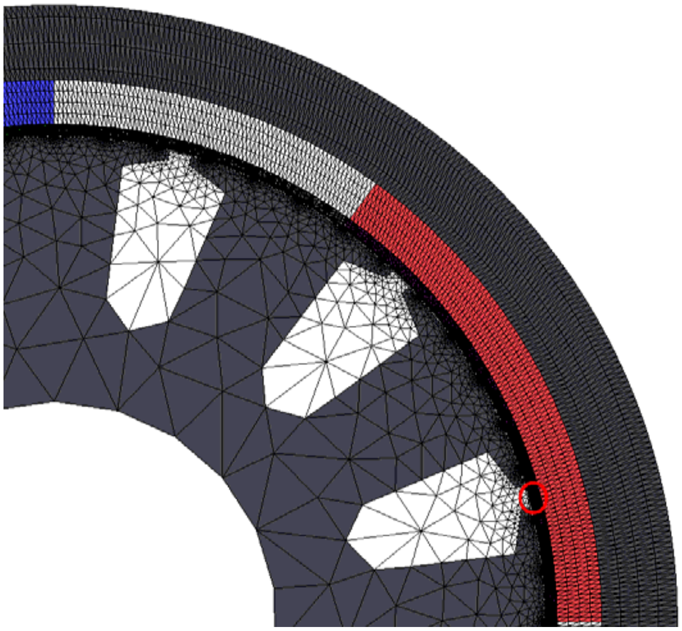


Fig. 6.19 The global view of the mesh on 1/4 part of motor.

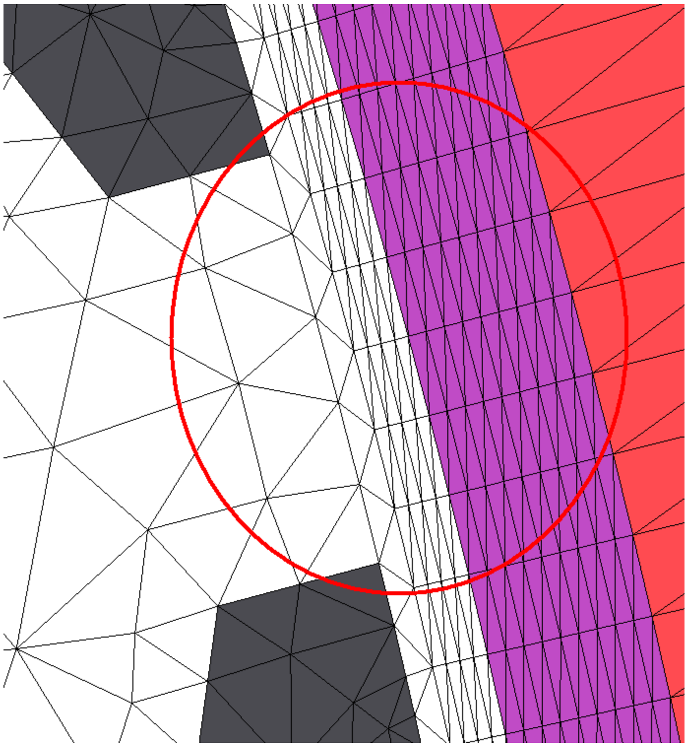


Fig. 6.20 Mesh on the zoon in local view of mesh on the air gap.

Because of the non-uniform air gap in the motor, the cogging torque T on each position as the rotor rotates through a complete cycle of 360 degrees needs to be computed. In the numeric experiment, it needs about 0.75 hour for computing 720 time steps if general FEM is used, however it needs only 0.45 second to obtain the motor torque using the proposed technique. Numerical experiment also shows that the results obtained by the two methods are the same.

In the numerical method, the cogging torque, flux and the quantity of the objective function with the full set of flexible PM blocks and without flexible PM blocks are compared and shown in TABLE 6-IV. The result shows the quantity of objective function of the former is bigger than that of the later, which means the quantity function encourages the use of more flexible PM blocks and therefore to ensure the flux crossing the air gaps are virtually unchanged.

TABLE 6-IV COMPARISON BETWEEN FULL OF AND NO FLEXIBLE PM BLOCKS

	Cogging torque (N·m)	*flux/L* (Wb/m)	Quantity of objective function
Full set of flexible PM blocks	1.770	0.0432	11.026
No flexible PM blocks	0.816	0.0360	3.485

The cogging torques when the PMs are of the general shape and at the optimized shape are shown in Fig. 6.21. By using the FEM and GA, the peak cogging torque decreases from 1.770 Nm to 0.716 Nm. The value of *flux/L* of the former and latter is 0.0432 and 0.0408, respectively. The optimal shape of each PM represented by the number of flexible PM blocks is given in table III.

Table. III
OPTIMAL SHAPE OF EACH PM REPRESENTED BY THE NUMBER OF FLEXIBLE PM BLOCKS

Block	I	II	III	IV	V	VI	VII	VIII	IX
A	1	3	1	3	3	3	3	2	1
B	0	2	3	1	3	2	1	3	2
C	1	3	2	1	3	2	1	2	1
D	0	3	3	3	3	3	2	2	0

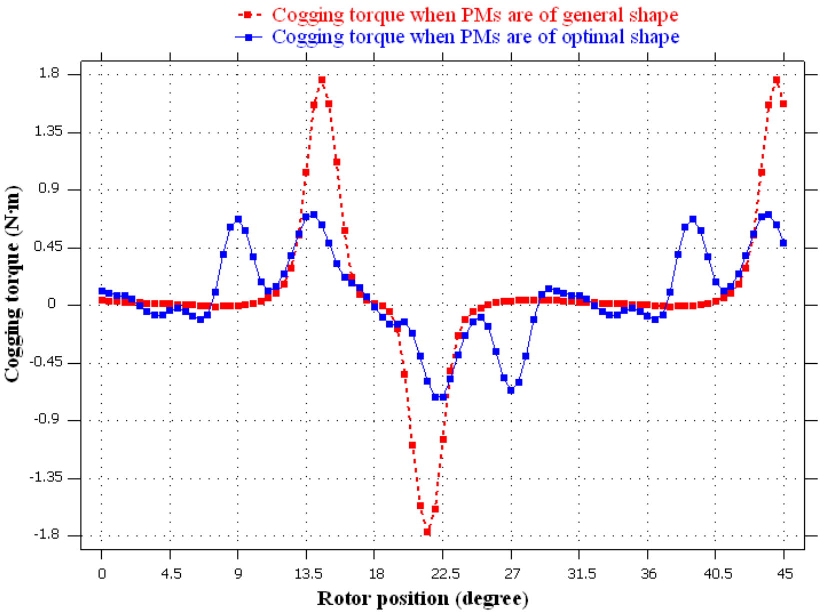


Fig. 6.21 Cogging torque when PMs are of the general shape and optimal shape.

6.1.6. SUMMARY

In this chapter, an efficient FEM based on finding the basis solutions of the magnetic field of PM slices and then combined to constitute the solutions for arbitrary PMs is proposed. It is then used to optimize the distribution of PMs in a surface-mounted PM motor and to optimize the shape of PMs in a surface-mounted PM motor so as to reduce the cogging torque. The computing time of the proposed FEM method for solving this problem is significantly reduced to only 0.06% of that required by using general FEM. GA is used in the optimization and after optimizing the position of the PMs in the motor. The maximum cogging torque is reduced to about 24.7% of that produced by a motor having evenly distributed PMs. Numerical experiment also shows that when the PMs are placed at the optimal position, the average magnetic flux is almost the same as that in a motor having the PM located in a general position.

6.2. OPTIMIZATION FOR MAGNETIC GEARS

6.2.1. INTRODUCTION

Permanent magnet materials can be used to produce a constant magnetic field and exert magnetic force without any energy consumption. They are widely used in numerous industry applications. However, with the skyrocketed price of rare earth PM materials, how to efficiently use the PMs and reduce the cost of the PM associated products becomes a practical and urgent problem to be resolved [63][64].

Magnetic gears (MGs) are one class of magnetic devices to transmit torque between two rotating parts with different magnetic pole pairs through flux modulation poles [65][66]. The reason why MGs gather more attention than ever before is that their contactless operation can effectively reduce mechanical loss and acoustic noise which is inevitable and harmful in mechanical gears. Unfortunately, under the worldwide shortage of rare earth PM supply, excessively using PM materials in MGs and PM motors leads to a severe increase in the cost of the products. Hence, it is necessary and important for researchers to find approaches to optimize the design of PM gears and improve the utilization rate of PM materials.

Finite element method has been successfully used in the design of electric devices and serves as indispensable tools in the static and dynamic performance analysis of electromagnetic devices. Compared with analytical method, the advantages of FEM are its applicable to any complicated geometrical structures, its high accuracy and reliability. However, when dealing with optimization problems, the numerical computation is time-consuming, because new meshes need to be regenerated repeatedly according to the variations of geometrical parameters of designs.

A PSO based mesh adjusting optimization approach is applied in the MG design to maximize the torque output with a limited usage of PMs. The coordinates of the meshes are adjusted according to our proposed dimension modification method, which is introduce in chapter 5. The dimension displacements are determined by the PSO algorithm. Consequently, the time-consuming geometry rebuilding and mesh regeneration process are avoided and a global optimum is reached at a relatively fast speed of convergence.

The design optimizations of a SPM MGs are presented, which realizes with the defined utilizations of PMs to reach a maximum of torque output. Optimal results confirm the proposed method. The optimized designs can effectively use the PM materials, as well as improve the torque density.

6.2.2. CONFIGURATION AND OPTIMIZATION PROBLEMS

Fig. 6.22 shows the constructions of the SPM MGs to be optimized. In this MG, the pole pair number of NdFeB PMs in the high speed rotor is $p_h = 4$ and that in the low speed rotor is $p_l = 22$, and the flux modulation pole number is $p_p = 26$. Consequently, the gear ratio is $G_r = p_l / p_h = -5.5$. It means that if the flux modulation poles are stationary and the high speed rotor rotates in positive direction, the low speed rotor will rotate in negative direction. The outside diameters of the MG are designed as $r_m = 126$ mm, the shaft diameter is $r_n = 30$ mm and the stack length $l = 50$ mm. The volume of PMs in the MG is set as 0.1125 m^3. Each of airgap length is 1 mm. These values are kept unchanged during the optimization process. According to [67], the torque on the low speed rotor can be expressed as:

$$T_l = \frac{lB_{rl}B_{rh}h_{mo}h_{mi}P_0P_1}{2\mu_0}\cos\left(p_h\theta_0^h - p_l\theta_0^l\right) \tag{6.7}$$

where, h_{mo} and h_{mi} are the thicknesses of the outer PMs and the inner PMs, respectively; B_{rl} and B_{rh} are the remanences of the outer and inner PMs; θ_0^h and θ_0^l are the initial position of low speed and high speed rotor. The magnetic permeance in radial direction is expressed as:

$$P(\theta) = P_0 + \sum_{k=1,2,3\cdots} P_k \cos\left(kp_p\theta\right) \tag{6.8}$$

where P_0 and P_1 are the coefficients of permeance. The width of slot is set to be equal to that of flux modulating pole to get the satisfactory modulation effect [67]. From (6.7), it is illustrated that the torque output is directly related to the dimensions of PMs and the flux modulating poles as well as the initial position of the rotor.

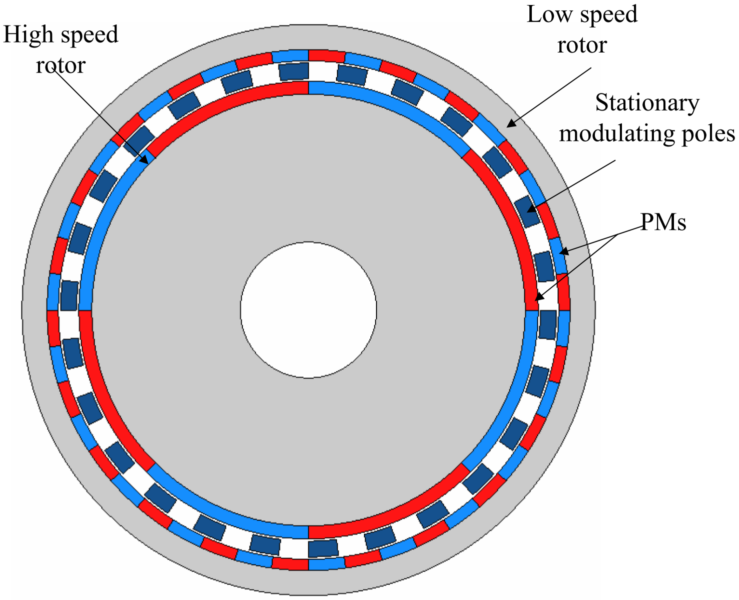
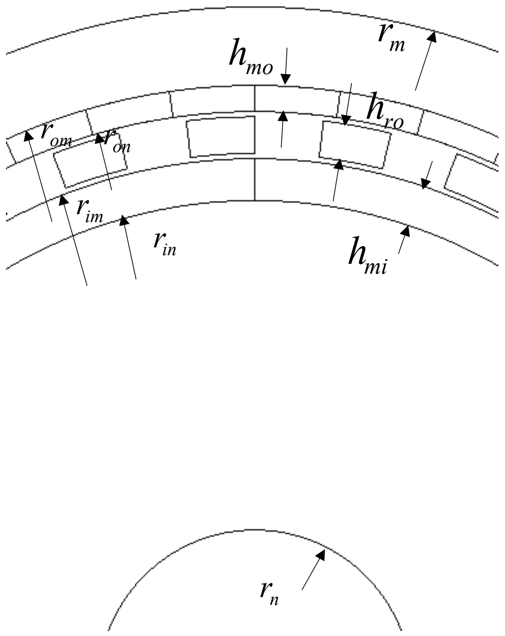


Fig. 6.22 The structure of MG.



(b)

Fig. 6.23 Design parameters of MG.

With the initial dimension of the MG, the static torque is obtained by FEM with the high rotor rotating at 150 rpm and the low rotor and the flux modulating poles stationary. The flux line distribution is shown in Fig. 6.24. The torque waveforms are shown in Fig. 6.25 and it is verified that the torque ratio is -5.5 almost at any position which is consistent with the principle of MG. It is shown that when the MG dimensions are predetermined, there is one position at which the output torque is the maximum. Then at this postion, the radial dimensions of PMs and modulating poles are need to be optimized to get the maximum value of the torque.

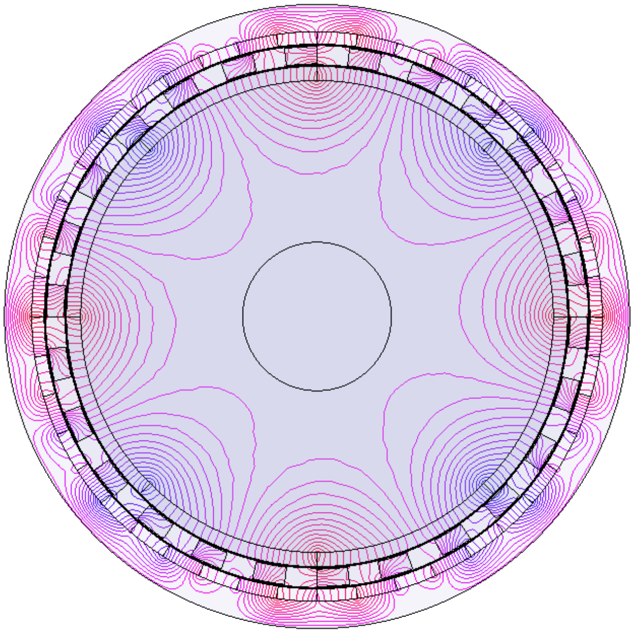


Fig. 6.24 Magnetic flux distribution in magnetic gears.

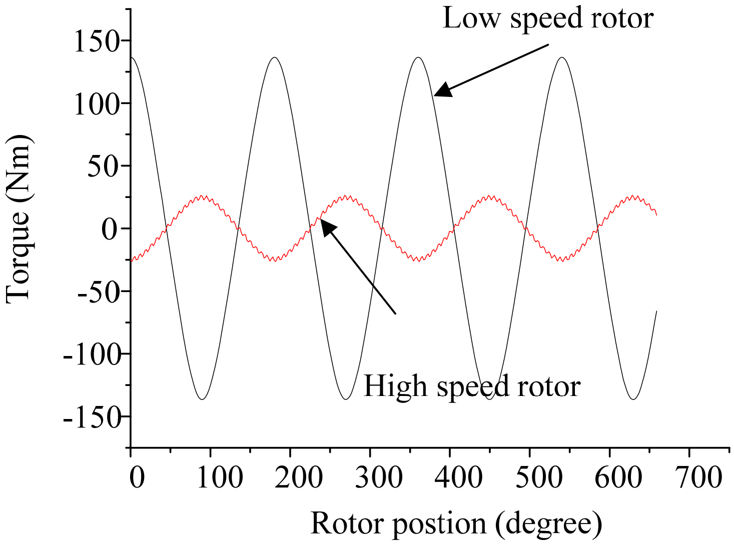


Fig. 6.25 Static torque waveforms.

6.2.3. Optimal Problem

The design constrains are that outer radius $r_m = 126\,\text{mm}$ and the area of PMs are $S_{pm} = 2250\,\text{mm}^2$. The design variables can be expressed as:

$$r_{om} = r_{01} + \Delta a + \Delta k \tag{6.9}$$

$$r_{on} = \sqrt{(r_{01} + \Delta a)^2 - S_{pm}/2} \tag{6.10}$$

$$r_{im} = r_{02} + \Delta b \tag{6.11}$$

$$\begin{aligned} r_{in} &= \sqrt{r_{im}^2 - \left[S_{pm} - \left(r_{om}^2 - r_{on}^2\right)\right]} \\ &= \sqrt{(r_{02} + \Delta b)^2 - S_{pm}/2 + (r_{01} + \Delta a + \Delta k)^2 - (r_{01} + \Delta a)^2} \end{aligned} \tag{6.12}$$

where, r_{om}, r_{on} are outer and inner radius of outer PMs and r_{im}, r_{in} are that of inner PMs. Assuming the initial PM areas in lower speed rotor and higher speed rotor are equal, r_{01} and r_{02} are initial dimensions of outer radius of inner and outer PMs. Δa and Δb are the adjustment of outer radius of outer PMs and inner PMs, while keeping the PM area unchanged; Δk is the additional adjustment of outer radius of outer PMs to change the PM area in inner and outer rotors. In this paper, the initial values are $\Delta a = \Delta b = \Delta k = 0$, $r_{om} = r_{01} = 115\,\text{mm}$, $r_{on} = 110\,\text{mm}$, $r_{im} = r_{02} = 101\,\text{mm}$ and $r_{im} = 95.26\,\text{mm}$. Since the airgap length is fixed, once the radii of inner and outer PMs are obtained, the whole dimensions of MG are determined. The optimal objective is simplified as:

$$\max T(x) = T(\Delta a, \Delta b, \Delta k) \tag{6.13}$$

6.2.4. PSO ALGORITHM

The PSO algorithm can be expressed as:

$$v_d^{k+1} = \omega_i v_i^k + c_1 r_1 \left(p_d^i - x_d^i(k)\right) + c_2 r_2 \left(g_d^i - x_d^i(k)\right) \tag{12}$$

$$x_d^i(k+1) = x_d^i(k) + v_d^i(k+1) \tag{13}$$

where, v_d^{k+1} and x_d^k are the velocity and position; g_d^i represents the best particle position among the entire population; c_1 and c_2 represent the acceleration coefficients; r_1 and r_2 are two uniform random functions; p_d^i is the previous best particle position; ω is the inertial weight to adjust the global and local optimization capability of the method.

In optimal problem of electromagnetic devices, the optimization variables are always within a certain range. Therefore, bound control should be imposed to ensure the variables are in the demanded range. If the position x_d^{k+1} of one particle is out of range, its speed is changed according to the following formular:

$$v_d^i(k) = sign(r_3) \cdot \alpha \cdot v_d^i(k) \qquad (14)$$

where, $\alpha < 1$, r_3 is a ramdom number in [0,1], *sign* is the symbolic function, which is defined as:

$$sign(r) = \begin{cases} 1 & (\mathrm{r} \geq 0.5) \\ -1 & (\mathrm{r} < 0.5) \end{cases} \qquad (15)$$

Then the new position x_d^{k+1} is updated according to (13).

The PSO method is a stochastic searching and optimization algorithm, in which the trajectory of each particle is gracefully adjusted towards its own best position and the global best position discovered by its neighbors, as well as the whole swarm [69][70]. Since the PSO method has the capability in searching the global optimum with high probability and its convergence rate is fast, it is advantageous to be combined in FEM to determine the dimension changes of MGs.

6.2.5. RESULTS

This mesh adjusting with weighted Laplacian smoothing method is convenient and effective to implement in the design optimization of MGs. During the optimization, the initial mesh of the MG mentioned is shown in Fig. 6.26, while the mesh changing in the MG during the optimal procedure is shown in Fig. 6.27. It is shown that, the mesh quality is still high, which guarantees the accuracy of numerical calculation.

There are three variables which need to be adjusted in the optimization procedure, namely Δa , Δb and Δk . The geometry constraints are listed below: $\Delta a \in [-5.0\,\text{mm}, 5.0\,\text{mm}]$, $\Delta b \in [-5.0\,\text{mm}, 5.0\,\text{mm}]$, $\Delta k \in [-3.0\,\text{mm}, 3.0\,\text{mm}]$.

The final optimized results are obtained as $\Delta a = 3.1\,\text{mm}$, $\Delta b = -4.0\,\text{mm}$, $\Delta k = 1.1\,\text{mm}$. With the optimized dimensions, as shown in Fig. 6.28, the torque performance of MG is shown in Fig. 6.29. The torque is 147.9 Nm. Compared with the initial design, the outer diameter is unchanged, but the torque density of the machine is improved by 10.3%.

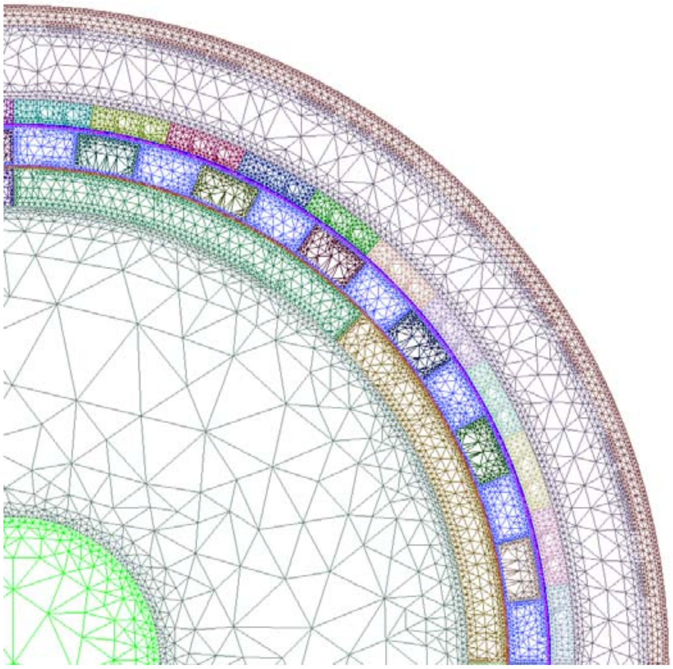


Fig. 6.26 Original mesh.

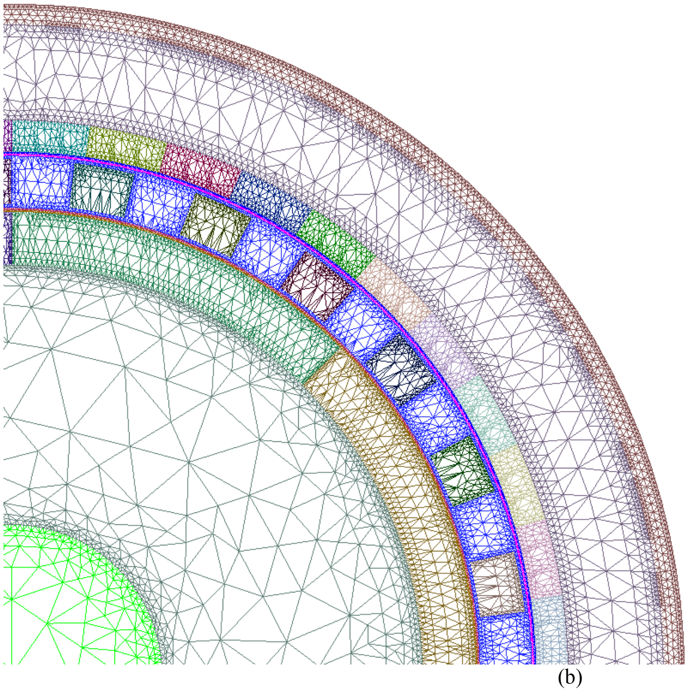


Fig. 6.27 Adjusted mesh result.

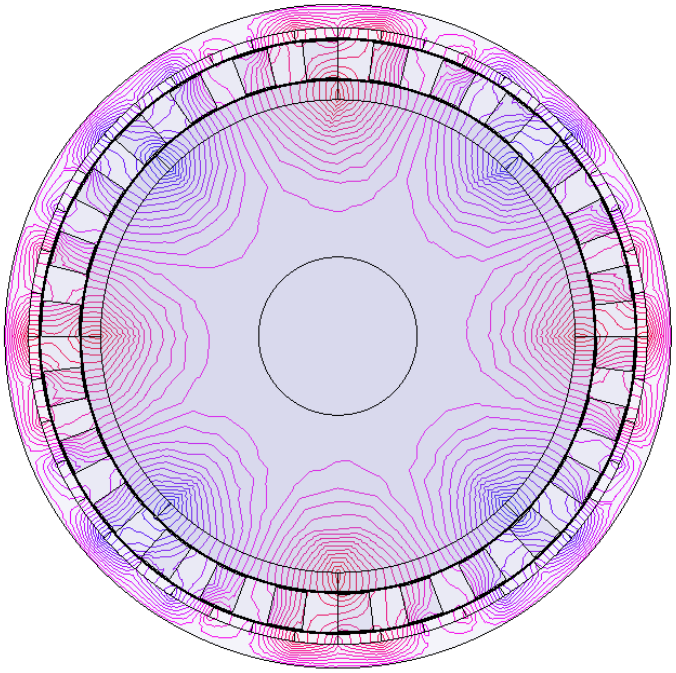


Fig. 6.28 The flux line results with the optimized dimensions.

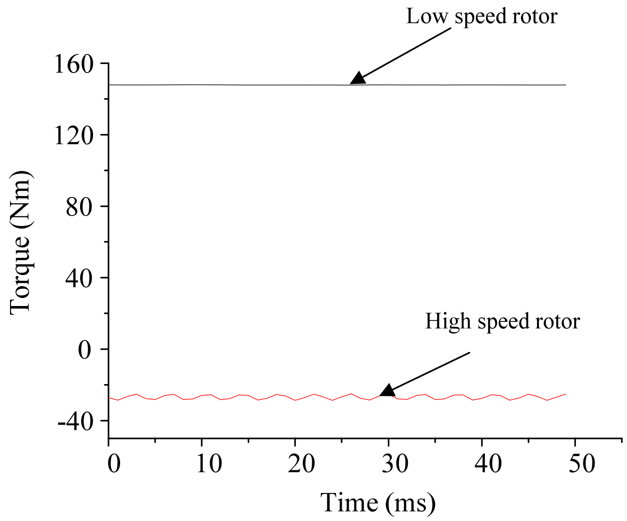


Fig. 6.29 The torque transient waveforms on the lower speed rotor while the lower speed rotor with -27.27 rpm and higher speed rotor with 150 rpm.

6.2.6. SUMMARY

A PSO based mesh adjustable finite element method is utilized to optimize a MG dimensions and the aim is to get the maximized torque output with defined utilization of PMs. With the mesh adjustable finite element algorithm, the coordinates of mesh vertexes can be moved according to the dimension changes, while the mesh quality can still kept high. The PSO algorithm guarantees a reliable convergence to the global optimum. With this method, the torque density of MG is improved by 10.3%. Optimal results confirm the validity and effectiveness of the proposed algorithm.

7. CONCLUSIONS AND RECOMMENDATIONS

In the thesis, three dimensional (3D) Finite element method (FEM) embedded global optimization method are applied to obtain optimal design electric devices. FEM with nodal basis function and edge basis function are used to analysis magnetic field and the eddy current problem in electric devices. Time step FEM with slave master technique, circuit coupling technique is applied to simulate the performance of electric motor. Global optimization method including Genetic algorithm (GA), simulated annealing (SA), taboo search (TA) and particle swarm optimization (PSO) are coupled with FEM program to find the parameters of the optimal design. Parameter extraction technique is applied to extract the mass parameters of electric motor to accelerate the computation speed of optimization process. Moving least square (MLS) based surface response model is applied to reduce the optimization time.

State-of-the-art programming techniques of implement of FEM by object oriented programming language for electromagnetic field computation are presented. It covers program structure, data structure and algebraic matrix equation. The advantages of the proposed program structure are that multi-developers are empowered to work on different solvers and share common algorithms.

Two-grid FEM algorithm is suggested to reduce the computation time and meanwhile conserves the computation accuracy of the simulation results. TEAM Workshop problem 13 is used to test the proposed method and the results obtained by using the proposed algorithm are compared with those using conventional methods. The computing time using the proposed method is only 75% of that by using the conventional method. Since the two-grid method requires less computing time, it can be effectively applied to study large-scale nonlinear problems.

In order to facilitate the procedure for obtaining a new mesh on the newly designed device defined by the optimized parameters and to reduce the time of regenerating mesh, moving mesh algorithm with fast solver is presented. TEAM Workshop problem 25 is used to test the resistance of mesh folding of the proposed algorithm and also to evaluate the computing time saved when using the proposed fast solver. The result shows that the moving mesh algorithm has successfully overcome the difficulty of data transfer when coupling FEA and optimization method.

An efficient FEM based on superposition principle of PMs is proposed. It is used to optimize the distribution of PMs in a surface-mounted PM motor and to optimize the shape of PMs in a surface-mounted PM motor so as to reduce the cogging torque. The computing time of the proposed FEM method for solving this problem is significantly reduced to only 0.06% of that required by using general FEM. A PSO based mesh adjustable finite element method is utilized to optimize a MG dimensions and the aim is to get the maximized torque output with defined utilization of PMs. With the mesh adjustable finite element algorithm, the coordinates of mesh vertexes can be moved according to the dimension changes, while the mesh quality can still kept high. The PSO algorithm guarantees a reliable convergence to the global optimum. With this method, the torque density of MG is improved by 10.3%. Optimal results confirm the validity and effectiveness of the proposed algorithm.

References

[1] H. Whitney, Geometric Integration theory. Princeton, NJ: Princeton University Press, 1957.

[2] J.C. Nedelec, “Mixed finite element in R^3,” *Numer. Meth*, vol. 35, pp. 315-341, 1980.

[3] M.L. Barton and Z. J. cendes, “New vector finite elements for three-dimensional magnetic field computation,” *J. Appl. Phys.*, vol 61, no. 8, pp. 3919-3921, April 1987.

[4] A. Bossavit and J. C. Verite, “A mixed FEM-BIEM method to solve 3-D eddy current problems,” *IEEE trans. Magn.*, vol. MAG-18, pp. 431-435, March 1982.

[5] K. Preis, I. Bardi, O. Biro, C. Magele, G. Vris, K.R. Richter, “Different finite element formulations of 3D magnetostatic fields,” *IEEE Trans. Magn.*, vol 28, No. 2, pp. 1056-1059, March 1992.

[6] 谢德馨, 三维涡流场的有限元分析, China Machine Press, 2007

[7] O. Biro and K. Preis, “On the use of the magnetic vector potential in the finite element analysis of 3-D eddy currents,” *IEEE Trans. Magn.* vol. 25, no. 4, pp. 3145-3159, 1989

[8] A. Bossavit, “How week is the "weak solution" in finite element methods,” *IEEE Trans. on Magn.*, vol. 34, no. 5, pp. 2429-2432, Sep. 1998.

[9] J.M. Jin, The finite element method on electromagntics, 2nd Edition, Wiley-IEEE Press, May 2002.

[10] X.D. Shi, Y. L. Menach, J.P. Ducreux, F. Piriou, “Comparison of slip surface and moving band techniques for modeling movement in 3D with FEM,” *COMPEL*, vol. 25, no. 1, pp. 17-30.

[11] W.N. Fu, P. Zhou, D. Lin, S. Stanton, and Z.J. Cendes, “Modeling of solid conductors in Two-Dimensional transient finite-element analysis and its application to electric machines,” *IEEE Trans. Magn.* vol. 40, no. 2, pp. 426-343, Mar. 2004

[12] S.L. Ho, Shuangxia Niu and W.N. Fu, “An Equivalent Parameter Extraction Method of Transient Electric Circuit and Magnetic Field Coupled Problems Based on Sensitivity Computation of System Equations, ” *IEEE Trans. on Magn.*, vol.47, no.8, pp.2068-2075, Aug. 2011.

[13] D. Chen, K. R. Shao, H. Yu, and J. D. Lavers, “A novel finite analytic element method for solving eddy current problems with moving conductors,” *IEEE Trans. Magn.*, vol. 37, no. 5, pt. 1, pp. 3150–3154, Sep. 2001.

[14] N. Burais, N. Foggia, and A. Nicolas, “Numerical solution of eddy currents problems including moving conducting parts,” *IEEE Trans. Magn.*, vol. MAG-20, no. 5, pp. 1995–1997, Sep. 1984.

[15] Wei Li, Bo Zhang, Rong Zeng and Jinliang He ,“Dynamic simulation of surge corona With time-dependent upwind difference method”, *IEEE Trans. Magn.*, vol. 46, no. 8, pp. 3109–3112, Aug. 2010.

[16] François Henrotte, Holger Heumann, Enno Lange, and Kay Hameyer, “Upwind 3-D vector potential formulation for electromagnetic braking simulations”, *IEEE Trans. Magn.*, vol. 46, no. 8, Aug. 2010.

[17] Yong Zhang, Guangyuan Yang, K. R. Shao, Youguang Guo, Jianguo Zhu, and J. D. Lavers, “Multiscale combined radial basis function collocation method for eddy currents analysis in high-speed moving conductors,” *IEEE Trans. Magn.*, vol. 45, no. 10, pp. 3973–3976, Oct. 2009.

[18] W. N. Fu and S.L. Ho, “Matrix analysis of 2-D eddy-current magnetic fields,” *IEEE Trans. Magn.*, vol. 45, no. 9, pp. 3343-3350, Sep. 2009.

[19] W. N. Fu and S. L. Ho, “Elimination of nonphysical solutions and implementation of adaptive step size algorithm in time-stepping finite-element method for magnetic

field–circuit–motion coupled problems”, *IEEE Transactions on Magnetics*, Vol. 46, No. 1, pp. 29-38, January, 2010.

[20] Y. S. Cho, S. Jun, S. I. Im and H. G. Kim, “An improved interface element with variable nodes for non-matching finite element meshes”*, Comput. Methods Appl. Mech. Engrg.* 194, pp. 3022-304, 2005.

[21] S. L. Ho, S. Niu and W. N. Fu, “Reduction of Numerical Errors of Time-Stepping Finite Element Analysis for Dynamic Simulation of Electric Machines,” *IEEE Trans. Applied Superconductivity,* vol. 20, no. 3, pp. 1864-1868, June 2010

[22] W. N. Fu and S. L. Ho, “Elimination of nonphysical solutions and implementation of adaptive step size algorithm in time-stepping finite-element method for magnetic field – circuit – motion coupled problems,” *IEEE Trans. Magn.*, vol. 46, no. 1, pp. 29-38, Jan. 2010.

[23] E. J. Silva and R.C. Mesquita, "Data management in finite element analysis programs using object-oriented techniques," *IEEE Trans. Magn.*, vol.32, no.3, pp.1445-1448, May 1996

[24] A. Promwungkwa, “Data structure and error estimation for an adaptive p-version finite element method in 2-D and 3-D solids,” Ph.D Dissertation, 1998.

[25] H. Karutz and W. B. Kraetzig, “A quadtree data structure for the coupled finite-element/element-free Galerkin method,” *International Journal for Numerical Methods in Engineering*, vol. 53, no.2, pp. 375-391, 2001.

[26] W. N. Fu, S. L. Ho, H. L. Li and H. C. Wong, “A multislice coupled finite-element method with uneven slice length division for the simulation study of electric machines,” *IEEE Trans. Magn.,* vol. 39, no. 3, pp. 1566-1569, May 2003.

[27] V. Chugunov, Y. Vassilevski, “Parallel multilevel data structures for a nonconforming finite element problem on unstructured meshes,” *Russian Journal of Numerical Analysis and Mathematical Modelling,* vol. 18, no. 1, pp. 1-11, Feb. 2003.

[28] S. Kumar, “Object-oriented finite element programming for engineering analysis in C++,” *Journal of Software*, vol. 5, no. 7, pp. 689-696, Jul. 2010.

[29] S. L. Ho, S. Y. Yang, G. Z. Ni, H. C. Wong, "A particle swarm optimization method with enhanced global search ability for design optimizations of electromagnetic devices," *IEEE Trans. Magn.*, vol. 42, no. 4, pp. 1107-1110, Apr. 2006.

[30] S. L. Ho, Shuangxia Niu, and W. N. Fu, "Design and analysis of a novel axial-flux-modulated machine," *IEEE Trans. Magn.*, accepted.

[31] S. L. Ho, Shuangxia Niu, and W. N. Fu, "Design and comparison of Vernier permanent magnet machines," *IEEE Trans. Magn.*, accepted.

[32] Shuangxia Niu, S. L. Ho, and W. N. Fu, "Performance analysis of a novel magnetic-geared tubular linear permanent magnet machine," *IEEE Trans. Magn.*, accepted.

[33] S. L. Ho, Ningning Chen, and W. N. Fu, "An Moving Mesh Embedded Algorithm in Finite Element Method for Optimal Design of Electromagnetic Devices," *IEEE Trans. Magn.*, accepted.

[34] J. Xu, "Two-Grid discretization technique for linear and nonlinear PDE," *SIAM Journal on Numerical Analysis,* vol. 33 , Issue 5, October 1996, pp.1759-1777

[35] K. Preis, I. Bardi, O. Biro, C. Magele, W. Renhart, K. R. Richter, G. Vrisk, "Numerical analysis of 3D magnetostatic fields", *IEEE Trans. Magn.*, vol. 77, No. 5, September 1991, pp. 3798-3803

[36] S. K. Chang, S. R. Jae, and F. Koji, "Convergence acceleration of the Newton–Raphson method using successive quadratic function approximation of residual," *IEEE Trans. Magn.* vol.42, No.4, April 2006, pp.611-614

[37] J. Fetzer, S. Kurz, G. Lehner, "Comparison of analytical and numerical integration techniques for the boundary integrals in the BEM-FEM coupling considering TEAM workshop problem No. 13," *IEEE Trans. Magn.* vol. 33, No. 2, March 1997, pp.1227-1230

[38] Dong-Joon Sim, Dong-Hyeok Cho, Jang-Sung Chun and Hyun-Kyo Jung and Tae-Kyoung Chung, "Efficiency optimization of interior permanent magnet

synchronous using genetic algorithms," *IEEE Trans. Magn.*, vol. 33, no. 2, pp. 1880-1883, Mar. 1997.

[39] Yoshiaki Kano, Kazuki Tonogi, Takashi Kosaka and Nobuyuki Matsui, "Optimization of axial-flux PM machines for improved torque density by simple non-linear magnetic analysis," *Electric Machines & Drives Conference, 2007. IEMDC '07. IEEE International*, pp. 27-33, May. 2007.

[40] Chang-Chou Hwang, Li-Yang Lyu, Cheng-Tsung Liu and Ping-Lun Li, "Optimal Design of an SPM Motor Using Genetic Algorithms and Taguchi Method," *IEEE Trans. Magn.*, vol. 44, no. 1, pp. 4325-4328, Nov. 2008.

[41] N. Bianchi and S. Bolognani, "Design optimisation of electric motors by genetic algorithms," *IEE Proc.- Electr. Power Appl.*, vol. 145, no. 5, pp. 475–483, Sep. 1998.

[42] K. S. Chai and C. Pollock, "Using genetic algorithms in design optimization of the flux switching motor, " *in Proc. Conf. Power Electron.*, Mach. Drives, Apr. 2002, pp. 540–545.

[43] N. Bianchi and A. Canova, "FEM analysis and optimisation design of an IPM synchronous motor," *in Proc. Conf. Power Electron., Mach. Drives*, Apr. 2002, pp. 49–54.

[44] M. Lukaniszyn, M. JagieLa, and R. Wrobel, "Optimization of permanent magnet shape for minimum cogging torque using a genetic algorithm," *IEEE Trans. Magn.*, vol. 40, no. 2, pp. 1228–1231, Mar. 2004.

[45] Stefan Giurgea, Daniel Fodorean, Giansalvo Cirrincione, Abdellatif Miraoui and Maurizio Cirrincione, "Multimodel Optimization Based on the Response Surface of the Reduced FEM Simulation Model With Application to a PMSM," *IEEE Trans. Magn.*, vol. 44, no. 9, pp. 1228–1231, Sep. 2008.

[46] Hadjout, L. Takorabet, N. Ibtiouen, R. Mezani and S. Ecole, El-Harrach, "Optimization of instantaneous torque shape of PM motors using artificial neural networks based on FE results," *IEEE Trans. Magn.*, vol. 42, no. 4, pp. 1283–1286, Apr. 2006.

[47] Pan Seok Shin, Han-Deul Kim, Gyo-Bum Chung, Hee Sung Yoon, Gwan-Soo Park and Chang Seop Koh, “Shape optimization of a large-Scale BLDC motor using an adaptive RSM utilizing design sensitivity analysis,” *IEEE Trans. Magn.*, vol. 43, no. 4, pp. 1653–1656, Apr. 2007.

[48] Liang Fang, Jae-woo Jung, Jung-Pyo Hong and Jung-Ho Lee, “Study on high-efficiency performance in interior permanent-magnet synchronous motor with double-layer PM design,” *IEEE Trans. Magn.*, vol. 44, no. 11, pp. 4393–4396, Nov. 2008.

[49] Pan Seok Shin, Sung Hyun Woo, Yanli Zhang and Chang Seop Koh, “An application of latin hypercube sampling strategy for cogging torque reduction of large-scale permanent magnet motor,” *IEEE Trans. Magn.*, vol. 44, no. 11, pp. 4421–4424, Nov. 2008.

[50] S. L. Ho, S. Y. Yang, G. Z. Ni, H. C. Wong, “A particle swarm optimization method with enhanced global search ability for design optimizations of electromagnetic devices,” *IEEE Trans. Magn.*, vol. 42, no. 4, pp. 1107-1110, Apr. 2006.

[51] P. Knupp and S. Steinberg, *The Fundamentals of Grid Generation*, CRC Press, Boca Raton FL, 1993.

[52] A,Winslow, “Numerical solution of the quasilinear Poisson equations in a nonuniform triangle mesh,” *J. Comp. Phys.,* 2, pp. 149-172, 1967.

[53] S. Shontz, S. Vavasis, “A mesh warping algorithm based on weighted Laplacian smoothing,” *12th International Meshing Roundtable*.

[54] K. Yamazaki, H. Ishigami, and A. Abe, “An adaptive finite element method for minor shape modification in optimization algorithms,” *IEEE Trans. Magn.*, vol. 44, no. 6, pp. 1102-1205, Jun. 2008.

[55] S. G. Nash, A. Sofer, *Linear and nonlinear programming*, McGraw-Hill, Inc., 1996.

[56] M. S. Floater, “Convex combination maps,” *Algorithms for Approximation IV,* 2001.

[57] TEAM optimization benchmark 25 [OL]. Available: http://www.compumag.org/jsite/team.html

[58] M. Aydin, Z. Q. Zhu, T. A. Lipo , and D. Howe, “Minimization of cogging torque in axial-flux permanent-Magnet machines: design concepts,” *IEEE Trans. Magn.*, vol. 43, no. 9, pp. 3614-3622, Sep. 2007.

[59] L. Dosiek and P. Pillay, “Cogging torque reduction in permanent magnet machines”, *IEEE Trans. Magn.*, vol. 43, no. 6, pp. 1565-1571, Nov./Dec. 2007.

[60] W. N. Fu, Z. J. Liu, and C. Bi, “A dynamic model of the disk drive spindle Motor and its applications,” *IEEE Trans. Magn.*, vol.38, no. 2, pp.973-976, Mar. 2002.

[61] K. J. Han, D. H. Cho, and H. K. Jung, “Optimal core shape design for cogging torque reduction of BLDC motor using genetic algorithm,” in *Proc. Compumag’99*, Sapporo, Japan, pp. 332–333, 1999.

[62] G. H. Kang, Y. D. Son, G. T. Kim and J. Hur, “A novel cogging torque reduction method for interior-type permanent-magnet motor,” *IEEE Trans. Magn.*, vol. 45, no. 1, pp. 161-167, Jan. /Feb. 2009.

[63] S. Vaez-Zadeh and A. R. Ghasemi, “Design optimization of permanent magnet synchronous motors for high torque capability and low magnet volume,” Elect. Power Syst. Res., vol. 74, pp. 307–313, Mar. 2005.

[64] S. Vaez-Zadeh and A. H. Isfahani, "Multiobjective design optimization of air-core linear permanent-magnet synchronous motors for improved thrust and low magnet consumption," *IEEE Trans. Magn* , vol. 42, no. 3, pp. 446- 452, Mar. 2006.

[65] K. Atallah and D. Howe, “A novel high-performance magnetic gear,” *IEEE Trans. Magn.*, vol. 37, no. 4, pp. 2844–2846, Jul. 2001.

[66] S. L. Ho. S. Niu and W. N. Fu, “Transient analysis of a magnetic gear integrated brushless permanent magnet machine using circuit-field-motion coupled time-stepping finite element method,” *IEEE Trans. Magn* , vol. 46, no. 6, pp. 2074-2077, Jun. 2010.

[67] L. Jian, K. T. Chau, W. Li and J. Li “A novel coaxial magnetic gear using bulk HTS for industrial applications,” *IEEE Trans. Applied Superconductivity,* vol. 20, no. 3, 981-984, Jun. 2010.

[68] K. Yamazaki, H. Ishigami, and A. Abe, “An adaptive finite element method for minor shape modification in optimization algorithms,” *IEEE Trans. Magn,* vol. 44, no. 6, pp. 1202-1205, Jun. 2008.

[69] M. Clerc and J. Kennedy, “The particle swarm—explosion, stability, and convergence in a multidimensional complex space,” *IEEE Trans. Evolutionary Computation*, vol. 6, no. 1, pp. 58-73, 2002.

[70] S. L. Ho, S. Y. Yang, G.Z. Ni, H. C. Wong, “A particle swarm optimization method with enhanced global search ability for design optimizations of electromagnetic devices,” *IEEE Trans. Magn.*, vol. 42, no. 4, pp. 1107-1110, Apr., 2006.