Efficient Parallel Adaptive Computation of 3D Time-Harmonic Maxwell's Equations Using the Toolbox PHG

Tao Cui*
State Key Laboratory of Scientific and Engineering Computing,
Institute of Computational Mathematics,
Academy of Mathematics and Systems Science, Chinese Academy of Sciences,
Beijing 100190, P.R. China.
tcui@lsec.cc.ac.cn

Abstract

PHG (Parallel Hierarchical Grid) is a scalable parallel adaptive finite element toolbox under active development at the State Key Laboratory of Scientific and Engineering Computing, Chinese Academy of Sciences. This paper demonstrates its application to adaptive finite element computations of electromagnetic problems. Two examples on solving the time harmonic Maxwell's equations are shown. Results of some large scale adaptive finite element simulations with up to 1 billion degrees of freedom and using up to 2048 CPUs are presented.

1. Introduction

Ever since the pioneering work of Babuška and Rheinboldt [1], the adaptive finite element method (AFEM) based on a posteriori error estimates has become a central theme in scientific and engineering computing. The AFEM is very efficient for problems with local singularities since it produces "quasi-optimal" meshes for the given problem by using reliable and efficient error estimates [4, 6, 14]. For steady state problems, the AFEM based on a posteriori error estimates is characterized by the *solve→estimate→mark→refine* loop and is described by the following algorithm.

Algorithm: *Starting from an initial mesh $\mathcal{T}_0$, let $\mathcal{T}_h = \mathcal{T}_0$.*

1. *Solve the problem* (1)-(2) *on $\mathcal{T}_h$.*
2. *Compute the local error estimate $\hat{\eta}_T$ for each element $T \in \mathcal{T}_h$ and the global error estimate $\mathcal{E}$. If $\mathcal{E}$ is smaller than the prescribed tolerance then stop.*
3. *Mark the elements whose local error estimate is large.*
4. *Refine the mesh T_h by dividing the marked elements and possibly some other elements, in order to maintain conformity of the mesh, into smaller elements.*
   Goto step 1.

It is well-known that the solutions of the time harmonic Maxwell's equations generally have very strong singularities, thus the AFEM is well suited for solving these problems. A framwork of the AFEM based on a posteriori error estimates for the time harmonic Maxwell's equations was presented in [6]. Extensive numerical experiments in [6] indicated that the AFEM based on the a posteriori error estimates has the very desirable quasi-optimality property: the energy error decays like $N^{-1/3}$, where N is the number of degrees of freedom, for the Nédélec lowest order edge element [9, 15], which has gain widespread popularity in numerical electromagnetic field computations by finite element methods [2, 3].

Unfortunately, parallel implementation of the AFEM on distributed memory parallel computers is very difficult because of the complexities of the mesh management and load balance issues. Also, highly efficient numerical methods for solving the linear system resulting from finite element discretization are required. For facilitating implementing the AFEM, we have developed the toolbox PHG, *Parallel Hierarchical Grid* [13]. The motivation of this toolbox is to support the research on AFEM algorithms and development of AFEM codes. PHG deals with conforming tetrahedral meshes and uses bisection for adaptive mesh refinement and MPI for message passing[10]. Using an object oriented design, the details of complex mesh management and parallelism are hidden from users. PHG provides supports for adaptive finite element computations, such as finite element bases (including the Nédélec edge elements for electromagnetic computations), numerical quadrature, and basic operations with finite element functions. For building, assem-

*Supported by the 973 Program under the grant 2005CB321702 and by China NSF under the grant 10531080

bling, and solving linear systems and eigenvalue problems resulting from finite element discretization, an unified linear algebra module for manipulating distributed sparse matrices stored in compressed sparse rows (CSR) and distributed vectors is provided, based on linear solvers including PCG and GMRES are built. Load balancing is achieved through mesh repartitioning and redistribution. PHG also provides optional interfaces to many well known open source linear solvers and eigen solvers, such as PETSc[12], HYPRE[8], MUMPS[17], and PARPACK[16].

In this paper, we demonstrate the application of PHG to electromagnetic computations with two examples in which 3D time harmonic Maxwell's equations are solved. The layout of the paper is organized as follows. In section 2, we present the numerical algorithm and the a posteriori error estimates for the time harmonic Maxwell's equations. In section 3, we give numerical results obtained with two problems. The first one is the so-called "screen problem", and the second one is an eddy current model with voltage excitations for complicated three dimensional structures. In the last section, section 4, some concluding remarks are given.

2. Time-harmonic Maxwell's equations and adaptive finite element computation

The general form of time harmonic Maxwell's equations we focus on in this paper is as follows:

$$\nabla \times (\frac{1}{\mu}\nabla \times \mathbf{E}) - k^2\mathbf{E} = \mathbf{f} \qquad \text{in } \Omega, \tag{1}$$

$$\mathbf{E} \times \mathbf{n} = \mathbf{g} \qquad \text{on } \partial\Omega, \tag{2}$$

where Ω is a bounded domain with Lipschitz boundary, $\mathbf{E}$, $\mathbf{f}$, and $\mathbf{g}$ are vector functions from $\mathscr{R}^3$ to $\mathscr{R}^3$ or $\mathscr{C}^3$, $\mathbf{E}$ is the solution to compute, $\mathbf{f}$ and $\mathbf{g}$ are known functions, and $\mathbf{n}$ is the unit outer normal of the boundary.

We introduce some notations here. $\mathcal{T}_h$ denotes a regular tetrahedral triangulations of Ω. $\mathcal{F}_h$ and $\mathcal{E}_h$ denote respectively the set of faces and the set of edges not lying on $\partial\Omega$. For any face $F \in \mathcal{F}_h$, assuming $F = T_1 \cap T_2$, T_1, $T_2 \in \mathcal{T}_h$ and the unit normal ν pointing from T_2 to T_1, we denote the jump of a function φ across F by $[\![\varphi]\!]_F := \varphi|_{T_1} - \varphi|_{T_2}$.

The Nédéléc lowest order finite element is used to discretize equations (1)-(2). Details of the finite element discretization for this problem can be found in P. Monk *et al.* [11]. Let $\mathbf{E}_h$ be the finite element solution of (1)-(2) on $\mathcal{T}_h$. The local a posteriori error estimate $\hat{\eta}_T$ for the solution $\mathbf{E}_h$ over an element $T \in \mathcal{T}_h$ is computed by (see e.g. Chen *et al.* [6]):

$$\hat{\eta}_T^2 := \eta_T^2 + \sum_{F\in\mathcal{F}_h\cap T} \eta_F^2, \tag{3}$$

where

$$\begin{aligned} \eta_T^2 &= h_T^2(\|f + k^2\mathbf{E}_h - \nabla \times \frac{1}{\mu}\nabla \times \mathbf{E}_h\|_{0,T}^2 \\ &\quad + \|\nabla \cdot (f + k^2\mathbf{E}_h)\|_{0,T}^2), \\ \eta_F^2 &= h_F(\|[\![\frac{1}{\mu}(\nabla \times \mathbf{E}_h) \times \nu]\!]\|_{0,F}^2 \\ &\quad + \|[\![(f + k^2\mathbf{E}_h) \cdot \nu]\!]\|_{0,F}^2. \end{aligned}$$

Here h_T and h_F are respectively the diameter of the element T and the face F.

The finite element discretization of problem (1)-(2) and the computations of the a posteriori error estimate (3) can be conveniently implemented based on PHG, resulting in an efficient parallel adaptive finite element code for solving the time harmonic Maxwell's equations.

3. Numerical examples

In this section we demonstrate our parallel adaptive finite element code for solving the time harmonic Maxwell's equations with two examples.

3.1. The screen problem

In this example, all functions and parameters have real values. The computational domain $\Omega = \Omega_0 \backslash \Sigma$, where $\Omega_0 = [-1, 1]^3$, and (the screen) $\Sigma = [-\frac{1}{2}, \frac{1}{2}] \times 0 \times [-\frac{1}{2}, \frac{1}{2}]$, as shown in Figure 1. The other parameters are $\mathbf{f} = (1, 1, 1)$, $k^2 = 1$, and $\mathbf{g} = (0, 0, 0)$.

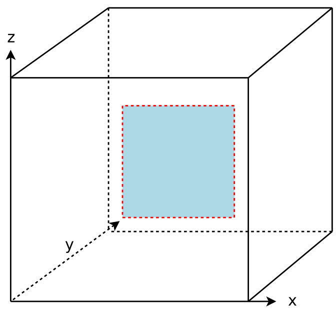

Figure 1. The screen problem.

The solution of this problem exhibits strong singularities near the edges and the corners of the screen. Thus this example can be used to check the performance of the adaptive strategy, as well as the correctness and robustness of our code.

Since $-k^2$ is negative, the linear system of equations resulting from the finite element discretization is symmetric indefinite. It is solved by a PCG method in which the preconditioning matrix is the finite element discretization of the same problem with k^2 set to -1. The preconditioning matrix is symmetric positive definite and is solved by a PCG method using the very efficient Hiptmair-Xu auxiliary space preconditioner [7] (the AMS preconditioner). The actual implementation of the AMS preconditioner used in this computation is from Hypre-2.2.0b [8], which uses the algebraic multigrid solver BoomerAMG, also available in Hypre, for solving the Poisson equations in the auxiliary spaces. In the computations, the preconditioning system is solved "exactly" to a prescribed tolerance in each outer PCG iteration.

The computations were performed on a home made massively parallel computer. Figure 2 shows the numbers of PCG iterations required to reduce the initial residual by a factor of 10^{-10} on 2048 CPUs, in which the number of degrees of freedom grows from 10K to 1G (one billion). The iteration numbers are stable with increasing mesh size for both adaptive and uniform mesh refinements, except some oscillations of the iteration numbers with adaptive mesh refinements.

Figure 3 is the log-log plot of the a posteriori error indicator with respect to the number of degrees of freedom. It shows the quasi-optimality of the adaptive method: the error decays as $O(N^{-1/3})$ with adaptively refined meshes, where N denotes the number of degrees of freedom, which is not true with uniformly refined meshes.

Firgure 4 shows a sample mesh on the plane $y = 0$ generated by the adaptive method. We observe that the adaptive mesh captures the singularities of the solution quite well.

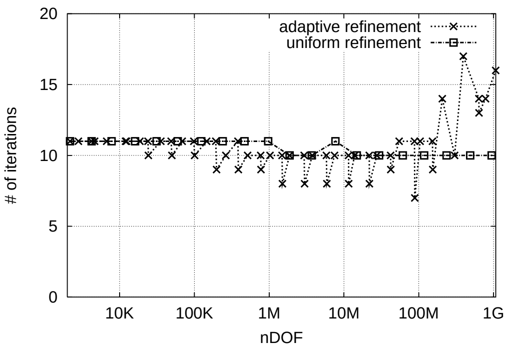

Figure 2. Number of PCG iterations on 2048 CPUs (the screen problem).

Table 1 lists the wall-clock time spent in different parts of the finite element solution procedure for a fixed problem size with 21,805,534 degrees of freedom. For this problem size our code scales well up to 256-512 CPUs.

Table2 lists the wall-clock time for the case in which the average number of elements is fixed to 393,216, and the global mesh size increases proportionally with the number of CPUs, using uniform mesh refinements. It is an indication on the scalability of our code.

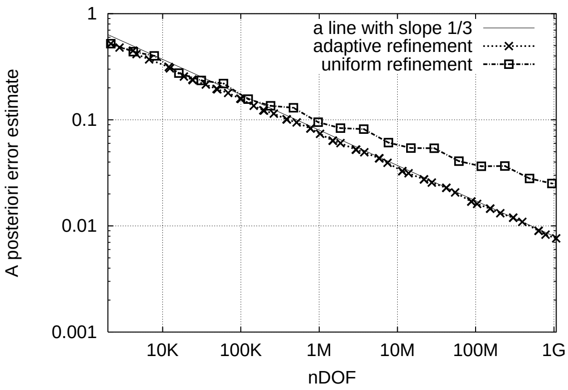

Figure 3. Log-log plot of the a posteriori error indicator with respect to the number of degrees of freedom (the screen problem).

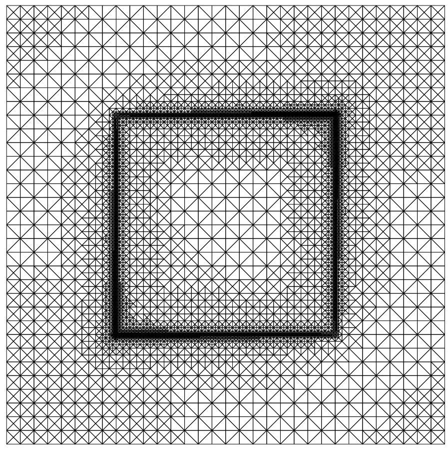

Figure 4. The adaptive mesh on the plane $y = 0$ (the screen problem)

3.2. The eddy current problem

The second example comes from the eddy current problem of circult/field coupling which has direct applications in circuit design. The dimensionless form of the $\mathbf{A} - \phi$ model of the problem is given by:

$$\begin{cases} \nabla \times \nabla \times \mathbf{A} + \mathbf{i}s^2\omega\mu\sigma\mathbf{A} = -s\sigma\mu\nabla\phi_0 & \text{in } \Omega, \\ \mathbf{A} \times \mathbf{n} = 0 & \text{on } \partial\Omega, \end{cases} \tag{4}$$

where s is the dimensionless factor. Please refer to [5] for details about the formulation of this problem. Note that here $\mathbf{A}$ is a complex valued function.

# of CPUs	Setup		Solve
	linear system	PC	
32	13.391	16.911	1839.0
64	7.0697	8.8474	872.02
128	3.3057	4.3237	398.52
256	1.7918	2.6003	231.26
512	0.9112	1.9709	177.75
1024	0.4461	2.7106	217.94
2048	0.2425	5.5601	344.31
4096	0.1210	15.107	565.93

Table 1. Wall-clock time (seconds) spent in different parts of the finite element solver on a fixed mesh containing 18,333,752 elements/21,805,534 DOFs (the screen problem)

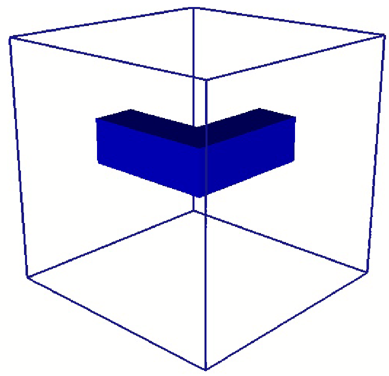

Figure 5. The conductor and the computational domain (the eddy current problem).

# of CPUs	# of elements	Setup		Solve
		linear system	PC	
32	12,582,912	10.338	11.096	902.45
64	25,165,824	9.7286	10.593	1138.9
128	50,331,648	9.7098	12.543	1023.9
256	100,663,296	9.7551	14.767	1130.9
512	201,326,592	9.9193	19.113	1563.9
1024	402,653,184	9.7997	30.252	1491.3
2048	805,306,368	10.180	53.440	2455.6

Table 2. Wall-clock time (seconds) spent in different parts of the finite element solver by fixing the average number of elements in each process to 393,216 (the screen problem).

The computational domain $\Omega = \Omega_c \cup \Omega_{nc} = [0,5]^3$, where Ω_c is an L-shaped conductor, and the surround of the conductor $\Omega_{nc} = \Omega \backslash \Omega_c$ is air. The computational domain is shown in Figure 5. The other parameters are: $\sigma = 5.8 \times 10^7$ in Ω_c and 0 in Ω_{nc}, $\mu = \mu_0 = 4\pi \times 10^{-7}$, $\omega = 2\pi \times 10^{10}$, $s = 10^{-6}$, ϕ_0 is any function satisfying $\phi_0|_{S_1} = 1$ and $\phi_0|_{S_2} = 0$, where $S_1 = 0 \times [2,3] \times [2,3]$ and $S_2 = [2,3] \times 0 \times [2,3]$ are the two ports of the conductor (it can be shown that the quantities of physical interests depend only on the values of ϕ_0 at the ports).

By separating the real and the imaginary parts, the linear system resulting from finite element discretization can be written in the following form:

$$\begin{pmatrix} K & -M \\ -M & -K \end{pmatrix} \begin{pmatrix} A_{re} \\ A_{im} \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix}, \tag{5}$$

where K is the stiffness matrix and M the mass matrix, A_{re} is the real part of the solution and A_{im} the imaginary part. This system is singular because $\sigma = 0$ in Ω_{nc}. It is solved directly by a preconditioned GMRES or MINRES method with the preconditioning matrix chosen as:

$$\begin{pmatrix} K+M & 0 \\ 0 & K+M \end{pmatrix}^{-1}. \tag{6}$$

The preconditioning matrix $K + M$ is symmetric semidefinite and corresponds to the finite element discretization of the following problem:

$$\begin{cases} \nabla \times \nabla \times \mathbf{A} + s^2 \omega\sigma\mu \mathbf{A} = \mathbf{f} & \text{in } \Omega, \\ \mathbf{A} \times \mathbf{n} = 0 & \text{on } \partial\Omega, \end{cases} \tag{7}$$

It can be efficiently solved by a PCG method using the AMS preconditioner.

Let $\mathbf{A}_h$ be the finite element solution of (4) on $\mathcal{T}_h$. The local a posteriori error estimate over an element $T \in \mathcal{T}_h$, which is a little different from the previous example, is given by (see [5]):

$$\begin{aligned} \hat{\eta}_T &:= \left(\eta_T^2 + \sum_{F \in \mathcal{F}_h, F \subset \Omega\mathrm{nc}, F \subset \partial T} \|J\|_{L^2(F)}^2 \right)^{1/2}, \quad (8) \\ J &:= h_F^{1/2} [\![(\mathbf{A}_h - \nabla\phi_h) \cdot \nu]\!]_F, \\ \eta_T^2 &:= h_T^2 \| - s\sigma\mu(\nabla\phi_0 + \mathbf{i}s\omega\mathbf{A}_h) \|_{L^2(T)}^2 \\ &+ h_T^2 \| s\mu\sigma\nabla \cdot (\nabla\phi_0 + \mathbf{i}s\omega\mathbf{A}_h) \|_{L^2(T)}^2 \\ &+ \sum_{F \in \mathcal{F}, F \subset \partial T} \Big(h_F \| [\![\nu \times \nabla \times \mathbf{A}_h]\!]_F \|_{L^2(F)}^2 \\ &\quad + h_F \| [\![s\sigma\mu(\nabla\phi_0 + \mathbf{i}s\omega\mathbf{A}_h) \cdot \nu]\!]_F \|_{L^2(F)}^2 \Big), \end{aligned}$$

where ϕ_h is a function satisfying:

$$\begin{cases} \nabla\phi_h = 0 & \text{in } \Omega_c, \\ \phi_h|_{\Omega_{nc}} \in V_h(\Omega_{nc}), \forall v_h \in V_h(\Omega_{nc}), & \\ (\nabla\phi_h, \nabla v_h)_{\Omega_{nc}} = (\mathbf{A}_h, \nabla v_h)_{\Omega_{nc}} & \text{in } \Omega_{nc}. \end{cases} \tag{9}$$

Here $V_h(\Omega_{nc})$ denotes the H^1 conforming linear finite element over Ω_{nc}.

The computations were performed on 64 CPUs of the cluster LSSC-II in the State Key Laboratory of Scientific and Engineering Computing of Chinese Academy of Sciences. LSSC-II is built of 512 Intel Pentium IV 2.0 GHz CPUs on 256 nodes. Each node has 1GB of memory and all nodes are connected by Myrinet2000 network. The preconditioned GMRES method was used and in each GMRES iteration the preconditioning system was solved by performing only a few (typically 3-5) PCG/AMS iterations such that the residual of the preconditioning system was reduced by a factor of 0.01.

Figure 6 is the log-log plot of the a posteriori error estimate with respect to the number of degrees of freedom, showing the quasi-optimality of the adaptive meshes. Figure 7 shows a sample adaptive mesh on the plane $z = 0.25$. The adaptive meshes capture the singularity of the solution very well.

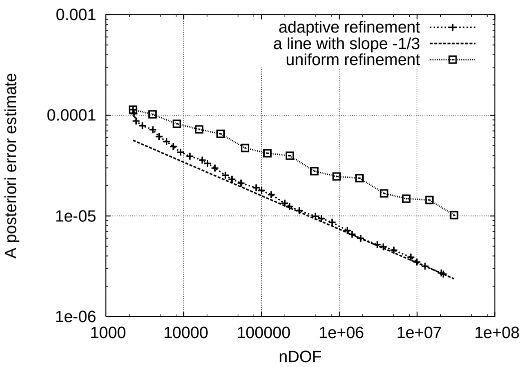

Figure 6. Log-log plot of the a posteriori error estimate with respect to the number of degrees of freedom (the eddy current problem).

Table 3 lists the relative error of the resistance RR, the inductance RL, and the number of the GMRES iterations and the time required to reduce the initial residual by a factor of 10^{-10} in solving the linear system of equations on adaptively refined meshes. Table 4 shows the results on uniform refined meshes. The relative error of the resistance RR is defined as RR=$|R-\hat{R}|/|\hat{R}|$, where $\hat{R}$ is the resistance obtained at the last adaptive refinement step which we take as the ”exact” solution. The relative error of the inductance RL is defined similarly. We observe that for the resistance, the numerical result obtained on an adaptive mesh with 41,752 degrees of freedom is similar to the result obtained on an uniform mesh with 1,821,040 degrees of freedom. This demonstrates the efficiency of the adaptive algorithm.

The fourth column of the tables shows the numbers of

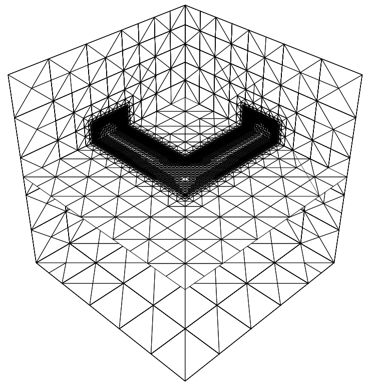

Figure 7. The adaptive mesh on the plane $z = 0.25$ with 3338028 degrees of freedom (the eddy current problem).

GMRES iterations performed in order to reduce the residual by a factor of 10^{-10}. The stable iteration numbers in Tables 3 and Tables 4 indicate that the preconditioner is nearly optimal.

4. Conclusion

We have presented some simulations for the time harmonic Maxwell’s equations with our parallel adaptive finite element code implemented by using the parallel adaptive finite element toolbox PHG. The results show that application of PHG to the numerical solution of the time harmonic Maxwell’s equations is successful. The resulting code is robust, efficient, scalable, and is capable of solving large scale problems in adaptive finite element computations with up to 1 billion degrees of freedom using thousands of CPUs.

Acknowledgement.

The work presented here was done at the State Key Laboratory of Scientific and Engineering Computing. The work on the eddy current problem was jointly done with Zhiming Chen, Junqing Chen, and Lin-bo Zhang.

References

[1] I. Babuška and C. Rheinboldt, Error estimates for adaptive finite element computations, *SIAM J. Numer. Anal.*, 15(1978).

[2] A. Bossavite and J-C. Vérité. A mixed fem-bem method to solve 3-d Eddy urrent problems. *IEEE Transactions on Magnetics*, 18:431-435, 1982.

DOF	$RR_k(\%)$	$RL_k(\%)$	ITs	Time
2230	8.47933	11.531	20	1.6328s
4028	16.715	4.96048	20	3.7396s
9146	5.26418	0.948607	19	9.7886s
17334	3.11843	1.47452	18	17.9832s
25290	1.05826	2.63336	18	15.6908s
41752	0.385401	2.94323	18	14.6994s
85546	0.932705	0.800637	18	16.2295s
199860	0.198863	1.18614	18	22.6697s
589742	0.138243	0.700146	18	38.3012s
1270730	0.0223211	0.372132	18	53.4644s
3081720	0.0299658	0.167859	18	99.0557s
8324122	0.012712	0.0193979	18	234.1677s
12681466	0.00267753	0.0296221	18	393.7277s
20471918	0.0260496	0.0168069	18	659.5128s

Table 3. The relative error of the resistance and inductance, the number of GMRES iterations and the time required for solving the linear system on adaptively refined meshes. The "exact" solution is $\hat{R} = 7.992448 \times 10^{-2}$(Ohm), $\hat{L} = 1.427988 \times 10^{-3}$(nH) (the eddy current problem).

DOF	$RR_k(\%)$	$RL_k(\%)$	ITs	Time
2230	8.47933	11.531	20	1.6126s
8180	17.7202	5.22462	19	4.4551s
15860	17.3377	7.93557	20	8.9316s
29860	9.14819	2.48735	19	9.3170s
119320	5.05059	1.7171	19	16.0965s
231320	2.62963	0.287397	19	18.8913s
925040	1.35844	0.327664	18	36.2513s
1821040	0.836252	0.0717093	18	47.2343s
7283680	0.3837	0.0652667	18	176.0440s
14451680	0.246207	0.00259106	18	348.4231s
29926880	0.112719	0.043978	18	758.2660s

Table 4. The relative error of the resistance and inductance, the number of GMRES iterations and the time required for solving the linear system on uniformly refined meshes. The "exact" solution is $\hat{R} = 7.992448 \times 10^{-2}$(Ohm), $\hat{L} = 1.427988 \times 10^{-3}$(nH) (the eddy current problem).

[3] A. Bossavite. Whitney forms: aclass of finite elements for three-dimensional computation in electromagnetism, *Inst. Elec. Eng. Proc.*, Part A, 135:493-500(8), 1988

[4] Z. Chen and F. Jia, An adaptive finite element method with reliable and efficient error control for linear parabolic problems, *Math. Comp.*, 73:1163-1197, 2004

[5] J. Chen, Z. Chen, T. Cui, and Lin-bo Zhang, An Adaptive Finite Element Method for the Eddy Current Model With Circuit/Field Couplings, submitted to *SIAM. J. Sci. Comput.*.

[6] Z. Chen, L. Wang and W. Zheng, An adaptive multilevel method for time harmonic Maxwell equations with singularities, *SIAM J. Sci. Comput.*, 29:118-138, 2007

[7] R. Hiptmair and J. Xu, Nodal auxiliary space preconditioning in H(curl) and H(div) spaces, *Research Report No. 2006-09*, Seminar für Angewandte Mathematik, Eidgenössische Technische Hochschule, CH-8092 Zürich, Wsitzerland, May 2006.

[8] HYPRE, *High performance preconditioners*, http://www.llnl.gov/CASC/hypre

[9] J.C.Nédélec. Mixed finite elements in R^3. *Numerische Mathematik*, 35:315-341, 1980.

[10] MPI, *Message Passing Interface*, http://www-unix.mcs.anl.gov/mpi

[11] P. Monk, *Finite Element Methods for Maxwell's Equations*, Carendon Press, Oxford, 2003.

[12] PETSC, *Portable, Extensible Toolkit for Scientific Computation*, http://www-unix.mcs.anl.gov/petsc/petsc-2

[13] PHG, *Parallel Hierachical Grid*, http://lsec.cc.ac.cn/phg

[14] R. Verfürth, *A Review of A Posteriori Error Estimation and Adaptive Mesh-Refinement Techniques*, Wiley-Teubner, CHichester, Stuttgart, 1996.

[15] H. Whitney. *Geometric Integration Theory*. Princeton University Press, 1957.

[16] R. B. Lehoucq and D. C. Sorensen and C. Yang, ARPACK, http://www.caam.rice.edu/software/ARPACK/

[17] P. Amestoy and A. Fevre and A. Guermouche and J.-Y. L'Excellent and S. Pralet, MUMPS: a MUltifrontal Massively Parallel sparse direct Solver, http://mumps.enseeiht.fr/