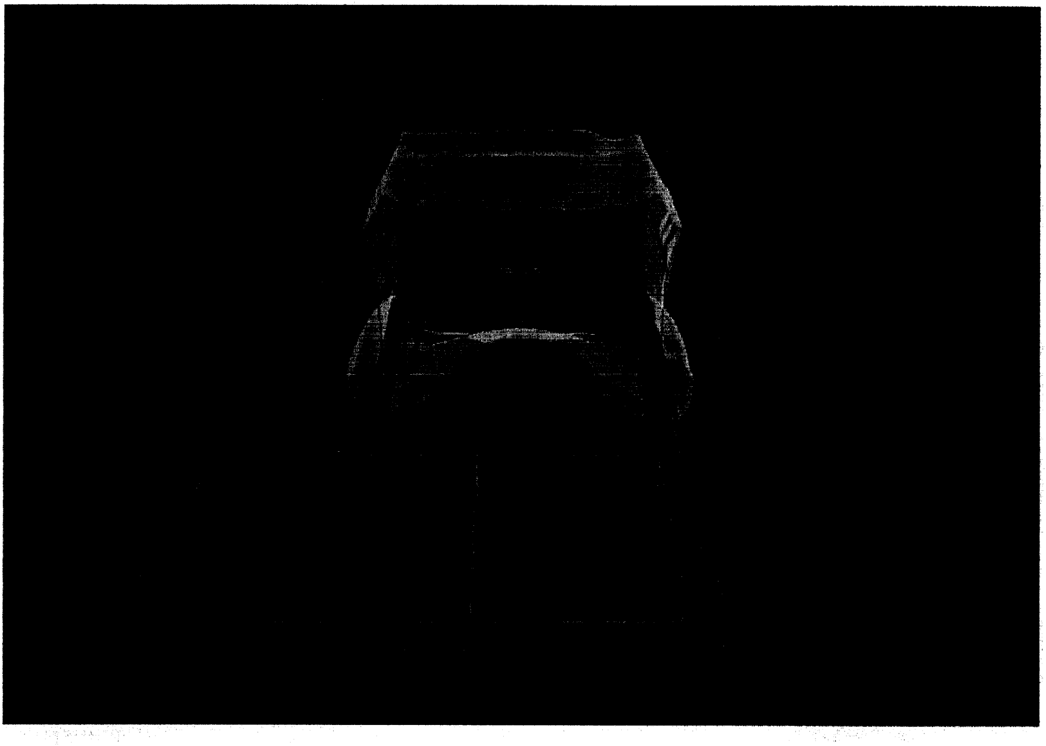
ISSN 1562-5370

7th International Journal of Theoretical Electrotechnics



Editors: A. Kost, L. Jänicke
Brandenburgische Technische Universität Cottbus

March 1999

Fast Electromagnetic Field Computations Using Multigrid Method Based on Nested Finite Element Meshes

Vlatko Čingoski and Hideo Yamashita
Faculty of Engineering, Hiroshima University, 1-4-1 Kagamiyama, Higashihiroshima 739-8527, JAPAN

***Abstract*—In this paper the investigation of the efficiency of the multigrid method as a solution method for large systems of algebraic equations that arise from ordinary finite element analysis is presented. The mathematical background for multigrid methods and some points regarding definition of restriction and prolongation matrices for multigrid finite element analysis based on nested meshes are also given. The convergence rate and computation speed of the V-cycle and W-cycle multigrid algorithms are discussed and compared with the ordinary ICCG method. A comparison of the computation speed between multigrid method and the ICCG method is also presented, showing that the multigrid method is very promising as a fast and accurate solution method for large systems of equations or as a solution method for adaptive finite element computations.**

***Index terms*—Finite element methods, multigrid methods, magnetostatics, iterative methods, relaxation methods, linear algebra.**

I. Introduction

For the past four decades tremendous developments in hardware and software technology have been witnessed. As a result, huge research work has been done in the field of numerical methods, improving their performance and broadening the area of their application. In the same time, a number of numerical methods already theoretically described, however not yet effectively used due to their numerical complexity, has emerged with large speed and great success mainly as a result of vigorous increase in computer's performances. The solution of large systems of linear algebraic equations is only one field where such vigorous research has been performed. It is well known that a huge class of physically based problems such as those which arise from boundary value problems can be easily solved after transforming them into a system of linear algebraic equations. Today, various direct and indirect (iterative or relaxation) methods for the solution of such systems of equations have already been successfully proposed [1]. However, for highly complex, especially three-dimensional simulations such as the computation of fluid dynamics, transient phenomena, propagations etc., still a large scale simulation is indispensable. These simulations are usually very time consuming, therefore scientists always look for new and improved procedures for increasing the computational speed.

One of the most popular methods for decreasing the computation time needed for solving linear systems of algebraic equations is the multigrid solution method. This method is theoretically known for some time, its practical application begins in 1970s with the work of Brandt [2]. Later on with the works of Briggs, Bramble, Rüde, and especially Hackbusch and McCormick, the multigrid method becomes very popular for the solution of various problems that arise when partial differential equations are discretized using regular or irregular meshes such as in the case of the finite difference or the finite element method [3–8]. It has been reported that the multigrid method can solve elliptic partial equations discretized on N grid points in $O(N)$ operations, which is much faster than any other rapid solution methods which can go as far as $O(N \log N)$ operations [9]. Multigrid methods show the same good convergence rate even for the solution of elliptic partial equations with nonlinear coefficients or for the solution of systems of nonlinear equations. However, on the other hand, there is not a single multigrid algorithm which can be successfully applied to a wide range of physical problems. Rather than that, the multigrid method gives an ideas on how to adjust the principles of multigrid algorithms to a particular problem [10].

Recently, in the field of electromagnetic computation large scale simulations have become available particularly for three-dimensional field computations and transient electromagnetic field analysis [11]. For such time consuming analysis it is interesting to investigate the application of multigrid algorithms as a solution method. While for the discretization of elliptic equations using the finite difference method, the multigrid method was widely applied, it is interesting to investigate its properties and benefits when applied to the finite element analysis. Recently, several authors presented some initial research results in using the multigrid method for electromagnetic field computation, however, only as a preconditoner but not as a solution method [12], [13].

In this paper, we present an application of the multigrid method as a full solution method for a system of linear algebraic equations generated by means of the finite element method. First, we briefly address the mathematical

Manuscript received November 15, 1998.
vlatko@eml.hiroshima-u.ac.jp

background of the multigrid method and describe why the multigrid method is faster than other relaxation methods. Next, we discuss some issues regarding the numerical implementation of the multigrid method over several finite element meshes with various mesh densities. We also discuss the types of multigrid algorithms and their differences, benefits and computational effects. Finally, we compare the convergence rate and the computational speed of several multigrid algorithms with the ICCG (Incomplete Cholesky Conjugate Gradient) method as one of the most commonly used iterative method for the solution of a large scale system of linear algebraic equations. The paper will be concluded with some final remarks and points for future research in this area.

II. Multigrid Methods

Today, for electromagnetic finite element analysis various linear system solvers are in common use, among them the ICCG (Incomplete Cholesky Conjugate Gradient) method is probably the most popular. However, for 3D nonlinear or time dependent problems, even the ICCG method results in lengthy computation. Moreover, as already mentioned in [13], the number of arithmetic operations for the ICCG method grows as much as $N^{3/2}$ for 2D and as much as $N^{4/3}$ for 3D problems, respectively, where N is the number of unknowns per finite element mesh.

Multigrid methods are set of techniques to solve a linear system of equations by using several grids with different mesh density levels. Their main advantage is that they show a much faster convergence rate than other iterative solution methods keeping the number of arithmetic operations per number of unknowns almost independent of the size of the problem. These two properties make multigrid methods very attractive for the solution of large systems of equations. As we already mentioned in the introduction of this paper, there is not a single multigrid method but there is rather a global multigrid method framework. Each multigrid method is based mainly on three numerical operations: *smoothing*, *restriction* and *prolongation*. Next, we will discuss each of these operations in detail using the simplest of all multigrid methods, the so-called two-grid method.

A. Two-grid Method

In order to describe the properties of the multigrid solution method, we will use the so-called two-grid solution method which has no practical meaning but is very suitable to explain the main idea that lies behind the multigrid methods. As the name suggests, the two-grid method is based on the development of two grids with different mesh densities, a coarse grid (later on *level #1)* and a dense grid (later on *level #2)*. These two grids in general can have an arbitrary number of elements with arbitrary shape, however, in this paper we will assume that those two grids are so-called *nested grids* which simply means that the nodes of the coarse grid coincide with some notes of the dense grid, as shown in Fig. 1. Even more, in this paper whenever we discuss meshes with different grid densities we refer to nested grids. The extension of the multigrid methods for non-nested meshes is possible and will be presented somewhere else.

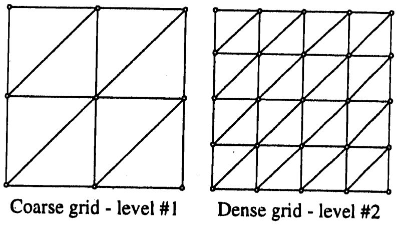


Fig. 1. Grid levels for two-grid method.

Since we already have two grids, the next step is to define a system of linear algebraic equations which has to be solved. We generate this system of equations using the usual finite element procedure only on the dense grid — *level #2.* Let

$$\mathbf{K}\,\mathbf{x} = \mathbf{f} \tag{1}$$

denote a system of linear equations where $\mathbf{K}$ is the matrix of the system, $\mathbf{x}$ is the unknown vector, and $\mathbf{f}$ stands for the source vector. We will use $\mathbf{x}$ to denote the exact solution of the above system, and $\widetilde{\mathbf{x}}$ to denote the approximation of the exact solution, usually obtained using some kind of iterative method. Therefore, for the algebraic error $\mathbf{e}$ the following equation is valid

$$\mathbf{e} = \mathbf{x} - \widetilde{\mathbf{x}} \quad . \tag{2}$$

This error $\mathbf{e}$ is also a vector and its norm can be easily established using any kind of standard vector norms [1]. However, what is most important for us is that since we don't know the exact solution $\mathbf{x}$ we are not able to find the error vector $\mathbf{e}$ either. Fortunately, we are able to compute the residual vector $\mathbf{r}$

$$\mathbf{r} = \mathbf{f} - \mathbf{K}\,\widetilde{\mathbf{x}} \quad , \tag{3}$$

which gives us some information on how well we approximated the exact solution $\mathbf{x}$ with $\widetilde{\mathbf{x}}$. It is easy to notice that if $\mathbf{r} = \mathbf{0}$, at the same time the error vector $\mathbf{e}$ also equals zero. By simple rearrangement of (3) and its subtraction from (1), a very important relationship between the error vector $\mathbf{e}$ and the residual vector $\mathbf{r}$ can be obtained

$$\mathbf{K}\,\mathbf{e} = \mathbf{r} \quad . \tag{4}$$

This means that the error $\mathbf{e}$ regarding the residual $\mathbf{r}$ satisfies the same system of equations as the unknown solution $\mathbf{x}$ regarding the source vector $\mathbf{f}$. Additionally, if we have the approximated solution $\widetilde{\mathbf{x}}$ and the error $\mathbf{e}$, we can compute the exact solution easily using

$$\mathbf{x} = \widetilde{\mathbf{x}} + \mathbf{e} \quad . \tag{5}$$

These are the two main components of any multigrid computation algorithm.

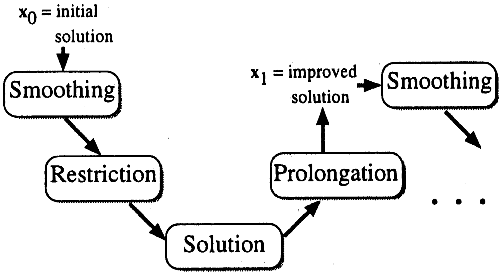


Fig. 2. One cycle of two-grid solution method.

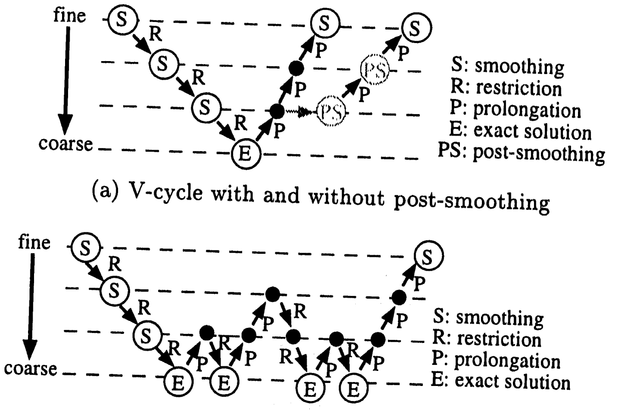
Now, before we proceed with the description of a two-grid method, let us briefly address the reason why the multigrid method is faster than other iterative solution methods. First, if one expands the error $\mathbf{e}$ into a discrete Fourier series, two sets of error components can be found: one in the lower half of the frequency spectrum which are usually called *smooth components*, and the other on the higher frequency range usually called *nonsmooth components*. It is known that when the size of the finite elements becomes very small (dense mesh), the relaxation converges very slowly as a result of a very small reduction of the smooth error components. On contrary, many relaxation methods reduce the amplitude of the nonsmooth components with each consecutive iteration by large factors, or as usually said, they smooth the error fast. Therefore, they are good smoothing operators. What we actually have in multigrid methods is this: we perform only few iteration steps on the dense grid level and reduce the nonsmooth error components by a large factor — that's what we call smoothing. Next, as a very important step, we restrict the error on the coarser grid, where the smooth error components from the dense mesh look less smooth on the coarse mesh, or in other words: they become nonsmooth error components on this coarse grid and are ready for further reduction by only several iteration steps using the ordinary iterative solution method. The procedure continues downward to the coarsest grid where even the smoothest error components from the densest grid become nonsmooth error components and can be rapidly reduced. With other words, when the relaxation process begins to stall on one grid level, we move to a coarser grid level on which the existing smooth error components could be reduced faster and where the relaxation procedure becomes more effective.

Let us now come back to our original problem (1). Let us first find the approximated solution $\widetilde{\mathbf{x}}$ for the system (1) by performing m iterative steps (usually $m = 1$ or 2) using some kind of stationary iterative process such as the Jacobi or the Gauss-Seidel iterative scheme. This process is called *smoothing*. Next, after we have the approximated solution $\widetilde{\mathbf{x}}_2$ we compute the residual vector $\mathbf{r_2}$ according to (3). The subscript 2 denotes that both the error and the residual vector are computed for the dense grid (*level #2*). Next, we try to estimate the residual vector $\mathbf{r_1}$ on the coarse grid (*level #1*). For that purpose we need a *restriction operator* $\mathbf{R}$. This matrix $\mathbf{R}$ restricts the residual error from the dense grid towards the coarse grid, therefore, this operation is called *restriction.* After we have the estimation of the residual error $\mathbf{r_1}$ on the coarse grid level, we could solve (4) exactly using any kind of solution method (direct or even better iterative) and obtain the correction vector $\mathbf{e_1}$. However, this correction vector is defined on the coarse grid only and has to be interpolated on the dense grid in order to be added to the existing approximated solution $\widetilde{\mathbf{x}}$. A new *interpolation (prolongation) matrix* $\mathbf{P}$ has to be defined which interpolates the correction vector from the coarse grid toward the dense grid. This operation is called *prolongation.* Finally, after the computation of the correction vector $\mathbf{e_2}$ for the dense mesh, a new and improved approximated solution $\widetilde{\mathbf{x}}_2$ can be computed according to (5). This closes one cycle of the two-grid iterative scheme. Later, the same procedure could be performed in cycles until the desired accuracy of the approximated solution $\widetilde{\mathbf{x}}$ is achieved. A simplified flowchart of one cycle of a two-grid algorithm is given in Fig. 2.

It is readily apparent that two-grid method does not have any practical meaning, because of two things: *(a)* if we need an accurate solution, a dense grid must be dense enough, which in turn means that the coarse grid, although coarser than the dense grid, will still be rather dense for the exact solution of (4), and *(b)* exactly solving (4) will not be much faster than the iterative solution of the governing equation (1) itself. Fortunately, one can easily see that (1) and (4) have the same shape, and the same global matrix, only different right hand side source vectors. Therefore, instead of solving (4) exactly we can recursively use the same way of thinking like in the two-grid method and try to solve (4) approximately instead of exactly by developing a new coarser grid than the original coarse grid – *level #1*. Let us call this new coarse grid *level #0*. Now we have three grid levels, dense grid (*level #2*), intermediate grid (*level #1*) and the coarse one (*level #0*). Therefore, our two-grid method is automatically extended into the three-grid method. Using the same reasoning we easily develop the idea of the multigrid solution method.

B. Types of Multigrid Solution Methods

As already mentioned, there is not a single multigrid algorithm which can be applied with same success to all problems. On contrary, the user must accommodate the problem and find the algorithm which is most adequate to his/her problem. Therefore, various multigrid algorithms have been established and can be found in the literature [2–8]. However, several algorithms have attracted the largest interest due to their efficiency and easy application: the *V-cycle*, the *W-cycle* and the *F-cycle*. While the first two obtained their names according to the shape of their main iterative cycle, the last one got its name from its full name, the *full multigrid method.* Typical cycles for V-cycle and W-cycle multigrid methods are given in Fig. 3. As can be seen, the number of smoothing steps m can be different and depends on the problem. Addition-



(a) V-cycle with and without post-smoothing

(b) W-cycle without post-smoothing

Fig. 3. V-cycle and W-cycle multigrid algorithms with four grids.

ally, smoothing steps can be performed before reaching the coarsest grid as well as after coarse grid computations, therefore two separate names exist — *pre-smoothing* and *post-smoothing* with two user defined numbers of iteration steps: m_1 – number of pre-smoothing iteration steps, and m_2 – number of post-smoothing iteration steps. In this paper, we will discuss only the V-cycle and W-cycle algorithms and the computer efficiency of both methods with and without post-smoothing.

C. Restriction and Prolongation Operators

From all of the above mentioned, one can easily see that the selection of a restriction matrix $\mathbf{R}$ and a prolongation matrix $\mathbf{P}$ have paramount importance in order to obtain a fast and stable multigrid solution algorithm. Various authors have already suggested several types of restriction and prolongation matrices such as *five-point star* or *nine-point star*, etc., which can be used effectively, however, mainly for finite difference computation schemes [5].

Figure 4 shows a common relationship between nodes of two nested 2D grids with adjoint levels l and $l-1$, where level l corresponds to the dense grid and level $l-1$ to the coarse grid. If we use first order triangular finite elements then the obtained results have C^0 continuity, i.e. they have linear continuity of the approximation function and constant values of its first derivatives at any point inside the finite elements. Therefore, at midpoints of all three edges of an element, the approximated function can be computed as a linear combination of its values at all three nodes. Using this reasoning, in our research for the definition of the restriction and prolongation matrices we used a simple ***straight injection method***, by which the value of the residual at each midpoint is equal to the half sum of the residuals at both terminal nodes of that edge. This simple prolongation method was chosen for two reasons: (a) because it has the same interpolation order as the order of the finite element approximation, and (b) because it is easy to implement numerically. After the prolongation matrix $\mathbf{P}$ is once defined as above, the restriction matrix $\mathbf{R}$ can easily be computed using the following relationship

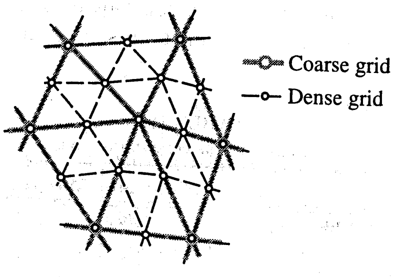


Fig. 4. Relationship between nodes of two adjoint grids.

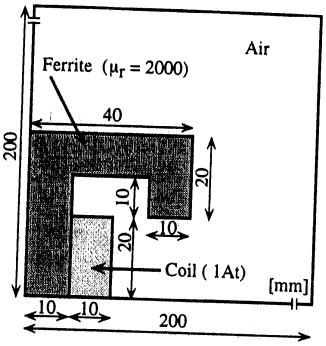


Fig. 5. Analyzed model of C-type ferrite and source coil.

$$\mathbf{R} = c\,\mathbf{P}^{\mathrm{T}} \quad , \tag{6}$$

where, c is a constant and the T stands for matrix transposition. Defining prolongation and restriction matrices according to (6) has one additional benefit: it provides a straightforward way for the computation of the system matrix for level $l-1$ if the system matrix for level l is already know as follows

$$\mathbf{K}_{l-1} = \mathbf{R}\,\mathbf{K}_l\,\mathbf{P} \quad . \tag{7}$$

Equation (7) opens new perspectives for the multigrid method, where instead of generating a set of grids with different grid density levels, one can generate only the densest grid level and all lower grid levels and respectively their system matrices can be generated automatically, or, starting from the coarsest grid one can adaptively increase the level of density using some kind of error evaluation method and adaptive mesh generation algorithm. In this paper, we use the former technique.

III. Investigation of the Efficiency of the Multigrid Method

A. Analyzed Model

Next, we investigate the efficiency of the multigrid solution method for electromagnetic field computations. A 2D magnetostatic model which consist of a C-type ferrite and a source coil is shown in Fig. 5. Four different nested meshes were automatically generated using the initial coarse mesh division. The number of nodes and finite elements for each grid are given in Table I.

TABLE I
NUMBER OF NODES AND ELEMENTS PER GRID LEVEL

Mesh	Number of nodes	Number of elements
level #1	78	129
level #2	284	516
level #3	1083	2064
level #4	4229	8256

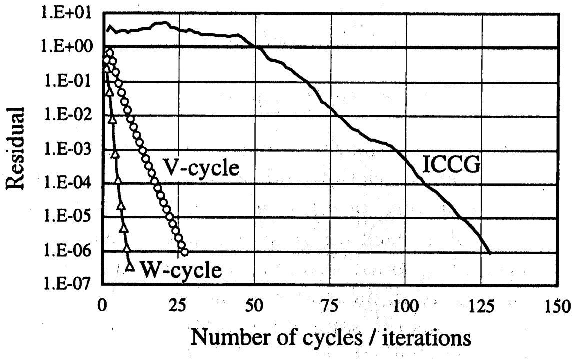


Fig. 6. Comparison of the convergence rate.

B. *V-cycle vs. W-cycle*

Initially, we made the comparison between the convergence rates and the computation speed for the V-cycle and the W-cycle without post-smoothing step on one side and the ordinary ICCG method as most widely used iterative method for the solution of a system of linear algebraic equations on the other side. Figure 6 shows the convergence characteristics for all three solution methods, V-cycle and W-cycle multigrid method and the ICCG method, respectively. It is easily visible that both multigrid methods exhibit fast and stable convergence with convergence rates larger than that of the ICCG method. Additionally, the W-cycle converges faster and with a smaller number of iteration cycles than the V-cycle multigrid method. The computation time for all three methods is presented in Table II. As can be seen, although the number of computation operations per cycle for the W-cycle multigrid is larger than that for the V-cycle multigrid, because these operations are performed on grid levels as much coarse as possible, (see Fig. 3(b)), W-cycle is still faster than V-cycle and should be preferred. Regarding memory usage both multigrid methods use approximately the same amount of computer memory which also prefers the W-cycle multigrid method as faster iterative solution method.

C. *Post-smoothing Influence on Convergence Rate*

As shown above, the W-cycle exhibits a faster convergence mainly as a result of the fact that most parts of the computation are performed on coarse grids. Therefore, one can easily conclude that the computation of highly accurate results on a coarse grid could automatically lead to a more accurate solution with a smaller number of iteration cycles and hence would be faster. To investigate this property of the multigrid method we made a comparison of the number of iterative steps, computation time and the convergence rate for W-cycles with and without post-smoothing, respectively. The obtained results are summarized in Table III.

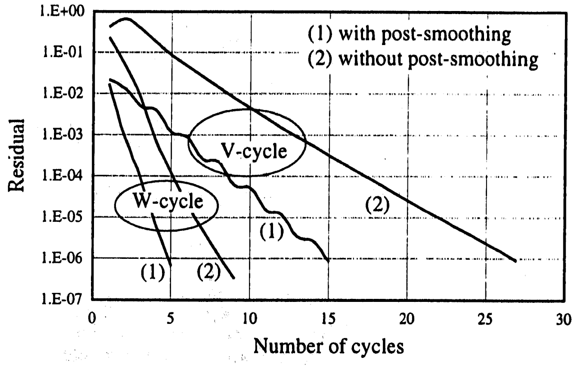


Fig. 7. Comparison of the convergence rates for V-cycle and W-cycle multigrid method with ($m_1 = m_2 = 2$) and without post-smoothing procedure ($m_1 = 2, m_2 = 0$).

As can be seen, the post-smoothing process could considerably improve the performance of the multigrid solution method. For a small number of relaxation steps ($m_1 = m_2 \leq 2$) the convergence rate improves and the number of cycles and the computation time decreases. On contrary, if the number of steps increases then multigrid without post-smoothing is faster than with post-smoothing. The computation time and the number of iteration cycles was optimal for $m_1 = m_2 = 2$, resulting in the W-cycle multigrid method being almost 5 times faster than the original ICCG method. Figure 7 shows the comparison of the convergence rate for the V-cycle and W-cycle multigrid methods with and without post-smoothing. It is apparent that the convergence rate improves using post-smoothing. The improvements, however, can be different for different problems and it is on the user to define the optimum value of relaxation steps according to his/her problem.

D. *Multigrid Method for Large Scale Problems*

We showed that even for a modest problem size the multigrid method (especially W-cycle with post-smoothing) is much faster than the currently widely used ICCG method. However, it is of paramount importance to investigate how this conclusion can be generalized for large scale problems, i.e. problems with a huge num-

TABLE II
COMPARISON OF THE COMPUTATION TIME

	V-cycle	W-cycle	ICCG
Computation time [sec]	0.96	0.47	1.52
Number of cycles/iterations	27	9	128
Memory usage [MB]	6.5	6.5	4.5

SGI Indigo2

TABLE III

POST-SMOOTHING INFLUENCE ON THE MULTIGRID ITERATIVE PROCESS

Number of relaxation steps	Without post-smoothing		With post-smoothing	
W-cycle	*Cycles*	*CPU*	*Cycles*	*CPU*
$m_1 = m_2 = 1$	15	0.63	8	0.42
$m_1 = m_2 = 2$	9	0.47	5	0.34
$m_1 = m_2 = 3$	7	0.42	7	0.46

CPU time in [sec], SGI Indigo2

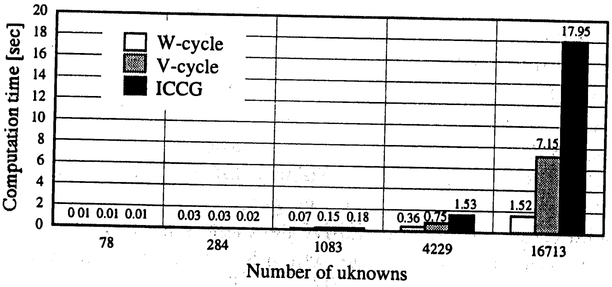


Fig. 8. Computation time of multigrid method vs. size of the problem.

ber of unknowns. These problems which need numerical solutions of systems of several hundred thousands up to millions simultaneous equations, recently appeared in numerical computation such as in very complex three dimensional field problems, fluid dynamics, transient phenomena analysis and simulations, etc. For such problems, a large amount of memory and computation time is necessary. We present here a simple investigation of the efficiency of the multigrid method over the ordinary ICCG method as a function of the size of the problem, i.e. the number of equations in the system. The obtained results are summarized in Fig. 8. It can be seen that the power of the multigrid method increases strongly with the increase of the size of the system that has to be solved. For example, if the multigrid is only five times faster than the ICCG method for the solution of a system of about $5,000$ unknowns, by doubling the number of unknowns the speed-up ratio becomes more than 10 in favor for the multigrid method. Therefore, one can conclude that the efficiency of the multigrid method increases with the increase of the size of a system of equations that has to be solved.

Regarding electromagnetic field analysis, multigrid has a wide potential, especially for the solution of problems that result in less sparse system matrices, or even with full matrices such as 3D eddy-current analysis, transient electromagnetic phenomena or simulations of coupled electromechanical or electro-thermal field problems. Additionally, the multigrid method opens a wide research area regarding adaptive solution methods and a possibility for a fast and accurate solution of a large class of problems in connection with appropriate error estimation criteria and mesh subdivision algorithms. It is also very important to extend the application area of multigrid methods using non-nested grids and to develop new and more effective restriction and prolongation methods adequate to one's own problem. A comparison of the efficiency of the multigrid method based on nested and non-nested finite element meshes is very interesting and is planned as a future work.

IV. Conclusions

We presented an investigation on the efficiency of the multigrid method for the solution of a system of linear algebraic equations obtained using the finite element method. The convergence rates and the computation speed of two main algorithms based on the so-called V-cycle and W-cycle multigrid were compared with that of the ordinary ICCG method. All multigrid solutions were based on nested finite element meshes due to its simplicity and computational efficiency. The influence of the post-smoothing procedure and the efficiency of the multigrid method in correlation with the number of unknowns were also investigated. Multigrid methods largely improve the convergence rate and decrease the computation time in comparison with the ICCG method. Additionally, the efficiency of multigrid methods increase strongly with the increase of the size of the problem which has to be solved.

References

[1] G. H. Golub and C. F. Von Loan, *Matrix Computations*, Third Edition, The Johns Hopkins University Press Ltd., London, 1996.

[2] A. Brandt, "Multi-level adaptive solutions to boundary value problems", *Mathematics of Computation*, 31, 1977, pp.333-390.

[3] W. Hackbusch and U. Trottenberg, *Multigrid Methods*, Springer-Verlag, Berlin, 1982.

[4] J. Bramble, *Multigrid Methods*, Pitman Research Notes in Mathematics, Longman Scientific and Technical, Harlow, United Kingdom, 1993.

[5] W. Hackbusch, *Multi-Grid Methods and Applications*, Springer-Verlag, Berlin, 1985.

[6] S. McCormick, *Multigrid Methods*, Volume 3 of the SIAM Frontiers Series, SIAM, Philadelphia, 1987.

[7] S. McCormick, *Multilevel Adaptive Methods for Partial Differential Equations*, Volume 6 of the SIAM Frontiers Series, SIAM, Philadelphia, 1989.

[8] U. Rüde, *Mathematical and Computational Techniques for Multilevel Adaptive Methods*, Volume 13 of the SIAM Frontiers Series, SIAM, Philadelphia, 1993.

[9] W. H. Press et al., *Numerical Recipes in FORTRAN*, The Art of Scientific Computing, Second Edition, Cambridge University Press, 1992.

[10] P. Wesseling, *An Introduction to Multigrid Methods*, John Wiley and Sons, 1992.

[11] V. Čingoski et al., "Computer Simulation of a Three-Phase Brushless Self-Exciting Synchronous Generator," Proceedings of the the 8th CEFC Conference, Tucson, Arizona, 1998, pp.303.

[12] R. Mertens et al., "An Algebraic Multigrid Method for Solving Very Large Electromagnetic Systems," *IEEE Trans. on Magnetics*, Vol. 34, No. 5, 1998, pp.3327-3330.

[13] I. Tsukerman et al., "Multigrid methods for computation of magnetostatic fields in magnetic recording problems," *Journal of Applied Physics*, Vol. 83, No.11, 1 June 1998, pp.6344-6346.