情報処理学会論文誌　コンピューティングシステム　Vol. 3　No. 3　189–198 (Sep. 2010)

マルチコアプロセッサシステムによる高速有限要素電磁界解析

美舩　健[†1]　廣谷　迪[†1]　岩下武史[†1]
村山敏夫[†2]　大谷秀樹[†2]

本論文では，マルチコアプロセッサシステム上における効率的な有限要素電磁界解析手法の開発を行う．有限要素法により導かれる連立一次方程式の解法として幾何マルチグリッド法を採用し，Arnold，Folk，Winther のスムーザを並列化するため，ブロックマルチカラーオーダリングの一種と解釈できるオーダリング手法を提案した．数値実験により，オーダリングのブロックサイズと色数を適切に設定することで，良好な並列性能が得られることが示された．

A Fast Finite Element Electromagnetic Analysis on Multi-core Processer System

Takeshi Mifune,[†1] Yu Hirotani,[†1] Takeshi Iwashita,[†1]
Toshio Murayama[†2] and Hideki Ohtani[†2]

This paper presents a fast finite element (FE) electromagnetic analysis on multicore processer systems. A geometric multigrid method is used to solve the linear equations arising from the FE formulation. A special ordering technique, based on the concept of block multicolor ordering, is proposed for parallelization of the Arnold, Folk and Winther's smoother. Numerical tests show a good parallel performance of the presented multigrid solver with an appropriate adjustment of the number of colors and block size.

†1 京都大学
Kyoto University
†2 ソニー株式会社
Sony Corporation

1. はじめに

実用的・先進的な電気電子機器の設計開発において有限要素法を用いる電磁界シミュレーション[1] が大きな役割を果たしている．有限要素解析には，その規模が大きくなるほど多大な計算コストが消費される．近年では計算機性能の向上にともなって数百万自由度以上の規模の有限要素解析を PC 上で行うことも可能となってきているが，中規模以上の有限要素解析に費やされる計算は依然として大きく，高速化が望まれている．一方最近ではマルチコアプロセッサの商用化・普及が進み，スレッド並列処理を有効活用した計算高速化の重要性が高まっている．そこで本研究では，マルチコアプロセッサを活用した高速有限要素電磁界解析について検討を行う．

本研究では特に，電界を未知数にとった有限要素法（E 法）による高周波電磁界解析[1] を扱う．最近では電気電子機器の発する電磁波の外界におよぼす影響が様々な場面で注目されていることもあり，高周波電磁界問題を対象とした数値解析の重要性は高まっている．また高周波電磁界解析においては，電磁波の波長に対して十分細密なメッシュ分割を全解析領域について行う必要があるため，解析の大規模化・高速化の必要性も高い．

規模の大きい有限要素解析の計算時間のほとんどは，偏微分方程式の離散化によって導出される連立一次方程式の求解に費やされる．したがってこの求解部を高速化することが，解析全体の高速化のために重要である．有限要素法により導かれる連立一次方程式の係数行列はほとんどの場合スパースであり，中規模以上の解析においては反復解法[2),3)] を用いた求解が有力である．本研究では，マルチグリッド法[4)–6)] を利用した反復解法を適用する．マルチグリッド法は有限要素法に現れる連立一次方程式に対する現在最も有力な解法の 1 つであり，その特徴は，自由度の異なる複数の補助的な連立一次方程式を利用して，解くべき連立一次方程式に対する反復の収束を加速させる点にある．ただし本研究では，マルチグリッド法を単独の反復解法としてではなく，クリロフ部分空間法の前処理[3)] として使用する．

マルチグリッド法のアルゴリズムの並列処理は，スムーザ[5)] と呼ばれる基礎的反復法の部分以外については，原理的には容易である．しかしながら，スムーザとしてたとえばガウスザイデル法のように本質的に逐次性を持つような解法を用いる場合には，その部分の並列化には特別な工夫が必要となる．高周波有限要素解析におけるスムーザとしては，curl 演算のカーネル成分を適切に扱う特殊な手法が必要となることが知られている[7)–10)]．本研究では文献 7) で提案されたスムーザ（以下では AFW（Arnold，Fork，Winther）スムーザと表記する）を採用するが，AFW スムーザはブロックガウスザイデル法の一種であるため本

質的に逐次的な計算が含まれている．そこで本研究では，AFW スムーザのスレッド並列処理を可能にするため，ブロックマルチカラーオーダリング[11)–14)] の一種と解釈できる特別なオーダリング手法を提案する．通常のブロックマルチカラーオーダリングでは，連立一次方程式の各自由度（未知数）に対して “色” と呼ばれる識別番号を与えてグループ（色）分けが行われる．これに対して提案手法では，色分けの対象が自由度そのものではない点が特徴である．

テストモデルについて OpenMP によるスレッド並列化を施した高周波有限要素解析を行い，提案手法のマルチコアプロセッサシステムにおける有効性を検証する．さらに提案手法中で使用する色数等のパラメータの適切な選択について考察する．

2. マルチグリッド法を用いた高周波有限要素解析

2.1 高周波有限要素解析

電界 $\boldsymbol{E}$ を未知数とした高周波解析の基礎方程式は

$$\nabla \times [\nu(\nabla \times \boldsymbol{E})] - \omega^2 \left(\epsilon + \frac{\sigma}{\mathrm{i}\omega}\right) \boldsymbol{E} = -\mathrm{i}\omega \boldsymbol{J}_0 \tag{1}$$

と表される[1)]．ここで，i，ω，σ，ϵ，ν，$\boldsymbol{J}_0$ はそれぞれ，虚数単位，励振角周波数，導電率，誘電率，磁気抵抗率，印加電流密度を表す．式 (1) について辺要素[1)] を用いた有限要素定式化を行うことにより，複素対称疎行列を係数行列とする連立一次方程式が導出される．辺要素を用いる場合，連立一次方程式の自由度は有限要素メッシュの各辺に割り当てられる．つまり，有限要素メッシュの辺の数が，連立一次方程式の自由度の数に等しくなる．

2.2 マルチグリッド法による求解

一般に有限要素解析では，解析規模が大きくなるほど係数行列が悪条件化し，従来の反復解法による求解の反復収束性が悪化する傾向がみられる．それに対してマルチグリッド法は，理想的には反復収束性が問題の規模に依存しないという性質を持っており，大規模な問題では，IC（Incomplete Cholesky）分解等を前処理とする標準的反復解法と比較して格段に高速であることが期待される．

マルチグリッド法では複数のグリッド G_m が設定され，それぞれのグリッドについて連立方程式が定義される．ここではグリッドの数を M とおき，それらの方程式を

$$A_m \boldsymbol{x}_m = \boldsymbol{b}_m \tag{2}$$

と表す（$1 \leq m \leq M$）．ここで A_m，$\boldsymbol{x}_m$，$\boldsymbol{b}_m$ はそれぞれ各方程式の係数行列，解ベクトル，右辺ベクトルを表し，

$$A_m \in \mathbb{C}^{N_m \times N_m}, \tag{3}$$

$$\boldsymbol{x}_m \in \mathbb{C}^{N_m}, \tag{4}$$

$$\boldsymbol{b}_m \in \mathbb{C}^{N_m} \tag{5}$$

である．N_m は，グリッド G_m における自由度の数を表す．$m = 1$ の方程式が有限要素法から導出された本来解くべき問題であり，これに対して高速な求解を行うために他の方程式が利用される．通常，$N_1 > N_2 > \ldots > N_M$ であり，2 つのグリッド Ω_m，Ω_{m+1} を比較して，Ω_m をファイングリッド，Ω_{m+1} をコースグリッドと呼ぶ．

V サイクル[5)] と呼ばれる代表的なマルチグリッドアルゴリズムは，グリッド G_m の方程式に対して近似解ベクトル $\boldsymbol{v}_m$ を更新する手続きとして次のように再帰的に記述できる．

アルゴリズム $\mathrm{MGV}(A_m, \boldsymbol{b}_m, \boldsymbol{v}_m)$

(1)　$m = M$ ならば，$A_M \boldsymbol{x}_M = \boldsymbol{b}_M$ の近似解 $\boldsymbol{v}_M$ を直接法等により求める．$m \neq M$ ならば (2) 以下を実行する．

(2)　$\mathrm{S}_{\mathrm{pre}}(A_m, \boldsymbol{b}_m, \boldsymbol{v}_m)$

(3)　$\boldsymbol{b}_{m+1} \leftarrow R_m(\boldsymbol{b}_m - A_m \boldsymbol{v}_m), \boldsymbol{v}_{m+1} \leftarrow 0$

(4)　コースグリッドにおける V サイクル：$\mathrm{MGV}(A_{m+1}, \boldsymbol{b}_{m+1}, \boldsymbol{v}_{m+1})$

(5)　$\boldsymbol{v}_m \leftarrow \boldsymbol{v}_m + P_m \boldsymbol{v}_{m+1}$

(6)　$\mathrm{S}_{\mathrm{post}}(A_m, \boldsymbol{b}_m, \boldsymbol{v}_m)$

上記のステップ (1) におけるグリッド G_M の方程式の求解は，N_M が十分小さくなるようグリッドを設定することで，適当な解法を用いて容易に行えるものとする．ステップ (2)，(6) に現れる手続き $\mathrm{S}_{\mathrm{pre}}$，$\mathrm{S}_{\mathrm{post}}$ は，適当な反復解法により近似解 $\boldsymbol{v}_m$ を更新する手続きであり，それぞれプリスムーザ，ポストスムーザと呼ばれる．またステップ (3)，(5) に現れる $R_m \in \mathbb{C}^{N_{m+1} \times N_m}$，$P_m \in \mathbb{C}^{N_m \times N_{m+1}}$ はそれぞれ，グリッド G_m から G_{m+1}，G_{m+1} から G_m の写像に用いられる行列である．

一般に，R_m，P_m の決定法の差異により，マルチグリッド法は幾何マルチグリッド法と代数マルチグリッド法の 2 つに分類される．本研究では幾何マルチグリッド法を採用するが，その特徴は R_m，P_m を決める際に解析対象の幾何情報を直接的に利用することである[6),15)]．この場合，疎密の異なる複数の有限要素メッシュを作成し，各グリッド G_m に割り当てる必要がある．

G_1 以外の係数行列 A_m（$m > 1$）は，本研究ではガラーキン法[5)]

$$A_{m+1} = R_m A_m P_m \tag{6}$$

によって決定する．

アルゴリズム MGV$(A_1, \boldsymbol{b}_1, \boldsymbol{v}_1)$ 自体を反復解法として用いることもできるが，適当なクリロフ部分空間法の前処理として用いることにより反復の収束を加速することも可能である．本研究ではクリロフ部分空間法の 1 つである COCR（Conjugate Orthogonal Conjugate Residual）法[16] を採用する．COCR 法を導入すると 1 反復あたりの計算コストが増加するため，収束加速の効果がそれを上回らなければ，期待に反して求解性能が低下することもありうる．しかしながら本研究では，$\mathrm{S}_{\mathrm{pre}}$，$\mathrm{S}_{\mathrm{post}}$ として用いる AFW スムーザの計算コストが大きいため，COCR 法の導入による計算コストの増分は相対的に小さい．このため，COCR 法の導入による計算コストの増加よりも収束性の改善の効果が大きいことが期待できる．

2.3　AFW スムーザ

辺要素を用いた高周波電磁界解析にマルチグリッド法を適用する場合，良好な収束性を得るためにはスムーザ $\mathrm{S}_{\mathrm{pre}}$，$\mathrm{S}_{\mathrm{post}}$ として，AFW スムーザ[7),10),⋆1] あるいは Hiptmair によるハイブリッドスムーザ[8] 等の特殊なスムーザを用いる必要がある．本研究では AFW スムーザを使用する．

以下では，グリッド G_m における有限要素メッシュの節点，辺の集合をそれぞれ Ψ_m，Ω_m で表し，各節点 $i \in \Psi_m$ に接続する辺の集合（節点パッチと呼ぶ）を $\Omega_{m,i} \subset \Omega_m$ とする．**図 1** に，直方体メッシュを用いた場合の，節点パッチの例を示す．

グリッド G_m における AFW スムーザでは，各節点 $i \in \Psi_m$ に対して順に，$\Omega_{m,i}$ をブロックとしてブロックガウスザイデル更新が行われる．

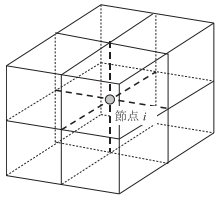

図 1　節点パッチ $\Omega_{m,i}$（破線で示した辺の集合）の例
Fig. 1　Example of node-based patch $\Omega_{m,i}$.

⋆1 文献 10) では，同手法を節点パッチガウスザイデルスムーザと呼んでいる．

3.　OpenMP を利用したスレッド並列処理

本章では，前章に述べた反復解法の OpenMP によるスレッド並列処理について述べる．マルチグリッド法を前処理とする COCR 法において計算コストの大部分を占める処理は，

- COCR 法におけるベクトルの加減算および内積計算，行列—ベクトル積
- マルチグリッドアルゴリズム MGV で用いる行列 A_m（$m > 1$）を求める行列—行列積（式 (6)）
- MGV 内の行列 R_m，P_m に関する行列—ベクトル積
- MGV 内のスムーザ $\mathrm{S}_{\mathrm{pre}}$，$\mathrm{S}_{\mathrm{post}}$（AFW スムーザ）

である．ただし，アルゴリズム MGV でグリッド G_M について使用される直接解法の計算コストは，N_m が十分小さくなるようコースグリッドを設定することで無視できるとする．

上にあげた処理のうち，スムーザ以外のベクトルの加減算および内積計算，行列—ベクトル積，行列—行列積は，原理的に並列処理に向いた計算であり，OpenMP によるスレッド並列化は比較的容易である．ただし，ベクトルの加減算および内積計算，行列—ベクトル積についてはコンパイラによる自動並列化が十分期待できるため，コンパイラの自動並列機能を利用する．

3.1　マルチカラー AFW スムーザ

AFW スムーザはブロックガウスザイデル法の一種であり，本質的に逐次的な処理を含むため，並列処理にはなんらかの工夫が必要である．本研究では，AFW スムーザの並列処理を可能にするために，各節点に関する更新処理の順序を適切に設定する．本手法は，ブロックマルチカラーオーダリングと呼ばれている手法の一種と考えられる．まず本節ではブロック化を行わないマルチカラーオーダリングの導入について説明し，次節でブロック化について述べる．

通常のガウスザイデル法をマルチカラーオーダリング[3),11)] を用いて並列化する場合，各自由度に “色” と呼ばれる識別番号を与え，同色に属する節点をグループとしてまとめる．このとき，すべての色について，その色に属するすべての自由度に関するガウスザイデル更新計算が互いに干渉しないように色番号が設定される．これにより，同色に属する自由度に関する更新を独立に行うことが可能になる．

本研究で使用する AFW スムーザでは，更新計算の単位が各節点に対する節点パッチとなっているので，各自由度に色を与える代わりに，各節点に色を与えることを考える．同色に属する節点パッチに関する更新計算が互いに依存関係を持たないような色分けを行えば，

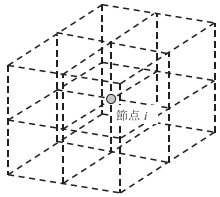

図 2　直方体要素を用いる際の $\bar{\Omega}_{m,i}$ $(=\bar{\Omega}^T_{m,i})$
Fig. 2　$\bar{\Omega}_{m,i}$ $(=\bar{\Omega}^T_{m,i})$ when using the brick elements.

並列処理が可能となる．しかしながら色の設定に際しては，後述するように AFW スムーザの具体的な実装法の違いによって，節点パッチ間の計算依存関係に差異が現れることに注意しなければならない．

ここで $[\cdot]_i$，$[\cdot]_{ij}$ はベクトルの i 成分，行列の ij 成分を表すものとし，次のような集合

$$\bar{\Omega}_{m,i} = \{j \in \Omega_m : \exists k \in \Omega_{m,i}, [A_m]_{kj} \neq 0\} \tag{7}$$

$$\bar{\Omega}^T_{m,i} = \{j \in \Omega_m : \exists k \in \Omega_{m,i}, [A_m]_{jk} \neq 0\} \tag{8}$$

を定義する．本研究で扱う係数行列のように対称性がある場合は，$\bar{\Omega}_{m,i} = \bar{\Omega}^T_{m,i}$ である．また，$[A_m]_{ij} \neq 0$ となるのは辺 i，j を両方含むような直方体要素が存在するときのみである．直方体要素を使用する場合の $\bar{\Omega}_{m,i}$（$=\bar{\Omega}^T_{m,i}$）を**図 2** に示す．

AFW スムーザの効率的な実装法としては以下の 2 通りが考えられる．以下では誤解の恐れがない場合には，グリッド番号を表す添字 m を省略する．

(a) 各節点パッチ Ω_i に関するブロックガウスザイデル更新を行う．つまりベクトル成分 $[\boldsymbol{v}]_j$（$j \in \Omega_i$）が更新される．この際，$[\boldsymbol{b}]_j$（$j \in \Omega_i$）と $[\boldsymbol{v}]_j$（$j \in \bar{\Omega}_i$）に対する参照が必要となる．

(b) まず残差 $\boldsymbol{r} = \boldsymbol{b} - A\boldsymbol{v}$ を求めた後，各節点 i について順次，以下の処理を行う．

（1） 節点パッチ Ω_i に関するブロックガウスザイデル更新を行う．つまり $[\boldsymbol{v}]_j$（$j \in \Omega_i$）を更新する．このとき，$[\boldsymbol{r}]_j$（$j \in \Omega_i$）が参照される．

（2） 残差ベクトルの更新を行う．ここでは $[\boldsymbol{r}]_j$（$j \in \Omega_i$ または $j \in \bar{\Omega}^T_i$）の更新と，$[\boldsymbol{v}]_j$（$j \in \Omega_i$）の参照が必要となる．

上記の実装 (a)，(b) において，同色に属する任意の異なる節点 i_1 と i_2 に関する更新計算が依存関係を持たない条件（すなわち並列化可能のための条件）は，

- i_1 に関して更新する成分と，i_2 に関して更新する成分
- i_1 に関して更新する成分と，i_2 に関して参照する成分

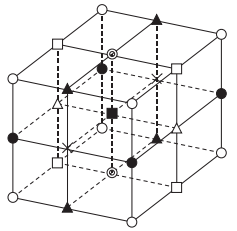

図 3　マルチカラーオーダリングの例（$l_x = l_y = l_z = 2$）
Fig. 3　Example of the multicolor ordering ($l_x = l_y = l_z = 2$).

の間に重なりがないことである．この条件は具体的には実装 (a) と (b) で異なることに注意が必要である．つまり，(a) では Ω_{i_1} と Ω_{i_2}，Ω_{i_1} と $\bar{\Omega}_{i_2}$，(b) では Ω_{i_1} と Ω_{i_2}，$\bar{\Omega}^T_{i_1}$ と $\bar{\Omega}^T_{i_2}$，Ω_{i_1} と $\bar{\Omega}^T_{i_2}$，それぞれに重なりがないことが並列化の条件となる．

直方体分割による有限要素メッシュの使用を仮定し，上の条件を満たす色分けを行うために次式を用いて節点 (i,j,k) の色を決定することを考える．ここで i，j，$k = 0, 1, \ldots$ は x，y，z 各方向について節点の位置を示すインデクスとする．

$$\mathrm{color}(i,j,k) = \mod(i, l_x) + l_x \mod(j, l_y) + l_x l_y \mod(k, l_z) \tag{9}$$

ここで l_x，l_y，l_z は適当な整数であり，上式は，ある節点に対して x，y，z 各方向にそれぞれ l_x，l_y，l_z の倍数だけストライドした位置の節点がすべて同色となることを意味する．使用する色の数は，$l_x l_y l_z$ となる．**図 3** に，$l_x = l_y = l_z = 2$ としたときの色分けの例を示す．

上の議論と図 1 および図 2 を考慮すれば，AFW スムーザの並列化のためには，実装 (a) では l_x，l_y，$l_z \geq 2$，実装 (b) では l_x，l_y，$l_z \geq 3$ が要求されることが分かる．

特に疎行列 A_m を CRS（Compressed Row Storage）形式[2] で格納する場合には実装 (a)，CCS（Compressed Column Storage）形式[2] で格納する場合には実装 (b) が効率的である．本研究では CRS 形式を採用するが，行列の対称性を利用すれば，実装 (a)，(b) どちらも効率的に実装することができる．ただしこの場合は，対称性を利用して行列の記憶に要するメモリを削減することはできない．

本研究では，MGV$(A_1, \boldsymbol{b}_1, \boldsymbol{v}_1)$ を前処理として用いるため，その初期近似解 $\boldsymbol{v}_1$ は零と考えてよい．さらにステップ (3) で $\boldsymbol{v}_{m+1} \leftarrow 0$ としているので，ステップ (2) でプリスムーザ $\mathrm{S_{pre}}$ を適用する際には $\boldsymbol{v}_m = 0$ である．この場合，プリスムーザ $\mathrm{S_{pre}}$ としては実装 (b) を使用するのが効率的である．なぜならば (b) で初期残差を計算する必要はなくなり，ステップ (3) に現れる $\boldsymbol{b}_m - A_m\boldsymbol{v}_m$ には (b) の終了時に保持している残差ベクトルを与えれ

ばよいからである．一方実装 (a) は，終了時に残差ベクトルを保持していない代わりに，初めに残差ベクトルを計算する必要がない．したがってポストスムーザ S_{post} として (a) を使用するのが適切である．実装 (a) と (b) を併用する場合には，並列処理のためには式 (9) で l_x，l_y，$l_z \geq 3$ とする必要がある．

3.2　ブロック化

マルチカラーオーダリングにおいて，色決めに際して複数の自由度をブロックとしてまとめることで，反復収束性およびキャッシュの利用効率を改善できることが知られている[12),13)]．ブロック化は前節に述べたマルチカラー AFW スムーザにおいても可能であり，少なくともキャッシュヒット率の改善が期待できる．本論文では，ブロックマルチカラーオーダリングを導入した AFW スムーザをブロックマルチカラー AFW スムーザと呼ぶことにする．

本研究ではブロックサイズを x，y，z 方向それぞれについて m_x，m_y，m_z とし，次式に従って節点 (i,j,k) の色を決定する．

$$\mathrm{color}(i,j,k) = \mod(\lfloor i/m_x \rfloor, l_x) + l_x \mod(\lfloor j/m_y \rfloor, l_y) + l_x l_y \mod(\lfloor k/m_z \rfloor, l_z) \tag{10}$$

ここで，$\lfloor\rfloor$ は床関数を表す．前節で述べたオーダリングは，式 (10) で $m_x = m_y = m_z = 1$ としたときに相当する．**図 4** に，$l_x = l_y = l_z = m_x = m_y = m_z = 2$ のときの色分けの例を示す．

各ブロック内の計算は逐次的に行う必要があるが，ブロックサイズを適当に調整することで，使用するスレッド数に見合った並列度を得つつキャッシュ効率の改善を図ることが可能である[12),13)]．ブロックサイズを 2 以上にとる場合には，前節で述べた実装 (b) を用いるときの条件は緩和され，l_x，l_y，$l_z \geq 2$ となる．

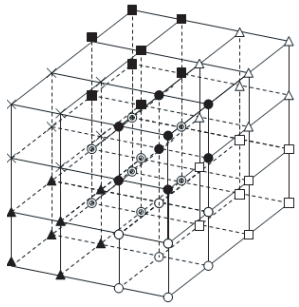

図 4　ブロックマルチカラーオーダリングの例（$l_x = l_y = l_z = m_x = m_y = m_z = 2$）
Fig. 4　Example of the block multicolor ordering ($l_x = l_y = l_z = m_x = m_y = m_z = 2$).

式 (10) による色分けを行うとき，少なくとも

$$p = \left\lfloor \frac{n_x + 1}{l_x \times m_x} \right\rfloor \times \left\lfloor \frac{n_y + 1}{l_y \times m_y} \right\rfloor \times \left\lfloor \frac{n_z + 1}{l_z \times m_z} \right\rfloor \tag{11}$$

の並列度（独立に処理してよいブロックの数）が得られる．ここで n_x，n_y，n_z は x，y，z 方向の分割数である．並列処理の際，スレッド数に対してある程度大きな並列度が得られていなければ，ロードバランスが悪化し，良好な並列性能が得られない可能性がある．色数とブロックサイズを設定する際には，このことに注意する必要がある．

4.　数値解析結果

数値計算に用いたマルチコアプロセッサによる PC の計算機仕様を**表 1** に示す．解析コードは Fortran 95/2003 により記述し，コンパイラとして Intel Visual Fortran 11.0（最適化オプション “/O2”，並列化時には “/Qopenmp/Qparallel” を追加）を使用した．反復解法の収束判定基準を，(相対残差ノルム) $< 10^{-10}$ とした．

3 章で述べたブロックマルチカラー AFW スムーザの色数・ブロックサイズは各グリッドごとに設定することが可能であるが，以下ではすべてのグリッドについて同じ色数とブロックサイズを用い，$l_x = l_y = l_z = l$，$m_x = m_y = m_z = m$ とした．

4.1　テストモデル

図 5 および**図 6** に示す 2 つのテストモデルについて高周波有限要素解析を行う．

テストモデル 1 は，真空中に線状電流ソースを配置したものである．対称性を考慮した 1/8 モデルであり，12 層の PML（Perfectly Mached Layer）[1)] による吸収境界を x，y，z 方向それぞれに設定している．x，y，z 方向それぞれ 48 分割したメッシュを使用し，有限

表 1　使用した計算機
Table 1　Computers.

	PC1	PC2
OS	Windows XP professional x64	Windows XP professional x64
CPU	Intel Core 2 Extreme QX9770 3.2 GHz	Intel Core i7 950 3.06 GHz
コア数	4	4
キャッシュ	L2: 6 MB × 2	L2: 256 KB × 4 L3: 8 MB
主記憶	8 GB	12 GB

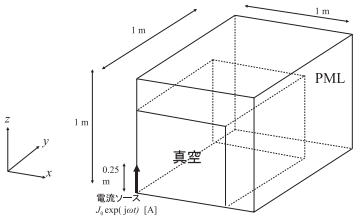

図 5　テストモデル 1
Fig. 5　Test model 1.

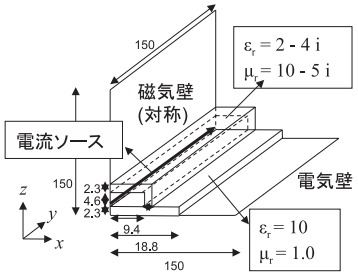

図 6　テストモデル 2 [mm]
Fig. 6　Test model 2 [mm].

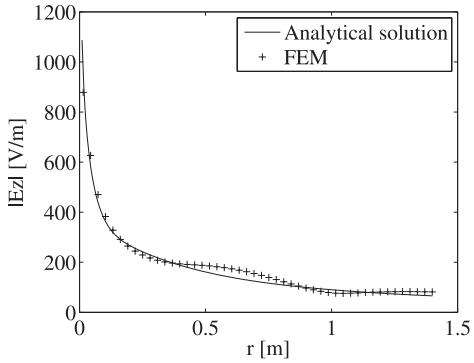

図 7　有限要素法による近似解と解析解との比較
Fig. 7　Comparizon between the FEM and analytical solution.

表 2　前処理の性能比較
Table 2　Performance comparison of the preconditioners.

前処理	マルチグリッド	IC 分解
テストモデル 1	24	2855
	(13.5)	(574.4)
テストモデル 2	15	18728
	(187.4)	(45732.0)

テストモデル 1，2 についてそれぞれ，PC1，PC2 を使用.
各欄の上段：反復回数，下段：求解時間 (s).

要素解析の自由度数は 345,744 である．ただし 4.6 節においては，メッシュ分割数および自由度数を変更した際の結果も提示する．

テストモデル 1 については解析解を計算することが可能であるので，解析コードの検証のため，有限要素法により得られる解との比較を行った．線状ソースからの距離を r として，電界の z 成分絶対値 $|E_z|$ を**図 7** に示す．PML の効果が不十分なためと考えられる誤差はみられるが，両者はおおむね一致しており解析コードの信頼性は確認できたと考えられる．

テストモデル 2（図 6）では，マイクロストリップラインを模擬しており，直線状電流ソースの周囲に損失性を有する材料を配置している．テストモデル 2 については，x，y，z 方向それぞれを 128 分割したメッシュを用いる．有限要素解析の自由度数は 6,390,144 である．

4.2　マルチグリッド前処理の効果

まず逐次計算において，高周波有限要素解析におけるマルチグリッド法の有効性を，従来広く使用されている IC 分解と比較することで確認する．テストモデル 1，2 について，マルチグリッド法および IC 分解を前処理とする COCR 法の求解性能を測定した結果を**表 2** に示す．使用した計算機はテストモデル 1，2 それぞれについて，PC1，PC2 である．マルチグリッド法の AFW スムーザについては，辞書式オーダリング（ブロックマルチカラーオーダリングにおいて m を 1 に，l をメッシュ分割数より大きく設定した場合に相当する）を使用する．

表 2 の比較に際して注意を要することは，実用上 IC 分解を用いる場合には，電界 E に加えて電気スカラポテンシャル ϕ を未知数として扱う E-ϕ 法[1] を使用するのが一般的であることである．IC 分解を前処理とした解法については，E 法に代えて E-ϕ 法を採用することで数倍程度の高速化が期待できる（たとえば文献 17) 等）．しかしながらこれを考慮しても，マルチグリッド法を使用した際の反復収束の速さと求解の高速さは顕著である．2.2 節にも述べたようにメッシュ分割が細密化するほどマルチグリッド法の優位性がさらに増すこ

とが予想されるため，大規模な高周波有限要素解析の高速化の観点においてマルチグリッド法の有用性は高い．

4.3　逐次計算時の求解性能

以下ではブロックマルチカラー AFW スムーザを使用した際のマルチグリッド解法の性能について検討を行う．まず本節では，スレッド並列化を行わない逐次計算において，ブロックマルチカラー AFW スムーザにおける色数とブロックサイズの求解性能への影響について調べる．

表 3，**表 4** に，テストモデル 1，2 についてそれぞれ PC1，PC2 上で解析を行ったときの反復回数および求解時間を示す．表 3，表 4 に共通する全体的傾向として，色数あるいはブロックサイズをある程度大きくとったときには，辞書式オーダリングと同等の良好な反復収束性が得られていることが分かる．本傾向は差分解析におけるブロックマルチカラーオーダリングの振舞いと一致しており，ある程度以上の色数・ブロックサイズにおいて反復回数の変動が少ないことは，パラメータ設定の容易さの観点から提案手法の長所といえる．

ブロックサイズ m を 1 とした場合，$m > 1$ のときと比較して反復回数が大きくなっているケースが多いが，加えて，求解時間が反復回数増加の割合以上に増加していることも分かる．これは，$m = 1$ では配列アクセスの連続性が失われキャッシュ効率が低下していることが原因と考えられる．ブロックサイズを 2 より増加させたときに求解時間と反復数の比が緩やかに減少していることも，キャッシュ効率の改善によるものと考えられる．計算時間の観点からは，色数にかかわらずブロックサイズをある程度大きくとれば，辞書式オーダリングのときと比較して遜色のない性能が得られている．

4.4　並列計算時の反復回数と計算時間

次に，スレッド並列計算時の求解性能について検討する．前節と同様にテストモデル 1，2 についてそれぞれ PC1，PC2 上で解析を行った結果を，**表 5**，**表 6** に示す．ただしテストモデル 1 についてはスレッド数 4，テストモデル 2 についてはスレッド数を 8（Core i7 のハイパースレッディング機能を使用）とした．3.1 節で述べたように，今回の実装では $l = 2$，$m = 1$ のときには並列処理を行うことはできない．

反復回数については，すべての設定において，逐次計算時と同一の結果が得られている．

並列性能の計測にあたっては，ブロックマルチカラー AFW スムーザにおいて得られる並列度 p が式 (11) で見積もれることを考慮して，グリッド G_1 においてスレッド数 N_{thread} が

$$p > 10N_{\mathrm{thread}} \tag{12}$$

を満たす範囲を，十分な並列度が得られる設定と考える．表 5，表 6 では，二重縦罫線より左側が式 (12) を満たす設定である．式 (12) を満たす範囲を超えて色数・ブロックサイズを大きくしたときに計算時間の観点から性能が低下していく傾向にあるのは，得られる並列度が不十分なためである．式 (12) を満たす範囲で色数・ブロックサイズを大きく設定すれば，おおむね良好な性能を得ることができている．

ブロックサイズを大きくしたときに求解時間/反復回数の比が改善されているのは，逐次計算と同様にキャッシュ効率の改善により説明できる．今回のテストでは顕著に現れていないが計算環境によっては，色数を増加させたときの同期コストの増加が深刻な影響を与える

表 3　逐次計算時の反復回数と求解時間（テストモデル 1）
Table 3　Number of iterations and solution time in sequential computing (Test model 1).

色数 l^3 ＼ブロックサイズ m	1	2	3	5	10	15
2^3	42 (28.9)	22 (14.9)	21 (13.3)	22 (13.1)	23 (13.3)	23 (12.9)
3^3	30 (23.7)	22 (15.2)	23 (14.4)	23 (14.3)	23 (13.0)	23 (12.9)
4^3	27 (23.3)	22 (15.2)	20 (13.3)	22 (13.1)	23 (12.8)	23 (12.8)
5^3	24 (21.3)	22 (14.5)	23 (13.7)	22 (12.5)	24 (13.1)	23 (12.8)
6^3	25 (20.0)	21 (12.8)	22 (12.7)	23 (12.9)	24 (13.1)	23 (12.8)

PC1 を使用．各欄の上段：反復回数，下段：求解時間 (s)．
（参考）辞書式オーダリングのときの反復回数：24 回，逐次求解時間：13.5 (s)

表 4　逐次計算時の反復回数と求解時間（テストモデル 2）
Table 4　Number of iterations and solution time in sequential computing (Test model 2).

色数 l^3 ＼ブロックサイズ m	1	2	3	5	10	15
2^3	15 (242.5)	16 (221.6)	15 (196.9)	16 (193.0)	16 (183.7)	16 (180.2)
3^3	19 (313.8)	15 (216.7)	16 (209.4)	15 (185.7)	17 (191.7)	17 (187.6)
4^3	20 (340.7)	16 (231.4)	15 (203.0)	16 (196.1)	17 (192.5)	15 (172.4)
5^3	17 (307.9)	17 (244.8)	15 (204.9)	15 (187.8)	16 (184.8)	16 (180.5)
6^3	17 (313.5)	17 (245.1)	17 (223.7)	15 (186.8)	16 (183.5)	16 (179.7)

PC2 を使用．各欄の上段：反復回数，下段：求解時間 (s)．
（参考）辞書式オーダリングのときの反復回数：15 回，逐次求解時間：187.4 (s)

表 5　並列計算時の反復回数と求解時間（4 スレッド，テストモデル 1）
Table 5　Number of iterations and solution time in parallel computing (Four threads, test model 1).

色数 l^3 ＼ブロックサイズ m	1	2	3	5	10	15
2^3	–	22	21	22	23	23
	(–)	(8.1)	(7.5)	(7.1)	(7.2)	(7.8)
3^3	30	22	23	23	23	23
	(14.7)	(8.5)	(8.3)	(7.4)	(8.1)	(10.2)
4^3	27	22	20	22	23	23
	(13.5)	(8.0)	(7.4)	(7.2)	(8.8)	(11.4)
5^3	24	22	23	22	24	23
	(10.1)	(7.9)	(7.3)	(7.1)	(11.5)	(11.3)
6^3	25	21	22	23	24	23
	(9.2)	(6.9)	(6.9)	(8.5)	(11.6)	(12.8)

PC1 を使用．各欄の上段：反復回数，下段：求解時間 (s).
（参考）辞書式オーダリングのときの反復回数：24 回，逐次求解時間：13.5 (s)
二重縦罫線より左の設定では，式 (12) が満たされる．

表 6　並列計算時の反復回数と求解時間（8 スレッド，テストモデル 2）
Table 6　Number of iterations and solution time in parallel computing (Eight threads, test model 2).

色数 l^3 ＼ブロックサイズ m	1	2	3	5	10	15
2^3	–	16	15	16	16	16
	(–)	(68.1)	(60.5)	(61.1)	(60.3)	(63.2)
3^3	19	15	16	15	17	17
	(96.6)	(65.5)	(64.4)	(58.9)	(64.4)	(69.0)
4^3	20	16	15	16	17	15
	(106.7)	(70.0)	(61.9)	(62.5)	(67.7)	(68.9)
5^3	17	17	15	15	16	16
	(96.8)	(74.1)	(62.0)	(60.2)	(68.0)	(78.6)
6^3	17	17	17	15	16	16
	(98.9)	(74.5)	(69.0)	(60.6)	(70.8)	(90.6)

PC2 を使用．各欄の上段：反復回数，下段：求解時間 (s).
（参考）辞書式オーダリングのときの反復回数：15 回，逐次求解時間：187.4 (s)
二重縦罫線より左の設定では，式 (12) が満たされる．

可能性が考えられる．したがってブロックマルチカラー AFW スムーザを使用する際には，色数よりもブロックサイズを大きくする設定が推奨できる．たとえばテストモデル 1 について $l = 2$，$m = 5$，テストモデル 2 について $l = 2$，$m = 10$ としたときの性能は，表中の最良な設定時とほぼ同等であり，辞書式オーダリングによる逐次計算時と比べて，それぞれ

表 7　ハイパースレッディングの効果（テストモデル 2）
Table 7　Effect of hyper-threading technology (Test model 2).

ハイパースレッディング ON（8 スレッド）	16
	(60.3)
ハイパースレッディング OFF（4 スレッド）	16
	(68.1)

PC2 を使用．各欄の上段：反復回数，下段：求解時間 (s).
色数 $l^3 = 2^3$，ブロックサイズ $m = 10$.

1.9 倍，3.1 倍程度の速度向上が得られている．

4.5　ハイパースレッディングの効果

テストモデル 2 で $l = 2$，$m = 10$ としたときについて，ハイパースレッディングの ON/OFF の影響を**表 7** に示す．逐次計算時と比較した速度向上比はそれぞれ 3.1 倍（ON 時），2.8 倍（OFF 時）であり，本計算実験においてはハイパースレッディングは計算速度向上に有効であることが示されている．ただし後述するように，今回適用した求解アルゴリズムには計算時間がメモリ転送性能に強く依存する部分が含まれるため，ハイパースレッディングにより大幅な性能改善を得ることは難しいと考えられる．

4.6　並列化による速度向上率

本節では，使用するスレッド数に対する速度向上率の変化について検討する．テストモデル 1 について PC1 および PC2 を使用して解析を行った結果を**図 8** に示す．ただし x，y，z 方向のメッシュ分割数を 48，64，96 とした 3 通りの解析を行っており，それぞれ 345,744 自由度，811,200 自由度，2,709,792 自由度の問題である．ハイパースレッディングを使用しているのは PC2 でスレッド数を 8 としたときのみである．スレッド数が 1 のときは辞書式オーダリング，それ以外ではブロックマルチカラーオーダリングを適用している．4.4 節で得た指針に従って，ブロックマルチカラーオーダリングの色数は 2^3 に固定し，問題規模および使用するスレッド数に応じて個別に，グリッド G_1 について式 (12) を満たす最大のブロックサイズを選択している．したがって図 8 において，2 から 8 の範囲のスレッド数の変更に対しても，反復回数の変動がともなっていることに注意されたい．

図 8 より，使用するスレッド数の増加にともなって相応の速度向上が得られていることが分かる．スレッド数 8 の結果については，PC2 においてハイパースレッディングを使用したものであり，物理的なコア数は 4 である．

速度向上率の理想的な値は使用するスレッド数に等しいが，実際に得られた速度向上率は理想的な値には及んでいない．主な原因としては，求解アルゴリズムに含まれるベクトルの

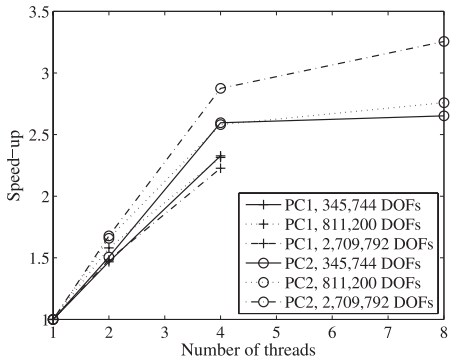

図 8　速度向上率（テストモデル 1）
Fig. 8　Speed-up ratio (Test model 1).

内積，スパース行列とベクトルの積等の並列化については，原理的には扱いが容易である一方で，特にマルチコアプロセッサシステムにおいてはメモリ転送性能がボトルネックとなる傾向があることが考えられる．

PC2 では PC1 と比べて，小容量高速な L2 キャッシュをコアごとに配置する 3 レベルのキャッシュ構造，メモリコントローラを CPU に内蔵する構成等が導入されている．PC2 において比較的良好な速度向上率が得られているのは，これらのアーキテクチャの改良により，PC1 と比べて実効的なメモリ転送性能の向上が得られているためと考えられる．

5. むすび

本論文では，マルチコアプロセッサシステム上で効率的に並列化された高周波電磁界解析を行うことを目的とし，AFW スムーザの並列化を可能にするためのブロックマルチカラーオーダリングを提案した．この際，並列化を可能にするための色分けの条件を詳しく検討するとともに，色数・ブロックサイズの適切な選択についても考察した．

テストモデルについてマルチコアプロセッサを搭載した PC 上で有限要素解析を行い，ブロックマルチカラー AFW スムーザを用いたマルチグリッド解法の並列性能について検証を行った．マルチグリッド解法の反復収束性については，色数あるいはブロックサイズをある程度大きく設定することで，良好な結果が得られた．キャッシュ効率・同期コストの影響を合わせて考慮すれば，スレッド数に比較して十分な並列度が得られる範囲で，色数を少なく，ブロックサイズを大きくした設定が推奨できると考えられる．この方針に従って選んだ設定では，スレッド数の増加に応じて良好な速度向上率が得られた．

本研究の計算実験では，色分けおよびブロック化の設定が容易な構造メッシュのみを扱ったが，たとえば文献 14) の手法を用いることで，一般の非構造メッシュにブロックマルチカラーオーダリングを適用することも可能である．

今後の課題としては，解析の大規模化を実現するための分散メモリ環境における並列処理があげられ，著者らの研究グループで取り組んでいる．本論文で述べたスレッド並列処理向けの手法を併用したプロセス・スレッド並列処理による解析について検討を行っており，その結果については別途報告する．

参考文献

1) Jin, J.: *The Finite Element Method in Electromagnetics*, *2nd edition*, John Wiley and Sons (2002).
2) Barrett, R., et al.（著），長谷川里美，長谷川秀彦，藤野清次（訳）：反復法 Templates，朝倉書店 (1996).
3) Saad, Y.: *Iterative Methods for Sparse Linear Systems*, *2nd edition*, SIAM (2003).
4) Hackbusch, W.: *Multigrid Methods and Applications*, Springer-Verlag (1985).
5) Briggs, W.L.: *A Multigrid Tutorial*, *2nd edition*, SIAM (2000).
6) Zhu, Y. and Cangellaris, A.C.: *Multigrid Finite Element Methods for Electromagnetic Field Modeling*, IEEE Press (2006).
7) Arnold, D., Falk, R. and Winther, R.: Multigrid in H(div) and H(curl), *Numer. Math.*, Vol.85, pp.197–218 (2000).
8) Hiptmair, R.: Multigrid Method for Maxwell's Equations, *SIAM J. Numer. Anal.*, Vol.36, pp.204–225 (1998).
9) Weiss, B., Biro, O., Caldera, P., Hollaus, K., Paoli, G. and Preis, K.: A Multigrid Solver for Time Harmonic Three-Dimensional Electromagnetic Wave Problems, *IEEE Trans. Magn.*, Vol.42, No.4, pp.639–642 (2006).
10) 用水邦明，岩下武史，森　倫也，小林英一：小規模 PC クラスタによる高周波電磁界解析のための並列マルチグリッドソルバ，電学論 B，Vol.127, No.8, pp.911–917 (2007).
11) Doi, S. and Washio, T.: Ordering Strategies and Related Techniques to Overcome the Trade-off between Parallelism and Convergence in Incomplete Factorizations, *Para. Comput.*, Vol.25, pp.1995–2014 (1999).
12) Iwashita, T. and Shimasaki, M.: Block Red-Black Ordering Method for Parallel Processing of ICCG Solver, Lecture Notes in Computer Science, Vol.2327, pp.175–189, Springer (2002).
13) Iwashita, T. and Shimasaki, M.: Algebraic Block Red-Black Ordering Method for Parallelized ICCG Solver with Fast Convergence and Low Communication Costs,

IEEE Trans. Magn., Vol.39, No.3, pp.1713–1716 (2003).
14) 岩下武史，高橋康人，中島　浩：代数ブロック化多色順序付け法による並列化 ICCG ソルバの性能評価，情報処理学会研究報告，Vol.2009-HPC-121, No.11 (2009).
15) Watanabe, K., Igarashi, H. and Honma, T.: Convergence of Multigrid Method for Edge-Based Finite-Element Method, *IEEE Trans. Magn.*, Vol.39, No.3 (2003).
16) Sogabe, T. and Zhang, S: A COCR Method for Solving Complex Symmetric Linear Systems, *J. Comput. Appl. Math.*, Vol.199, No.2, pp.297–303 (2007).
17) Mifune, T., Takahashi, Y. and Iwashita T.: Folded Preconditioner: A New Class of Preconditioners for Krylov Subspace Methods to Solve Redundancy-Reduced Linear Systems of Equations, *IEEE Trans. Magn.*, Vol.45, No.5, pp.2068–2075 (2009).

(平成 22 年 1 月 26 日受付)
(平成 22 年 5 月 4 日採録)

美舩　健

2003 年京都大学大学院工学研究科電気工学専攻博士後期課程修了．博士（工学）．同年より京都大学大学院工学研究科助手（2007 年職名変更により同助教），現在に至る．電磁界解析，線形反復法に関する研究に従事．IEEE，電気学会会員．

廣谷　迪

2008 年京都大学工学部電気電子工学科卒業．同年より京都大学大学院工学研究科電気工学専攻修士課程在籍，現在に至る．

岩下　武史（正会員）

1998 年京都大学大学院工学研究科電気工学専攻博士課程修了．博士（工学）．1998 年京都大学大学院工学研究科リサーチ・アソシエイト（日本学術振興会未来開拓学術研究推進事業 PD），2000 年同大学大型計算機センター助手を経て，2003 年より同大学学術情報メディアセンター助教授（2007 年職名変更により同准教授），現在に至る．高性能計算，線形反復法，電磁界解析，並列処理に関する研究に従事．IEEE，SIAM，電気学会，日本応用数理学会，日本 AEM 学会，日本計算工学会各会員．

村山　敏夫

1987 年東京大学工学部計数工学科卒業．ソニー株式会社入社．1991 年 University of California Santa Barbara，Department of Computer Science 修了．東京大学工学系研究科システム創成学専攻博士後期課程在籍．電気学会，日本計算工学会各会員．大規模高周波電磁界解析技術の研究に従事．

大谷　秀樹

1986 年電気通信大学電気通信学部機械工学科（固体力学講座）卒業．1988 年同大学大学院電気通信学研究科機械工学専攻（流体力学講座）修了，ソニー株式会社入社，現在に至る．半導体プロセスデバイス，熱応力，電磁界シミュレーション技術，デジタル放送技術の研究開発に従事．