

高速なモーメント法に関する研究

今野 佳祐, 陳 強 (東北大学大学院工学研究科)

概要: モーメント法は, 有力な電磁界数値解析法の 1 つとして知られている. 近年, モーメント法を大規模問題に適用するための高速化に関する研究が盛んに進められている. 本報告では, モーメント法の高速化に関するこれまでの我々の研究成果を紹介する.

キーワード: モーメント法, CG-FMM, CBFM, GPU

1. まえがき

近年, 計算機の性能が飛躍的に向上したことに伴い, アンテナや散乱体等の特性を数値的に得る電磁界数値解析法が普及している. 電磁界数値解析法は, マクスウェル方程式及びそれらから導出された電磁界の微積分方程式を数値的に解くための手法を指す. 代表的な電磁界数値解析法として, 有限差分時間領域 (Finite Difference Time Domain, FDTD) 法, 有限要素法 (Finite Element Method, FEM), モーメント法 (Method of Moments, MoM) 等が挙げられる.

FDTD 法は, 微分形式のマクスウェル方程式を差分して時間領域で解くという手法である [1]. FDTD 法では, アンテナや誘電体などを含む有限領域が立方体のセルに分割され, マクスウェルの方程式に従ってセル上の電磁界が時間ステップごとに更新される. 最終的に得られた電磁界の値をフーリエ変換すれば, アンテナの入力インピーダンスや遠方界等の値が得られる. FDTD 法は時間領域の数値解析法であるので, パルス応答を求めるのが容易であり, 解析モデルの広帯域な特性が一度の数値解析で得られるという長所がある. また, セルごとに媒質定数を設定すれば複雑な媒質も容易に取り扱うことができるのも FDTD 法の長所である. その一方で, 特殊な場合を除いてセルの形状が立方体であるため, 曲面形状の解析モデルは階段近似せざるを得ず, 誤差が大きくなるという短所がある. また, セルの電磁界の値を全て保存しておく必要があるため, 大規模な問題の取り扱いには大型の計算機メモリを有するスーパーコンピュータが必須である.

FEM は, 電磁界に関する微分方程式の境界値問題を直接解く代わりに, それと等価な汎関数の極値問題を数値的に解くという手法である [2]. 汎関数の極値問題を数値的に解く際, 領域を小さな要素に分割し, 要素内の未知関数を重み付き多項式で表現する. そして, リッツ法あるいはガラーキン法を用いて行列方程式を生成し, それを解くことで未知の多項式の重みを得る. FEM は変分原理に基づいており, 得られる解の精度が高いという長所がある. また, 共振器や導波管など, 閉領域の電磁界の数値計算が非常に得意である. その一方で, 有限領域の数値解析法であるため, 放射問題や大規模問題はあまり得意としないという短所がある.

MoM は, 電界積分方程式を離散化して解くための手法

であり, 重み付き残差法とも呼ばれる [3]. MoM では, アンテナや散乱体表面の電流分布を適当な基底関数で表現し, 重み関数との内積を取ることで積分方程式を行列方程式に変換する. そして, 逆行列を求めることで行列方程式を解き, 未知の電流値を求める. 基底関数と重み関数を同じ関数とするガラーキン法を用いた MoM は, 変分法と等価なため, 非常に高精度の結果が得られる. また, 有限領域内の電磁界を未知数とする FDTD 法や FEM のような数値解析法と異なり, MoM での未知数はアンテナや散乱体上の電流であり, 空間に未知数を配置する必要がない. 従って, 同じ規模の問題であれば, MoM の未知数の数は FDTD 法や FEM と比較して少ない. その一方で, モーメント法の定式化は非常に複雑であり, 特に面上導体や誘電体を含むモデルにおいては多重積分の数値計算の処理が問題となる. また, 逆行列を求めるための計算時間が未知数の 3 乗に比例するため, 大規模問題の数値解析には高速化が不可欠である. 以上 3 つの電磁界数値解析法の特徴を表 1 にまとめる.

本報告では, MoM の高速化に関する筆者のこれまでの研究成果をまとめたので報告する.

2. MoM

MoM では, 電界積分方程式が以下に示すような行列方程式に変換される.

$$\mathbf{Z}\mathbf{I} = \mathbf{V} \quad (1)$$

ここで, \mathbf{Z} は $N \times N$ の既知のインピーダンス行列, \mathbf{V} は既知の N 元電圧ベクトル, \mathbf{I} は未知の N 元電流ベクトル, N はセグメント数である. インピーダンス行列 \mathbf{Z} の逆行列を求め, (1) 式の両辺に乗じると, 未知の電流ベクトル \mathbf{I} は以下のように求められる.

$$\mathbf{I} = \mathbf{Z}^{-1}\mathbf{V} \quad (2)$$

逆行列を求めるための計算時間は $O(N^3)$ なので, N が 10 倍になると計算時間は 1000 倍になる. その一方で, 逆行列を求める時間と比較すると, インピーダンス行列や電圧ベクトルの計算に要する時間は非常に小さく, それぞれ $O(N^2)$ と $O(N)$ である. 従って, N が大きな大規模問題の数値解析には, 行列方程式を解く部分の高速化が不可欠である. すなわち, MoM の高速化は, 行列方程式の解を高速に得ることと同義である.

3. CBFM による高速化

大きな行列方程式を高速に解くための手法の 1 つとして, 元の行列方程式を小さく圧縮してから解くという CBFM (Characteristic Basis Function Method) が挙げられる [4]. CBFM では, まず (2) 式で表される行列方程式

表 1: 電磁界数値解析法の比較.

手法	有限差分時間領域法 (FDTD 法)	有限要素法 (FEM)	モーメント法 (MoM)
計算領域	時間領域	周波数領域	周波数領域
方程式の種類	マクスウェル方程式 (微分形)	電磁界の微分方程式	電界 (磁界) 積分方程式
未知数の種類	有限領域内の電磁界	有限領域内の電磁界	導体上・誘電体内の電磁流
仮想境界	必要	必要	不要
原理	差分法	重み付き残差法	重み付き残差法
得意な問題	面状アンテナ, 誘電体	共振器, 周期構造	線状アンテナ
不得意な問題	線状アンテナ, 曲面構造 長距離の伝搬問題	放射問題, 非線形素子 大規模問題	線・面・立体が混在する問題 誘電体, 非線形素子
商用シミュレータ	SEMCAD-X(speag 社) EMPro(Agilent Technology 社)	HFSS(ANSYS 社)	FEKO(EMSS 社) WIPL-D(WIPL-D 社)

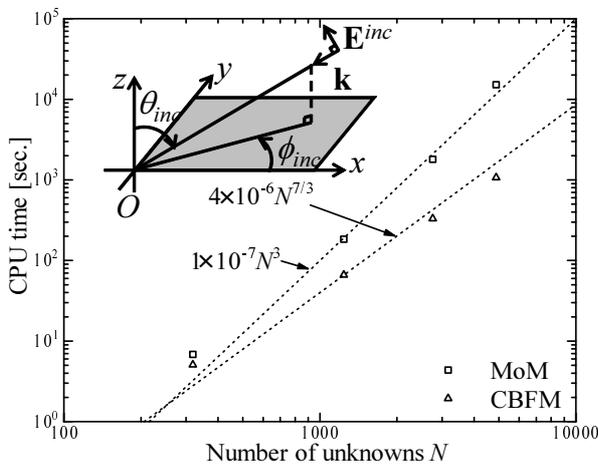


図 1: CBFM による板状導体の散乱問題の数値解析時間.

を複数のブロック行列方程式に分割する. 次に, 自己・相互インピーダンス行列に対応するブロック行列方程式をそれぞれ解き, ブロック内の電流ベクトルを求める. このようにして得られた電流ベクトルはブロック内の電流分布を近似的に表す基底関数とみなせるので, これらを用いてインピーダンス行列を新たに計算し, 元の行列方程式を圧縮する. 最後に, 圧縮した行列方程式をガウス消去法で解いて得られた重みを, 最初に求めた電流ベクトルに乗じて足し合わせることで元の (2) 式の解を得る.

CBFM の計算時間を決める最も重要なパラメータはブロック数である. 筆者らのグループは, CBFM のブロック数 M が以下の関係式を満たすとき, 計算時間が最小の $O(N^{7/3})$ になることを数値的に明らかにした [5].

$$M = 0.9N^{1/3} \quad (3)$$

CBFM によって板状導体の散乱問題を数値解析するのに要した計算時間を図 1 に示す. ガウス消去法によって求めた逆行列によって解を得る計算時間が $O(N^3)$ となって

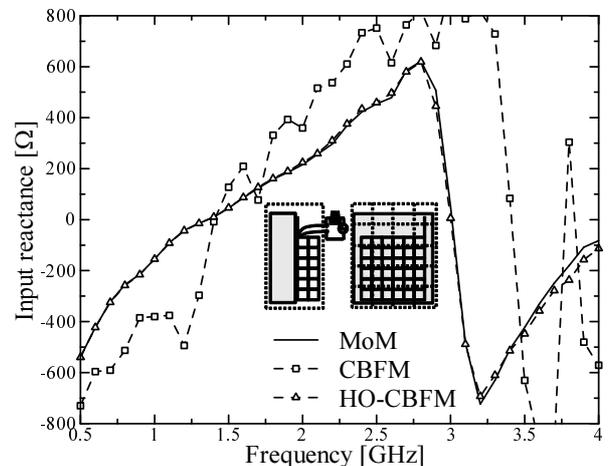
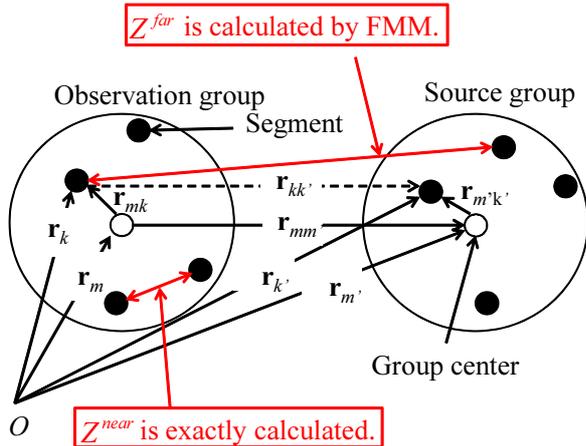


図 2: 誘電体近傍筐体上モノポールアンテナの入力リアクタンス.

いるのに対し, CBFM の計算時間が $O(N^{7/3})$ となっていることが分かる.

一方, CBFM では, ブロック間の相互結合によって生じる各ブロック内の電流分布を表す基底関数が少なく, 実際の電流分布を表すのに不十分なことがある. その結果, CBFM で得られる解の精度が著しく落ちることがあり, 特に放射問題の数値解析で解の精度が大きく悪化するという問題があった [6]. そこで, ブロック間の相互結合を厳密に考慮した CBFM として, 筆者らは高次の CBFM(Higher-Order CBFM, HO-CBFM) を提案した [7]. HO-CBFM では, ブロック間の相互結合を複数回計算することで, ブロック間の相互結合によって生じる電流を表す基底関数を多数求める. 得られた基底関数の数が多いので, 実際の電流分布を高い精度で求めることができ, 結果的に解の精度が上がる. 従来 of CBFM 及び HO-CBFM を用いて, 誘電体近傍アンテナの数値解析を行った結果を図 2 に示す. 従来 of CBFM で得られた入力リアクタンスが MoM によ



m : Group center of m th observation group
 m' : Group center of m' th source group
 k : k th segment in m th observation group
 k' : k' th segment in m' th source group

図 3: FMM の概念.

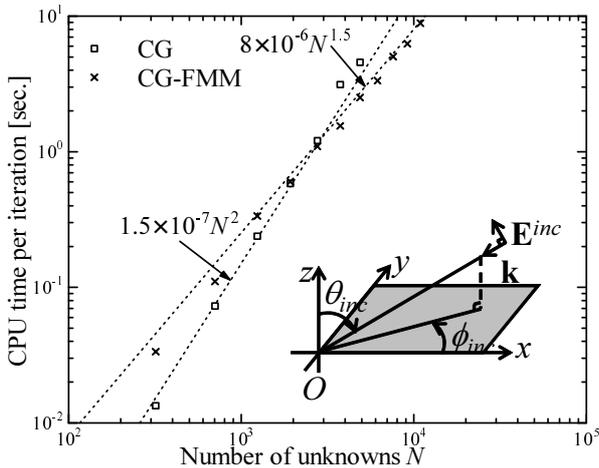


図 4: CG-FMM の反復 1 回当たりの計算時間 (板状導体).

る厳密解から大きくずれている一方で, HO-CBFM で得られた解は厳密解とよく一致していることが分かる.

4. CG-FMM による高速化

CBFM が行列方程式を圧縮した後でガウス消去法により解を求めるのに対し, 行列方程式の解を反復的に求めるという高速 MoM がある. CG-FMM (Conjugate Gradient-Fast Multipole Method) は, 反復法である共役勾配 (Conjugate Gradient, CG) 法と, 高速多重極法 (Fast Multipole Method, FMM) を組み合わせた高速 MoM である [8]. CG 法は, 以下の式で示される残差ベクトル \mathbf{R}_i の大きさが一定以下の値になるまで解を更新する [9], [10].

$$\mathbf{R}_i = \mathbf{V} - \mathbf{Z}\mathbf{I}_i \quad \text{where} \quad i = 1, 2, \dots, N_{it}. \quad (4)$$

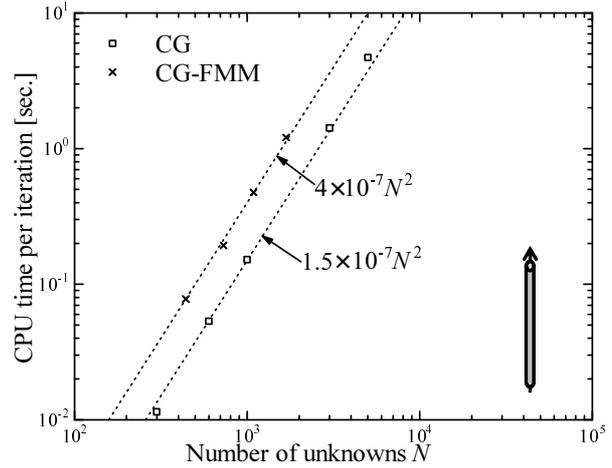


図 5: CG-FMM の反復 1 回当たりの計算時間 (線状ダイポールアンテナ).

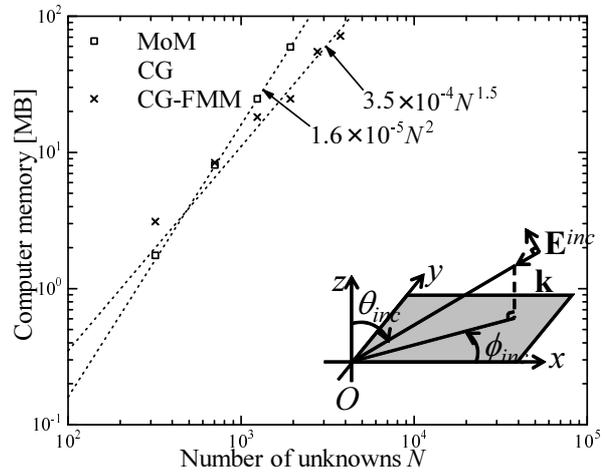


図 6: CG-FMM の計算機メモリ (板状導体).

ここで, \mathbf{I}_i は第 i 回の反復後の近似解, N_{it} は反復回数である. CG 法では, (4) 式における近似解 \mathbf{I}_i を更新する際に, $N \times N$ のインピーダンス行列 \mathbf{Z} と N 元ベクトルとの行列-ベクトル積を繰り返し行う. CG 法の反復処理の中で最も大きな計算時間を要するのはこの行列-ベクトル積であり, その計算時間は $O(N^2)$ となる. 従って, CG 法の計算時間は $N_{it} \times O(N^2)$ となり, $N_{it} \leq N$ の場合はガウス消去法の $O(N^3)$ よりも高速になる.

CG 法を更に高速化するための手法が FMM である. FMM の概念を図 3 に示す. FMM は, 加法定理に基づいた相互インピーダンスの表示式を利用することで, 互いに遠方にあるセグメント間の相互インピーダンスをグループ単位でまとめて計算するという手法である [11]. 相互インピーダンスをまとめて計算する際, グループ内に含まれるセグメントの情報をそれぞれのグループ中心に集め, グループ中心を經由して相互インピーダンスを計算する. 近傍にあるセグメント間の自己・相互インピーダンスは,

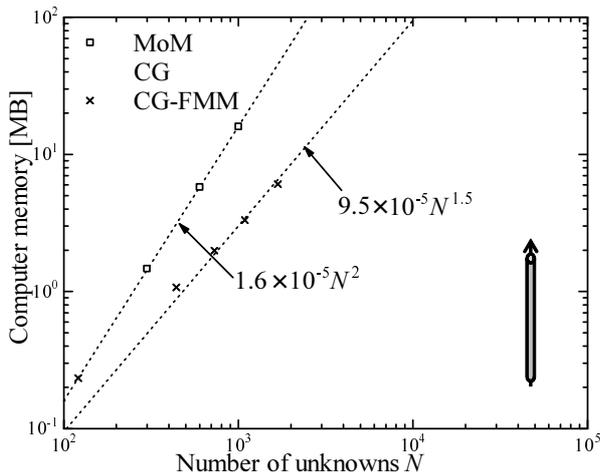


図 7: CG-FMM の計算機メモリ (線状ダイポールアンテナ).

従来の MoM と同様に 1 つ 1 つ計算する. 例えば, 波源グループと観測グループにそれぞれ 30 個のセグメントがあったとすると, 従来の MoM では相互インピーダンスの計算回数は $30 \times 30 = 900$ 回となる. 一方, FMM を用いた場合の相互インピーダンスの計算回数は, 観測セグメント-観測グループ中心間で 30 回, 波源セグメント-波源グループ中心間で 30 回, 観測グループ中心-波源グループ中心間で 1 回の 61 回で済む. FMM を CG 法に適用する (CG-FMM) と, CG 法の行列-ベクトル積をまとめて計算することが可能になり, 計算時間を $O(N^2)$ より抑えることが可能になる.

従来の研究では, CG-FMM の反復 1 回あたりの計算時間はモデルの形状によらず $O(N^{1.5})$ であると言われてきた [8]. 一方, 筆者らのグループは, CG-FMM の反復 1 回あたりの計算時間は必ずしも $O(N^{1.5})$ とはならず, それがモデルの形状に依存することをはじめて明らかにした [12]. 図 4 及び 5 に, CG-FMM の反復 1 回あたりの計算時間を示す. 行列-ベクトル積の計算時間が $O(N^2)$ であるため, CG 法の反復 1 回あたりの計算時間はどちらも $O(N^2)$ となっている. その一方で, CG-FMM の反復 1 回あたりの計算時間は解析モデルによって異なり, 板状導体では $O(N^{1.5})$, 線状ダイポールアンテナでは $O(N^2)$ となっていることが分かる. CG-FMM では, 行列-ベクトル積の一部を FMM によりまとめて計算しているが, その計算時間はグループの大きさ (図 3 の円の直径に相当) とグループ内に含まれるセグメント数の関係で決まる. 線状ダイポールアンテナは 1 次元の形状であり, グループが含むセグメント数がグループの大きさに対して少ないため, 反復 1 回あたりの計算時間は CG 法と同じ $O(N^2)$ のままである. その一方, 板状導体は 2 次元の形状であり, グループが含むセグメント数がグループの大きさに対して十分多いため, 反復 1 回あたりの計算時間は $O(N^{1.5})$ に抑えられている.

図 6 及び 7 に, CG-FMM の計算機メモリを示す. MoM

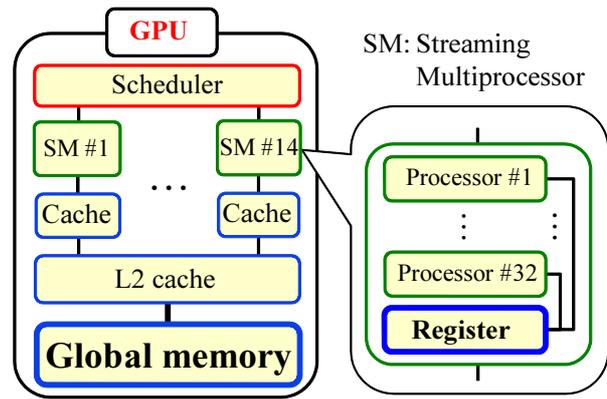


図 8: GPU Tesla C2075 の構造.

や CG 法ではインピーダンス行列全体を保持しておく必要があるため, 数値解析に必要な計算機メモリが $O(N^2)$ となっている. 一方の CG-FMM では, 互いに遠方にあるセグメント間の相互インピーダンスを 1 つ 1 つ保存しておく必要がないため, 必要な計算機メモリが結果的に小さくなる. 従って, CG-FMM による数値解析に必要な計算機メモリは $O(N^2)$ より小さくなっている. 数値解析に必要な計算機メモリを小さくできるということは CG-FMM の大きな特徴の 1 つであり, このことが MoM を用いた商用シミュレータに CG-FMM が搭載されている大きな理由であると考えられる.

5. GPU による高速化

近年, 画像処理用の GPU (Graphics Processing Unit) を数値計算に利用するという GPU コンピューティングが盛んに行われている [13], [14]. 図 8 に, NVIDIA Tesla C2075 GPU の構造を示す. GPU には, SM (Streaming Multiprocessor) と呼ばれる演算部が存在する. SM は, CUDA (Compute Unified Device Architecture) コアと呼ばれる演算装置と, シェアドメモリやキャッシュと呼ばれる記憶装置から構成される. 例えば, Tesla C2075 は 14 個の SM から成り, 1 つの SM 当たり 32 個の CUDA コアが含まれているため, 合計の CUDA コア数は 448 個である. また, 1 つの SM 当たり 64KB のシェアドメモリ (L1 キャッシュと兼用), GPU 全体で 768KB の L2 キャッシュが Tesla C2075 には搭載されている. シェアドメモリとキャッシュは, 記憶容量は小さいものの, アクセス速度が速いという長所がある. その一方, GPU には他にもグローバルメモリと呼ばれる記憶装置があり, 大規模なデータを格納できる (Tesla C2075 の場合は, 容量 6GB) が, シェアドメモリやキャッシュと比較するとアクセス速度が遅い.

GPU を用いて数値計算を高速化するためには, 以上に述べたような GPU の構造を踏まえてプログラムをチューニングする必要がある. これまでに, GPU を用いた数値計算の高速化に効果的とされているチューニング法は以下の 3 つである [15], [16].

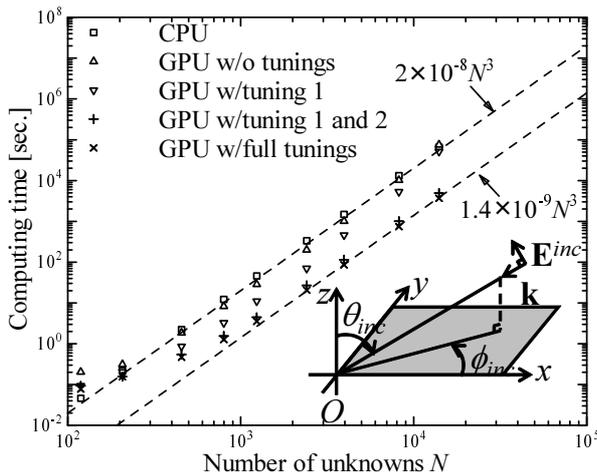


図 9: GPU によるガウス消去法の高速度化。

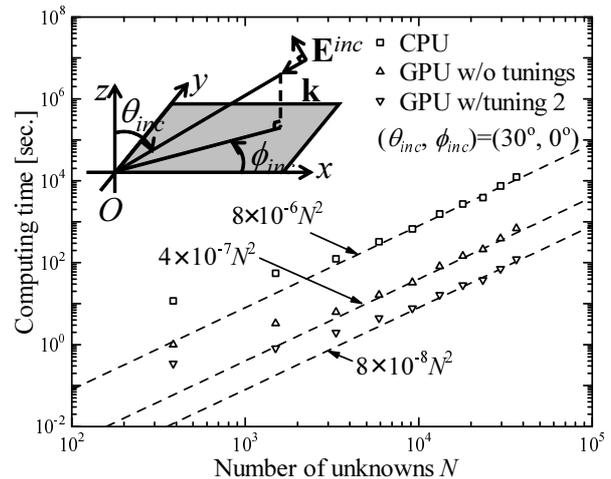


図 10: GPU による CG-FMM の高速度化。

1. CPU と GPU との間のデータ転送量を最小限にすること。
2. GPU のグローバルメモリへのアクセスはコアレスアクセス (Coalesced Access) とすること。
3. 1 ブロックあたりのスレッド数 (Threads/Block) を 32 の倍数にすること。

ここで、スレッドは CUDA コアが実行する演算の最小単位で、ブロックはスレッドの集まりである。GPU は、複数の CUDA コアを用いて演算を並列的に実行することで高速化を図る。従って、全ての CUDA コアが常に演算している状態を維持し、かつ CUDA コアが演算に必要な情報を取得するための時間を最小化することが重要である。Tesla C2075 には 1 つの SM 当たり 32 個の CUDA コアが含まれているため、全ての CUDA コアが常に演算している状態を維持するためには、1 ブロックあたりのスレッド数 (Threads/Block) を 32 の倍数にするのが最も効率が良い。また、データ転送に要する時間を最小化することも重要である。特に、グローバルメモリに 1 度アクセスするだけで複数のデータを取得できるコアレスアクセスは、データ転送に要する時間を減らすために有効とされている。

筆者らは、GPU を用いた数値計算の高速化に効果的とされているチューニング法を MoM に適用し、その効果を定量的に明らかにした [17]。図 9 に、GPU 上でガウス消去法を実行したときの計算時間を示す。その他のチューニング法と比較して、チューニング 2 (コアレスアクセス) が最も高速化の効果が高いことが分かる。また、CPU (Intel Xeon 2.27 GHz) 上でガウス消去法を実行した際の計算時間と比較すると、GPU 上でガウス消去法を実行する方が十数倍程度高速なことが分かる。図 10 に、GPU 上で CG-FMM を実行したときの計算時間を示す。コアレスアクセスを適用すると、計算時間が 1/5 になることが分かる。また、CPU 上で CG-FMM を実行した際の計算時間と比較すると、GPU 上で CG-FMM を実行する方が 100

倍程度高速なことが分かる。CG-FMM のような反復法では、並列化の容易な行列-ベクトル積の計算時間が大半であるため、ガウス消去法のような直接法と比較して GPU による高速化の効果が高い傾向にある。その一方で、計算時間のオーダー自体は MoM の種類で決まるため、CPU と GPU のどちらでも同じになっていることも分かる。

6. むすび

本報告では、MoM の高速化に関する筆者のこれまでの研究成果をまとめた。本報告で紹介した手法の特徴、長所・短所を表 2 にまとめる。MoM を高速化するための様々な手法があるが、その長所と短所をよく理解した上でそれらを数値解析に利用することが大切である。

参考文献

- [1] K.S. Yee, "Numerical solution of initial boundary value problems involving maxwell's equations in isotropic media," IEEE Trans. Antennas Propag., vol.14, no.4, pp.302-307, May 1966.
- [2] 小柴正則, 光・波動のための有限要素法の基礎, 森北出版, 1990.
- [3] R.F. Harrington, Field Computation by Moment Methods, New York, Macmillan, 1968.
- [4] V.V.S. Prakash and R. Mittra, "Characteristic basis function method: A new technique for efficient solution of method of moments matrix equations," Microw. Opt. Technol. Lett., vol.36, no.2, pp.95-100, Janu. 2003.
- [5] K. Konno, Q. Chen, K. Sawaya, and T. Sezai, "Optimization of block size for CBFM in MoM," IEEE Trans. Antennas Propag., vol.60, no.10, pp.4719-4724, Oct. 2012.

表 2: 高速 MoM の比較.

手法	CBFM(HO-CBFM)	CG-FMM	GPU
高速化技術	ソフトウェア的	ソフトウェア的	ハードウェア的
行列方程式の解法	直接法	反復法	直接法あるいは反復法
最小計算時間の オーダー	$O(N^{7/3})$	$N_{it} \times O(N^{1.5})$	実行した手法と同じ
長所	既存の MoM への適用が容易 計算時間と精度の調整が容易	計算機メモリが削減可能	既存の MoM への適用が容易 精度を損なわず高速化が可能
短所	高速化の効果が比較的小さい 計算機メモリの削減効果はない	定式化が複雑 計算コストが問題に依存	GPU の購入費が必要 計算機メモリの削減効果はない

- [6] K. Konno, Q. Chen, K. Sawaya, and T. Sezai, "Analysis of linear antenna near dielectric object by CBFM," Proc. IEEE AP-S Int. Symp., 263, pp.1-2, July 2012.
- [7] 今野 佳祐, 陳 強, "高次の CBFM を用いた誘電体近傍アンテナの数値解析," 信学技報, vol.113, no.261, EST2013-77, pp.147-152, 2013 年 10 月.
- [8] R. Coifuman, V. Rokhlin, and S. Wandzura, "The fast multipole method for the wave equation: A pedestrian prescription," IEEE Antennas Propag. Mag., vol.35, no.3, pp.7-12, June 1993.
- [9] T.K. Sarker and S.M. Rao, "The application of the conjugate gradient method for the solution of electromagnetic scattering from arbitrarily oriented wire antennas," IEEE Trans. Antennas Propag., vol.32, no.4, pp.398-403, April 1984.
- [10] T.K. Sarker, "The conjugate gradient method as applied to electromagnetic field problems," IEEE Antennas Propagation Society Newsletter, vol.28, no.4, pp.4-14, Aug. 1986.
- [11] M. Abramowitz and I.A. Stegun, Handbook of Mathematical Functions, National Bureau of Standards, p.440, 1972.
- [12] K. Konno, Q. Chen, and K. Sawaya, "Quantitative evaluation for computational cost of CG-FMM on typical wiregrid models," IEICE Trans. Commun., vol.E93-B, no.10, pp.2611-2618, Oct. 2010.
- [13] S. Peng and Z. Nie, "Acceleration of the method of moments calculations by using graphics processing units," IEEE Trans. Antennas Propag., vol.56, no.7, pp.2130-2133, July 2008.
- [14] E. Lezar and D.B. Davidson, "GPU-based LU decomposition for large method of moments problems," Electron. Lett., vol.46, no.17, pp.1194-1196, Aug. 2010.
- [15] NVIDIA Corporation, "CUDA C Programming Guide," ver.5.0, NVIDIA Corp., Oct. 2012.
- [16] The Portland Group, "CUDA Fortran Programming Guide and Reference," ver.13.1, The Portland Group, Jan. 2013.
- [17] K. Konno, H. Katsuda, K. Yokokawa, Q. Chen, K. Sawaya, and Q. Yuan, "Quantitative study of computing time of direct/iterative solver for MoM by GPU computing," IEICE Commun. Express, vol. 2, no. 8, pp. 359-364, 2013.