GPUによるモーメント法の高速化に関する研究

Research on acceleration of method of moments using GPU

横川佳[*1] 齋藤優輝[*1] 袁巧微[*1] 瀬在俊浩[*2] 勝田肇[*3] 今野佳祐[*3] 陳強[*3] 澤谷邦男[*3]

Kei Yokokawa Yuki Saito Qiaowei Yuan Toshihiro Sezai Hajime Katsuda Keisuke Konno Qiang Chen Kunio Sawaya

[*1]仙台高等専門学校 [*2]宇宙航空研究開発機 [*3]東北大学大学院工学研究科

Sendai National College of technology JAXA Graduate School of Engineering, Tohoku University

1. まえがき

モーメント法を用いた大規模アンテナの電磁界数値解析には，連立一次程式を解く際に多大な CPU 時間が必要であるという問題がある．そこで近年，描画処理に用いるGPU(Graphics Processing Unit)をモーメント法に応用し，連立一次程式を高速に解くという試みが注目されている．本報告では掃き出し法と CGNR(Conjugate Gradient method applied to the Normal equations where the Residue is minimized)法[1]を用いて[2][3]，CPU 及び GPU 上で連立一次方程式を解いたときの計算時間を比較し GPU の有効性を定量的に示す．

2．モーメント法

モーメント法[4]は，微分方程式・積分方程式を代数方程式に変換し，それを解く数学的な手法である。モーメント法によるアンテナ解析手法とは，モーメント法を用いてアンテナの電磁特性を表すマクスウェル方程式と境界条件式を解くことにより，アンテナの電磁特性を明らかにすることである．具体的に導体アンテナに対して,下記導体表面で電界の接線成分が0になるという境界条件から，以下の積分法定式が得られる．

$$\hat{\mathbf{n}}\times[-j\omega\mu_0\int_s \overline{G}(\mathbf{r},\mathbf{r}')\cdot\mathbf{J}(\mathbf{r}')dS'+\mathbf{E}^i(\mathbf{r})]=0 \quad (1)$$

(1)式より電流分布を求めれば放射指向性や利得などアンテナの特性を求める事ができる．また，アンテナの給電電圧と給電点からの電流により入力インピーダンスが求めることができる．(1)式の未知電流分布 $\mathbf{J}(\mathbf{r}')$ を，展開数と呼ばれる関数で展開し，更に重み関数と内積をとることで連立一次方程式(2)に変換する事ができる．この過程がモーメント法の過程であり，連立一次方程式を解くことでアンテナの電流分布を得る事ができる．

$$\begin{bmatrix} z_{11} & z_{12} & \cdots & z_{1N} \\ z_{21} & z_{22} & \cdots & z_{2N} \\ \vdots & & \ddots & \vdots \\ z_{N1} & z_{N2} & \cdots & z_{NN} \end{bmatrix}\begin{bmatrix} i_1 \\ i_2 \\ \vdots \\ i_N \end{bmatrix}=\begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{bmatrix} \quad (2)$$

しかし，アンテナが大規模になるに連れ，連立一次方程式の未知数の数 N が増加してしまい電流分布を求めるまでの時間が非常にかかる問題点がある．本報告では GPU を用いて連立一次方程式を解く過程の高速化の可能性を示す．また，式(2)は式(3)のように行列変数で表すことができる．

$$\mathbf{ZI}=\mathbf{V} \quad (3)$$

ここで $\mathbf{Z}$ は $N\times N$のインピーダンス行列で，アンテナの形状，展開関数と重み関数から求められる既知の行列である．$\mathbf{V}$ は既知の N 元電圧ベクトル，$\mathbf{I}$ は未知の N 元電流ベクトルである．

3. 連立一次方程式の解法

モーメント法で得られた連立一次方程式 $\mathbf{ZI}=\mathbf{V}$ を解くための解法には大きく分けて直接法と反復法がある．本報告では，その連立一次方程式を直接法である掃き出し法，反復法である CGNR 法を使用する．

3.1 掃き出し法

直接法である掃き出し法は以下に示すように，まず連立一次方程式の拡大係数行列である$[\mathbf{Z}\mid\mathbf{V}]$を考える．

$$\begin{bmatrix} z_{11} & z_{12} & \cdots & z_{1n} & v_1 \\ z_{21} & z_{22} & \cdots & z_{2n} & v_2 \\ \vdots & & \ddots & \vdots & \vdots \\ z_{n1} & z_{n2} & \cdots & z_{nn} & v_n \end{bmatrix}$$

図 1.拡大係数行列

この行列 $\mathbf{Z}$ 部分を行列の基本変形を用いて単位行列に変形することで右端の列部分が解となる算法である．掃き出し法のアルゴリズムとして，大きく分けてピボット操作と掃き出し操作に分けられる．この 2 つの操作を繰り返すことで単位行列を生成することができる．図 2 に掃き出し法のアルゴリズムを示す．図 2 のループ B の掃き出し部分を GPU で並列計算させることにより高速化を図る．更にグローバルメモリ転送回数を減らすためにループ A 部分も GPU 上でシングルスレッドとして計算させることにより更に高速になった．

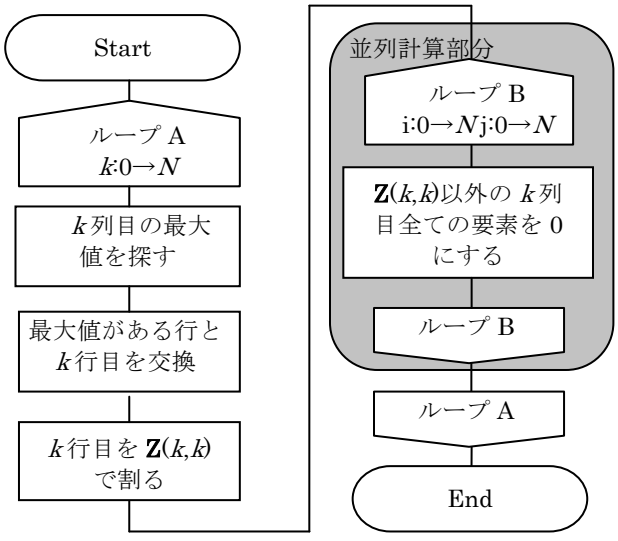

図 2. 掃き出し法のアルゴリズム

ここで $\mathbf{Z}(k,k)$とは行列 $\mathbf{Z}$ の k 行 k 列目の要素を表してい

る．この解法は計算量が $O(N^3)$,メモリが $O(N^2)$に比例するが, 確実に解を求めることができるという特徴がある.

3.2 CGNR 法

モーメント法で得られる行列 $\mathbf{Z}$ は非エルミート行列であるため, そのままでは反復法である共役勾配法(CG 法)を使用し解くことができない．そこで $\mathbf{ZI} = \mathbf{V}$ の左方から, 複素共役の転置をとった $\mathbf{Z}$ との行列積を行った行列を $\mathbf{Z}'$, $\mathbf{V}'$ とすることで, (5)に示す. $\mathbf{Z}'$ がエルミート行列になり, CG 法を適応することができる．この方法を CGNR 法と呼ぶ.

$$\mathbf{Z}^*\mathbf{ZI} = \mathbf{Z}^*\mathbf{V} \quad (4)$$

$$\mathbf{Z}'\mathbf{I} = \mathbf{V}' \quad (5)$$

CGNR 法の反復部のアルゴリズムを図 3 に示す．CGNR 法のアルゴリズムでは，行列とベクトルの積の計算量が最も大きく計算時間を費やしている．そこで $\mathbf{w}_{i-1}$ と $\tilde{\mathbf{r}}_i$ の演算処理を GPU に計算させることで CGNR 法の高速化を試みる.

While $i < N$ *and* $\varepsilon < 10^{-8}$

$\boxed{\mathbf{w}_{i-1} = \mathbf{Z}\mathbf{p}_{i-1}}$:GPU 計算部

$$\alpha_{i-1} = \frac{(\tilde{\mathbf{r}}_{i-1}^* \cdot \tilde{\mathbf{r}}_{i-1})}{\|\mathbf{w}_{i-1}\|_2^2}$$

$$\mathbf{I}_i = \mathbf{I}_{i-1} + \alpha_{i-1}\mathbf{p}_{i-1}$$

$$\mathbf{r}_i = \mathbf{r}_{i-1} - \alpha_{i-1}\mathbf{w}_{i-1}$$

$\boxed{\tilde{\mathbf{r}}_i = \mathbf{Z}^*\mathbf{r}_i}$:GPU 計算部

$$\beta_{i-1} = \frac{(\tilde{\mathbf{r}}_i^* \cdot \tilde{\mathbf{r}}_i)}{(\tilde{\mathbf{r}}_{i-1}^* \cdot \tilde{\mathbf{r}}_{i-1})}$$

$$\mathbf{p}_i = \mathbf{r}_i + \beta_{i-1}\mathbf{p}_{i-1}$$

end

図 3. CGNR 法のアルゴリズム

4. GPU の数値計算

GPU とは描画処理する際に用いるチップであり， GPU を CUDA(Compute Unified Device Architecture)を用いて並列計算する事で高速に計算できる事がわかってきている．CUDA では GPU 上で実行するプログラムをカーネルと呼ぶ. カーネル実行の流れとして CPU 側が GPU 側にプログラムを読み込ませ, メインメモリに格納されているデータを GPU 側にあるグローバルメモリに転送する. GPU は CPU と比較すると, 行列間の積または行列とベクトルの積が高速に行える．表 1 は GPU と CPU による行列ベクトル積を行い, 計算時間を比較した比率を示している. CPU と比較すると最大で約 161 倍速い結果が得られている.

表 1.GPU･CPU による行列ベクトル積の計算時間比

Size of matrix	Ratio of GPU to CPU
1200×1200	153.8
2700×2700	139.8
4800×4800	161.9
7500×7500	153.3

5. 数値解析結果

解析モデルとして半波長ダイポールアレーアンテナを用い, 図 4 に示した. x, y 方向にそれぞれ M 個半波長ダイポール素子を配置し, 各々のダイポールアンテナ素子を 3 セグメントで分割した. 全部のセグメント数は $3M^2$ になる. つまり(3)のような連立一次方程式の未知数の数 N は $3M^2$ に等しい.

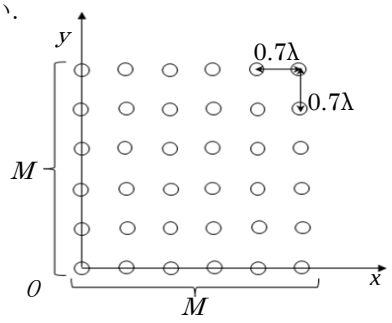

図 4. 半波長ダイポールアレー

このモデルにモーメント法を応用することで, 密な非エルミートなインピーダンス行列 $\mathbf{Z}$ を得ることができる. ベクトル $\mathbf{V}$ はアレーアンテナの励振に関係し, 本計算では 1 セグメントのみ給電されるモデルとなる.

GPU による掃き出し法において, グローバルメモリへの転送回数が多くなったことから, GPU 上でループ A とループ B 共に実行させることで更に高速化した．一方 CGNR 法では図 3 に示すアルゴリズムの $\mathbf{w}_{i-1}$ と $\tilde{\mathbf{r}}_i$ の演算処理を GPU で行った. この GPU と CPU の計算時間比を表 2 に示す. 計算環境として GPU には Tesla C2050, CPU には XeonE5607, コンパイラとして nvcc を用いた.

表 2.GPU･CPU による連立一次方程式を解く計算時間比

	Ratio of GPU to CPU	
Number of unknowns	CGNR 法	掃き出し法
1200(M=20)	110.8	93.0
2700(M=30)	139.6	126.0
4800(M=40)	177.3	149.9
7500(M=50)	171.3	152.0

GPU を用いることで掃き出し法では最大 152 倍, CGNR 法では最大 171.3 倍の高速化になったことが明らかになった．CGNR 法は掃き出し法よりも GPU による高速化の効果が大きいことが確認できた.

6. まとめ

掃き出し法において N^3 に比例する掃き出し部分を GPU で実行することによって, 計算速度が約 100 倍になることが明らかになった. GPU による並列計算が高速化の観点で, 非常に有効であることが確認できた. また, 行列サイズが大きくなるにつれて CPU と GPU の処理時間の差も拡大するため, 大規模な行列ほど GPU が有効であることがわかった. そして, GPU は掃き出し法よりも CGNR 法が更に有効であることもわかった.

参考文献

[1] Walton C. Gibson, “ The Method of Moments in Electromagnetics, pp.53-54,”Chapman & Hall/CRC, Boca Raton, 2008.

[2] 横川 佳, 齋藤 優輝, 袁 巧微, 瀬在 俊彦, 陳 強, 澤谷 邦夫, “GPU による共役勾配法を用いたモーメント法の高速化に関する研究,” *電子情報通信学会 2012 年総合大会講演論文集*,ISSN 1349-1377,2012.3.20-3.23

[3] 横川 佳, 齋藤 優輝, 袁 巧微, 瀬在 俊彦, 勝田 肇, 今野 佳祐, 陳 強, 澤谷 邦男,“GPU によるモーメント法の高速化に関する研究,” *信学技報*, vol.112, no.7, pp27-32, AP 2012-6.

[4] 陳 強,”モーメント法によるアンテナ解析の基礎,” *アンテナ・伝搬における設計・解析手法ワークショップ*,pp.1-10,2009.