B-003

ＧＰＵを用いた２次元ＦＤＴＤ法の計算高速化

A high-speed two-dimensional FDTD method using graphics processing unit

高田直樹† 増田信之†† 田中喬††
阿部幸男†† 伊藤智義†† 下馬場朋禄†††
Naoki Takada Nobuyuki Masuda Takashi Tanaka
Yukio Abe Tomoyoshi Ito Tomoyoshi Shimobaba

１．まえがき

様々な電磁波シミュレーション手法が存在する中で、有限差分時間領域法（FDTD 法：Finite-Difference Time-Domain method）[1]を用いた電磁波問題解析に関する研究は著しく発展を遂げている。FDTD 法は、計算効率に優れ、電磁波問題のモデル化が容易であり、実験値と良く一致することから実用化されている。FDTD 法の計算高速化は電磁波関連機器の開発時間短縮につながるため重要である。また、電磁波は目には見えないため、現象を把握することは困難であり、電磁界現象の可視化も重要である

FDTD 法の計算高速化に関する研究として、PC クラスタを用いた分散並列 FDTD 法[2][3]がある。しかし、PC クラスタはコストが高く、大規模となる欠点がある。

近年、GPU(Graphics processing unit)の計算速度は著しく向上している。最近の GPU は、単精度浮動小数点演算を行うことができ、CPU(Central processing unit)の計算速度を超えている。本研究では、FDTD 法の差分式を GPU に適用することを目的とする。計算モデルとして、基本的な電磁波問題である平面波の金属による散乱問題を扱い、GPU による FDTD 法の計算高速化を検証する。

２．２次元 FDTD 法

FDTD 法[1]は、1966 年に Yee により考案された。マクスウェル方程式を時間及び空間について離散化し、中心差分により導出された差分式を用いて電磁界成分を求める方法である。リープフロッグアルゴリズムを用い、電界 $\vec{E}$ 及び磁界 $\vec{H}$ は、時間に対して交互に計算される。

FDTD 法の差分式は次式となる。

$$H_x^{n+1/2}(i,j+1/2)=H_x^{n-1/2}(i,j+1/2)-\frac{\Delta t}{\mu\Delta y}\left\{E_z^n(i,j+1)-E_z^n(i,j)\right\} \quad (1)$$

$$H_y^{n+1/2}(i+1/2,j)=H_y^{n-1/2}(i+1/2,j)+\frac{\Delta t}{\mu\Delta x}\left\{E_z^n(i+1,j)-E_z^n(i,j)\right\} \quad (2)$$

$$E_z^{n+1}(i,j)=E_z^n(i,j)-\frac{\Delta t}{\varepsilon\Delta y}\left\{H_x^{n+1/2}(i,j+1/2)-H_x^{n+1/2}(i,j-1/2)\right\}+\frac{\Delta t}{\varepsilon\Delta x}\left\{H_y^{n+1/2}(i+1/2,j)-H_y^{n+1/2}(i-1/2,j)\right\} \quad (3)$$

ここで、空間離散間隔を $\Delta x, \Delta y$ とし、時間離散間隔を Δt とする。

式(1),(2)の磁界成分計算終了後に、タイムステップを進め、式(3)の電界計算を行う（図１）。

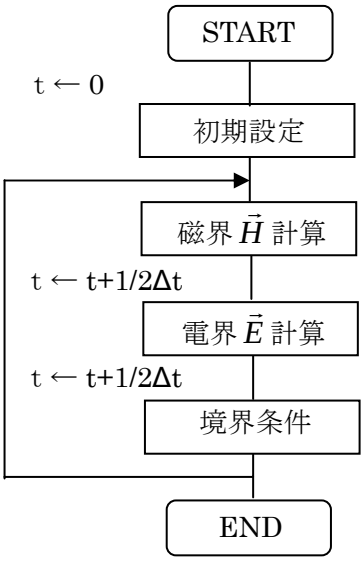


図１　FDTD 法のフローチャート

例えば、散乱体が損失のない金属（完全導体）の場合は、次式を用い、散乱現象を発生させる。

$$E_z(i_a,j_a)=-{}^{inc}E_z(i_a,j_a) \quad (4)$$

ここで、${}^{inc}E$ は入射波、(i_a,j_a) は散乱体上の座標を示す。

３．GPU を用いた散乱界 FDTD 法

最近の GPU[4]は、単精度浮動小数点演算ユニットを複数もち、ベクトル演算が可能である。CPU を上まわる高い演算性能を発揮している。GPU 内部には、高速なコンピュータグラフィックス（CG）処理を行うために、２つのパイプラインが存在する。頂点処理を目的としたパイプライン「Vertex Shader」と、ピクセル処理を目的としたパイプライン「Pixel Shader」である。現在、「Vertex Shader」,「Pixel Shader」はプログラム可能となっている。

本研究では、GPU として nVidia Geforce 7800 GTX (表１)を使用した。グラフィックス API として、DirectX9.0c を使用し、上位レベルシェーダ言語として、HLSL (High Level Shader Language)を用いた。FDTD 法の差

† 湘北短期大学情報メディア学科　†† 千葉大学工学部
††† 山形大学工学部

分計算には「Pixel Shader」のみ使用した。電界成分(E_z)のテクスチャーと、磁界成分(H_x,H_y)のテクスチャーに分け、次ステップの電界または磁界が計算を行う際に、２つのテクスチャーを GPU に渡し、「Pixel Shader」において差分計算を行わせる。境界条件を効率良く取り扱うため、タイムステップの加算の仕方を改良した。従来の方法(図１)では、磁界及び電界計算の終了後に$1/2\Delta t$だけタイムステップを進め、その後に境界条件を計算する。しかし、これでは電磁界成分のテクスチャーを GPU に送る回数が増えてしまう。そこで、本手法では、磁界計算終了後にタイムステップをΔt進め、電界計算と境界条件の計算を１度に行うようにした。

表１　nVidia Geforce 7800 GTX

Core Clock	430 MHz
Memory	256 MB
Memory Clock	1.2G Hz
Vertex Shader	8
Pixel Shader	24

４．計算モデル

本研究では、基本的な散乱現象を計算モデルとして扱い、本手法と従来の FDTD 法の計算時間を比較する。計算モデルとして、完全導体角柱(１辺の大きさ:$ka = 2\pi a/\lambda = 5.0$)に平面波による散乱を用いる。完全導体角柱の境界条件は次式となる。

$$E_z(i_a, j_a) = -\sin\{\omega(n+1)\Delta t\} \qquad (5)$$

ここで、$\omega = 2\pi/T$，$\Delta x = \Delta y = \Delta$，$\Delta t = 0.5\Delta$，$\lambda = 20\Delta$とする。なお、$C = \lambda/T = 1$として規格化した。完全導体角柱は、解析領域($L\Delta x \times L\Delta y$)の中央に配置した(図２)。

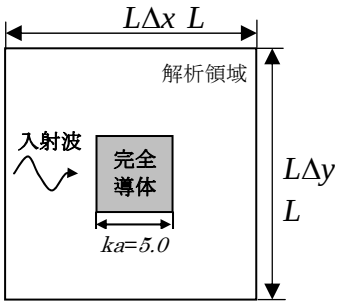


図２　計算モデル

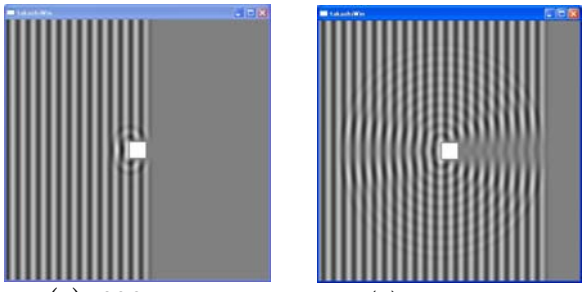
５．結果

本手法による計算高速化を検証するため、CPU を用いて従来の FDTD 法で計算した場合と、本手法により GPU を用いて計算した場合と、両者について計算時間を測定した（表２）。計算システムには、同じ PC（CPU: Intel Pentium4 3.4 GHz, メモリ：2.0GB）を用いた。CPU の計算においては、２つのオペレーティングシステム（OS）：Microsoft Windows Xp、Fedra Core 4 で計算時間を測定した。GPU の計算では、Windows Xp を使用した。グラフィックス API として、DirectX9.0c を、上位レベルシェーダ言語として、HLSL (High Level Shader Language)を用いた。また、コンパイラは、Windows の場合、Visual C++ .Net 2003 を使用した。最適化オプションに “-O2” を使用した。Linux の場合は、gcc 4.0 を使用し、最適化オプションに “-O3” を用いた。表２において、“CPU (Win)”は、OS に Windows Xp を用いて CPU で計算した場合の計算時間を、“CPU (Linux)”は Linux で CPU を用いた場合の計算時間を示している。”GPU (Win)”は、Windows で GPU を用いた場合の計算時間を示している。なお、それぞれ 1000 step までの計算時間を測定した。CPU において FDTD 法の計算には、単精度浮動小数点演算を使用した。

表２　散乱界 FDTD 法の計算時間　(1000 step)

	Calculation time of 1000 steps (ms)		
解析領域	CPU (Win)	CPU (Linux)	GPU (Win)
256x256	2047	2138	431
512x512	13763	14020	1655
1024x1024	74841	70230	6340

表２より、解析領域1024×1024において、GPU での計算は、Linux 上で CPU を用いて計算した場合に比べ、約 11 倍計算高速化された。また、解析領域が大きくなるにつれ、計算高速化がなされることが示された。1000 step 後の計算結果を比較したところ、GPU と CPU の計算結果は、有効数字６桁一致しており、十分な計算精度が得られている。また、GPU による計算結果（電界成分 E_z）を図３に示す。



(a) 600 steps　　(b) 900 steps

図３　完全導体角柱による散乱（電界 E_z）

６．まとめ

２次元 FDTD 法を GPU に適用した。解析領域1024×1024において、従来の CPU による計算時間に比べ、本手法により約１１倍計算高速化なされた。今後は、FDTD 法で一般的に使用されている PML 吸収境界条件の適用と、３次元化を行う予定である。

７．文献

[1] K.S. Yee, *IEEE Trans. Antennas Propag.* **AP-14** (1966) 302.
[2] D. P. Rodohan, S. R. Saunders, R. J. Glover, *Int. J. Numerical Modelling: Electronic Networks, Devises and Fields* 8 (1995) 221.
[3] 高田直樹, 安藤勝規, 本島邦行, 伊藤智義, 上崎省吾, “新たな分散 FDTD 法アルゴリズム”, 電子情報通信学会論文誌 C-I, Vol. J80-C-I, No. 2, 1997.
[4] M. Pharr, “GPU Gems 2”, ボーンデジタル (2005)