

科学研究費助成事業（科学研究費補助金）研究成果報告書

平成24年 6月 5日現在

機関番号：13901

研究種目：若手研究(B)

研究期間：2010～2011

課題番号：22760062

研究課題名（和文）

波動問題に対する高速多重極積分方程式法のGPUを利用した並列計算手法の開発

研究課題名（英文）

Development of GPU-based parallel algorithm for fast multipole integral equation method in wave problems

研究代表者

高橋 徹 (TAKAHASHI TORU)

名古屋大学・工学研究科・講師

研究者番号：90360578

研究成果の概要（和文）：本研究の目的は、3次元 Helmholtz 方程式に対する低周波領域における高速多重極境界積分方程式法をマルチスレッド型メニーコアプロセッサであるグラフィカル・プロセッシング・ユニット（GPU）によって効率的に実行することである。特に計算負荷の高い M2L 計算に注目し、算術演算に対するメモリ演算の比率を低減可能とする高性能アルゴリズムを提案し、NVIDIA 社 Fermi シリーズ GPU に向けた計算プログラムを開発した。数値解析において計算性能の向上を検証し、約 560 万自由度の音響散乱解析を約 2.8 時間で実行することができた。

研究成果の概要（英文）：The present study purposed to perform the low-frequency fast-multipole integral equation method for three dimensional Helmholtz' equation efficiently on a graphical processing unit (GPU), which is regarded as a multi-threaded many-core processor. Focusing on the multipole-to-local (M2L) operation that is the most time-consuming in the whole computation, we proposed high-performance algorithms that can decrease the memory-to-flop ratio in the operation and implemented a computer programme optimised for NVIDIA' s Fermi GPUs. In the numerical analysis, we verified the improvement in computational performance and successfully performed an acoustic scattering problem of 560 million degree of freedom in 2.8 hours.

交付決定額

(金額単位：円)

	直接経費	間接経費	合計
2010 年度	1,700,000	510,000	2,210,000
2011 年度	700,000	210,000	910,000
総計	2,400,000	720,000	3,120,000

研究分野：工学

科研費の分科・細目：応用物理学・工学基礎

キーワード：アルゴリズム、計算物理、シミュレーション工学、数理工学

1. 研究開始当初の背景

境界積分方程式法（境界要素法やモーメント法を含む）は波動現象に関連する境界値問題に対する高精度な数値解析手法として古くから研究され、諸問題（音響学、動弾性学、電磁気学等）へ応用されてきた。しかし、従来の境界積分方程式法は計算コストが高いという欠点を抱えているために、大規模な境界

値問題を解くことは困難である。その原因は N 個の境界要素によって離散化された層ポテンシャル（境界積分）を評価する際に $O(N^2)$ の計算量を必要とする（境界要素間の相互作用を計算するという一種の古典的な多体問題になっているため）ことにある。

この計算コストの問題は、1980 年代後半に提案された高速多重極法（FFT のような分割

統治型の高速アルゴリズム)が契機となって解消に向かいつつある。高速多重極法を組み込んだ境界積分方程式法(以下、高速多重極積分方程式法)の計算量は $O(N)$ 程度で済む(かつ、得られる数値解は所望の近似誤差の範囲で正しい)。その結果、境界積分方程式法によって大規模な波動解析が実行可能となった。

現在、高速多重極積分方程式法に関する研究の重要な課題の一つは、さらなる加速手法を探究することである。その計算量は N でスケールするとは言え、その比例定数はアルゴリズムの複雑さに起因して必ずしも小さいとは言えないからである。これまでに、数理的な方面では、「adaptive」や「new version」と言った加速手法の研究が進展した。一方、計算技法的な方面でも研究が進められ、近年では、並列処理が有効な加速手法と考えられている。その背景には、過去約10年間に計算機のクラスタ化技術が発達したのに加えて、ここ数年間でマルチコアCPUが急速に普及したことがある。マルチコアプロセッサの研究開発は今後も発展することは間違いなく、マルチコアプロセッサを積極的に活用する研究は今後もますます重要である。

2. 研究の目的

そこで本研究は、波動問題に対する高速多重極積分方程式法の並列計算をGPUを利用して実現することを目的とする。昨今のGPUは、基本的にSIMD演算(一つの命令を、異なる複数のデータに対して並列に実行する演算)用の演算器(計算コア)を数十基程度も搭載したマルチコアプロセッサの一種である。GPUの中でも、NVIDIA社の「CUDA」はマルチスレッド方式で自由にプログラムが可能である。そのハイエンドの製品もグラフィックボードの形態で安価(10万円程度)に入手できる上に、その理論演算性能は1テラflop/s(CPUの約10倍)以上もある。現在、GPUは多種多様な方面でアクセラレータ(計算エンジン)としての利用価値を見い出されつつある。

ただし、あらゆる種類の計算がGPUによって高速化できるわけではない。基本的にはSIMD演算に合致したデータパラレル型の計算である必要がある。その上、GPUのハードウェアとコンパイラの詳細は非公開のため、GPUのプログラミングには経験に基づく知識とスキルが必要である。

研究代表者はこれまでに高速多重極積分方程式法を含む境界積分方程式法の研究に従事しており、近年は、GPUを用いた高速計算に取り組んで来た経緯がある。平成20年度は、3次元Helmholtz方程式に対する境界積分方程式法について、約5万円のグラフィックボード1基を使って約20倍のスピード

アップに成功した。平成21年度は、多体問題に関する高速多重極法(ブラックボックス高速多重極法)をGPUによって加速する研究を行なった。これらは本研究の先行研究に位置付けられる。

研究開始当初の時点で、高速多重極積分方程式法をGPUによって高速化した研究例はない。ただし、境界積分方程式法ではなくて多体問題に関する高速多重極法の分野では研究が少ないながらもなされていた。しかし、これらの並列化手法は特定の相互作用(クーロン力など)に限定されており、本研究の境界積分方程式法における波動問題への拡張は比較的困難である。他方、本先行研究であるブラックボックス高速多重極法は、主として多体問題分野に属する研究であるが、取り扱える相互作用に一定の任意性を許している。そのため、直接利用する事はできないとしても、本波動問題の高速多重極法の出発点とすることができる。

3. 研究の方法

3次元Helmholtz方程式に対する低周波領域用の高速多重極法を研究対象とし、計算負荷が特に高いM2L計算(多重極モーメントを局所展開係数へ変換する計算)を、先行研究で得た知見を基にして、GPUによって効率的に実行するためのマルチスレッド型並列アルゴリズム(スキーム)を構成する。このアルゴリズムを組み込んだ、高速多重極積分方程式法プログラムを開発する。具体的には次のように研究を遂行する。

まず、M2L計算を行列ベクトル積の形式に表現する。このように構造化された計算は、GPUによって比較的高速化が容易なSIMD演算と関連付け易くなるからである。実際、M2L計算は次のように表すことがわかった： $L(T) = \sum_S K(T, S)M(S)$ 。ここで、 L (局所展開係数)と M (多重極モーメント)は p 次元ベクトル、 K は p 次正方行列である。定数 p は近似精度を表わす入力パラメータである。 T は境界要素の集合(ターゲットセルと言う)を表し、集合の総数は N に比例して増大する。また、 S は T と相互作用する189個の集合(ソースセルと言う)を表す。すなわち、M2L計算とは、189回の行列ベクトル積(KM)を積算(Σ)することにより、 T 毎に L を求める計算である。

次に、M2L計算の並列アルゴリズムを検討する。上記のM2L計算には、(i)セル T に関する並列性、(ii)縦ベクトル L の行に関する並列性が含まれている。そこで、 T 毎に p 個のスレッドを割り当てて、各スレッドに1つの行を処理させるというアルゴリズムが考えられる。このような素朴なアルゴリズムは先行研究のブラックボックス高速多重極法において調査しており、CPUよりもGPUの方

が高速に M2L 計算を実行できている。

続いて、M2L 計算のメモリ演算（データの読み書き）の低減を試みる。M2L 計算に含まれる浮動小数点演算の総数は減らしようがないが、メモリ演算を減らすことは可能である。このような低減化も先行研究において検討しており、性能向上を確認している。その基本的なアイデアは、複数のセルをグループ化（ブロッキング）して、各グループにおいて共通なデータ（K や M）を一斉にキャッシュ（正確にはシェアードメモリ）上に読み込み、その後は、キャッシュ上のデータを使ってグループ内の各セルの M2L 計算を並列に実行する、というものである。さらに、兄弟セルの M2L 計算を同時実行する方法や、S に関する和を行列ベクトル積の列に関する和に先行させる方法と併用することが有効であることも見いだしている。これらを本研究でも検討する。

上記のアルゴリズムに従って M2L 計算を実行できるように GPU をプログラムする。開発したプログラムは、GPU を利用しないオリジナルのプログラムと比較して、計算時間と計算精度の面から検証を行なう。

最後に、M2L 計算と並んで計算負荷の高い「直接計算」（境界要素と境界選点との相互作用を直接評価する計算）の並列処理についても検討し、さらなる高速化を目指す。

4. 研究成果

(1) 主な研究成果

① 高性能 M2L スキームの開発

低周波領域に対する高速多重極法について、中でも計算負荷の高い M2L 計算をマルチスレッドプロセッサである GPU によって効率的に処理可能な計算アルゴリズム（スキーム）を提案した。そして、その計算コードを NVIDIA 社の GPU（Fermi シリーズ、モデル C2050）上で CUDA（当該 GPU 用の C/C++ コンパイラを含む開発環境）を用いて実装した。

本計算スキームは先行研究であるブラックボックス高速多重極法の研究成果を基とした。基本的には、GPU を利用した計算において陥り易いメモリ帯域の問題を回避するために、M2L 計算に現れる数種のループ（階層レベル、ターゲットセル、ソースセル、行、列などに関するループ）の順序を変更し、各種データ（M2L 行列など）のブロッキングを行った。具体的には、各プロセッサで GPU のメインメモリから一度ロードしたデータを 8 つの兄弟セルの間で再利用することを可能とする sibling blocking スキーム、および、多数のターゲットセル間でデータの共有を可能とする ij-blocking スキームを開発した。

他方、開発した GPU コードの比較対象である CPU コードでは、sibling blocking スキ

ームを採用した。ij-blocking スキームの実装は本質的にキャッシュプログラミングとなるために、そのコード開発は難しく、本研究では未実装である。

なお、波動問題に関する高速多重極積分方程式法では、計算精度の上で単精度では不十分（計算が破綻してしまう）であることが判明し、CPU および GPU コードは共に倍精度型で実装した。

② 高速多重極境界積分方程式法の加速

開発した GPU コードの計算精度および速度性能の検証を行った。計算精度については CPU コードと変わらない（計算順序の変更や丸め誤差に伴う誤差のレベルでは異なる）ことを確認した。速度性能の検証にあたっては、境界要素の偏在の度合いが低い場合（計算領域の全体に概ね一様分布している場合）と度合いが高い場合の二ケースを考慮した。

まず、前者の場合は階層構造として導入される八分木は概ね一様な深さを持つ。そのため、粒度の高い（すなわち、多数のセルを一括する）ブロッキングである ij-blocking スキームが有効であると考えられる。M2L 計算の実行性能を図 1 に示す。横軸にプロットした N は境界要素数（自由度）を表す。図中の「GPU $_x$ 」と「CPU $_x$ 」はそれぞれ GPU コードと CPU コード（12 コア利用；ピーク性能は合計約 160Gflop/s）の結果であることを示し、番号 $x=1$ は基本スキーム（ブロッキングを行わず、自明な並列化のみを行う）、番号 2 は sibling スキーム、番号 4 は ij-blocking スキームをそれぞれ利用していることを示す。CPU2 がせいぜい 30Gflop/s であるのに対して、GPU2 は約 100Gflop/s、GPU4 では 250Gflop/s を達成している。後者はピーク性能（515Gflop/s）の約 5 割であり、このような高い効率性は GPU 計算の分野でも特筆すべきことである。

ここで、図中の「GPU4/2」は 2 つのスキームを適宜切り替える、ハイブリッド型スキームの結果を示す。GPU4 単独の性能は、幾つかの N に対しては極端に低下してしまった。そのような N は、八分木の深さがちょうど変化する付近（図 3 に深さ κ を示す）である。この場合、最下層のセルには空のセルが多数存在するので、多数（実査には 512 個）のセルをブロックする ij-blocking スキームには無駄が発生しやすく、そのため性能が低下してしまう。他方、ブロッキングの粒度が小さい（実際には 8 個のみをブロックする）sibling-blocking スキームは有効となる。そこで、ある八分木レベルのセルに含まれる境界要素数の平均値が所定の値（具体的には 8 とした）よりも少ない場合、したがって、空セルの存在率も高いレベルについては、ij ではなく sibling blocking スキームを使う

こととした。

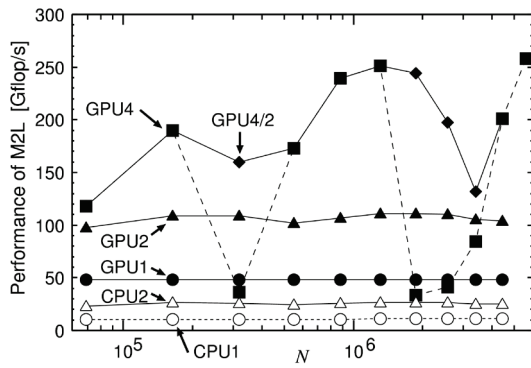


図 1. M2L 計算の性能比較 (境界要素の偏在度合いが低い場合)

以上の結果として、境界要素の偏在の度合いが低い場合 (計算領域全体的に分布している場合) において、本研究は約 560 万自由度の解析を、GPU ボード 1 基を搭載したデスクトップ計算機によって実現できた。計算時間は約 2.8 時間である。なお、本計算は応用上重要なフォノン結晶における音響散乱問題を模擬したものである。図 2 は $26 \times 26 \times 26$ 個の球形散乱体群 (奥側半分のみを表示) の下方向から平面波を入射した場合の全波動場の強度を示している。

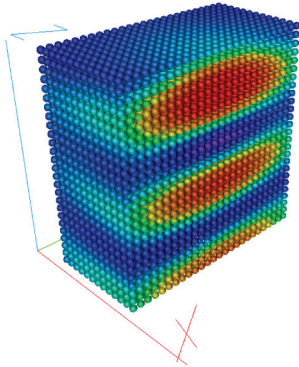


図 2. 560 万自由度モデル (半分のみ表示) における音響散乱解析 (全波動場の強度)

他方、境界要素の偏在の度合いが高い場合では、 N のいかに拘らず空のセルが多数発生するために、 ij -blocking スキームの性能は一律に低い結果となった (図 3)。これは上記のハイブリッドスキームを用いても解消できなかった。しかし、CPU コード (図 3 の CPU1 および CPU2) との比較においては、 $sibling$ -blocking スキームによって約 4 倍の高速化を達成できた。ここで、GPU2 が示す 80Gflop/s 程度の性能値は GPU のメインメモリのバンド幅の上限であることが理論計算から推量されている。

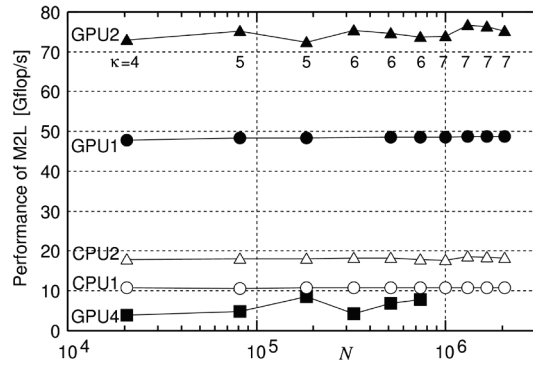


図 3. M2L 計算の性能比較 (境界要素の偏在度合いが高い場合)

③直接計算等の GPU への実装

M2L計算と並んで計算コストが高い直接計算部分についてもGPUへの実装を検討した。まずブラックボックス高速多重極法について検討し、3倍程度の高速化を達成した。その並列アルゴリズム自体は既存の研究に散見するものと大差はない。しかし、本研究はPTXコード (アセンブラコードの一種) を解析することによって理論性能を算出し、それが実効性能とよく一致することを示した。このような性能解析はGPUアプリケーションをチューニングする際に重要であることが普通であるが、本研究を通じて一つの具体例を挙げるのができたのは有用と考えられる。

続いて、上記の直接計算のフレームワークを当該の低周波用高速多重極積分方程式法の直接計算に組み込んだ。ところが、直接計算に現れる三角関数の処理時間が、CUDAによって提供される組み込みルーチンを用いても、また、自作したルーチンを用いても計算コスト上のネックとなり、CPUコードに比較して大きな性能向上を得ることができなかった。

一方、その他の主要な計算 (上向きパスにおける境界要素からモーメントの変換、および、モーメントの変換; 下向きパスにおける局所係数の変換、および、局所係数による評価) もGPUに実装し、それぞれが正しく動作することを確認した。しかし、特筆すべき性能の向上は得られなかった。その理由として、特殊関数 (球ベッセル関数) のように条件分岐を含まざるを得ない関数が並列計算に本来不向きであること、また、各計算の全体に占める割合が元々低いために、GPUの利用に伴うオーバーヘッドが無視できないことが挙げられる。この結果は、高速多重極法を構成する全ての計算をGPUで実行することが必ずしも最善ではないことを示唆している。むしろ、成功裡に加速できたM2L計算のように、特定の箇所だけをGPUで実行するような、ピンポイント的なGPUの利用方法が (少なくとも現状のGPUハードウェアにおいては) 妥当であると結論できる。

(2) 国内外における位置付け

本研究実施期間中、国外の数グループによって波動問題に対する高速多重極法をGPUによって加速する研究がなされた。これらの研究はCPUコードをGPUへ移植することに成功したと言うことはできるが、GPUの特性に見合う計算手法を提案しているとは必ずしも言い難い。特にM2L計算に関しては新しいスキーム(アルゴリズム)が提案されているわけではなく、この点で本研究はイニシアチブを得ていると考えられる。

(3) 今後の展望

本研究は3次元Helmholtz方程式に対する境界積分方程式法について研究したが、本手法は周波数領域の波動問題に対する種々の高速多重極境界積分方程式法に対しても適用可能である。その一旦として、2次元動弾性問題に対する高速多重極境界積分方程式法について基礎的検討を始めた。

また、本研究は低周波領域に対する高速多重極法を対象とした。理論的には高周波領域にも適用できるが、周波数が高い場合には計算コスト上は不利になる(0(N)ではなくなる)。本研究が開発したGPUによる高速化手法によってその計算コストは緩和することは可能である。しかし、さらに高い周波数を扱うためには、対角形式に基づく高周波領域用の高速多重極法に基づくことが避けられない。今後は高周波領域用高速多重極法に対するGPU用並列アルゴリズムの開発が重要になると思われる。

さらに、本研究の母体となったブラックボックス高速多重極法に対するGPU計算については、米国スタンフォード大学のグループに加え、フランスの研究機関INRIAと共同研究の輪が広がっており、その共同研究によって得られる成果が波動問題に対する高速多重極法に還元できるものと期待される。

5. 主な発表論文等

(研究代表者、研究分担者及び連携研究者には下線)

[雑誌論文] (計8件)

- ① Toru Takahashi, Cris Cecka, Eric Darve, Optimization of the parallel black-box fast multipole method on CUDA, InPar2012 (Innovative Parallel Computing), 査読有, (in CDROM), 2012
- ② Toru Takahashi, Cris Cecka, Eric Darve, The fast multipole method on CPU and GPU processors, Program Guide of GPU Technology Conference, 査読無, 2012, p. 78
- ③ Toru Takahashi, A wideband fast

multipole accelerated boundary integral equation method for time-harmonic elastodynamics in two dimensions, International Journal for Numerical Methods in Engineering, 査読有, 2012, DOI:10.1002/nme.4288

- ④ Toru Takahashi, Cris Cecka, Eric Darve, Optimizing the multipole-to-local operator in the fast multipole method for graphical processing units, International Journal for Numerical Methods in Engineering, 査読有, Vol.89, Issue 1, 2012, pp.105-133, DOI:10.1002/nme.3240
- ⑤ Toru Takahashi, Cris Cecka, Eric Darve, An acceleration of multipole-to-local operation for low-frequency fast multipole BIEM with GPU, Extended Abstracts for IABEM 2011 (Symposium of the International Association for Boundary Element Methods), 査読無, 2011, pp.303-308
- ⑥ Toru Takahashi, Cris Cecka, Eric Darve, An implementation of multipole-to-local (M2L) operation for low-frequency fast multipole BIEM on GPU, Proceedings of The 11th U.S. National Congress on Computational Mechanics (in CDROM), 査読無, 2011, No.213538
- ⑦ Eric Darve, Cris Cecka, Toru Takahashi, The fast multipole method on parallel clusters, multicore processors, and graphics processing units, Comptes Rendus Mechanics, 査読有, Vol.339, 2011, pp.185-193, DOI: 10.1016/j.crme.2010.12.005
- ⑧ Toru Takahashi, Cris Cecka, Eric Darve, An implementation of low-frequency fast multipole BIEM for Helmholtz' equation on GPU, 計算力学講演会講演論文集 (in CDROM), 査読無, 2010-09-23, 2010, pp.319-321

[学会発表] (計6件)

- ① Toru Takahashi, Cris Cecka, Eric Darve, The fast multipole method on CPU and GPU processors, GPU Technology Conference 2012, 2012.5.14-17, San Jose, California, USA
- ② Toru Takahashi, Cris Cecka, Eric Darve, Optimization of the parallel black-box fast multipole method on CUDA, InPar2012 (Innovative Parallel Computing), 2012.5.13-14, San Jose, California, USA
- ③ Toru Takahashi, Cris Cecka, Eric Darve,

An acceleration of multipole-to-local operation for low-frequency fast multipole BIEM with GPU, IABEM 2011 (Symposium of the International Association for Boundary Element Methods), 2011.9.5-8, Brescia, Italy

- ④ Toru Takahashi, Cris Cecka, Eric Darve, An implementation of multipole-to-local (M2L) operation for low-frequency fast multipole BIEM on GPU, The 11th U.S. National Congress on Computational Mechanics, 2011.7.25-28, Minneapolis, Minnesota, USA
- ⑤ 高橋徹, 計算力学最前線 : GPU, 高大連携講座(出前授業), 2010.10.25, 愛知県立横須賀高等学校
- ⑥ Toru Takahashi, Cris Cecka, Eric Darve, An implementation of low-frequency fast multipole BIEM for Helmholtz' equation on GPU, 日本機械学会第23回計算力学講演会, 2010.9.23-24, 北見工業大学

[その他]

本研究で作成した計算プログラムの一部は下記のホームページで公開している。

<http://sourceforge.net/projects/bbfmmgpu>

6. 研究組織

(1) 研究代表者

高橋 徹 (TAKAHASHI TORU)
名古屋大学・工学研究科・講師
研究者番号 : 90360578

(2) 研究分担者なし

(3) 連携研究者なし