

Diagonal form 多重極積分方程式法の並列化

A Parallel Implementaion of Diagonal Fast Multipole BIEM

吉田 研一¹⁾, 西村 直志²⁾

Ken-ichi YOSHIDA and Naoshi NISHIMURA

- 1) 京都大学学術情報メディアセンター (〒 606-8501 京都市左京区吉田本町, E-mail: yoshida@gspsun1.gee.kyoto-u.ac.jp)
 2) 京都大学学術情報メディアセンター (〒 606-8501 京都市左京区吉田本町, E-mail: nchml@media.kyoto-u.ac.jp)

This paper discusses a parallel implementation of the diagonal form fast multipole BIEM. Diagonal form FM-BIEMs are suitable for parallelization because the M2L operations are dianonalized. Taking this advantage, we parallelize an implementation of the diagonal form fast multipole BIEM with MPI. In the numerical analysis we deal with crack problems for three-dimensional Helmholtz' equation. We use a PC cluster to test the parallel implementation. The numerical results show the efficiency of the proposed method.

Key Words: Diagonal Form Fast Multipole BIEM, MPI

1. はじめに

多重極法は未知数の多い問題に有利な数値計算の手法のひとつとして、積分方程式や、多体問題に関連して Rokhlin⁽¹⁾, Greengard⁽²⁾ らによって研究され、発達してきた。Helmholtz 方程式への多重極法の適用に関する研究は、diagonal form⁽³⁾ を用いたものが盛んである。Helmholtz 方程式を扱う場合、波数や問題の規模が大きくなると、M2L の計算量や多重極モーメント、局所展開係数等を記憶するのに必要なメモリが非常に多くなる。このため、大規模問題の解析では並列計算を視野に入れる必要が出てくる。幸い、Diagonal form は多重極法で導入するツリー構造を意識することなく、容易に並列化することができる。本論文では PC クラスタを用いて、Helmholtz 方程式のクラック問題を扱った diagonal form 多重極積分方程式法のプログラムの並列計算を行ない、並列化効率を検証する。

2. 積分方程式の定式化

無限領域 R^3 中のクラック S による散乱問題は次のように定式化される。

$$\begin{aligned} \text{支配方程式} \quad & (\Delta + k^2) u(x) = 0 \quad \text{in } R^3 \setminus \bar{S} \\ \text{境界条件} \quad & \frac{\partial u^\pm}{\partial n}(x) = 0 \quad \text{on } S \\ \text{正則条件} \quad & \phi(x) = 0 \quad \text{on } \partial S \end{aligned}$$

遠方で $u - u_I$ は放射条件を満たす。

ここに、クラックとは、自分自身と交わらない、縁 ∂S を有する、一般には複数の滑らかな曲面であり、 Δ は Laplacian, k は波数、 u は全波動場、 u_I は入射波である。また、 ϕ はク

ラック上での u の不連続量であり、

$$\phi(x) := u^+(x) - u^-(x)$$

と表される。上つきの $+$ ($-$) はクラックの単位法線ベクトル \mathbf{n} の正(負)の向きからの極限値を意味する。解の積分表示は次のようになる。

$$u(x) = u_I(x) + \int_S \frac{\partial G(x-y)}{\partial n_y} \phi(y) dS_y \quad (1)$$

$$x \in R^3 \setminus \bar{S}$$

ここに、 $\partial/\partial n_y$ はクラック上の点 y における法線微分を表し、 $G(x-y)$ は 3 次元 Helmholtz 方程式の基本解である。また、式 (1) の積分の dS_y は $y \in S$ に関する面積分を意味する。時間依存が $e^{-i\omega t}$ の時、 $G(x-y)$ は

$$G(x-y) = \frac{e^{ik|x-y|}}{4\pi|x-y|}$$

と書ける。次に、式 (1) を x で法線微分した後に、領域内の点 x をクラック上の点へと極限移行し、境界条件を考慮すると次の超特異積分方程式が得られる。

$$-\frac{\partial u_I}{\partial n_x}(x) = \text{p.f.} \int_S \frac{\partial^2 G}{\partial n_x \partial n_y}(x-y) \phi(y) dS_y \quad (2)$$

$$x \in S$$

ここに、p.f. は発散積分の有限部分を表す。

3. Diagonal form 多重極積分方程式法

3.1. 多重極モーメント及び局所展開係数

まず、Helmholtz 方程式の基本解の積分表示⁽⁴⁾

$$\frac{e^{ikr}}{4\pi r} = \frac{ik}{16\pi^2} \int_{S_0} e^{ik\hat{\mathbf{k}} \cdot \vec{x}_0} \mathcal{T}(k, \hat{\mathbf{k}}, \vec{y}_0 \vec{x}_0) e^{-ik\hat{\mathbf{k}} \cdot \vec{y}_0} dS_{\hat{\mathbf{k}}}, \quad (3)$$

を用いる．ここに，不等式 $|x - x_0 + y_0 - y| < |x_0 - y_0|$ を仮定した．式 (3) において， $\hat{\mathbf{k}}$ は単位ベクトル

$$(\cos \phi \sin \theta, \sin \phi \sin \theta, \cos \theta), \quad (0 \leq \theta \leq \pi, 0 \leq \phi \leq 2\pi),$$

を表し， S^0 は単位球面である．また， $T(k, \hat{\mathbf{k}}, \overrightarrow{y_0 x_0})$ は

$$T(k, \hat{\mathbf{k}}, \overrightarrow{y_0 x_0}) = \sum_{n=0}^q (2n+1) i^n h_n^{(1)}(k|\overrightarrow{y_0 x_0}|) P_n \left(\frac{\overrightarrow{y_0 x_0}}{|\overrightarrow{y_0 x_0}|} \cdot \hat{\mathbf{k}} \right). \quad (4)$$

のように級数で表される．式 (4) において， $h_n^{(1)}$, P_n はそれぞれ，球 Hankel 関数，Legendre 関数であり，打ち切り項数 q は Song et al.⁽⁵⁾ に従って決定する．

多重積分方程式法では，ソース点 y と観測点 x が近い時は基本解の特異性を考慮して従来と同じように計算を行ない，遠い時は多重極法を用いて効率良く計算を行なう．つまり，多重積分方程式法では，ソース点 y と観測点 x が離れていると想定して定式化を行なう．今，クラック S の部分 S_y の積分 (2) への寄与を評価する．ただし， x は S_y から十分はなれているとする．この時，式 (3) より次式を得る．

$$\begin{aligned} & \int_{S_y} \frac{\partial^2 G}{\partial n_x \partial n_y} (x-y) \phi(y) dS_y \\ &= \frac{ik^3}{16\pi^2} \int_{S^0} \hat{\mathbf{k}} \cdot \mathbf{n}(x) e^{ik\hat{\mathbf{k}} \cdot \overrightarrow{x_0 \hat{x}}} L(\hat{\mathbf{k}}, x_0) dS_{\hat{\mathbf{k}}} \end{aligned} \quad (5)$$

ここに， L は局所展開係数であり，多重極モーメント M により次のように表される．

$$L(\hat{\mathbf{k}}; x_0) = T(k_T, \hat{\mathbf{k}}, \overrightarrow{y_0 x_0}) M(\hat{\mathbf{k}}; y_0), \quad (6)$$

また，多重極モーメントは次のような積分で与えられる．

$$M(\hat{\mathbf{k}}; y_0) = \int_{S_y} \hat{\mathbf{k}} \cdot \mathbf{n}(y) e^{-ik\hat{\mathbf{k}} \cdot \overrightarrow{y_0 \hat{y}}} \phi(y) dS \quad (7)$$

3.2. 単位球面 S^0 上での数値積分

本論文では，単位球面上の積分は， θ 方向は Gauss 積分， ϕ 方向は台形公式を用いて，次のように数値的に行なっている．

$$\begin{aligned} \int_{S^0} f(\hat{\mathbf{k}}) dS_{\hat{\mathbf{k}}} &= \int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} f(\theta, \phi) \sin \theta d\theta d\phi \\ &= \int_{\phi=0}^{2\pi} \int_{x=-1}^1 f(\cos^{-1}(x), \phi) dx d\phi \\ &= \sum_{J=0}^{2p} \sum_{I=1}^{p+1} w_I^p \eta_J^p f(\theta_I^p, \phi_J^p) \end{aligned}$$

ここに， $f(\theta, \phi)$ は球面上の関数， θ_I^p と w_I^p は $p+1$ 点の Gauss 積分公式の I 番目の分点の逆余弦と重みである．また， $\phi_J = 2\pi J / (2p+1)$, $\eta_J^p = 2\pi / (2p+1)$ である．式 (5) の積分を数値的に評価する時には，球面上の有限の標本点での局所展開係数を用いて行なう．

3.3. 多重極モーメント及び局所展開係数の移動公式

式 (5) の積分の精度を保つためには，より高 (低) い周波数または，大きい (小さい) 領域を扱う時は，必要に応じて多重極モーメントや局所展開係数の標本点の分布を密 (疎) にすればよい．多重極法での M2M (L2L) では，ツリー構造

を昇る (降りる) につれて扱うセルのサイズが大きく (小さくなる) ので，多重極モーメント (局所展開係数) を far-field transform^(1, 6) を用いて補間する．

3.3.1 Far-field transform

球面上の関数 f の far-field transform，即ち球面調和関数による Fourier 係数は次のように定義される．

$$\tilde{f}_{n,m} = \int_{S^0} Y_{n,m}(\theta, \phi) f(\theta, \phi) dS_{\hat{\mathbf{k}}} \quad (8)$$

ここに， $Y_{n,m}(\theta, \phi)$ は正規化された球面調和関数である．また，逆変換は

$$f(\theta, \phi) = \sum_{n=0}^{\infty} \sum_{m=-n}^n Y_{n,m}^*(\theta, \phi) \tilde{f}_{n,m} \quad (9)$$

で表される．ここに， $*$ は複素共役を表す．

式 (8) の積分は 3.2 での数値積分の方法と同様に次のように評価される．

$$\begin{aligned} \tilde{f}_{n,m} &= \int_{S^0} Y_{n,m}(\theta, \phi) f(\theta, \phi) dS_{\hat{\mathbf{k}}} \\ &\approx \sum_{J=0}^{2p} \sum_{I=1}^{p+1} w_I^p \eta_J^p Y_{n,m}(\theta_I^p, \phi_J^p) f(\theta_I^p, \phi_J^p) \end{aligned} \quad (10)$$

式 (10) の J に関する和には高速 Fourier 変換を用いることが出来る．

結局，球面上の有限な標本点での関数値の集合 $\{f(\theta_I^p, \phi_J^p) | I = 1, \dots, p+1, J = 0, \dots, 2p\}$ を用いて，他の標本点での関数値の集合 $\{f(\theta_I^{p'}, \phi_J^{p'}) | I = 1, \dots, p'+1, J = 0, \dots, 2p'\}$ を得るには

$$\tilde{f}_{n,m} = \sum_{J=0}^{2p} \sum_{I=1}^{p+1} w_I^p \eta_J^p Y_{n,m}(\theta_I^p, \phi_J^p) f(\theta_I^p, \phi_J^p) \quad (11)$$

$$f(\theta_I^{p'}, \phi_J^{p'}) = \sum_{n=0}^p \sum_{m=-n}^n Y_{n,m}^*(\theta_I^{p'}, \phi_J^{p'}) \tilde{f}_{n,m} \quad (12)$$

を用いて計算する．M2M (L2L) では $p < p'$ ($p > p'$) となる．

3.3.2 多重極モーメント，局所展開係数の移動

多重極展開の中心を y_0 から y_1 に移動する時は次の 2 つの作業を行なう．

- 多重極モーメントを次の式で移動する．

$$M(\hat{\mathbf{k}}; y_1) = M(\hat{\mathbf{k}}; y_0) e^{-ik\hat{\mathbf{k}} \cdot \overrightarrow{y_1 y_0}}.$$

- 式 (11) と (12) で多重極モーメントを補間する．

また，局所展開の中心を x_0 から x_1 に移動する時は多重極モーメントと同様に以下の作業行なう．

- 式 (11) と (12) で局所展開係数を補間する．

- 局所展開係数を次の式で移動する．

$$L(\hat{\mathbf{k}}; x_1) = L(\hat{\mathbf{k}}; x_0) e^{ik\hat{\mathbf{k}} \cdot \overrightarrow{x_0 x_1}}.$$

4. Diagonal form 多重極積分方程式法の並列化

4.1. 多重極積分方程式法について

多重極積分方程式法では、線形方程式のソルバーに反復法を用いる。反復法では $Ax = b$ の解を求める時に一般に次のようなことが繰り返し行なわれている。

- x に x_0 を与える。
- $b_0 = Ax_0$ を計算する。
- 残差ノルム $\|b_0 - b\|$ を小さくするように x_0 を更新する。

境界積分方程式法では、多重極法は通常未知数の2乗のオーダーのコストを要する Ax_0 の演算量を軽減するのに用いられている。多重極法では観測点から離れたソース点から境界積分への寄与を局所展開係数を用いて式(5)によって評価し、観測点の近傍のソース点からの寄与は直接計算で評価する。それぞれを足し合わせることによって、式(5)の左辺つまり、 b_0 を計算する。ここで、多重極法で行なう計算を整理すると次のようになる。

- A 式(7)による多重極モーメントの計算
- B 式(6)による M2L
- C 式(11), (12)による M2M, L2L の計算
- D 式(5)により遠方からの境界積分への寄与 b_0^{far} を評価し、 $b_0 = b_0 + b_0^{far}$ で足し込んでいく。
- E 式(2)により近傍からの境界積分への寄与 b_0^{near} を評価し、 $b_0 = b_0 + b_0^{near}$ で足し込んでいく。

本論文では、 Ax_0 の計算の部分のみを並列化しており、ソルバー自体は並列化していないので、各プロセッサでは全く同じ x_0 の更新作業が行なわれていることになる。各プロセッサでは b_0 に対応する記憶領域が確保されており、 I 番目のプロセッサのそれを b_0^I と書くことにする。

4.2. 多重極モーメントの計算の並列化 (A の並列化)

単位球面上を θ 方向にプロセッサの数だけ分割し、それぞれのプロセッサが対応する部分の多重極モーメントを計算し、それらを記憶する。つまり、プロセッサ数を N_p とすると、 I 番目のプロセッサは $\frac{(I-1)\pi}{N_p} \leq \theta \leq \frac{I\pi}{N_p}$, $0 \leq \phi \leq 2\pi$ 上にある標本点での多重極モーメントを計算し、記憶すれば良い。この様に分割すれば、式(10)でのFFTの計算に影響は出ない。以下では、 I 番目のプロセッサが持つ多重極モーメントを M^I と書く。なお、このように多重極モーメントを \hat{k} の方向ごとに各プロセッサに分散させる方法は、Chew らが Maxwell 方程式の多重極 code である ScaleME⁽⁷⁾ で上位のセルに対して用いているが、本論文の方式はこれを全レベルで用いている点で異なっている。

4.3. M2L の並列化 (B の並列化)

M2L は各プロセッサのが記憶している多重極モーメントに関してのみ行なえば良く、通信をする必要は全くない。

4.4. M2M 及び L2L の並列化 (C の並列化)

M2M と L2L の並列化の手順は同じであるので、ここでは M2M の並列化について述べる。M2M を行なう際に、式(11)

を計算するためには、全単位球面上の標本点での多重極モーメントが必要となる。しかし、多重極モーメントはそれぞれのプロセッサに分割されて記憶されているので、通信が必要になる。そこで、次のように並列化を行なう。

- まず、各プロセッサの持つ多重極モーメント $M^I(\theta, \phi)$ ($I = 1 \dots N_p$) を用いて $\tilde{f}_{n,m}$ への寄与 $\tilde{f}_{n,m}^I$ ($I = 1 \dots N_p$) を計算する。 $\tilde{f}_{n,m}$ と $\tilde{f}_{n,m}^I$ ($I = 1 \dots N_p$) の関係は

$$\tilde{f}_{n,m} = \sum_{I=1}^{N_p} \tilde{f}_{n,m}^I$$

である。

- 次に、各プロセッサで計算した $\tilde{f}_{n,m}^I$ を MPI ライブラリの `MPI_ALLREDUCE`⁽⁸⁾ を用いて足し合わせ、各プロセッサが式(12)の計算に必要な $\tilde{f}_{n,m}$ を得る。`MPI_ALLREDUCE` は各プロセッサのベクトルを足し合わせつつ、結果をブロードキャストする MPI ライブラリ subroutine である。
- 最後に、通信により得た $\tilde{f}_{n,m}$ を用いて、各プロセッサに割り当てられた標本点での $M^I(\theta, \phi)$ を計算する。

4.5. 局所展開係数による評価の並列化 (D の並列化)

式(5)の右辺の積分は単位球面上の標本点での局所展開係数を用いて数値積分で評価する。各プロセッサでは自分の記憶している局所展開を用いて b_0^{far} への寄与を計算し、 b_0^I に足し込んでいけば良い。ここで、通信を行なう必要はない。

4.6. 直接計算の並列化 (E の並列化)

多重極法の直接計算では二つのセル内の境界要素間での係数行列の計算と、得られた係数行列と対応する x_0 の部分との積を計算することになる。ここでは、係数行列の行方向を N_p 分割し、各プロセッサが担当する係数行列を計算し、対応する x_0 の部分との積を計算して b_0^{near} への寄与を求め、 b_0^I に足し込む。ここでも、通信を行なう必要はない。

4.7. 最終的な通信

x_0 の更新に必要な b_0 を得るには各プロセッサが持つ b_0^I ($I = 1 \dots N_p$) を足し合わせる必要がある。これを `MPI_ALLREDUCE` によっておこなう。この作業により各プロセッサでの b_0^I の値が b_0 の値となり、各プロセッサで x_0 の更新作業を行なうことが可能となる。

5. 数値実験

ここでは、以上に述べた並列化の効率を調べるために行なった数値実験について述べる。

5.1. 計算環境

計算機には DEC ALPHA 21264(666MHz) を2個、メモリを2GBを搭載したSMP計算機4台をmyrinetで結合したPCクラスタを使用した(1プロセッサ当たりのメモリは1GB)。使用した通信ライブラリはMPIの仕様⁽⁹⁾に準じたフリーで配布されている実装の一つであるmpich⁽¹⁰⁾である。また、高速Fourier変換にはFFTW⁽¹¹⁾を用い、線形方程式のソルバーにはNetlibより入手可能なBICGSTABを用いている。

5.2. 数値計算例

無限領域に、半径 a のランダムな法線ベクトルを持つ円形クラックが各座標方向に 4 個ずつ、計 64 個 (総未知数 42,496) 存在し、各座標方向に $4a$ の間隔で規則的に並んでいるものとする。そこに次のような平面波 ($ka = 10$) が入射する場合を考える。

$$u_I(x) = u_0 e^{ikz}, \quad u_0 : \text{定数}$$

その時の開口変位を計算するプログラムを並列化し、CPU の数が 1, 2, 4, 6, 8 個の時の計算時間プロットしたのが Fig.1、計算時間を CPU の数が 1 個の場合の計算時間で割ったものをプロットしたのが Fig.2 である。

これらの結果より、並列化による計算効率の向上が達成されている事がわかる。

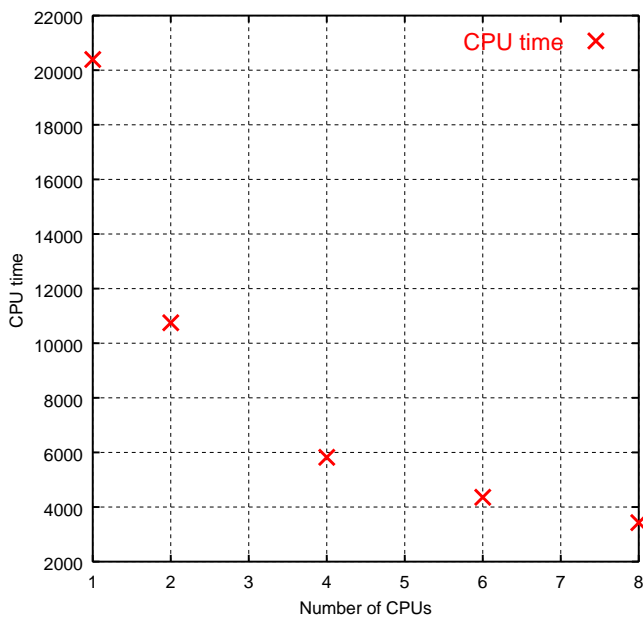


Fig. 1 計算時間

6. 結言

今回の結果では、8CPU の時で約 5.9 倍の高速化となり、一定の効率化を達成する事はできたが、更なる改善の余地も残す結果となった。今回の並列化では、M2M, L2L を行なうたびに通信を行なっているので通信回数が多く、通信によるオーバーヘッドがかなり影響していると考えられる。このため、CPU の数が増えると並列化効率が頭打ちになる事は容易に予想される。ただし、今回の並列化手法では、演算の並列化に必要な MPI のライブラリは MPI_ALLREDUCE だけで済むので非常に簡単であり、ツリー構造からロードバランスを考慮して並列化を行なうよりは手軽に並列化することが出来ると言えよう。今後は、通信回数を軽減し、並列化効率をあげて行きたい。

謝辞: 本研究は日本学術振興会特別研究員奨励費の援助を受けている。

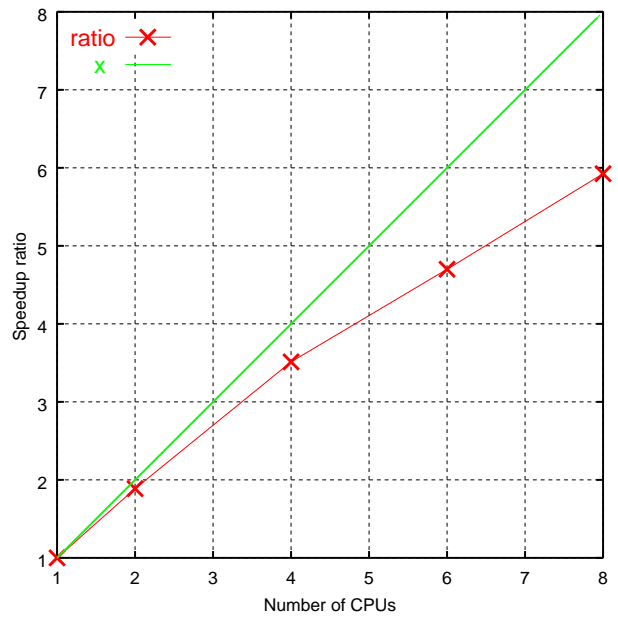


Fig. 2 並列化効率

参考文献

- (1) V. Rokhlin : Rapid solution of integral equations of classical potential theory, *J. Comp. Phys.*, 60, 187–207, 1985.
- (2) L. Greengard: The Rapid Evaluation of Potential Fields in Particle Systems, 1987, The MIT Press.
- (3) V. Rokhlin : Diagonal forms of translation operators of the Helmholtz equation in three dimensions, *Appl. Comp. Harmonic Anal.*, 1, 82–93, 1993.
- (4) M.A. Epton and B. Dembart : Multipole translation theory for the three-dimensional Laplace and Helmholtz equations, *SIAM J. Sci. Comput.*, 16, 865–897, 1995.
- (5) J.M. Song, C.C. Lu and W.C. Chew : Multilevel fast multipole algorithm for electromagnetic scattering by large complex objects, *IEEE Trans. Antennas Propagation*, 45(10), 1997.
- (6) M.F. Gyure and M.A. Stalzer : A prescription for the multilevel Helmholtz FMM, *IEEE Comp. Sci. Eng.*, 5(3), 39–47, 1998.
- (7) <http://www.ccem.uiuc.edu/reschew12.html>
- (8) 青山幸也 : 並列プログラミング虎の巻 MPI 版, 日本 IBM
- (9) <http://www.mpi-forum.org/>
- (10) <http://www-unix.mcs.anl.gov/mpi/mpich/>
- (11) <http://www.fftw.org/>