

共有メモリ計算機における有限要素法の並列化

京都大学大学院工学研究科

学生会員 ○堀翔一

京都大学学術情報メディアセンター

正会員

西村直志

1. はじめに

有限要素法は、固体力学や流体力学などの様々な分野で使用されている微分方程式の数値解析法で、大規模問題にて適用可能であり、今日、計算力学の代表的な数値解析手法となっている。

有限要素法の大規模問題への適用においては並列化を行うことが重要であり、GeoFEM⁽¹⁾ や Adventure⁽²⁾ などに代表される並列化コードが公開されている。これらのコードでは領域分割法、element-by-element 法などが用いられ、並列化には PC クラスタ等の分散メモリ環境を想定して MPI が用いられている。一方、共有メモリ並列計算機における API として OpenMP が標準的に用いられており、既存の逐次処理プログラムに対して指示文を使ってプログラムを並列化することができる。共有メモリ環境における並列化は、その容易さが最大の特徴である。しかし大規模共有メモリ計算機があまり存在しなかったこともあって有限要素法の並列化に関する研究も多くない。そこで本研究では OpenMP のみを用いた共有メモリ計算機における有限要素法の並列化手法について研究する。

本研究では簡単のために 2 次元ラプラス問題をとりあげ、ソルバとしては element-by-element 法に基づいて並列化した共役勾配法⁽³⁾ を用いる。並列化手法の影響や台数効果を調べ、MPI を用いたコードとの比較を行う。

2. 定式化

2 次元領域 Ω でのラプラス方程式の境界値問題は次のように表される。

$$\begin{aligned}\phi_{,ii} &= 0 & \text{in } \Omega \\ \phi &= \bar{\phi} & \text{on } \Gamma_\phi \\ q = \frac{\partial \phi}{\partial n} &= \bar{q} & \text{on } \Gamma_q\end{aligned}$$

ここに n は Ω の境界 $\Gamma = \Gamma_\phi + \Gamma_q$ の外向き単位法線ベクトルである。

等価変分方程式は次の式である。

$$\sum_{e=1}^N \int_{\Omega_e} \frac{\partial \phi^*}{\partial x_i} \frac{\partial \tilde{\phi}}{\partial x_i} d\Omega = \int_{\Gamma_q} \phi^* \bar{q} d\Gamma$$

ϕ^* : 重み関数

キーワード:有限要素法, 共役勾配法, 並列化, OpenMP, ATOMIC
連絡先:〒 606-8501 京都市左京区吉田本町 京都大学学術情報メディアセンターメディアコンピューティング分野 TEL: 075-753-7457 FAX: 075-753-7450

ϕ に有限要素補間を用いると次の有限要素方程式が得られる。

$$K\phi = F$$

ここで係数マトリックス K と $\bar{q}$ の等価節点ベクトル F は次式で表される。

$$\begin{aligned}K &= \sum_{e=1}^N \int_{\Omega_e} \frac{\partial \Phi_\alpha}{\partial x_i} \frac{\partial \Phi_\beta}{\partial x_i} d\Omega \\ F_i &= \int_{\Gamma_q} \Phi_i \bar{q} d\Gamma\end{aligned}$$

ここに Φ_α, Φ_β は形状関数である。

得られた有限要素方程式の解法として、並列処理向きである element-by-element 共役勾配法を用いる。

3. 有限要素法の並列化

OpenMP で element-by-element 法を並列化する場合、鍵となるのは言うまでもなく element-by-element 処理である。これを MPI で並列化する場合、図 1 のように要素をプロセスに振り分ける。各プロセスでは受け持った要素からの $K\phi$ への寄与を計算し、通信でこれを加えあわせることにより $K\phi$ を求める。同様の並列化を OpenMP で実行すると、2 つのスレッドの境界にある節点では、 $K\phi$ を格納する配列に 2 つのスレッドからアクセスが発生する。そのため、正しい答えを得るためには ATOMIC 構文等を用いて排他アクセスを行う必要がある。

しかし一般的には ATOMIC の処理は大変時間がかかり並列化の効果が現れにくい。このため図 2 のように節点をスレッドに割り当て、関連する要素の要素剛性行列の計算を各スレッドに割り当てる事にする。このとき、いくつかの要素は複数のスレッドに割り当てられることになるが、 $K\phi$ の計算におけるメモリ書き込みのアクセス競合は発生しない。

数値計算においては、ATOMIC を用いた OpenMP 版の並列コード、ATOMIC を用いない OpenMP 版並列コード、及び MPI によるコードの性能を比較した。

4. 数値計算結果

計算例として図 3 のような長方形領域を考える。

境界条件は図 3 に示す。また、要素分割は縦方向に 100 節点で固定し、横方向は 10 ~ 10000 節点の範囲で変化した。横方向 10000 節点のとき、全自由度は 1000000、要素数は 1979802 である。共役勾配法の収束判定は 1.0E-04 とし前処理は行っていない。計算には京都大学学術情報

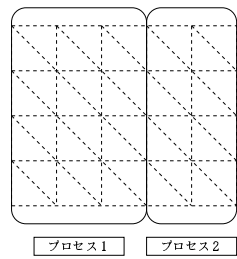


図 1: 各プロセスの受け持つ要素

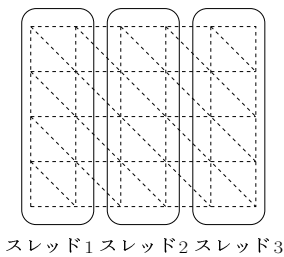


図 2: 各スレッドに割り当て関連する要素

メディアセンターの hpc2500 を用いた。繰り返し回数について横列の節点数を変えた場合、表 1 のようになった。

表 1: 節点数と繰り返し回数

横列の節点数	繰り返し回数
10	36
100	992
1000	1408
10000	10044

CPU の数と CPU time、スピードアップの関係を、ATOMIC なしの OpenMP(表 2)、MPI で並列化した場合(表 3) について示す。ATOMIC ありの OpenMP の場合にはかなりの時間がかかると思われるために、途中 (1 時間程度) で計算を打ち切った。

表 2: ATOMIC なしの OpenMP 並列の場合

CPU 数	CPU time	スピードアップ
1	1585	1.00
2	853	1.86
5	455	3.48
8	368	4.31
10	300	5.28
12	256	6.19
16	207	7.66

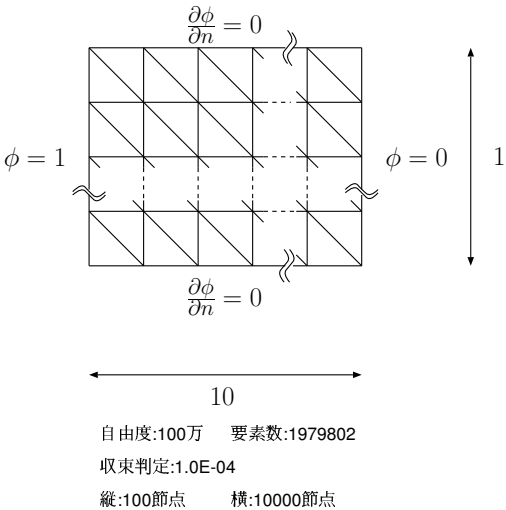


図 3: 長方形領域

表 3: MPI で並列化した場合

CPU 数	CPUtime	スピードアップ
1	1889	1.00
2	1045	1.81
5	504	3.75
8	379	4.98
10	303	6.23
12	247	7.65
16	195	9.69

これより ATOMIC を用いるとほとんど並列化の効果がでないことがわかる。また、ATOMIC を用いない場合、スレッド数を増やせば CPU time は短縮される。しかし、スレッド数多くなるとスピードアップは頭うちとなる。このことは MPI で並列した場合でも同じことがいえる。この例では MPI の方はやや高速となっているが、大きな差は見られない。

5. 結論

OpenMP を用いて有限要素法を element-by-element 法で並列化を行い、MPI に比べて遜色のない結果を得ることが出来た。今後、より一般的な問題への適用を検討する予定である。

参考文献

(1) GeoFEM <http://www.geofem.tokyo.rist.or.jp/>
(2) Adventure <http://www.adventure.q.t.u-tokyo.ac.jp/>
(3) 櫻山和男, 西村直志, 牛島省: 並列計算法入門, 丸善, 2003.