

分散メモリ型並列計算機上の有限要素法のための要素自動分割

7L-8

河野 洋一 福盛 秀雄 安江 俊明 村岡 洋一
早稲田大学

1 はじめに

粗分割要素をPEに割り当てる要素分割の並列化に必要な、高速三角形粗分割アルゴリズムを提案する。また有限要素法の前処理の要素分割部を並列化する際に、本アルゴリズムを使用し高速化を実現する。

本稿では、入力図形を逐次に三角形粗分割したのち最終的な節点数が各PEに均等になるように粗分割要素を割り当て、各PEが並列に要素分割を行なう並列化要素分割について述べる。なお本稿では要素分割に三角形分割を適用する。

2 要素分割の従来の研究

有限要素法のための要素分割の並列化の手法には、はじめに逐次処理部で粗分割し、その粗分割要素をPEに割り当てる方法がある。例えば、[3]では、あらかじめ粗分割した四角形の節点数を計算し、各四角形を節点数が均等になるようにPEに割り当てる。PEは割り当てられた粗分割四角形を並列に要素分割する。節点数の偏りがあれば、節点数の多いPEから節点数の少ないPEへ要素を通信し節点数の偏りを減少する方法を提案している。

本稿では[3]で提案されている<節点数の偏りを減少させるため、はじめに節点数の計算をする>アイデアを用い、64台のPEでの並列化要素分割を提案する。

3 並列化要素分割部

分散メモリ型並列計算機を対象とした要素分割部の特徴を説明する。処理の流れは、(A)逐次粗分割部、(B)PEへの割り当て、(C)並列要素分割部、の3つに分けられる(図1)。

まず(A)の逐次粗分割部においては、PEに処理を割り当てるために必要な粗分割三角形数 n は(a)問題によって、(b)PEの使用台数によって変わる。そのため条件により自由に決定できるものとした。

本三角形粗分割アルゴリズムは、この n の数まで要素数を高速に増やすことが可能である。[1]では、新しい節点の属する三角形の情報を得るためにソートを必要としていたが、本アルゴリズムでは、三角形の辺の長さ

から新しい三角形を作るためソートが不必要という特徴がある。

ところがこのままでは[1][2][3]に比べて質の悪い三角形(つまり鈍角三角形や極端に鋭い鋭角三角形)ができる可能性がある。そこで三角形の補正のため、ラプリアン法[1]とローソンのスワッピング・アルゴリズム[1]を採用し三角形の質を維持する。ローソンのスワッピング・アルゴリズムにはソートが必要である[1]が、本方式の逐次粗分割部では三角形数 n が小さいので、 $n \times n$ の配列を使うことによりソートを回避し、計算量をさらに削減している。

次に(B)のPEへの割り当てでは各粗分割三角形の要素分割後にできる節点数(三角形数)を計算し、節点数が均等になるようにPEに割り当てる[3]。

(C)の並列要素分割部では、通信をせずにPE間の境界部分の節点の整合をとるため、[2]のラベルのアイデアを採用する。このラベル付けは(A)の後に行なっておく。

このラベルは各節点に与えられる。ラベルの大きさで各節点の属する辺をいくつに分割するか決める。このラベルと(A)の三角形粗分割アルゴリズムを使って並列要素分割を行なう。

並列要素分割(C)の方法は、(B)の段階で分割後の節点数が計算できることが必要である。節点数は、三角形の数の半分である(三角形の数が十分に大きいとき)ので、並列要素分割後の三角形の数を計算できれば良い。本稿では並列化要素分割に先の三角形粗分割アルゴリズムを用いる。この場合(α)ラベルと、(β)三角形粗分割アルゴリズムの新しい三角形を作り出す過程の性質を用いて、並列要素分割後の三角形の数を計算した¹。

```

入力図形を三角形で覆う;
while(PEへの割り当てに必要な
      三角形数になるまで) :A
{
    三角形粗分割アルゴリズム(逐次);
    ラプリアン法;
    スワッピング アルゴリズム;
}
ラベル付け;
節点数の計算とPEへの割り当て; :B
並列要素分割; :C

```

図1: 要素分割部の流れ

¹粗分割三角形の各辺の長さが、ラベルの示す値の長さに対して平均 α 倍であれば、各辺を α 個の短い辺になると考える。このとき α^2 を並列要素分割後の三角形の数として計算する。

4 三角形粗分割アルゴリズム

計算量が $O(n)$ の三角形粗分割アルゴリズム(ただし n を三角形の数とする)を説明する。図形はあらかじめその図形の各頂点を適当に結んだ三角形で覆われているとする。これらの三角形の中で一番長い辺の半分の長さを<基準>とする。すべての辺の中で<基準>より長い辺は半分のところに

新しい点を打つ。この点を使って新しい

三角形を作る。作り

方は、図2の4種類

がある。

本アルゴリズムは三

角形を作るときに先

述のようにソートを

必要としないため、

計算量は $O(n)$ とな

る。また一回の操作

で三角形の数が最大

で4倍に増えていく

性質がある。

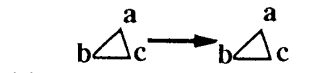
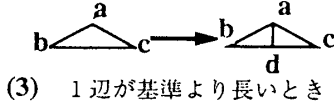
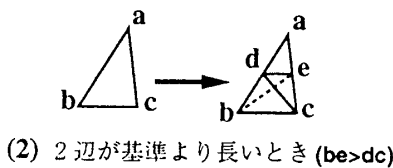
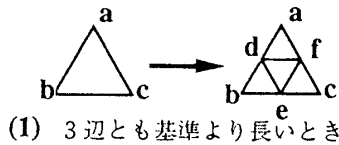


図2: 三角形粗分割アルゴリズム

5 評価

使用した富士通のAP1000は 8×8 の64台構成であり、ピーク性能5.56MFLOPS/cellである。評価の例として、部分ごとに分割の密度をあらかじめ指定した2次元の図3を要素分割する(図3はPEへの割り付けを示し、中心と左下を細かく分割するように指定してある)。図3を要素分割する際、粗分割三角形の数 n が大きい場合(例1: $n=4204$)と、その n が小さい場合(例2: $n=1074$)の結果を表1に示す。例1例2とも、要素分割結果はほぼ同じとなった。

つぎに複数の図形を要素分割し、粗分割三角形数と節点数の偏りの関係を図4に示す(ただしPE=64台、総要素数約1万個のとき)。図4よりPE間の節点数の偏りを抑えるためには、粗分割の三角形数 n は大きい方が良いことが分かる。しかし一回の逐次粗分割につき n は4倍ごとに増えるため、逐次の処理時間も一回の粗分割につき4倍ごとに増える(表1)。

64台のPEを使用して要素分割をするとき全体の処理時間を短くするには、できるだけ粗分割数 n を小さくすれば良い。しかし n が小さいと節点数の偏りは大きくなり、この後につながる有限要素法の並列化ソルバ部の処理時間の増大につながる。

したがって図4のような粗分割数 n と節点数の偏りの関係をもとに n を決定することとする。図4から、節点数1万点、PEは64台の場合、節点数の偏りを約1.2倍にするには n を1000個にすると良いことが分かる。

表1: AP1000での評価結果(64台、-O4 option)

(粗分割数)	time[s]	並列化率	偏り	要素数/節点数
例1(4204)	7.05	12.37	1.10	12854/6600
例2(1074)	1.17	7.08	1.33	13186/6768

ただし偏りとは節点数の偏りのことである。以下で節点数の偏りと並列化率を定義する。

節点数の偏り = (節点数の最大値/各PEの平均の節点数)

並列化率 = (1台での実行時間/64台での実行時間)

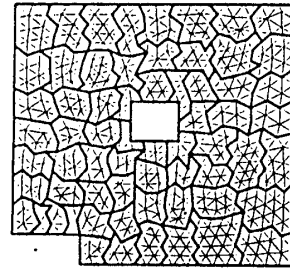


図3: PEへの割り付け(例2)

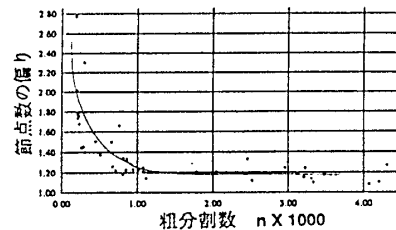


図4: 粗分割数 n と節点数の偏りの関係

6 おわりに

有限要素法の要素分割から解析までを実行するトータルシステムを分散メモリ型並列計算機を使って高速化させるための一段階として、三角形粗分割アルゴリズムを提案し、要素分割部の高速化を行なった。AP1000で評価し、PE64台のとき並列化率は平均10倍、PE間の節点数の偏りは平均1.3倍の結果が得られた。

本並列要素分割部は、粗分割数 n が小さいとき実行時間が最小となる。しかしこの場合各PEに割り当てられる節点数に偏りが生じ、並列化ソルバ部を含めた実行時間に影響が生じる。したがって n はシステム全体の求める負荷分散を考慮して決定されなければならない。本稿では節点数の偏りと n の関係を示した。

本要素分割部は、有限要素法の領域分割による並列化を対象としたソルバ部に接続することが可能であり、本研究室で開発中のシステムとして組み込む予定である。

謝辞

本研究に当たり、AP1000を使わせていただいたことを(株)富士通および並列処理センターに深く感謝いたします。

参考文献

- [1] 谷口健男: FEMのための要素自動分割 デローニー三角形分割の利用, 森北出版株式会社 (1992).
- [2] Fuhua Cheng, Jerzy W. Jaromczik, Junnin-Ren Lin, Shyue-Shian Chang and Jei-Yeou Lu: A Parallel Mesh Generation Algorithm Based on The Vertex Label Assignment Scheme, *International Journal for Numerical Methods in Engineering*, vol 28, pp.1429-1448 (1989).
- [3] I.St.Doltsinis and S.Noelting: Studies on parallel processing for coupled field problems, *Computer Methods in Applied Mechanics and Engineering* 89, pp. 497-521 (1991).