ハイパフォーマンスコンピューティング 60－8
（1996. 3. 4）

有限要素法要素分割の並列化 –ボロノイ分割の並列化–

河野 洋一　西松 研　福盛 秀雄　村岡 洋一

早稲田大学

有限要素法の要素分割には、有限要素法の解に信頼性が増すため、ボロノイ分割と双対な delaunay 三角形 (delaunay 四面体) が非常に有効である。しかしボロノイ分割の処理は非常に重いため、ボロノイ分割の高速化が望まれている。そこで、本論文ではボロノイ分割の高速化のため、並列計算機を用いたアルゴリズムを提案する。さらに、要素数の多い時に有効な要素分割の並列化手法の提案を行なう。本手法を富士通の分散メモリ型並列計算機 AP1000 に実装した結果、母点数 6000 個の 2 次元ボロノイ分割を、PE 数 32 台で約 11 倍の高速化を達成することができた。

Parallel Mesh Generation of FEM –Parallel constructing of Voronoi Diagram–

Yoichi KONO　Ken NISHIMATSU　Hideo FUKUMORI　Yoichi MURAOKA

School of Science and Engineering , WASWDA University
3-4-1 Ohkubo Shinjuku-ku , Tokyo 169 , JAPAN

This paper presents two parallel algorithms for constructing Voronoi Diagram and Delaunay triangulation. One algorithm using both a incremental method for inside PEs, and a divide-and-conquer method for merging Voronoi Diagrams between neighboring PEs is proposed. And the other algorithm for many generators which needs only one step of communication for merging Voronoi Diagrams is proposed. With 32 PEs configuration, the second method achieved speed-up ratio of 11 for 6000 generators.

1　はじめに

FEM(有限要素法) において、要素分割の結果は、FEM の解の信頼性 (誤差が少ない) や解の収束性に大きな役割を持っている。そのため従来から、幾何学的に特殊な性質を持つボロノイ分割 (もしくはボロノイ分割と双対な delaunay 分割) を用いた要素分割が提案されている [1][4]。ボロノイ分割の性質として、双対である delaunay 分割 (2 次元であれば三角形、3 次元であれば 4 面体) はその外接円内には他のいずれの節点も含まれない、という特徴がある。この性質は FEM で要求されている要素分割の性質に一致するため、ボロノイ分割は FEM の要素分割に非常に有用である。

従来から、ボロノイ分割の手法は盛んに研究されており逐次添加法 [2] [3] [4] や再帰二分法 [2] などの手法が開発されている。なかでも、逐次添加法は平均的な処理速度が $O(n)$ であり、非常に有効な手法である。しかし、例えば 10000 点を越えるような節点からなる大規模なボロノイ分割を行なう場合には、非常に時間がかかり、また大規模な記憶領域を必要とする。

そこで本論文では、並列にボロノイ分割を行なうことによって、ボロノイ分割を高速に実行する手法を提案する。さらに節点数が非常に多い場合には、FEM 要素分割として、**「要素分割を行なうためには、すべての PE 間の完全なボロノイ分割を行なう必要はない」**ということに着目して、さらに高速に要素分割する方法も提案する。

2　ボロノイ分割

平面上に n 個の点 $P_i(x_i, y_i)$ $(i = 1, \cdots, n)$ が与えられたとき、点 P_i の "勢力圏" $V_n(P_i)$ を

$$V_n(P_i) = \bigcap_{j \neq i} \{P | d(P, P_i) < d(P, P_j)\}$$

($d(P, P_i)$ は点 P と点 P_i の $Euclid$ 距離とする) で定義し、これを点 P_i に対するボロノイ多角形という。$V_n(P_i)$ による平面の分割を、ボロノイ分割 (ボロノイ図) と呼ぶ [2]。(以下、P_i を母点と呼ぶ。)

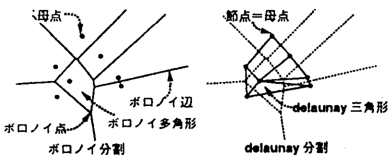

図 1: ボロノイ分割と delaunay 分割

ボロノイ図において点 P_i と点 P_j のそれぞれに対応するボロノイ多角形が共有辺を持つ時、点 P_i と点 P_j とを線分で結ぶことによって、$P_i(i = 1, \cdots, n)$ の凸包の三角形分割が得られる。これはボロノイ図を平面図と見た時の双対グラフであり、delaunay 三角形分割と呼ばれる。この時、各母点が三角形の節点となる。3 次元ボロノイ図と delaunay 四面体も同じように定義できる。

ボロノイ図を効率的に作成する方法は大別して、逐次添加法と再帰二分法の 2 つに分けられる。

2.1 逐次添加法

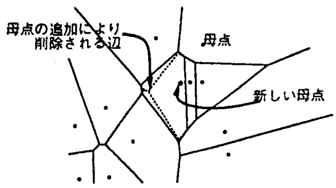

図 2: 逐次添加法

逐次添加法 [2] [3] [4] では、(1) 新たに追加される母点の含まれるボロノイ多角形を探し、(2) その多角形の周囲の母点との垂直 2 等分線を引き、新たなボロノイ多角形を作成する、という方法で、ボロノイ図形を作成していく。

この逐次添加法の計算量は最悪 $O(n^2)$ であるが、四分木を用いて新たに追加する母点の順番を工夫することによって、平均の計算量は $O(n)$ とすることができ、最も効率の良いボロノイ分割法として知られている。

2.2 再帰二分法

再帰二分法 [2] では、n 個の母点を x 座標に従って左右にほぼ $n/2$ 個ずつに分け、それぞれ別にボロノイ図を作ってから、それらを併合して全体のボロノイ図を作ることを再帰的に繰り返していく。

一般に、この再帰二分法は逐次添加法に比べて効率が悪いが、計算量は最悪・平均とも $O(nlogn)$ でボロノイ分割を作成することができる。

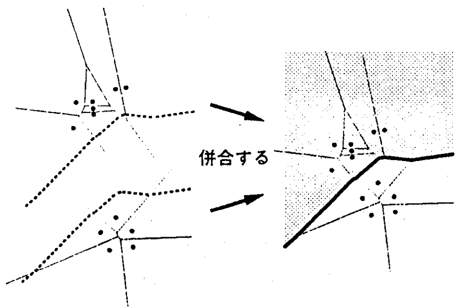

図 3: 再帰二分法

2.3 従来のボロノイ分割の並列化

ボロノイ分割の並列化に関する研究は、再帰二分法を用いた David J. EVANS[5] などがある。しかしそれらは母点の数 n に対して、n 台の PE を用いた場合の理論的研究であり、実際の並列計算機に実装した研究はない。

3 本論文で提案する手法

本論文では、分散メモリ型並列計算機を使って、ボロノイ分割を用いた要素分割を高速化する手法を提案する。

■並列化ボロノイ分割

本手法では、はじめに入力された節点を x 軸でソートし、各 PE に均等に割り当てる。各 PE は逐次添加法を用いて担当節点からボロノイ分割を行なう。この処理は完全に並列に実行できる。次に再帰二分法を用いて、PE 間のボロノイ図を併合する。この時必要な通信回数は $O(\log PE)$ 回である。

■改良した PE 間ボロノイ分割

さらに本論文では、節点数が多い場合、「要素分割を行なうためには、すべての PE 間の完全なボロノイ分割を行なう必要はない」という性質を利用して、各 PE は隣接する PE 間のみの併合を行なう手法を提案する。この時必要な通信回数は 1 回であるため、さらに高速に要素分割を行なうことができる。

ボロノイ分割終了後、ボロノイ分割からその双対である delaunay 分割を行なう。

以下並列化ボロノイ分割、および改良した PE 間ボロノイ分割について述べる。

4 並列化ボロノイ分割

始めに入力された母点(節点)を各PEに割り当てる。次に逐次添加法を用いて、PE内のボロノイ分割を行ない、その後再帰二分法を用いてPE間のボロノイ分割を行なう。

4.1 各PEへの母点の割り当て

本手法では、図4のように、x座標の値で母点をソートして、各PEの担当する母点数が均等になるように各PEに割り当てる。これにより、ボロノイ分割を行なう際に必要な通信は、両側の隣接する2台のPE間のみとなる。

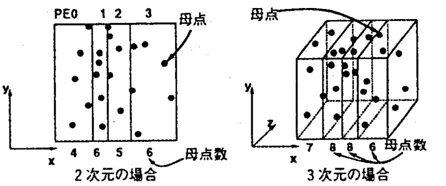

図 4: 各PEへの母点の割り当て

4.2 PE間のボロノイ分割：再帰二分法

再帰二分法の計算量は最悪・平均とも $O(nlogn)$ であり、一般に逐次添加法に比べて効率が悪いことが知られている。しかし、2個のボロノイ図を併合していくというアルゴリズムは、PE間の境界のボロノイ図を作成するのに有効であるため、図5のように、再帰二分法を使用する。各PEがボロノイ分割を次々と併合していき、$\log PE$回の併合後、全体のボロノイ図が完成する。

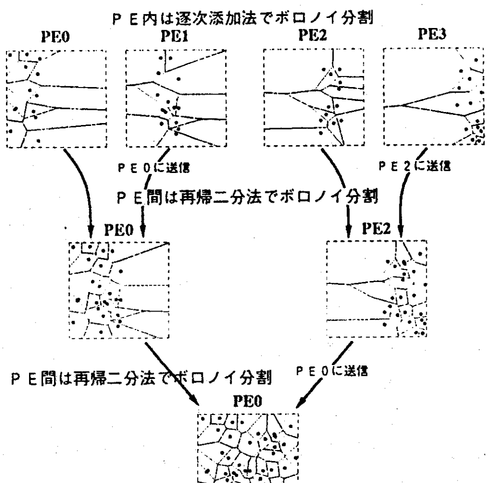

図 5: PE間のボロノイ分割：並列化再帰二分法

ボロノイ分割終了後、PE0はdelaunay分割を行なう。図5のボロノイ分割をdelaunay三角形分割する例を図6に示す。

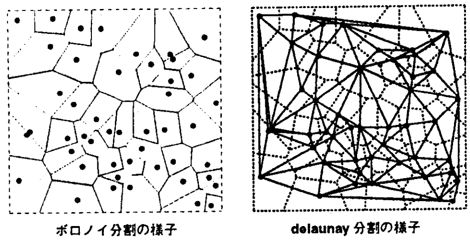

図 6: ボロノイ分割とdelaunay三角形の例

5 改良したPE間ボロノイ分割

さらに通信回数を削減する手法を提案する。

図6を見ると、delaunay分割を構成する三角形のうち、領域の内部の大半の三角形は、近傍の母点(節点)を三角形の頂点として持っている。この三角形を作成するには、近傍のボロノイ図のみで十分ある。つまり、FEMに必要なdelaunay分割をする際には、必ずしも完全なボロノイ分割は必要ない。そこで本論文では、隣接するPE間で1回だけ併合を行なう手法を提案する(図7)。

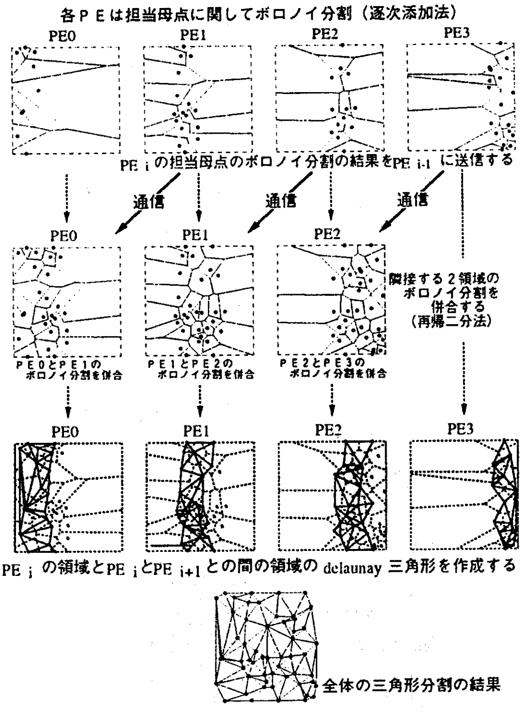

図 7: 改良したPE間ボロノイ分割の様子:PE間での併合の様子

第4章の「並列化ボロノイ分割」と同じく、各PEに母点を割り当て、PEは担当領域を逐次添加法によりボロノイ分割を行なう。逐次添加法の終了後、PE_i は PE_{i-1} にボロノイ分割の結果を送信する。PE_j は PE_{j+1} の隣接するPEからボロノイ分割の結果を受信し、PE_j は PE_j と PE_{j+1} の担当領域のボロノイ分割の結果を再帰二分法を使って併合する。併合終了後 PE_j は PE_j の領域と PE_j と PE_{j+1} の間の領域 のdelaunay分割を作成する。本手法は通信回数が1回と少ないため、並列化率が高いと考えられる。

また、本手法では、間違ったボロノイ分割を作成しないように、始めに各PEの領域の上部と下部に母点を打つ。仮に上部と下部に点を打たない場合の例を示す。図8の右側は、PE1とPE2の領域が重なりあってしまっている。これに対し、各PEの領域の上下に母点をあらかじめ設定しておくと、図8の左側のように正しい分割となる。

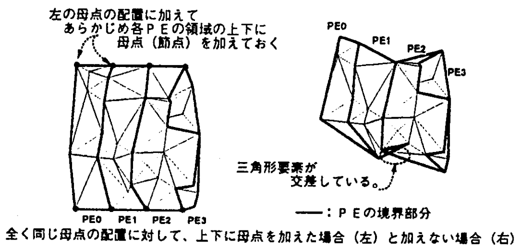

図8: 各PEの上下にあらかじめ母点を打つ理由

6 三角形(四面体)要素の交換

ソルバの負荷分散のため各PEの節点数を均等にする必要があり、かつソルバの通信量を減らすため、PE間の境界節点数を削減する必要がある。

そのため、本手法では、ボロノイ図から三角形(四面体)を作成した後、PEの担当する領域が四角形(立方体)になるように、PE間で三角形(四角形)を交換する(図9)。

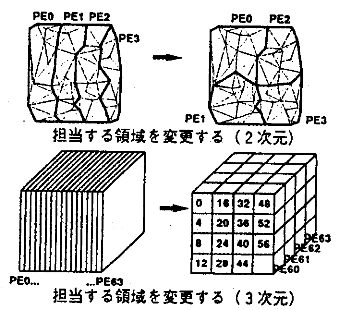

図9: 担当する三角形要素・四面体要素の変更

7 評価

本手法を富士通の分散メモリ型並列計算機AP1000に実装した。入力するデータは、2次元、3次元の空間上にある $0 \leq x, y, z \leq 1$ の一様実数乱数とする。それらの母点に対するボロノイ分割の並列化の実行時間、スピードアップを示す。並列化の手法の比較として、次の3種類を評価した。

■ **[並列化1]：PE内、PE間ともに再帰二分法を使って単純に並列化した場合** (実行時間を図10図12に、逐次処理の再帰二分法に対するスピードアップを図11図13に示す)

■ **[並列化2]：PE内に逐次添加法を使い、PE間に再帰二分法を用いて並列化した場合** (実行時間を図14図16に、逐次処理の逐次添加法に対するスピードアップを図15図17に示す)

■ **[改良並列化]：改良PE間ボロノイ分割(PE内は逐次添加法で、隣接PE間は再帰二分法で1回だけ併合)の場合** (実行時間を図18図20に、逐次処理の逐次添加法に対するスピードアップを図19図21に示す)

表1: 2次元の場合の各手法の実行時間

PE数	並列化1[sec]	並列化2[sec]	改良並列化[sec]
1	26.587	12.157	12.157
2	15.280	8.508	8.613
4	9.351	6.158	4.711
8	6.473	5.015	2.677
16	5.108	4.484	1.628
32	4.532	4.313	1.078
64	4.312	4.350	—

(母点数n=6000点)

表2: 3次元の場合の各手法の実行時間

PE数	並列化1[sec]	並列化2[sec]	改良並列化[sec]
1	418.260	142.486	142.486
2	225.801	109.455	109.455
3	—	—	81.787
4	143.623	97.974	—
8	112.785	100.782	—
16	102.552	—	—
32	99.319	—	—
64	98.465	—	—

(母点数n=1500点)

[並列化1] と、本論文で提案した **[並列化2]** を比較すると、**[並列化2]** の方が少ないPEで高い並列化を達成できている。しかし、節点数n=6000点程度では、**[並列化2]** はPE数が多くなるにつれて、スピードアップが飽和し、逆に遅くなってしまう。**[並列化1]** より遅くなってしまう原因は、**[並列化2]** が逐次添加法から再帰二分法へとデータを変換しているためである。

次に **[並列化2]** と、本論文で提案した **[改良並列化]** を比較する。**[改良並列化]** は、**[並列化1]** **[並**

列化 2] とは異なり、ほとんど全ての PE が稼働しているため、非常に高い並列化が現れている。しかし、評価した入力データの条件では、PE 数が増えると、担当する領域が狭くなり 2 次元では PE 数 32 台を越えると正しい結果が得られなくなる。そこで、入力節点数を 40000 点まで上げたところ、PE 数 64 台で、実行時間 2.83 秒、スピードアップ 35.2 倍の高い並列化が達成できた。3 次元でも同様の理由で、[改良並列化] は評価データでは PE 数 4 台以上でボロノイ分割を行なうことはできなかった。

しかし表 1 表 2 から、2 次元 3 次元のいずれの場合にも、[改良並列化] の手法が、もっとも効率の良い手法であることわかる。

8 まとめ

本論文では、ボロノイ分割を用いた FEM 要素分割の並列化方法を提案した。入力する節点数が多い場合には、PE 間で 1 回だけ併合する手法が有効であり、2 次元の場合には 32 台で 11 倍の高速化が達成できた。現在は、要素数を多くすると数値誤差の影響で再帰二分法による併合が正しく行なわれないことがあるため、今後この問題点を解決する予定である。

謝辞

本研究に当たり、AP1000 を使わせて頂いたことを (株) 富士通および並列処理センターに感謝致します。

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] 伊理 (監):計算幾何学と地図情報処理,bit 別冊, 共立出版 (1986).

[3] 杉原厚吉 "幾何アルゴリズムの数値的破綻とその対策" 応用数理,Vol.1,No.4,pp.280-299(1991).

[4] 稲垣宏, 杉原厚吉：退化を許す 3 次元 Delaunay 図構成算法, 情報処理学会研究会報告グラフィックスと CAD 67-2,pp.9-16(1994/2)

[5] David J.EVANS : On parallel computation of Voronoi Diagrams, Parallel Computing 12,pp.121-125(1989)

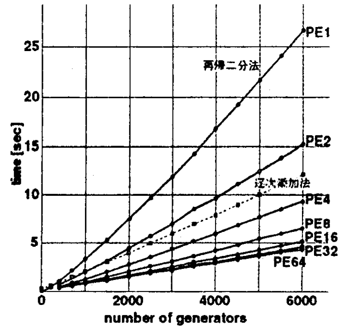

図 10: 2 次元:PE 間 PE 内ともに再帰二分法を使って並列化した場合の実行時間

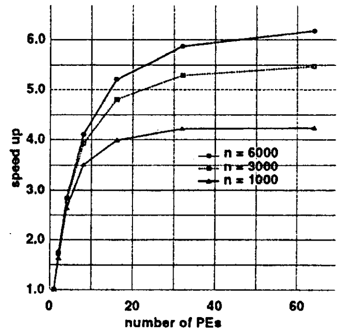

図 11: 2 次元:PE 間 PE 内ともに再帰二分法を使って並列化した場合のスピードアップ

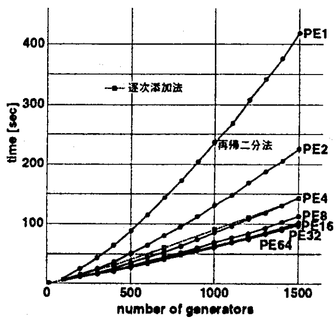

図 12: 3 次元:PE 間 PE 内ともに再帰二分法を使って並列化した場合の実行時間

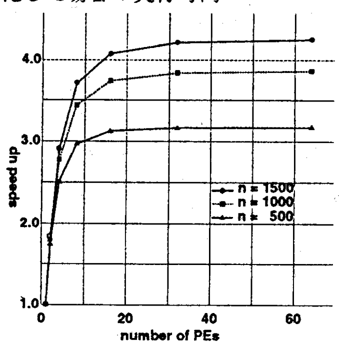

図 13: 3 次元:PE 間 PE 内ともに再帰二分法を使って並列化した場合のスピードアップ

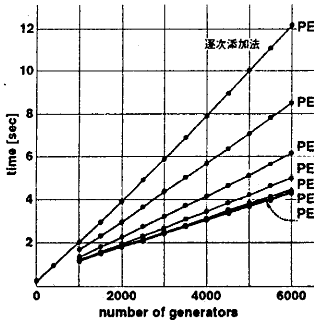

図 14: 2 次元:PE 内で逐次添加法、PE 間で再帰二分法を使って並列化した場合の実行時間

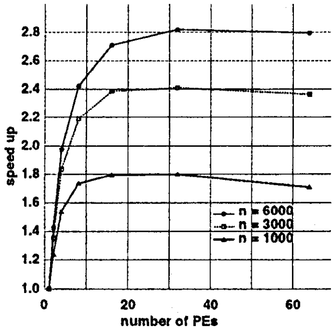

図 15: 2 次元:PE 内で逐次添加法、PE 間で再帰二分法を使って並列化した場合のスピードアップ

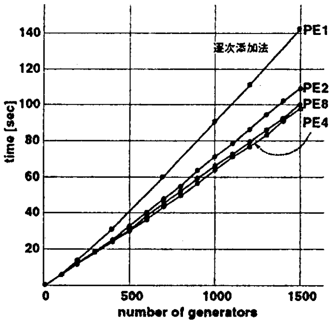

図 16: 3 次元:PE 内で逐次添加法、PE 間で再帰二分法を使って並列化した場合の実行時間

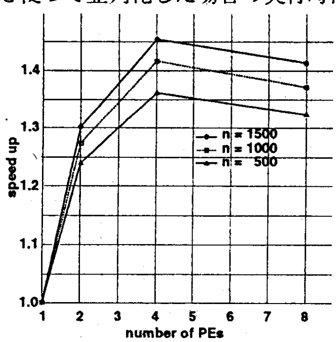

図 17: 3 次元:PE 内で逐次添加法、PE 間で再帰二分法を使って並列化した場合のスピードアップ

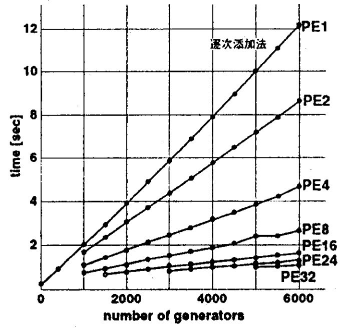

図 18: 2 次元:改良 PE 間ボロノイ分割の実行時間

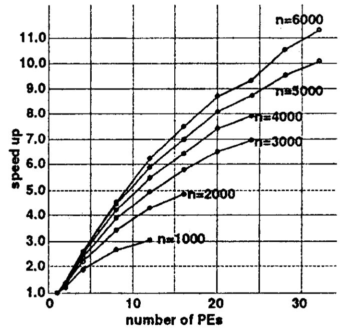

図 19: 2 次元:改良 PE 間ボロノイ分割のスピードアップ

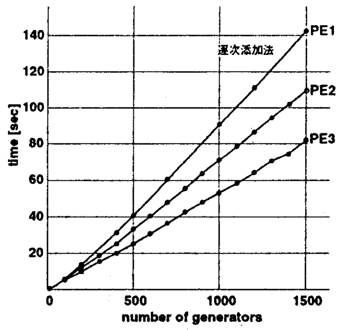

図 20: 3 次元:改良 PE 間ボロノイ分割の実行時間

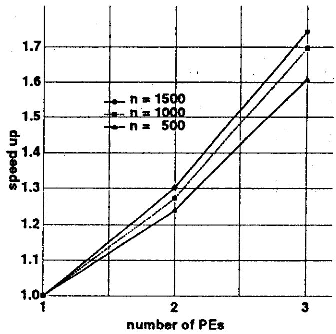

図 21: 3 次元:改良 PE 間ボロノイ分割のスピードアップ