

有限要素モデルのアレイ型マルチプロセッサへのマッピング手法

岡部 公治 , 浅原 重夫 , 鷺島敬之
松下電器産業株式会社 無線研究所

本報告は、有限要素モデルのアレイ型マルチプロセッサへのマッピング手法について、その概要を述べ、その効果について述べる。この手法は、有限要素モデルのアレイ型マルチプロセッサへのマッピングを行う際に、ノード接続行列の帯幅を小さくし、帯幅の小さい行列を優先的に処理することによって、帯幅の大きい行列の処理時間を短縮し、全体の処理時間を短縮する。また、帯幅の小さい行列を優先的に処理することによって、帯幅の大きい行列の処理時間を短縮し、全体の処理時間を短縮する。また、帯幅の小さい行列を優先的に処理することによって、帯幅の大きい行列の処理時間を短縮し、全体の処理時間を短縮する。

An algorithm for the mapping of the finite element models to array processors

Koji Okabe , Shigeo Asahara , Takayuki Sagishima
Matsushita Electric Industrial Co., Ltd.
Kadoma 1006, Kadoma-shi, Osaka, 571 Japan

An efficient algorithm for the mapping of the finite element models to array processors is described. The algorithm gets an initial mapping by the band width reduction of node connection information matrix for the finite element model, and then repeats the refinement of the mapping based on several estimation functions for the interprocessor data transfer time.

In the case that node connection information matrix has a homogeneous band after the band width reduction, we show that the optimum solution is obtained for 1-dimensional array processors and that each estimation function is a good approximation of the interprocessor data transfer time for 2-dimensional and 3-dimensional array processors. Finally, from the simulation results of mappings for several finite element models onto 1-, 2- and 3-dimensional array processors, this algorithm was proven to be efficient enough.

1 まえがき

有限要素法は構造解析、熱伝導解析などの技術計算で広く用いられており、数千～数万節点の大規模計算は、スーパーコンピュータなどで処理されることが多いが、この場合、十分なベクトル性能が発揮されない。このため最近では、マルチプロセッサによる並列処理が試みられるようになってきている。⁽¹⁾

多くの疎結合マルチプロセッサは不完全結合である。これは完全結合の場合、プロセッサ間の接続数と各プロセッサの I-Oポート数がそれぞれプロセッサ数の2乗、1乗に比例して増大し、実現が困難となるためである。したがって本文では、不完全結合のマルチプロセッサを対象とする。

有限要素問題は、互いに通信を行いつつ並列に解かれる複数のモジュールに分解できる。このようなモジュールを不完全結合マルチプロセッサにマッピングする時、互いに通信の必要なモジュールの対をできるだけ直接結合するプロセッサにマッピングすることが、通信のオーバーヘッドを減らし、並列処理効率を上げるうえで重要である。⁽²⁾

特に、負荷均等となるようなモジュール分割を求める問題は、物理領域を格子状あるいは有限要素に区切って問題を解く偏微分方程式などであらわれるが⁽³⁾、領域を2進分割する方法が2進木型マルチプロセッサ、アレイ型マルチプロセッサ⁽⁴⁾、⁽⁵⁾、ハイパーキューブ型マルチプロセッサ⁽⁵⁾に対して考察されている。

しかしながら、従来の方法では節点の接続関係を考慮していないため、十分な処理効率を得られていない。

本文では、プロセッサの負荷を均等にしつつ、通信が最小になるように、有限要素モデルをアレイ型マルチプロセッサにマッピングする問題を取り扱う。

まず2章で、本文で扱う有限要素モデルと対象とするアレイ型マルチプロセッサについて述べ、マッピング問題について定式化を行う。3章では、節点接続行列を基にしてバンド幅縮小化を行った後、いくつかの評価関数のもとで反復改良を行うマッピングアルゴリズムを提案する。次に4章で、バンド内の非ゼロ確率が一定という仮定のもとでの本アルゴリズムの有効性を示す。さらに5章で、接続次元が1, 2, 3のアレイ型マルチプロセッサに対して本アルゴリズムを適用した場合のシミュレーション結果を示し、その性能評価を通じて本アルゴリズムの有用性を明らかにする。6章では本報告のまとめを行い、今後の課題を提示する。

2 マッピング問題

2.1 有限要素モデル

有限要素法は解析対象となる物体(領域)を図1のように有限個(M)の要素に分割し、偏微分方程式を離散化して物体を有限個のパラメータに代表させて解析する方法である。

各要素は通常、三角形又は四角形などで形成されるが、その頂点は節点(N:節点数)と呼ばれる。(図1の頂点1~14)又、同一要素を形成する節点(例えば1と5, 2と6など)間の直線は、枝と呼ばれ、これらの

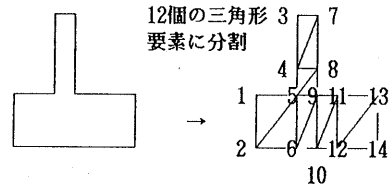


図1 領域の分割

節点は、接続されていると言う。

本文で取り扱うモデルは、上述のように他のいくつかの節点と接続するN個の節点から構成される有限要素モデルである。

ここで、節点接続行列 $M(m_{ij})$ を次のように定義する。

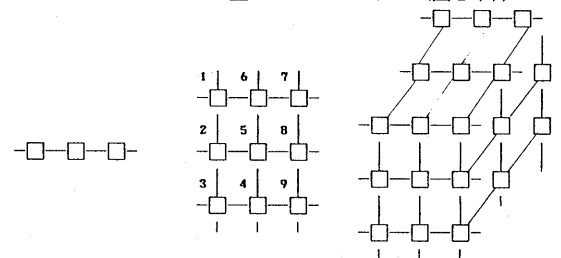
$$m_{ij} = \begin{cases} 1 & \text{節点 } i, j \text{ が接続時 } (1 \leq i, j \leq N) \\ 0 & \text{それ以外の場合} \end{cases}$$

2.2 アレイ型マルチプロセッサ

大規模科学技術計算の必要性などから、従来より種々のアーキテクチャのマルチプロセッサが考案されており、最近ではマイクロコンピュータ技術の発達などにより、実用化も進んでいる。(例えば⁽⁶⁾)

その中でも、各プロセッサエレメント(以下PEと省略する)が隣接する上下左右の4方向のPEと接続する2次元アレイ型マルチプロセッサ(図2(b))では行列演算や動的計画法を並列処理するアルゴリズムが提案されており⁽⁷⁾、ILLIAC IV, PAX 等で実現されている。⁽⁸⁾

有限要素モデルの規模・形状によっては、前後の接続も加味した3次元アレイ接続(図2(c))や左右方向の接続のみの1次元アレイ型マルチプロセッサ(図2(a))



(a) 1次元アレイ (b) 2次元アレイ (c) 3次元アレイ

図2 アレイ型マルチプロセッサ

の効率がよい場合もあると考えられるから、本文では、図2に示される3種類(1~3次元)の隣接結合されたP個のPEから構成されるアレイ型マルチプロセッサを対象とする。

アレイ形マルチプロセッサに対して下記の2つの行列を定義する。

$$\begin{aligned} & \cdot D(d_{ij}) \text{ PE間距離行列} & 1 \leq i, j \leq P \\ & & d_{ij}: \text{ PE } i, j \text{ 間の距離} \\ & \cdot E(e_{ij}) \text{ PE間接続行列} & 1 \leq i, j \leq P \\ & & e_{ij} = 1 \quad \text{PE } i, j \text{ が接続時} \\ & & = 0 \quad \text{それ以外の時} \end{aligned}$$

ここで、PE間の距離とは、2つのPE間でデータ通信を最短経路で行う時に通過する枝の数のことである。

2.3 マッピング問題

有限要素モデルの解析をアレイ型マルチプロセッサ上で実行する際、各PEの計算負荷を均等にするため、N個の節点はP個のPEに均等(N/P個)に割り当てられ、独立に処理されるものとする。但し、接続節点間は、データのやりとりが必要であり、互いに接続する節点が異なるPEに割り当てられた場合、PE間通信が必要となる。

上記の操作は、有限要素法で各節点の特性値を反復法等で求める際に必要となる。

データ通信に関しては、以下を仮定する。

- 各PEは接続するすべてのPE間との送受信を同時に行う。
- ルーチングは静的である。(例えば2次元接続の場合水平方向の転送がすべて終わってから垂直方向の転送を行う。)
- あるPEからあるPEに送られるデータ群はパケットと呼ばれ、通信はパケット単位で行われる。パケットの大きさをデータ量というが、これはPE間の節点接続数に等しい。すなわち、すべての接続する2節点間のデータ転送量は等しい。
- 1つのデータが隣接PE間を移動する時間はどこも等しく、パケットの移動時間はそのデータ量と移動距離に比例する。

マルチプロセッサ利用の目的は、負荷分散による処理の高速化である。処理時間Tは

$$T = \text{計算時間}(T_i) + \text{通信時間}(T_c)$$

であるが、ここでは均等(PE当たりN/P)なマッピングを仮定しているため、 T_i は一定である。したがってマッピング問題は、

$$T_c \rightarrow \text{最小}$$

となるようなマッピングを求める問題となる。

3 マッピングアルゴリズム

PEの番号が図2(b)のように付けられていると、Eはバンド行列であり、行列Mが行列Eと似るように節点の再番号付けを行う操作は、Mのバンド幅縮小と同様である。

一方、MがEと似ていると、多くの接続する節点を同一のPE又は直接結合するPEに割り当てることが可能となり、通信時間を小さくできる。

マッピング問題に対する多項式時間のアルゴリズム(polynomial order algorithm)が存在すれば、グラフの同形問題に対して多項式時間のアルゴリズムの存在することが既に示されている。⁽²⁾

グラフ同形問題は、現在のところ多くの研究努力にもかかわらず、多項式時間のアルゴリズムは見つかっていない。

従って、マッピング問題の最適解を得るのは容易でないと考えられるから、本文では次の2段階のアルゴリズム

で近似解を得る。

- ① 行列Mのバンド幅縮小化、及びグルーピングにより初期マッピングを得る
- ② 適当な評価関数に基づく反復改良を行う

行列Mのバンド幅縮小化及びグルーピングによる初期マッピング

バンド幅縮小化問題(行列Mの各列のバンド幅の最大値を最小にする問題)はNP完全であることがわかっている。⁽⁹⁾しかし、よい発見的手法があり^{(2), (4), (10)}ここではRCM法⁽¹⁰⁾を採用する。(付録参照)

次に、再順序付けされた節点をN/P個づつにグルーピング(簡単のためNはPの倍数と考える)する。初期マッピングはこのグループをその順にPEに割り当てる方法である。

評価関数に基づく反復改良

バンド幅縮小化により、特定の方向への転送量は減少できたが期待できるが、さらに節点のPEへ割り付けの交換により、他の方向への転送量の減少が可能となるであろう。

あるマッピングに対する実際の転送時間の算出は、膨大な計算時間を要するので、ここではいくつかの評価関数を導入し、その値が改善されるような交換を反復して行う。

評価関数の定義に先立ち、一つのマッピングに対して以下の量を定義する。

- $T(t_{ij})$ 転送量行列 $1 \leq i, j \leq P$
 t_{ij} : PE i, j間転送データ量
- $B(b_{ij})$ PE転送負荷行列 $1 \leq i, j \leq P$
 $b_{ij} = d_{ij} \times t_{ij}$
- $C(c_{ij})$ 転送路負荷行列 $1 \leq i, j \leq P$
 c_{ij} : PE i, j間通過データ量
- P_i : PE iからある1方向への重み付データ量
転送量 $P_i = \sum_{i=1}^P b_{ij} / 2d$

但し、 d は各PE結合の次元である。

本文では、下記(A)~(F)の評価関数を用いた。

- (A) $\max_i P_i$ PEの最大重み付データ量
- (B) $\sum_i P_i$ 総重み付データ量
- (C) $\max_{ij} b_{ij}$ PE間の最大負荷
- (D) $\max_{ij} c_{ij}$ PE間の最大通過データ量
- (E) $(\max_{ij} t_{ij}) \times (\max_{kl} d_{kl}) \times P$
($t_{kl} > 0$)

すべてのPEで最大データ量のパケットが最大距離を動いた場合の転送時間

- (F) 2次元の場合は線(3次元は面)を横切るデータ量の最大値(図3)

P_x の変化を ΔP_x とすると、その時の $\sum P_i$ の変化は $2 \Delta P_x$ である。

【証明】 省略

〔補題1〕

2次元あるいは3次元アレイにたいしては $\sum P_i$ は最小とならない。

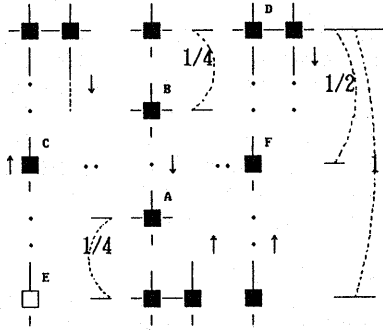
【証明】

2次元アレイの場合で反例を示せば十分である。図7において、PE Aからの転送がC~D、PE Bからの転送がE~FとするとPE AとBの割り当てを交換することにより、

交換後の重み付データ転送量を'を付けて示すと

$$Pa' - Pa = 2 \cdot \sum_{t=1}^{(p)^{1/2}/4} t - 2 \cdot \sum_{t=(p)^{1/2}/4}^{(p)^{1/2}/2} t$$

$$= -p/16$$



(p)^{1/2} = 1 としている
矢印はPEの番号順を示す

図 7 PE交換による転送量の減少

同様に、 $P_b' - P_b = -p/16$

$$\therefore Pa' + Pb' < Pa + Pb$$

これと命題1より $\sum P_i$ は増加する。 証明終

〔補題2〕

2次元アレイの場合、 $w=2\sqrt{m}$ (m は整数)が成立する時のみ、 $\sum P_i$ が最小となる。

【証明】 省略

〔定理2〕

評価関数 (B), (D) は実際の転送時間 T_c と (P)^D の最高次のオーダーで一致する。

【証明】

2次元で証明する。まず T_c を求めるが、各転送ステップでの転送量(時間)は表2のようになる。

ここで $k=w \cdot (p)^{-1/2}$, $q=(p)^{1/2}/2$ とおき $p/2 > w > 2q$ と仮定している。

$$T_c = \max(q, (k-1)q) + \max(2q-1, (k-2)q) + \dots +$$

$$\max((k-1)q + (k-1)(k-2)/2, q) + (kq - k(k-1)/2)$$

$$+ \dots + k(k+1)/2 + \dots + 1$$

$$= kq^2/2 - (k^2-k-1)q/2 + k(k-1)(k^2-k-3)/8$$

$$+ (2k-n)q/2 + (k^2-k-n^2+n)q/2$$

$$- k(k-1)(k^2-k-2)/8 + n(n-1)(n^2-n-2)/8$$

$$= kq^2/2 + (2k-2n+1)q/2 + 5k(k-1)/8 +$$

$$n(n-1)(n^2-n-2)/8$$

一方、評価関数 (A), (D) はそれぞれ、ある最大値を与えるような組 i, j に対して

$$P_i = \sum_j d_{i,j} \cdot t_{i,j}/4$$

$$= k(q^2 + q)/2 + k^2q/2$$

表 2 転送データ量の遷移

ステップ	1	2	k-1	k	k+1	q	q+1	q+k+1
縦方向 転送量	q	q-1	...	q-k+2	q-k+1	q-k	...	1			
		q	...	q-k+3	q-k+2	q-k+1	...	2	1	
					
				q	q-1	q-2		.	.		
					q	q-1		k	k-1		1
小計	q	2q-1	$kq - \frac{k(k-1)}{2}$	$\frac{k(k+1)}{2}$	1
T _{cv}											
横 T _{ch}	(k-1)q	(k-2)q	q	0					

(但し、n は $T_{cv}=T_{ch}$ となるステップ数)

$$= kq^2/2 + (k + k^2)q/2$$

$$c_{ij} = k(1 + \dots + q)$$

$$= kq^2/2 + kq/2$$

証明終

(補題3)

転送時間T cに対して評価関数(C) (D)は下限、(E)は上限を与える。

(証明) 省略

バンド幅小化を適用した理由は、3章で述べたとおりであるが、定理1でその正当性が保証された。(A), (C), (D) および(E)も定理2・補題3より、転送時間の評価関数として妥当であることがわかる。

5 シミュレーション結果

いくつかの有限要素モデルに対し、前章のアルゴリズムを用いてアレイ型マルチプロセッサにマッピングした場合の転送時間をシミュレートした。使用計算機は、Balance 8000で使用言語はFortran とC言語である。

ここで使用したモデルは、節点数が24~3280のものであり、図8に1例を示す。また対象のアレイ型マルチプロセッサは表3に示す6種類である。

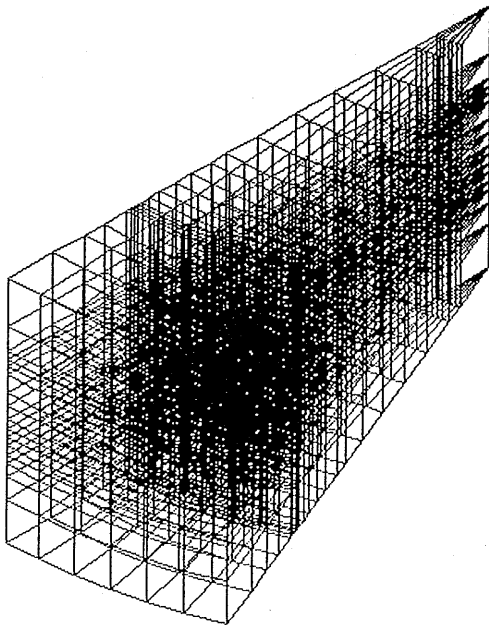


図 8 有限要素モデルのメッシュ分割図

表3 対象のアレイ型マルチプロセッサ

	接続次元	PE数
1	1	2 4
2	1	6 4
3	2	2 5
4	2	6 4
5	3	2 7
6	3	6 4

図9はバンド幅縮小化による通信時間減少の割合を示すもので、バンド幅縮小化をほとんど前の転送時間を1としている。これはモデルの形状や初期節点番号付け方法にも依存するが、節点数/PE数が大きいほどバンド幅縮小化の効果は大きい。なお、図9~図12は実際の測定データに対して適当な回帰曲線により補間を行っている。

図10は評価関数(A), (B), (D)及び(F)を利用した場合の通信時間減少の割合を示すものである。2次元と3次元接続における違いはほとんどみられなかった。ここでは両方の値を用いている。(C), (E)はそれぞれ転送時間の下限と上限をあたえるが、現実的でないものと考え対象から除いた。この図より、(D), (F)を評価関数として用いてもほとんど効果がなく(B)が最も適することがわかる。またその効果は節点数/PE数が小さいほどあらわれる。

図11は評価関数(B)を用いてマッピングを行った時の転送時間と節点数/PE数の関係を示す図である。転送時間の増加率は節点数/PE数が増加するほど小さくなる。

図9, 図11から、1, 2及び3次元アレイ間の転送時間の比率が読み取れる。1次元→2次元, 2次元→3次元の変更によって転送時間はそれぞれ50~70%, 60~80%に減少され、その差は節点数/PE数が小さいほど大きい。

3章でも述べたように、各節点に対する計算時間は一定と考えられるが、ここで1節点に対する値の更新に7回の演算を要し、(1データの隣接PE間通信時間) = (1データの1演算時間)と仮定した場合の計算時間(Ti)/通信時間(Tc)を示したのが図12である。プロセッサはPE数27の3次元アレイで評価関数(B)を用いたマッピングを行った時のものである。またTc/(Ti+Tc)は通信のオーバーヘッドを示す。

この仮定は必ずしも一般的ではないが、(1データの隣接PE間通信時間) = n * (1データの1演算時間)ならば縦軸をn倍することで一般化できる。

この図から、節点数の増加に伴い通信のオーバーヘッドが減少することがわかる。また、例えば節点数が1000の場合、Ti/Tc=2であるから、通信のオーバーヘッドを50%以下にしようと思えば、プロセッサ設計の際、(1データの隣接PE間通信時間) ≤ 2 * (1データの1演算時間)となるようにしなければならない。

今日の集積回路技術の進歩は目覚ましく、今後より小型化、高集積化したVLSIが安価に供給されると考えられるが、アレイ型マルチプロセッサを設計して有限要

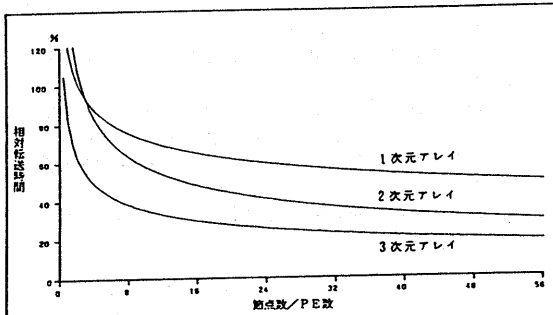


図9 バンド幅縮小による転送時間の変化

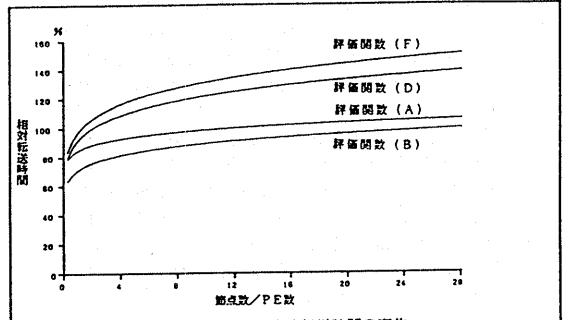


図10 評価関数利用による転送時間の変化

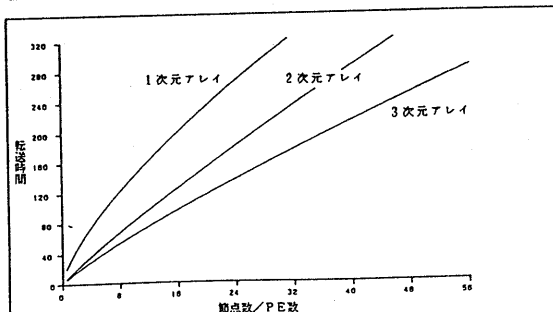


図11 節点数/PE数 - 転送時間

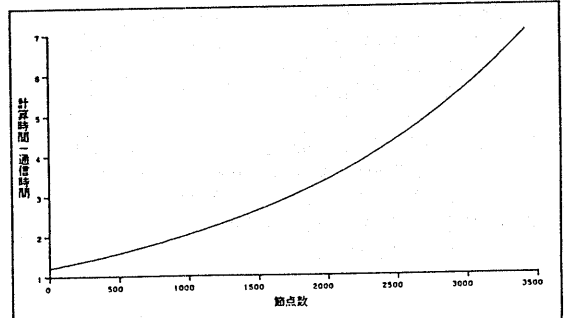


図12 計算時間/転送時間 - 節点数

素解析を行う際、本アルゴリズムと本シミュレーション結果は、一つの指標を与えていると思われる。

現在、集積回路技術の進歩により10MIPSのマイクロプロセッサ、16MFLOPSの浮動小数点演算チップが入手可能であるが、これらのチップを用いてPEを構成し、64個のPEを64MB/Sの高速通信路で3次元結合することにより、節点数2400程度の有限要素モデルに対して、50%の効率、実行速度500MFLOPSを実現できると考えられる。いかに多くのプロセッサをいかに高速の通信路により接続することができるかは、ハードウェアの実装技術にかかっていると考えられる。

6 むすび

有限要素モデル解析時の、アレイ型マルチプロセッサへのマッピングアルゴリズムを提案し、評価を行った。

その結果、

- ① バンド幅縮小化による再番号付けは、理論的にも実験においても効果が大きい。
- ② 評価関数としては、総転送量を最小にするようなマッピングがいちばんよい。
- ③ PE当たりの節点数が大きいほど通信のオーバーヘッドが小さい。
- ④ PEの接続次元は大きいほど転送に有利であるが、その効果は、節点数/PE数が小さい時ほど大きい、ことがわかった。

従って有限要素モデルの解析においては、バンド幅縮小化+総転送量最小化によるマッピングアルゴリズムが有効であり、PEの個数・接続次元は対象モデルの大きさ(N)と1データに対する演算と通信の時間比ならびに要求される通信オーバーヘッドの上限から決定すべきである。

シミュレーション結果によれば、現在のマイクロプロセッサを用いてアレイ型マルチプロセッサを構成し、スーパーコンピュータを上回る性能を出す可能性が示された。

しかしながら、バンド幅縮小化は基本的には1次元アレイに適したものであるから、n次元有限要素モデルに対してはn次元アレイに直接マッピングするようなアルゴリズムが本来望まれるところであり、今後の課題とするところである。

〔謝辞〕

シミュレーションに際して、貴重なデータを提供して頂いた、松下電器産業(株) 無線研究所 機構第4開発室の井ノ上裕人氏に感謝致します。

参考文献

- (1) 佐藤善行, 上村健, 星野力: 並列計算機PAXによる2次元弾性問題の有限要素解析, 情報処理学

- 会論文誌, Vol.26, No.4, pp.584-590 (1985)
- (2) Bokhari, S.H.: On the Mapping Problem, IEEE Trans. Computers C-30, pp.207-214 (1981)
 - (3) Berger, M. and Olinger, J.: Adaptive mesh refinement for hyperbolic partial differential equations, J. Comput. Phys., Vol.53 (1984)
 - (4) Berger, M.J. and Bokhari, S.H.: A Partitioning Strategy for PDEs across Multiprocessors, Proc. Int. Conf. Parallel Processing (1985)
 - (5) Berger, M. and Bokhari, S.: A Partitioning Strategy for Nonuniform Problems on Multiprocessors, IEEE Trans. Computers C-36, pp.570-580 (1987)
 - (6) 特集: 並列マシン, 情報処理学会誌, Vol.28, No.1 (1987)
 - (7) Hwang, K. and Briggs, F.A.: Computer Architecture and Parallel Processing, McGraw-Hill, pp.355-361, 769-774 (1985)
 - (8) 加藤清左夫, 苗村憲司: 並列計算機, オーム社, pp.149-151 (1976)
 - (9) Tarjan, R.E.: Graph theory and Gaussian elimination, Sparse Matrix Computations, Bunch and Rose, Eds., New York, pp.3-22 (1976)
 - (10) 村田健郎, 小国力, 唐木幸比古: スーパーコンピュータ, 丸善, pp.115-117 (1985)

付録

RCM法⁽¹⁰⁾

行列A (a_{ij})において, バンド幅 w は各列の枢軸位置 i と, その位置の最初の非ゼロ要素の位置 k との差 $w_i = i - k$ の最大値として定義できる.

$$w = \max_i w_i, \quad w_i = i - \min_{j; a_{ij} \neq 0} j$$

w_i の和 $P = \sum_i w_i$ をプロフィール(profile)と呼ぶ.

RCM法はバンド幅をするための Cuthill-McKee 法に George が改良を加えて, プロフィール $\sum_i w_i$ を少なくなるようにした方法である. 手順を説明する前にグラフと行列の関係の説明をする.

有限要素で分割したのち節点番号を付けた物理領域をグラフと呼ぶ. 特定の頂点 i から出発する枝の数を自由度または次数といい, ρ_i または $\text{Deg}(i)$ であらわす.

このとき, 与えられた行列Aに対するバンド幅縮小のための RCM法の手順はつぎのとおりである.

(手順1) 行列Aによって決まるグラフの各節点の次数 ρ_i を求める.

(手順2) $\rho_{i_1} = \min \rho_i$ となる i_1 を選び, この頂点 i_1 を根といい, 番号1をふる.

(手順3) この番号1の節点に接続する節点をレベル1の節点といい, その次数が増す順にレベル1の節点に接続して番号をふる.

(手順4) レベル1の各節点につながる節点をレベル2にあるという. レベル2の各節点について手順2~3を順に行う. ただし, レベル2の頂点につける新しい番号はレベル1の節点の最終番号の次から始まる番号

を付ける.

(手順5) 上の手続きをグラフのN個の節点すべてが番号付けられるまで, 各々の連続したレベルにある節点について行う.

(手順6) 新しくふられた番号 $i = 1, 2, \dots, N$ に対し, $N-i+1 \rightarrow i$ によって再び各節点の番号を付けかえる. この手順6がRCM法で Reverseという名が付いた理由である.

(手順7) 最後に再番号付けに応じて行列Aの各行と各列を交換する.