

有限要素モデルのアレイ型マルチプロセッサへの マッチング手法†

岡部 公治^{††} 浅原 重夫^{††} 鷲島 敬之^{††}

本文では有限要素モデルをアレイ型マルチプロセッサ上で効率よく処理するためのマッピングアルゴリズムを提案する。このアルゴリズムは、対象モデルの節点接続行列をバンド幅縮小化することにより初期マッピングを求め、通信時間に対する評価関数を利用して反復改良を行うものである。バンド幅縮小化の結果、節点接続行列が均質なバンド行列となる場合、1次元アレイに対して最適解が得られ、2次元、3次元アレイに対しては、評価関数が通信時間の良い近似となっていることを示す。さらに、幾つかの有限要素モデルを1次元、2次元、3次元アレイにマッピングするシミュレーションを行った結果、本アルゴリズムの実用性が示された。

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⁾。

さらに、各プロセッサの計算負荷が均等となるように、格子状あるいは有限要素に区切られた物理領域を、モジュール分割することも重要である³⁾。これに対しては、格子状に区切られた物理領域を、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⁶⁾ に対して考察されている。

しかしながら、従来の方法では計算負荷と通信オーバーヘッドが同時に考慮されておらず、かつ一般の有限要素モデルに対してはあまり研究が進んでいない。

本文では、プロセッサの負荷を均等にしつつ、通信量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。

まず第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。第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。次に第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。さらに第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。第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、三角形また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) また、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

† An Algorithm for the Mapping of the Finite Element Models to Array Processors by KOJI OKABE, SHIGEO ASAHARA and TAKAYUKI SAGISHIMA (Information Systems Research Laboratory, Matsushita Electric Industrial Co., Ltd.).

†† 松下電器産業(株)情報システム研究所

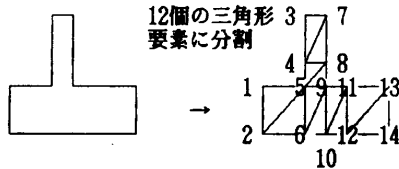


図 1 有限要素モデル例
Fig. 1 Example of finite element model.

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。

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

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、文献 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) 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 に示されるような 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。

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

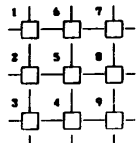


図 2 2次元アレイ型マルチプロセッサ
Fig. 2 2-Dimensional array processor.

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。ただし、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通信がすべて終わってから垂直方向の通信を行う。)
- 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言うが、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 節点間のデータ通信量は等しい。
- 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 のように付けられているとすると、 E はバンド行列であり、行列 M が行列 E と似ると、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 または直接結合する 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。

したがって、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、およ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のバンド幅縮小化およ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、よい発見の手法があり^{2), 4), 10)}、ここでは RCM 法¹⁰⁾を採用する。(付録参照)

RCM 法により再順序付けされた節点を、N/P 個ずつにグルーピング (簡単のためNはPの倍数と考える) し、各グループをその順に PE に割り当てることにより、初期マッピングを得る。

評価関数に基づく反復改良

バンド幅縮小化により、特定の方向への通信量は減少できたと期待できるが、さらに節点の PE への割り付けの交換により、他の方向への通信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通信時間の算出は、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。

- $T(t_{ij})$ 通信量行列 $1 \leq i, j \leq P$
 t_{ij} : PE i, j 間通信データ量
- $B(b_{ij})$ PE 通信負荷行列 $1 \leq i, j \leq P$
 $b_{ij} = d_{ij} \times t_{ij}$
- $C(c_{ij})$ 通信路負荷行列 $1 \leq i, j \leq P$
 c_{ij} : PE i, j 間通信データ量
- P_i : PE i からある1方向への重み付データ量

$$P_i = \sum_{j=1}^P b_{ij} / 2d$$

ただし、 d はプロセッサ結合の次元である。

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) $(\max_{ij} t_{ij}) \times (\max_{(t_{ki}>0)} d_{ki}) \times P$

すべての PE で最大データ量のペケットが最大距離を動いた場合の通信時間

(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)

2次元アレイの場合、プロセッサ番号を (水平方向の位置座標, 垂直方向の位置座標) の形式の2次元表示に変換すると、評価関数(F)の評価式は次のようになる。

$$\max \left[\begin{array}{l} \max_{k=1, 2, \dots, (P)^{1/2}} \sum_{i=1}^{(P)^{1/2}-1} C(i, k)(i+1, k), \\ \max_{k=1, 2, \dots, (P)^{1/2}} \sum_{i=1}^{(P)^{1/2}-1} C(k, i)(k, i+1) \end{array} \right]$$

交換は先のグループ単位で行う。ただしこれは PE 接続の第2, 第3次元方向の接続を生かすためであるから1次元アレイでは行わず、例えば2次元アレイでは図4のように PE11に初期マッピングされたグループは、PE 6~PE 15 の範囲との交換のみを考える。

行列Tに基づいて(A)~(F)の値を求めるには表1で示される計算量を要する。

実際の処理では(A)~(F)のいずれか1つまたは複数を組み合わせて評価関数とするが、第5章のシミュ

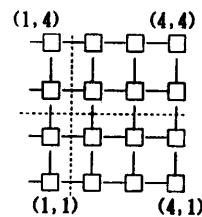


図3 PE 群間の通信データ
Fig. 3 Data transferred between PE groups.

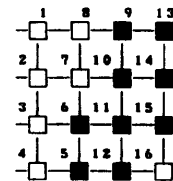


図4 PE の交換範囲
Fig. 4 Range of the exchange of PE.

表 1 評価関数の計算量

Table 1 Computation complexity of estimation function.

(A)	(B)	(C)	(D)	(E)	(F)
P^2	P^2	P^2	P^2	$2P^2$	$P^{2.5}$

レーションではそれぞれを用いた時を比較している。

4. アルゴリズムの理論的評価

本アルゴリズムの一般的な場合について解の良さを評価するのは難しいので、ある仮定のもとで、本アルゴリズムの有効性を示す。

均一性の仮定

M はバンド幅縮小化により、バンド幅 w 、バンド内要素の非零確率 $= \beta$ ($0 < \beta \leq 1$) なるバンド内が均質な行列となる。

β の値は以下の定理に影響がないので、 $\beta=1$ とする。

[定理 1]

1次元アレイ型マルチプロセッサに対しては、 N/P 個ずつの節点を番号順にグルーピングし、PE にその番号順に割り当てるのが総重み付データ量 $\sum P_i$ を最小にするアルゴリズムである。

[証明]

マッピングの変更によって、 $\sum P_i$ が減少しないことを証明する。

1次元アレイであるから、PE 間距離行列 D は図 5 のようになる。

そこで、仮定より PE 通信負荷行列 B は図 6 のようになる。

$2 \cdot \sum P_i$ は行列 B の各要素の総和であるが、要素の最大値 ($\max b_{ij}$) は w である。

マッピングの変更を行うと、行列 B は非ゼロ要素 (バンド内) の幾つか (その数を n_1 とする) がゼロになると同時にゼロ要素 (行列の右上と左下) の幾つか (その数を n_2 とする) が非ゼロとなり、その値は要素の位置から考えて $w+1$ 以上になる。

またマッピングの変更は、単なる行列 T の行と列の入れ替えにすぎないから、行列 B における非ゼロ要素数はマッピング方法に関係なく一定であり、 $n_1=n_2$ である。

したがって、マッピングの変更によって $\sum P_i$ は減少しない。 証明終

[命題 1]

n 節点モデルのアレイ型マルチプロセッサへのマッ

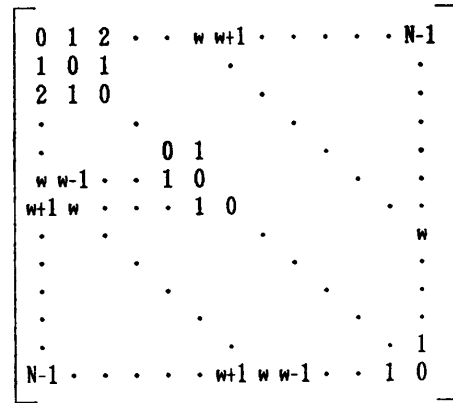


図 5 行列 D
Fig. 5 Matrix D .

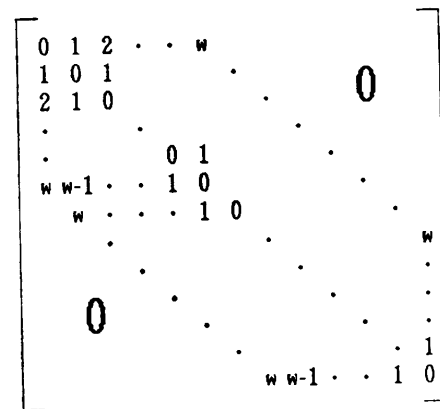


図 6 行列 B
Fig. 6 Matrix B .

ピングにおいて、任意の節点 x のマッピングの変化による 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次元アレイに対しては $\sum P_i$ 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。 $P=25, w=6$ とすると、図 7 において、PE 7 からは PE 1~PE 13 に対して、PE 9 からは PE 3~PE 15 に対して通信が必要である。この時 PE 7 と 9 の割り当てを交換し、交換後の重み付データ通信量を $'$ を付けて示すと

$$P_9' - P_9 = 2+1+2+1+0+1+2+2+3+3+2+1+2$$

(3) (15) ←

相手 PE 番号

表 2 通信データ量の遷移
Table 2 Transition of transferred data quantum.

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	q-k+2 q-k+3 ⋮ q	q-k+1 q-k+2 ⋮ q-1 q	q-k q-k+1 ⋮ q-2 q-1	1 2 ⋮ k	1 ⋮ k-1	1
	小計 Tcv	q	2q-1	kq- k(k-1) 2	k(k+1) 2
横 Tch	(k-1)q	(k-2)q	q	0					

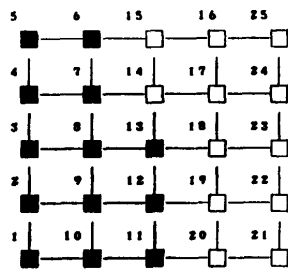


図 7 PE 交換による通信量の減少
Fig. 7 Reduction of data transfer quantity.

$$-(2+3+3+2+2+1+0+1+2+1+2+3+3)$$

$$(3) \dots \dots \dots (15)$$

相手 PE 番号

$$=-3$$

同様にして, $P'_i - P_i = -3$

$$\therefore P'_9 + P'_7 < P_9 + P_7$$

これと命題 1 より $\sum P_i$ は減少する. 証明終

【補題 2】

2次元アレイの場合, $w=2(P)^{1/2}m$ (m は整数) が成立する時のみ, $\sum P_i$ が最小となる.

【証明】 付録参照

【定理 2】

評価関数 (A), (D) は実際の通信時間 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, 各通信ステップでの通信量 (時間) は表 2 のようになる.

$$\text{ここで } k=w \cdot (P)^{-1/2}, q=(P)^{1/2}/2 \text{ とおき}$$

$$P/2 > w > 2q$$

と仮定している.

$$T_{cv} = T_{ch} \text{ となるステップ数を } n \text{ とすると,}$$

$$T_c = \max(q, (k-1)q) + \max(2q-1, (k-2)q) + \dots + \max((k-1)q + (k-1)(k-2)/2, q) + (kq - k(k-1)/2) + \dots + k(k+1)/2 + \dots + 1 = kq^2/2 - (k^2 - k - 1)q/2 + k(k-1)(k^2 - k - 3)/8 + (2k - n)nq/2 + (k^2 - k - n^2 + n)q/2 - k(k-1)(k^2 - k - 2)/8 + n(n-1)(n^2 - n - 2)/8 = kq^2/2 + (2k - 2n + 1)nq/2 + 5k(k-1)/8 + n(n-1)(n^2 - n - 2)/8$$

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

$$P_i = \sum_j d_{ij} \cdot t_{ij}/4 = k(q^2 + q)/2 + k^2q/2 = kq^2/2 + (k + k^2)q/2$$

$$c_{ij} = k(1 + \dots + q) = kq^2/2 + kq/2$$

証明終

【補題 3】

通信時間 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】 付録参照

バンド幅縮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 および定理 2 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, 通信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通信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.

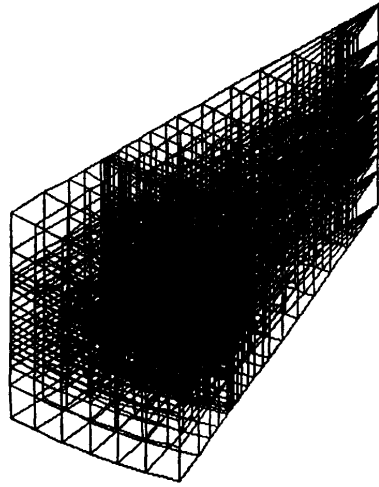


図 8 有限要素モデルのメッシュ分割図
Fig. 8 Mesh divided finite element model.

表 3 対象のアレイ型マルチプロセッサ
Table 3 Numbers of PE's of used array processors.
(a) 接続次元=1 (b) 接続次元=2 (c) 接続次元=3

No.	PE 数	No.	PE 数	No.	PE 数
1	24	7	25	13	27
2	64	8	64	14	64
3	125	9	121	15	125
4	220	10	225	16	216
5	520	11	529	17	512
6	980	12	961	18	1000

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 18 種類である。

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、バンド幅縮小化を施す前の通信時間を、100% 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。また、1, 2 および 3 次元アレイ間の通信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通信時間はそれぞれ 50~70%, 60~80% に減少され、その差は節点数/PE 数が小さい時ほど大きい。なお、図 9, 10, 12, 13 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) および (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ため、ここでは両方の値を用いている。(C), (E) はそれぞれ通信時間の下限と上限を与えるが、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 数が小さいほど現れる。

次に、PE 数の増加に伴う並列処理効率の変化を考察する。並列処理効率を

$$\text{並列処理効率} = \frac{\text{計算時間}}{\text{通信時間} + \text{計算時間}}$$

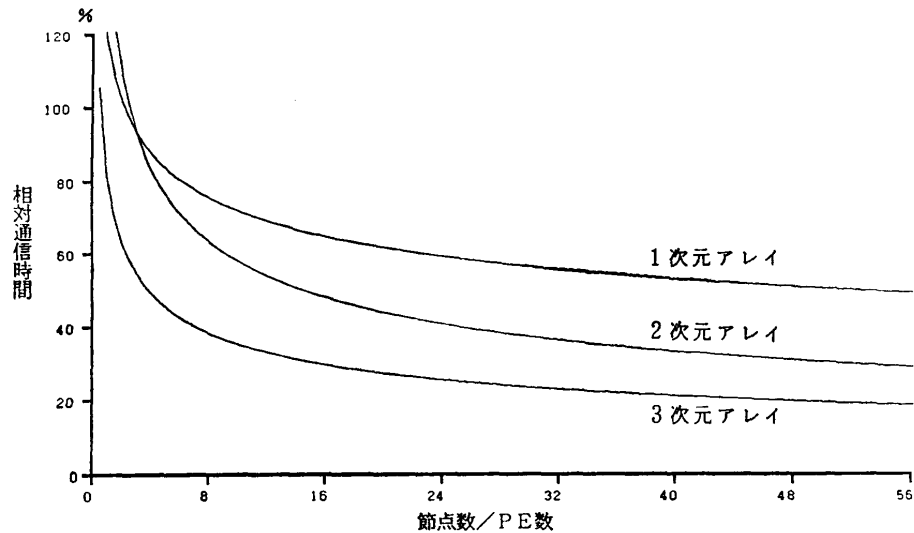


図 9 バンド幅縮小化による通信時間の変化
Fig. 9 Data transfer time after band width reduction.

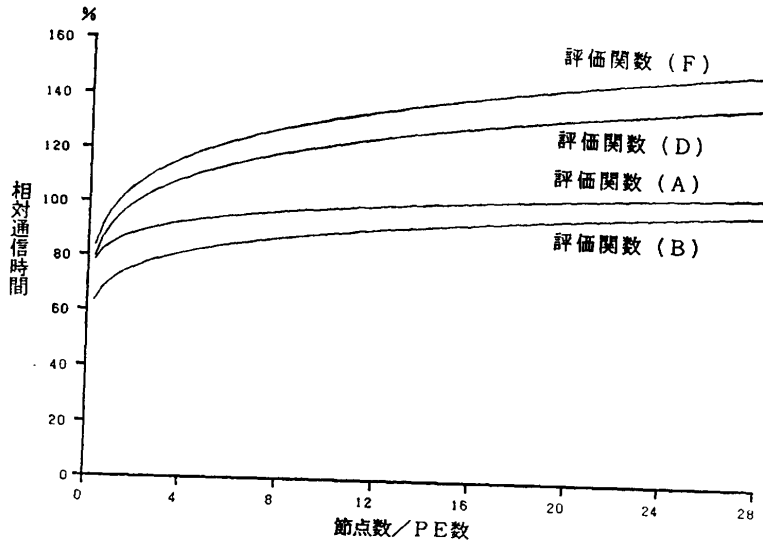


図 10 評価関数による通信時間の変化
Fig. 10 Difference of data transfer time between several estimation functions.

で定義する。3章でも述べたように、各 PE の受け持つ節点数は同一 (節点数/PE 数) であるので、各 PE の計算時間はすべて同じになる。その結果

$$\text{計算時間} = \frac{\text{節点数}}{\text{PE 数}} \times (\text{1 節点に対する演算量}) \times (\text{1 データの 1 演算時間})$$

で示される。

1 節点に対する演算量は次のようにして求められる。

5 点差分の反復法で処理すると、図 11 で示される

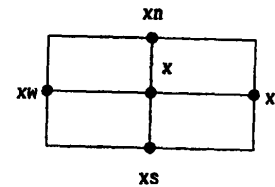


図 11 2次元四角形有限要素モデル
Fig. 11 2-Dimensional square finite element model.

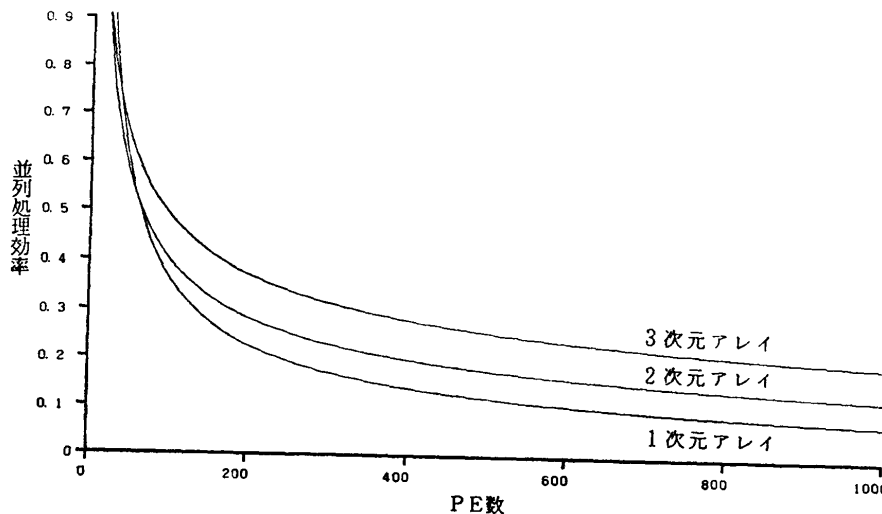


図 12 並列処理効率—PE 数
Fig. 12 Parallel processing efficiency versus number of PEs.

有限要素モデルの節点 x^{i+1} ($i+1$ は反復回数を示す) の計算式は

$$x^{i+1} = \frac{xn^i + xs^i + xw^i + xe^i}{4} + (1-w) \cdot x^i$$

となるので、1 節点に対する演算量 = 6 である。(w は適当な係数)

図 12 は (1 データの隣接 PE 間通信時間) = (1 データの 1 演算時間) とした場合の PE 数の増加に対する並列処理効率の変化を示す図である。これは節点数 3280 の有限要素モデルに対するもので、比較的並列処理効率の良かった場合である。モデルによっては、並列処理効率が、この 1/2~1/3 に低下する。

図 13 は上記の仮定のもとで、1 PE 当たりの節点数の増加に対する並列処理効率の変化を示す図である。1 PE 当たりの節点数が小さいほど並列処理効率は低下し、特に

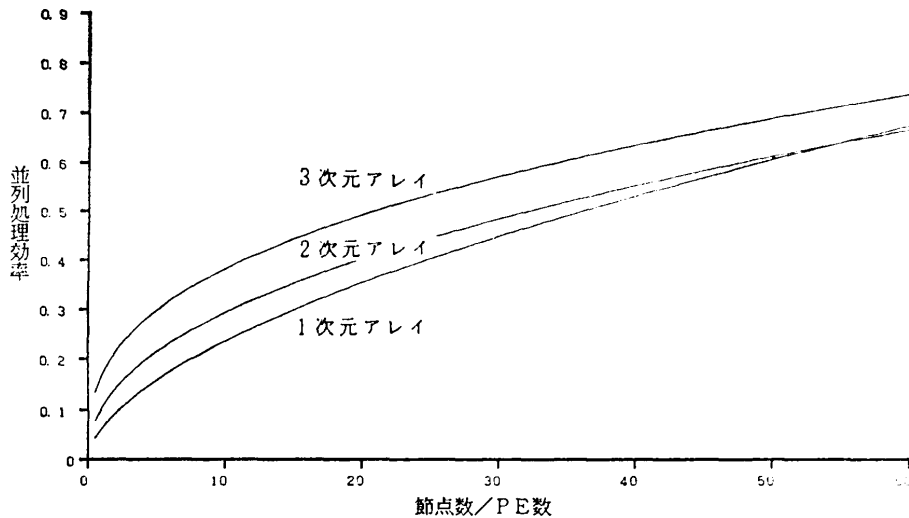


図 13 並列処理効率—節点数/PE 数

Fig. 13 Parallel processing efficiency versus number of nodes per PE.

10以下では高い並列処理効率を得られない。

図 12, 13 は (1 データの隣接 PE 間通信時間) = $n * (1 \text{ データの 1 演算時間})$ ならば縦軸座標 y を $y / ((1-y)n + y)$ に変換することで一般化できる。

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 10 MIPS のマイクロプロセッサ、16 MFLOPS 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、1,000 個の PE を 64 MB/S の高速通信路で 3次元結合することにより、図 12, 13 より節点数 3280 の有限要素モデルに対して、30% の効率、実行速度 4.8 GFLOPS 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、総通信量を最小にするような

マッピングが一番よい。

- ③ 並列処理効率は PE 数の増加に伴い低下する。
- ④ PE の接続次元は大きいほど通信に有利であるが、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。

したがって有限要素モデルの解析においては、バンド幅縮小化+総通信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いただいた、松下電器産業(株) 開発研究所 機構第 3 開発室の井ノ上裕人氏に感謝いたします。

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. Comput.*, Vol. 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, pp. 484-512 (1984).
 - 4) Berger, M.J. and Bokhari, S.H.: A Partitioning Strategy for PDEs across Multiprocessors, *Proc. Int. Conf. Parallel Processing*, pp. 166-170 (1985).
 - 5) Berger, M. and Bokhari, S.: A Partitioning Strategy for Nonuniform Problems on Multiprocessors, *IEEE Trans. Comput.*, Vol. C-36, pp. 570-580 (1987).
 - 6) 特集: 並列マシン, 情報処理, Vol. 28, No. 1 (1987).
 - 7) Hwang, K. and Briggs, F.A.: *Computer Architecture and Parallel Processing*, pp. 355-361, 769-774, McGraw-Hill, Singapore (1985).
 - 8) 加藤満左夫, 苗村憲司: 並列計算機, pp. 149-151, オーム社, 東京 (1976).
 - 9) Tarjan, R.E.: Graph Theory and Gaussian Elimination, in Bunch and Rose, eds., *Sparse Matrix Computations*, pp. 3-22, New York (1976).
 - 10) 村田健郎, 小園 力, 唐木幸比古: スーパーコンピュータ, pp. 115-117, 丸善, 東京 (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.

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

w_i の和 $P = \sum_i w_i$ をプロフィール

(profile) と呼ぶ.

RCM 法はバンド幅を小さくするための 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を示す.

有限要素で分割したのち節点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言ひ, ρ_i または $\text{Deg}(i)$ で表す.

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 を根と言ひ, 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 の節点と言ひ, その次数が増す順にレベル 1 の節点に連続して番号をふる.

[手順 4] レベル 1 の各節点につながる節点をレベル 2 にあると言ひ. 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 の頂点に付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 個の節点の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.

[補題 2 の証明]

$$(1) \quad 2(P)^{1/2}m < \text{プロセッサ番号 } A < P - 2(P)^{1/2}m$$

図 14-(a) のように, PE A から通信の必要な範囲は PE B~PE C である. $w = 2(P)^{1/2}m$ であれば, PE B, PE C は PE A とアレイ上の y 座標が等しい. また, 通信が必要な PE 数は y 座標 = 0, 1, 3, 4 で各 4 個, $y = 2$ (PE A の位置) では 5 個と他より一つ多

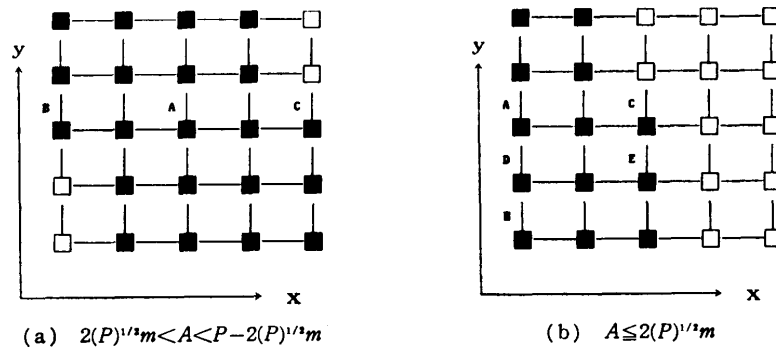


図 14 $w = 2(P)^{1/2}m$ の時のデータ通信
Fig. 14 Data communication at $w = 2(P)^{1/2}m$.

い。プロセッサアレイはトーラス状結合であるから、プロセッサAのマップング変更により、通信量 P_A は増加する。このことは、Aの座標、 m の値に無関係に成立する。

- (2) プロセッサ番号 $A \leq 2(P)^{1/2}m$ または
プロセッサ番号 $A \geq P - 2(P)^{1/2}m$

図 14-(b)のように、PE A から通信の必要な範囲は PE B~PE C である。PE A の y 座標を a とすると、PE B~PE C では、 y 座標 = $0 \sim a$ までの PE が y 座標 = $a+1 \sim (P)^{1/2}$ の PE より一つ多いから、 y 座標 = $a+1 \sim (P)^{1/2}$ の PE との交換により通信量 P_A が減少しないのは自明である。そこで y 座標 = d ($0 \leq d < a$) の PE D との交換を試みる。

交換後の重み付データ量を $'$ を付けて示すと

$$P'_A - P_A = \sum_{i=1}^d i + \sum_{i=1}^{a-d} i - \sum_{i=1}^a i$$

$$P'_D - P_D = \sum_{i=a-d+1}^a i - \sum_{i=1}^d i$$

$$\therefore P'_A + P'_D = P_A + P_D$$

これと命題 1 より P_i は減少しない。 証明終

[補題 3 の証明]

今、データ通信は 2.3 節で述べたような方式である。評価関数(C)は2つの PE 間での通信データ量と通信距離の積の最大値であり、評価関数(D)は一つの通信路を通過するデータ総量の最大値である。したがって、これらが通信時間 T_c の下限となるのは明らかである。

評価関数(E)はすべての PE で最大データ量 ($\max t_{ij}$) のパケットが最大距離 ($\max d_{ij}$) を移動し、しかもすべてのパケットが同一通信路を通過 (P) する場合の通信時間を示しているのだから、これが T_c の上限を示すのも自明である。

(昭和 63 年 1 月 29 日受付)

(昭和 63 年 6 月 24 日採録)



岡部 公治 (正会員)

昭和 33 年生。昭和 56 年京都大学工学部数理工学科卒業。昭和 58 年同大学院修士課程修了。同年松下電器産業(株)入社。現在情報システム研究所第 3 システム開発室技師。

EDP システムの構築。並列計算機アーキテクチャ、自動並列化コンパイラ等の研究、開発に従事。最近では並列処理アルゴリズム、マルチプロセッサのスケジューリングに興味を持つ。



浅原 重夫 (正会員)

昭和 28 年生。昭和 51 年大阪大学工学部電子工学科卒業。昭和 53 年同大学院博士課程修了。工学博士。同年松下電器産業(株)入社。現在情報システム研究所第 3 システム開発

室主任技師。グラフ理論の CAD への応用、ナビゲーションシステム、3次元画像生成用マルチプロセッサシステム等の研究、開発に従事。現在、並列処理アルゴリズムと3次元グラフィックスの応用に興味を持つ。IEEE, 電子情報通信学会各会員。



鷲島 敬之 (正会員)

1944 年 2 月 9 日生。1968 年 3 月大阪大学工学部電子工学科修士課程修了。同年 4 月松下電器産業(株)入社無線研究所配属、カラーテレビジョンの研究開発を経て情報関係技術

の研究開発に従事。現在同社技術本部情報システム研究所所属。計算機アーキテクチャ、コンピュータグラフィックスに興味を持つ。電子情報通信学会、ACM、IEEE CS。