

## FIFOメモリを使用したプロセッサアレイ方式

小松 俊雄 (NTT 電気通信研究所)

### 1. はじめに

マイクロプロセッサ 8086 を用いて 16 台のプロセッサエレメント (PE) からなるプロセッサアレイ装置 ESPAR<sup>[1]~[2]</sup> を以前、試作した。この目的は、プロセッサアレイ方式の有効性を確認するとともにプログラムの作成面に対して、総合的に評価する事にあった。対象とした問題は逆行列を反復法で解くものであり、並列処理効果の高いケースがネットワーク結合に適した特殊なスペース行列 (例えば 2 次元の 2 階の偏微分方程式を差分法で近似した際生じる行列) に限定される事が問題であった。また、数値計算をねらいとした世の中のプロセッサアレイ装置も、並列処理効果が高いのは主に特殊なスペース行列に限定されている。

そこで、プロセッサアレイの適用範囲を拡大するために、一般的な行列、すなわち密、帶、スペース行列に対しても並列処理効果が高くシンプルな処理が可能な方式 (以下、ESPAR-2 と称する) を提案する。ESPAR-2 の特徴は、

- ① MIMD 可能
- ② ネットワークの終端は行方向に一次元ループ、列方向は列単位にループ
- ③ PE は演算系と転送系からなり、それぞれ独立に動作可能
- ④ 演算系と転送系間に FIFO の通信メモリを設定
- ⑤ PE 間転送は FIFO のメモリ転送である。

以下に ESPAR の概要ならびに ESPAR-2 の装置構成、処理方法および机上評価結果を示す。

### 2. ESPAR の概要

ESPAR はハードウェアシミュレータとしての性格も持たせており、ハード的には冗長構成となっている。ネットワーク結合は現実的で、比較的適用範囲の広いメッシュ型としている。また、プログラムのうち使用頻度の高い個所はサブルーチン化している。言語は性能評価時、ハードとソフトのきりわけを明確にするためアセンブリとした。

#### (1) ハードウェア構成

ESPAR は 16 台の PE 群とこれらを制御するコントロールユニット (CU) からな

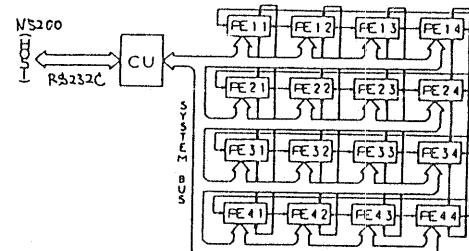


図1 システム構成

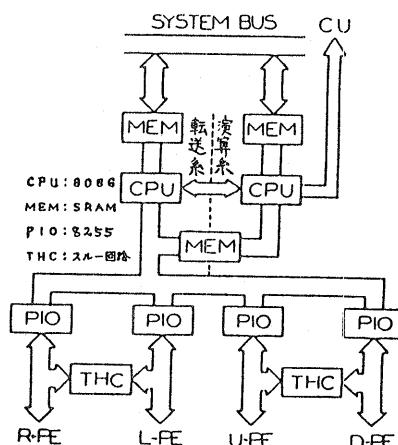


図2 PE のハードウェア構成

り、バックエンドとして位置付けられる。ホストはパソコンN5200であり、RS232CインターフェースでESPARと接続されている。図1にシステム構成、図2にPE構成を示す。本装置の特徴は次の通りである。

①MIMD化が可能：PEにプロセッサ(8086)とメモリ(計84KB)を持たせ、CU→PE間制御として i) 初期設定、ii) スタート、iii) 再スタート、iv) ストップ、v) 再構成指示が同時に任意のPE群に対して送出可能である。また、任意のPE群から同時に種々の報告が可能である。

②PEを演算系と転送系に分離：各系に対してプロセッサとメモリを個別に、また両系間に通信メモリを設ける事により演算と転送の同時動作が可能である。なお、両系間の通信はソフトによるハンドシェイクである。

③スルーパスの設定：障害時、予備PE群に切り替え可能とする。

④PEメモリのビジブル化：CUは独自のメモリを持つとともに、任意のPEメモリをアクセス可能である。

## (2) ソフトウェア構成

ソフトウェアは、ホストプログラム、CUプログラム、PE演算系プログラム、PE転送系プログラムからなる。図3にプログラム構成を示す。プログラム作成時、必要に応じてこれらの各プログラムをコールする。プログラムの主なフローは次の通りである。①CUはホストから各オブジェクトファイルを受け取り、各PEに対してプログラムを初期設定し、スタート指示を出す。②スタート指示後、各PEは自メモリからプログラムを読み出し実行を開始する。PE間転送時、演算系は転送系に対して転送指示を出す。③PEからの演算終了報告後、CUは演算結果を編集してホストに報告する。

ESPAR上で特殊なスペース行列以外の行列を実行する場合、①ネットワークの終端接続に柔軟性がない ②演算系と転送系間の通信速度が遅い、また両系間の独立性が低い ③PE間の転送速度が遅い 等が問題となる。

## 3. ESPAR-2の装置構成

一般に、マイクロプロセッサベースで数値計算用のプロセッサアレイ装置の構成は、主に次の点より分類できる。①演算系と転送系の分離度 ②両系間の通信方法 ③ネットワークの終端接続 ④ネットワークのオプション ⑤PE間の転送方法 ⑥制御情報、状態情報の種類)

### (1) 演算系と転送系の分離

演算系と転送系を分離せず、1つのプロセッサが演算の一部(浮動小数点演算は専用プロセッサで実行)と転送を行う場合、演算と転送処理はシーケンシャルにプログラム化されるので両者の同時動作は実行できない。従って、PE間転送を行って必要なデータがそ

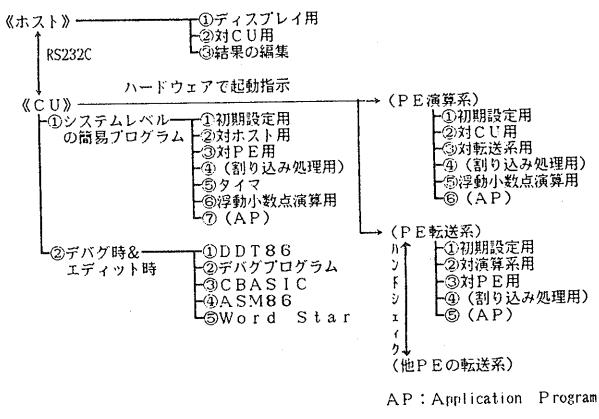


図3. プログラム構成

ろった時点で演算処理を開始するので、転送処理がオーバーヘッドとなる。ESPAR-2ではESPARと同様、それぞれの系にプロセッサとメモリを設ける。

## (2) 両系間の通信方法

ESPARの通信メモリに対する通信方式はソフトによるハンドシェイクであり、次の点が問題となる。  
①演算系と転送系が全く独立に同時動作できない。例えば、転送系が通信メモリにwrite(read)する場合、write(read)して良いかどうかをチェックする必要がある。演算系がアクセス中はその間待たされる。  
②演算系または転送系のどちらかに通信メモリのアクセス権があるので、両系は互いにアクセス権をソフトで管理する必要がある。

そこで、通信メモリに対する競合制御を簡単化して、両系の独立性を高めるとともに通信速度を向上させる必要がある。

ESPAR-2では、転送パターンが実行前にわかっている（静的である）事に着目して、通信メモリにFIFOメモリを適用する。これにより、演算系および転送系からのwriteは単にFIFOメモリに書き込むだけで済む。従って、writeが可能かどうかのチェックが不要となり、相手がread中であってもwrite可能である。また、readも単にFIFOのEF（エンディティフラグ）をチェックして、ONであればデータがreadできるので、演算系と転送系の独立性が高まり、通信速度も向上する。

①構成：図4にFIFOメモリによる通信メモリの構成を示す。通信メモリはFIFO1とFIFO2からなる。FIFO1は演算系（転送系）からのwrite(read)専用の64ビット幅のFIFOメモリである。FIFO1の語数は $\ell$ 語（ $\ell = PE$ 当たりの変数xの割り付け数）である。FIFO2は演算系（転送系）からのread(write)専用の64ビット幅のFIFOメモリである。FIFO2には転送系が順次、転送データを書き込むので語数は最大 $\ell * PE$ 数語（転送されたデータは演算系が逐次、処理するので実際はこれよりかなり小さい値となる）必要となる。

②動作：i) 演算系はまず転送データとして演算結果をFIFO1に書き込む。この書き込みが転送指示となる。次にFIFO2のEFビットを参照して転送データが転送系から書き込まれているかどうかをチェックする。書き込まれているならば、FIFO2からデータをreadして演算を実行する。ii) 転送系はFIFO1のEFビットを参照して転送データが演算系から書き込まれたかどうかをチェックする。書き込まれているならば、FIFO1からデータをreadしてPE間転送を開始する。隣接PEから転送されたデータは、必要に応じてFIFO2に書き込むとともに隣接PEに転送する。

## (3) ネットワークの終端接続

主に次の3方法がある。  
①行、列単位に開  
②行、列単位にループ、  
③行方向は一次元ループで列方向は列単位にループ。  
特殊な2次元のスペース行列を対象とする場合、両端のPEは互いの演算結果を必要としないので、いずれの終端でも良い。しかし、一般

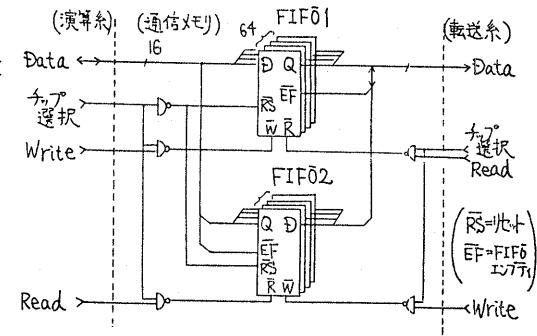


図4 FIFOの通信メモリ構成

的なスペース、特殊なスペース（次元数が3以上）や帶行列を対象とする場合、①、②の終端ではP E間転送がP E間で異なってくるため、転送が非効率となる。一方、③は各P E、基本的に、左、右方向のP E間転送を主体にする事により、効率良く転送ができるので、種々の行列に適している。

#### (4) ネットワークのオプション

①スルーパスについて：E S P A Rでは障害対策として上一下、左一右方向のスルーパスを設けている。障害時、スルーパスを使用して行、列単位のループパスを確保している。スルーパス状態のP Eは単にパスとして見える。E S P A R - 2ではネットワークの終端が（3）③であるので障害時、行方向、列方向のループを確保するためには、上一下、左一右、上一右、左一下方向のスルーパスを設ける必要がある、しかしこれは非現実的である。そこで、障害が生じた場合、行方向の一次元ループのみを再構成する。

②バイパスについて：遠距離のP E間転送を高速に行うため、新たにバイパスを設ける方法がある。しかし、次の理由により設けない。i) 転送幅が64ビットなのでピンネックの恐れがある。ii) 上、下方向のパスをバイパスとして使用できる。iii) さらに演算系と転送系は同時動作を行うので、新たにバイパスを設けるメリットは小さい。

#### (5) P E間の転送

E S P A Rの転送方法はハンドシェイクのI/O転送であるため、転送速度が遅い。従って、スペース率の高いスペース行列を対象とする場合、転送処理がかなり実行時間に見えてくる問題がある。P E間転送として、一般的にI/O転送以外にメモリ転送がある。E S P A R - 2では（2）と同様にFIFOメモリをメモリ転送に使用する。なお、RAMメモリについては、隣接P E間のマシンサイクルを互いに半周期ずらしてメモリ競合を無くす方法がある。しかし、この方法では、論理矛盾をなくすためどこかで（例えばCU）同期を取り、またはRAMメモリにデータの有効性を示すVビットエリアを設けてソフトで制御してP E間転送を行う必要があり、オーバーヘッドが大きくなる。また、アービタをメモリ制御として付加する方法もあるが、付加回路とバス制御のオーバーヘッドが大変である。

①構成：図5にFIFOメモリ周辺の構成を示す。FIFOメモリを隣接P E毎に持つ。FIFOメモリは自P Eから見ればread専用で他P Eから見ればwrite専用である。FIFOメモリのビット幅は64ビットで語数は $\ell * r$ （各P E同一の転送なので $0 < r \leq 1$ ）

②動作：i) 転送系はFIFO1のデータを転送すべき隣接P EのFIFOメモリにwrite

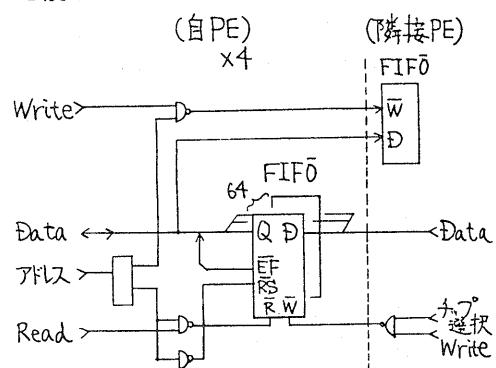


図5 P E間転送用のFIFO周辺

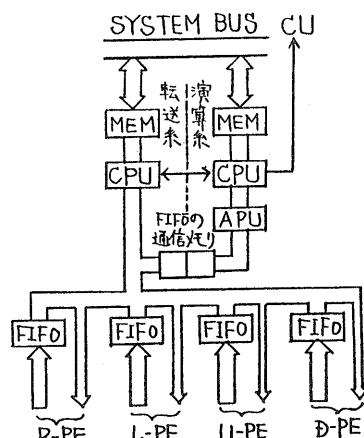


図6 E S P A R - 2 のP E構成

$t_e$ する。この書き込みが転送指示となる。ii) 次に自FIFOメモリ(転送方向は前もってわかっている)の8バイト目の $\overline{EF}$ ビットをチェックする。転送データが書き込まれていれば、必要に応じてそのデータをFIFO2に書き込むとともに次のPE間転送として、そのデータを転送すべき隣接PEのFIFOメモリに書き込む。iii) 以下、一定回数ii) を繰り返す。

図6に、ESPAR-2のPE構成を示す。

#### 4. 处理方法

以下、 $AX = B$ の反復法としてヤコビ法を一例としてその処理方法を示す。第 $k+1$ 近似解は次の式で表される。

$$X^{k+1} = D^{-1} \{ B - (A - D) X^k \}$$

(Dは $a_{11}, a_{22}, \dots, a_{mm}$ を要素を持つ対角行列)

##### (1) 基本処理

①ダミー $X' = 0$  ( $X'$ の次元数は $\alpha$ で $0 \leq \alpha < n$ ) を追加して、 $m + \alpha = \ell * n$  ( $m$ は $X$ の次元数、 $\ell$ は自然数) とする。すなわち、

$$\begin{bmatrix} a_{11} a_{12} \dots a_{1m} \\ a_{m1} a_{m2} \dots a_{mm} \\ 0 \end{bmatrix} \begin{bmatrix} X \\ X' \end{bmatrix} = \begin{bmatrix} B \\ 0 \end{bmatrix}$$

以下、説明を簡単にするため $\alpha = 0$ とする。なお、特殊なスペースの場合は、上、下、左、右方向の転送を行うので $X$ の中にダミーを分散させる。これにより、各PEに割り付ける変数の数を $\ell$ とする事によって、各PEのプログラムを同一とする。なお、初期データは各PE個別に設定する。

② $\ell \geq 2$ の場合、変数間の転送は実際のPE間転送(実転送)とPE内の変数間転送(仮想転送)からなる。処理の高速化のため、実転送と仮想転送はオーバーラップさせ、演算系は仮想転送を実行後、実転送された転送データを処理する。なお、 $\ell = 1$ の場合、変数間の転送は実転送のみからなる。

③種々の行列に対して、PE間転送は基本的に左方向の転送の繰り返しと右方向の転送の繰り返しで行う。例外として、特殊なスペースの場合、バイパスとして上、下方向の隣接転送を行う。

④遠隔転送は隣接転送を繰り返す事によって行う。この際、中間時の転送データを演算系が順次、処理する事により見かけ上、遠隔転送を隣接転送に見せる。

⑤一般的なスペース行列は帯または密行列とみなす。なお、 $a_{ij} = 0$ の演算 $a_{ij} * x_j$ はスキップさせる。

⑥特殊なスペース行列は帯行列とみなして、転送系は必要なデータのみを通信メモリに書き込む。

##### (2) 密行列

(n-1)回左方向に $\ell$ 個のデータを転送する。 $\ell = 1$ の場合、PE(i)への初期データは次の順で設定する。

$$a_{i,1}, a_{i,1+1}, a_{i,1+2}, \dots, a_{i,m}, a_{i,1}, a_{i,2}, \dots, a_{i,i-1}$$

##### (3) 帯行列

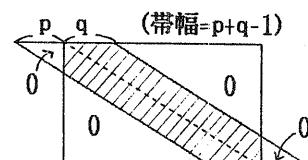


図7 帯行列の場合

演算系と転送系のプログラムを各P E同一とするため、図7に示すようにダミー(0)を加えて  $a_{i,j}$  を各P Eに初期設定する。 $\ell = 1$  の場合、q回左方向に、p回右方向に転送する。 $\ell \geq 2$  の場合、 $q = a * \ell$  ならば a 回、 $q \neq a * \ell$  ならば  $a + 1$  回左方向に、 $p = b * \ell$  ならば b 回、 $p \neq b * \ell$  ならば  $b + 1$  回右方向に転送する。 $\ell = 1$  の場合 P E(i)への初期データは次の順で設定する。

$$a_{i,i}, a_{i,i+1}, a_{i,i+2}, \dots, a_{i,i+q}, a_{i,i-1}, \dots, a_{i,i-p}$$

$\ell \geq 2$  の場合、変数間によって、実転送のデータの扱いおよび仮想転送の扱いが異なるので、プログラムの作成が若干複雑となる。

#### (4) 一般的なスパース行列

図8(a)のようなスパース行列の場合、 $p = \max_i(c_i)$ 、 $q = \max_j(d_j)$  の帶行列として扱う。また、(b)のようなスパース行列は密行列として扱う。

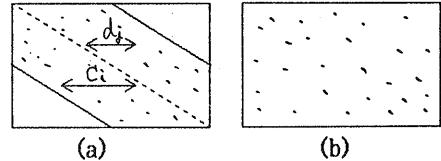


図8 一般的なスパース行列の場合

#### (5) 特殊なスパース行列

① (3) の改良：一般的なスパース行列の場合、 $a_{i,j} = 0$  である  $a_{i,j} * x_j$  の計算をスキップさせて性能向上を図っている。しかし、以下の処理がオーバーヘッドとなる。i)  $a_{i,j} = 0$  である  $a_{i,j} * x_j$  のスキップ処理。すなわち、 $a_{i,j} = 0$  かどうかの判定と  $a_{i,j}$  のポインタ更新、ii)  $a_{i,j} = 0$  である  $x_j$  を FIFO 2 に write, read する処理。

そこで、スパースに規則性がある事（例えば、n次元の特殊なスパースの場合、 $a_{i,j} (\neq 0)$  は対角線上から横方向に  $\pm \sum_{i=0}^{f-1} h^i$  ( $f$  は次元数、 $h$  は次元の一辺の大きさ) の位置にある）に着目して上記 i), ii) を省略する。手順は次の通りである。i) 図9に示すようにダミー(0)を加えて、各P Eのプログラムが同一となるようにする。ii) 斜線部で示す各要素を初期データとして演算系に設定する。iii) P E間転送として上、下方向の転送をバイパスとして使用する。iv) 転送系は  $a_{i,j} \neq 0$  でない  $x_j$  を受信したとき、FIFO 2 に  $x_j$  を書き込む。なお、上、下方向の転送を行う場合、転送系のプログラムは若干、複雑となる。



図9 特殊なスパース行列

②物理的解法：ある境界領域を持つ2次元の偏微分方程式を差分法で近似した場合、行列イメージで解くのではなく、物理的イメージで格子を各P Eに割りつけて解く方法がある。2次元の場合、P E間転送は上、下、左、右方向の四回の隣接転送でよく、非常に効率的に解ける。しかし、 $n (\geq 3)$  次元の場合、ネットワークは2次元であるので、P E数が多くてもP Eを有効に使用する事ができなく、P Eに対する格子割りつけ数が大きくなる。また、次元数が大きくなると割りつけ方（n次元→2次元への変換）が難しくなる問題も生じる。

### 5. 性能評価

#### (1) 評価条件

ESPARのプログラムをベースに、各評価用プログラムを作成し、実行時間を机上シミュレーションした。言語は8086アセンブリであり、各命令の実行時間は[6]を参考している。主な条件としては、①収束判定の処理時間は小さいので無視、②実行時間は

各命令の実行時間を加算, ③オペランドは偶数アドレスに存在, ④浮動小数点命令の実行時間は  $t_{typical}$  値等である。

## (2) 評価結果

並列処理効果 ( $P$ ) を次のように定義する。

$$P = \text{逐次の実行時間} (T_{tik}) / E S P A R - 2 \text{での実行時間} (T_{ES})$$

$$T_{ES} = \max \{ \text{演算系上での実行時間} (T_{EN}), \text{転送系上での実行時間} (T_{TEN}) \}$$

(ただし、各系上の実行時間は他系の処理待ちの時間を含まない)

$$\text{また, } t_{EN} = T_{EN} / \text{演算} (a_{ij} * x_j) \text{ 回数}$$

$$t_{TEN} = 8B \text{ 転送当たりの転送時間}$$

なお、逐次の場合、行列の種類により  $a_{ij}$  のポインタ操作が少し異なってくるので、演算回数当たりの実行時間は若干異なるが、程  $404 \sim 427\tau$  に収束する。

①密行列の場合:  $t_{EN} \geq 413\tau$ ,  $t_{TEN} \approx 145\tau$  であるので、  $T_{ES} \approx T_{EN}$  である。  $P$  を表1に示す。  $\ell = 1$  の場合、  $P$  は程PE数に比例する。

②帯行列の場合:  $t_{EN} \geq 414\tau$  であるので①と同様に  $T_{ES} \approx T_{EN}$  である。表1より、  $\ell = 1$  の場合、  $P$  は程PE数に比例する。

③一般的なスパース行列: スパース率 = 1 の場合、  $t_{EN}$  は程  $137\tau$  であり、 スパース率 = 0 の場合、 程  $413\tau$  である。  $t_{TEN} \approx 145\tau$  であるので、

$$\text{スパース率} \geq 0.03 \text{ の場合: } T_{ES} \approx T_{EN}$$

$$\text{スパース率} < 0.03 \text{ の場合: } T_{ES} \approx T_{TEN}$$

なお、  $\ell \geq 2$  の場合、  $T_{TEN} (\ell \geq 2) \approx T_{EN}$  ( $\ell = 1$ ) \*  $1/\ell$  となるので、  $T_{ES} \approx T_{EN}$  である。  $P$  を図10に示す。スパース率 = 0.2で、  $P$  はPE数の程0.45倍である。(なお、逐次は帯幅 =  $\max_{j=1}^m a_{ij} (\neq 0)$  数) の帯行列と見なしているが、實際上は  $a_{ij}$  のポインタ操作のため実行時間は増加するので、  $P$  はPE数の0.45倍以上になると予想される) また、図11にスキップ処理の効果を示している。演算回数が充分大きく、スパース率が高くなると、スキップ処理を行う方がそうでない方より、  $P$  は最大で程3倍向上する。

④特殊なスパース行列: i) ③の改良:  $t_{EN} = (448 + 375 * a) / a$  ( $a = 2 * M$ ,  $M$  は次元数)

$$t_{TEN} = 99 * \text{スパース率} + 162 * \text{スパース率}$$

従って、上、下方向の転送を行なわない場合、  $T_{EN} \geq T_{TEN}$  となる条件は

$$t_{EN} \geq t_{TEN} * (1 / \text{スパース率}) * 1 / \ell$$

である。(図12)  $\ell$  が大きくなると  $T_{ES} = T_{EN}$  の領域

表1 密、帯行列の  $P$  (PE数=1,000)

	密行列	帯行列	
		p, q=32	p, q=256
$\ell = 1$	979	1,047	912
$\ell = 16$	795	879	893
$\ell = 256$	817	941	955

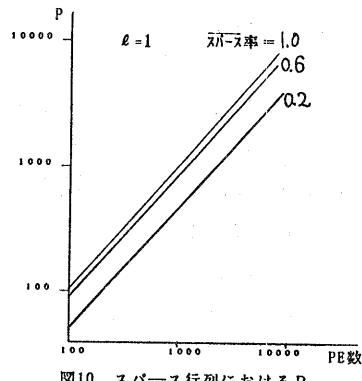


図10 スパース行列における  $P$

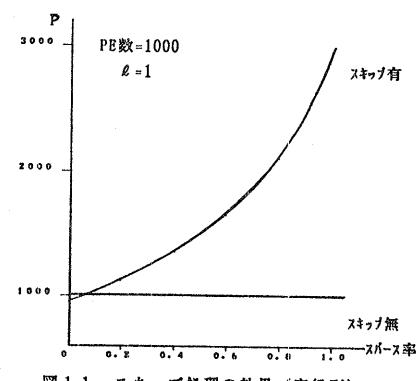


図11 スキップ処理の効果 (密行列)

が拡がる。また、上、下方向の転送を行うとさらに拡張できる。Pを図13に示す。 $\ell$ が大きくなると実転送の頻度が小さくなるので、Pは向上する。また、上、下方向の転送を行うとPは向上し、4次元で $\ell=1$ の場合、Pは程250倍となる。

ii) 物理的解法：  
 $T_{ES} = T_{EN}$ である。M=1, 2の場合、Pは程PE数に比例する。M≥3の場合、格子数を $r^M$  ( $r^M = \ell * n$ ) とすれば、 $r < \sqrt{n}$ の場合、Pは程 $r^2$ に比例する。 $r \geq \sqrt{n}$ の場合、Pは程nに比例する。

## 6. おわりに

種々の行列の反復法に対して、ESPAR-2の適用性を示した。主な結果は、①密、帯行列についてはPE数の0.8倍以上のPが期待できる。②一般的なスパース行列については、スパース率が高くなるとP(並列処理効果)は低下するが、スキップ処理を行う事により、行なわない場合より最大3倍位Pを向上させる事ができる。③特殊なスパース行列については、物理的解法と行列的解法((3)の改良)の2種の方法がある。物理的解法では、1, 2次元の場合、PE数に比例したPが期待できる。しかし、3次元以上の場合、割りつけ方によってPが異なってくる。また、(3)の改良では、上、下方向の転送をバイパスとして使用する事により、また $\ell$ を大きくする事により(3)より程(スキップ効果 \*  $\ell$  \* バイパスの効果)倍Pを向上させる事ができる。

以上、ESPAR-2では、Pが若干低下する場合もあるが、概して種々の行列の反復法を高速に実行できる。また、ESPAR-2は、PE間転送が高速でかつ演算系と転送系の独立性が高く両系間の通信も高速なので、反復法以外に種々の応用に対しても有効であると考える。

今後、 $\ell \geq 2$ の場合も含めて、処理系の検討等を行っていく予定である。

## 参考文献

- (1) 石川、小松：“プロセッサアレイ型のMIMD化に関する検討” S57電子通信学会総合全国大会 1687
- (2) 石川、小松：“プロセッサアレイ実験装置(ESPAR)の性能評価” S60前期情報処理全国大会 5B-2
- (3) 上村、星野：“並列計算機PAXによるADI法の実行と評価” 情報処理学会誌 VOL26 NO1
- (4) 信友、高橋：“2進木構造並列処理システムCORALプロトタイプの性能評価” 計算機アーキテクチャ44-1
- (5) “並列処理システムの性能を左右する相互結合ネットワーク” 日経エレクトロニクス81 DEC. 21 P.P. 88~108
- (6) “intel IAPX 86/88, 186/188 Users Manual” Programmers Reference

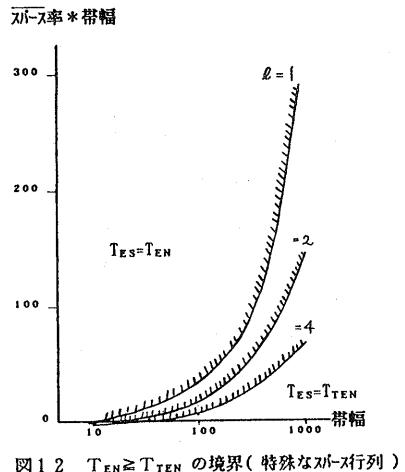


図12  $T_{EN} \geq T_{TN}$  の境界(特殊なスパース行列)

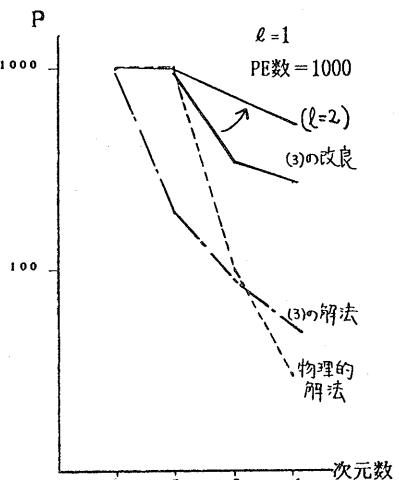


図13 特殊なスパース行列のP