マトリクスブロードキャストメモリ結合形並列計算機による n 元連立一次方程式の $O(n)$ 時間計算†

金田 悠紀夫†† 小畑 正貴†† 前川 禎男††

高速にそして低コストで大次元の連立一次方程式の数値計算を行いたいという要求はきわめて強いものがある．これらの要望に答えるものとしてパイプライン制御を用いたアレイ計算機がある．しかしながら，このような単一演算ユニットの高速化を指向したものでは n 元の連立一次方程式を解くのに直接法を用いた場合 $O(n^3)$ の時間が，係数行列が帯行列（帯幅 $2m+1$）の場合でも $O(nm^2)$ の時間を必要とする．ここではマトリクスブロードキャストメモリと呼ぶ特殊な共有メモリにより結合された多重プロセッサシステムにより並列計算を行い，ガウス消去法や変形コレスキー法を用いた計算が $O(n)$ 時間で実行できることを示す．また ILLIAC Ⅳ やシストリックアレイなどの他のシステムとの比較についても論じる．

1. まえがき

科学技術計算の分野において大形の連立一次方程式を高速計算で解きたいという要請はきわめて大きい．一般に直接法で n 元連立一次方程式を解く場合その計算時間はオーダ $O(n^3)$ となる．また n 元で帯幅が $2m+1$ なる帯係数行列をもつ連立一次方程式の場合その計算時間はオーダ $O(nm^2)$ となる．

ここではブロードキャストメモリという特殊な共有メモリを用いてマトリクス状に結合された計算機群によって並行計算を行うことにより，ガウス消去法および変形コレスキー法のいずれのアルゴリズムを用いても計算時間を $O(n)$ とすることが可能なことを示す．

他の $O(n)$ の能力をもつ ILLIAC Ⅳ やシストリックアレイとの比較についても論じる．

2. マトリクスブロードキャストメモリ結合形並列計算機システム

提案している並列計算機システムは多数台の演算プロセッサをブロードキャストメモリと呼ぶ特殊な共有メモリによってマトリクス状に結合したシステムで，$p \times q$ 台の多重プロセッサシステムである（**図1**）．

ブロードキャストメモリは**図2**に示すような構成のマルチポートのメモリシステムで，内部にはポート数だけのパケットメモリ (PKM) と呼ぶメモリバンクが設置されている．各パケットメモリ PKM_i $(i=1\sim p)$ にはそれぞれ同一のアドレス付けが行われている．読出しは全ポート並行して別々のアドレスのデータに対して行うことができる．一方，書込みを行う場合には書き込まれたデータは内部バスを介して全 PKM に伝達され，同一アドレス領域に格納される．したがって任意の PKM に演算プロセッサがデータを書き込むとそのデータは全 PKM の同一アドレス領域に同時に書き込まれることになりデータのブロードキャスティングが行われることになる（図2）．

各演算プロセッサにはローカルメモリが付加されており，各演算プロセッサにローカルなデータとプログラムが格納されている．以後 i 行 j 列の演算プロセッサ（以後 PE と呼ぶ）を p_{ij} で示し，i 列の PE アレイを列 PE アレイ $\boldsymbol{p}_i$，j 行の PE アレイを行 PE アレイ $\boldsymbol{p}_j$ とよぶことにする．

3. 連立一次方程式の並列計算

ここでは直接法のアルゴリズムであるガウス消去法と変形コレスキー法をとりあげる．両者とも図1に示したシステムで $O(n)$ 時間で計算可能であるが，ここではそれぞれについて論じる．対象の n 元連立一次方程式は $Ax=B$（A は $n\times n$，B は $n\times l$ のマトリクス）で表現されるとする．なお A の ij 要素を a_{ij}，B の st 要素を便宜上 $a_{s,n+t}$ と表現する．

3.1 ガウス消去法の場合

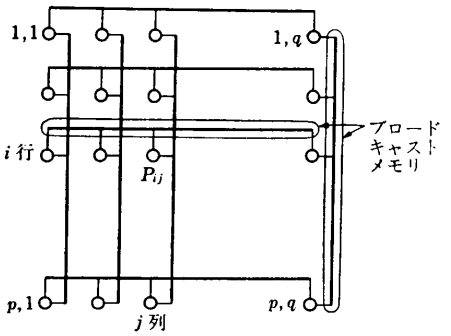
ガウス消去には前進消去と後退代入の手順がある．

前進消去の手順は i, j, k を整数変数として，

```
for k:=1 to n-1 do
  begin
```

† The $O(n)$ Time Computing Method of n Linear Simultaneous Equations on the Matrix-Broadcast-Memory Connected Array Processor System by YUKIO KANEDA, MASAKI KOHATA and SADAO MAEKAWA (Systems Engineering, Kobe University).

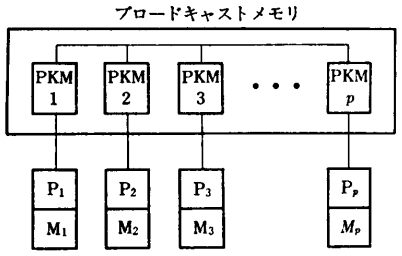
†† 神戸大学工学部システム工学科



○：演算プロセッサ

図 1 マトリクスブロードキャストメモリ結合形並列計算機システム

Fig. 1 Two dimensionally arranged multiple processor system with the matrix broadcast memory.



PKM：パケットメモリ
P　：演算部
M　：ローカルメモリ

図 2 ブロードキャストメモリの構成

Fig. 2 Structure of the broadcast memory.

```
for j:=k to n+l do
    a_kj:=a_kj/a_kk;
  for i:=k+1 to n do
    for j:=k to n+l do
      a_ij:=a_ij−a_ik*a_kj;
end
```

となる．

前進消去により $k-1$ 列まで消去が進んだ時点の A の状態を**図 3** に示す．$\{a_{ij}\}$ $(1\leq i\leq k-1,\ i\leq j\leq n+l)$ は消去操作によって確定した係数行列と定数項行列の部分行列である．$\{a_{ij}\}$ $(k\leq i\leq n,\ k\leq j\leq n+l)$ は消去操作の完了していない部分行列である．$n\leq p$, $n+l\leq q$ とし，a_{ij} を p_{ij} に割り当てることにすると，第 k 行をピボット行とする消去を進める場合の消去作業の並列化は以下のようになる（図 3）．a_{kk} は p_{kk} のローカルメモリに存在するので消去手順は以下のようになる．

ステップ 1：a_{kk}〜a_{nk} の値を p_{kk}〜p_{nk} が並行して行方向にブロードキャスティングする．

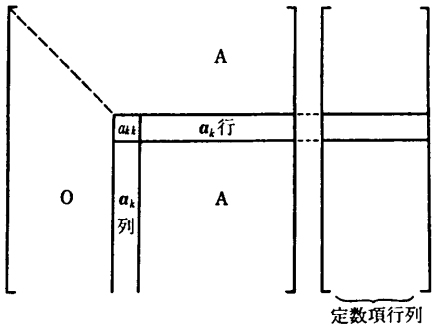


図 3 ガウス消去の前進消去過程

Fig. 3 Forward elimination on Gaussian method.

ステップ 2：$p_{k\,k+1}$〜$p_{k\,n+l}$ が並行して

$a_{kj}:=a_{kj}/a_{kk}$ $(k\leq j\leq n+l)$

を計算する．

ステップ 3：a_{kk}〜$a_{k\,n+l}$ の値を p_{kk}〜$p_{k\,n+l}$ が並行して列方向にブロードキャスティングする．

ステップ 4：各 p_{ij} $(k+1\leq i\leq n,\ k\leq j\leq n+l)$ が並行して

$a_{ij}:=a_{ij}-a_{ik}\cdot a_{kj}$

の計算を行う．

以上の四つのステップを $k=1$〜$n-1$ に対して順次行っていけば前進消去が完了する．各ステップに要する時間は n と独立に一定であり，かつステップ 1，3 におけるブロードキャストメモリの書込みにおいて競合が発生していないから計算時間は $O(n)$ となる．

一方，後退代入の手順は係数行列の対角要素がすべて 1 となっているからすべての t に対して

$x_{nt}:=a_{n\,n+t}$

とし，$i=n-1$ から 1 まで i を 1 ずつ減じながら

$$x_{it}:=a_{i\,n+t}-\sum_{j=i+1}^{n}a_{ij}x_{jt}$$

を計算することになる．

$x_{n\,t}, x_{n-1\,t}, \cdots, x_{1\,t}$ と順次値を計算していくことになるが，x_{it} $(1\leq t\leq l)$ の値は t に関して互いに独立なので t を固定して考える．解ベクトル $\boldsymbol{x}_t$ の計算は以下のステップを $k=n$〜1 に対して k を 1 ずつ減じながら実行していくことにより計算される．

ステップ 1：列 PE アレイ $\boldsymbol{p}_k$ が並行して $\boldsymbol{a}_k$ 列要素中の a_{ik} $(1\leq i\leq k-1)$ を行方向にブロードキャスティングする．

ステップ 2：$p_{k\,n+t}$ が

$x_{kt}:=a_{k\,n+t}$

としその値を列方向にブロードキャスティングする．

ステップ 3：列 PE アレイ p_{n+t} が並行して

$a_{i\,n+t}:=a_{i\,n+t}-a_{i\,k}x_{k\,t}$ $(1\leq i\leq k-1)$

の計算を行う．

いずれのステップでもブロードキャストメモリへの書込みの競合は発生していない．ステップ2,3に着目すると列 PE アレイ p_{n+t} のみ計算を担当している．

したがって $1\leq t\leq l$ に関して列 PE アレイ群 $p_{n+1}\sim p_{n+t}$ が並行して計算を進めることができその計算時間は $O(n)$ となる（**図4**）．

3.2 変形コレスキー法の場合

係数行列が正定値対称の場合に用いられる手法で，主要な計算は $Ax=B$ を変形し A を右上三角行列に変換する前進消去で，後退代入はガウス消去の場合とほぼ同一の手順となる．

前進消去の手順（**図5**）は a_1 行の値は変更せずに，$i=2\sim n$ について i を1ずつ増しながら $i\leq j\leq n+l$ なる全 j について，

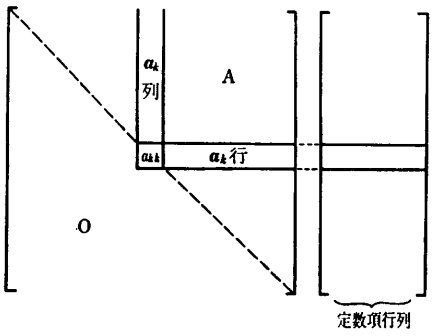


図 4 後退代入の手順
Fig. 4 Backward substitution method.

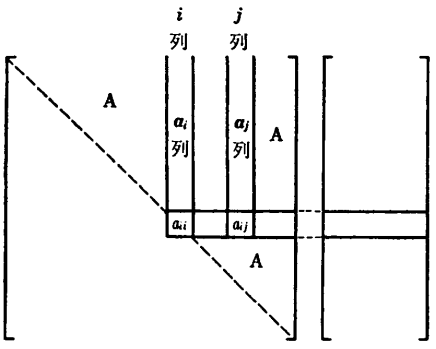


図 5 変形コレスキー法における前進消去
Fig. 5 Forward elimination on the modified Choleski method.

$$a_{ij}:=a_{ij}-\sum_{k=1}^{i-1}\frac{a_{ki}\cdot a_{kj}}{a_{kk}}$$

の値を計算することになる．

実際の計算は A と同一サイズの行列 A' を導入し，

$a_{1i}':=a_{1i}/a_{11}$ $(1\leq i\leq n+l)$

とし以下の計算を行う．

i を2から n まで1ずつ増しながら $i\leq j\leq n$ かつ $j\leq k\leq n+l$ なるすべての j,k に関して，

$a_{j\,k}:=a_{j\,k}-a'_{i-1\,j}*a_{i-1\,k}$

$a'_{ij}:=a_{i\,j}/a_{i\,i}$

を計算していくことになる．

プログラムの形で記述すると，

```
for i:=1 to n+l do a'1i:=a1i/a11;
  for i:=2 to n do
    for j:=i to n do
      begin for k:=j to n+l do
          ajk:=ajk-a'i-1 j * ai-1 k;
          a'ij:=aij/aii
      end
```

となる．a_{ij},a'_{ij} を p_{ij} のローカルメモリに割り当てておくと，以下のように並列計算可能となる．

$i-1$ 列まで消去が進み i 列の消去を行う計算は，

ステップ 1：$i-1$ 列の消去の段階で求められた a_{i-1} 行，a'_{i-1} 行の値を $p_{i-1\,i}\sim p_{i-1\,n+l}$ が並行して列方向にブロードキャスティングする．引き続きそれらデータを受けた p_{kk} $(i\leq k\leq n)$ が $a'_{i-1\,k}$ を行方向にブロードキャスティングする．

ステップ 2：ステップ1の操作により各 p_{jk} $(i\leq k\leq n+l,\ i<j\leq n)$ は $a'_{i-1\,j}$ と $a_{i-1\,k}$ がアクセス可能となるから並行して

$a_{jk}:=a_{jk}-a'_{i-1\,j}\cdot a_{i-1\,k}$

の計算を行う．

ステップ 3：p_{ii} が a_{ii} を行方向にブロードキャスティングする．

ステップ 4：行 PE アレイ p_i が並行して，$i\leq j\leq n$ について

$a'_{ij}:=a_{ij}/a_{ii}$

を計算する．

ステップ 1〜4 のいずれの計算時間も n と独立で一定であり，ブロードキャストメモリへの書込みの競合も発生しないから $i=2\sim n$ の計算時間は $O(n)$ となる．

3.3 $p<n, q<n+l$ の場合の並列計算

一般に大形連立一次方程式の場合には $p<n$,

$q<n+l$ となるのが普通である．この場合には各 PE は複数要素の計算を引き受けて行うことになる．この場合，各 PE の負荷のバランスを考え要素計算の割当を工夫し a_{ij} は $p_{(i-1) \bmod p+1\ (j-1) \bmod q+1}$ に担当させるようにする．

3.3.1 ガウス消去の場合

（i） 前進消去のステップ

第 $k-1$ 行まで消去が進み，第 k 行をピボット行として消去を進める場合を考える．$\boldsymbol{a}_k$ 列要素は列 PE アレイ $\boldsymbol{p}_{(k-1) \bmod q+1}$ に存在するので消去手順は以下のようになる．

ステップ 1：$\boldsymbol{a}_k$ 列の a_{kk} から a_{nk} の値を PE アレイ $p_{(k-1) \bmod q+1}$ が並行して行方向にブロードキャスティングする．

ステップ 2：行 PE アレイ $\boldsymbol{p}_{(k-1) \bmod p+1}$ が並行して $k \leq j \leq n+l$ なるすべての j について

$$a_{kj} := a_{kj} / a_{kk}$$

を計算する．

ステップ 3：新たに計算した a_{kj} $(k \leq j \leq n+l)$ の値を行 PE アレイ $\boldsymbol{p}_{(k-1) \bmod p+1}$ が並行して列方向にブロードキャスティングする．

ステップ 4：全 PE は並行して担当する全要素 a_{ij} $(k+1 \leq i \leq n,\ k \leq j \leq n+l)$ の更新を

$$a_{ij} := a_{ij} - a_{ik} \cdot a_{kj}$$

の計算によって行う．

以上の四つのステップを $k=1 \sim n-1$ に対して順次行えば前進消去が完了する．計算時間は 1 台のプロセッサが約 $n(n+l)/pq$ 要素担当するので $O(n^2(n+l)/pq)$ となる．

（ii） 後退代入のステップ

後退代入の手順についても基本的には 3.1 節で示した後退代入の手順と同一となる．

解ベクトル $\boldsymbol{x}_t$ の計算は $k=n \sim 1$ に関して k の値を 1 ずつ減じながらの以下の計算となる．

ステップ 1：列 PE アレイ $\boldsymbol{p}_{(k-1) \bmod q+1}$ が並行して a_{ik} $(1 \leq i \leq k-1)$ を行方向にブロードキャスティングする．

ステップ 2：$p_{(k-1) \bmod p+1\ (n+t-1) \bmod q+1}$ が

$$x_{kt} := a_{k\,n+t}$$

を求め値を列方向にブロードキャスティングする．

ステップ 3：列 PE アレイ $\boldsymbol{p}_{(n+t-1) \bmod q+1}$ が並行して

$$a_{i\,n+t} := a_{i\,n+t} - a_{ik} x_{kt} \quad (1 \leq i \leq k-1)$$

の計算を行う．

いずれのステップもブロードキャストメモリへの書込みの競合は発生しない．以上のステップは各 t $(1 \leq t \leq l)$ について並行に進めることができるから全体の計算時間は $l \leq q$ の場合には $O(n^2/p)$ 時間 $l>q$ の場合には $O(n^2 l/pq)$ 時間となる．

3.3.2 変形コレスキー法の場合

変形コレスキー法の場合もガウス消去法の場合と同様に並列計算が可能である．

いま i 行まで前進消去が進んだとすると次のステップは以下のようになる．

ステップ 1：$i-1$ 列の消去の終了で求まった $\boldsymbol{a}_{i-1}$ 行，$\boldsymbol{a}'_{i-1}$ 行の値を行プロセッサアレイ $\boldsymbol{p}_{(i-1) \bmod q+1}$ が並行して列方向にブロードキャスティングし引き続きそのデータを受けた対角プロセッサ群が $\boldsymbol{a}'_{i-1}$ 列の値を行方向にブロードキャスティングする．この操作により a_{jk} の計算を行うプロセッサが直接 $a'_{i-1\,j}$ と $a_{i-1\,k}$ とをアクセスできるようになる．

ステップ 2：各 PE が担当する

$$a_{jk} := a_{j\,k} - a_{i-1\,j} \cdot a_{i-1\,k}$$

の計算を $i \leq j \leq n+l$，$j \leq k \leq n$ なるすべての j，k について並行計算する．

ステップ 3：$p_{(i-1) \bmod p+1\ (i-1) \bmod q+1}$ により求められた a_{ii} を行方向にブロードキャスティングし行 PE アレイ $\boldsymbol{p}_{(i-1) \bmod p+1}$ の働きにより並行して $1 \leq j \leq n$ なる j について

$$a_{ij} := a_{ij} / a_{ii}$$

を計算する．

計算時間は $O(n^2(n+l)/pq)$ となる．後退代入はガウス消去の場合と同一手順となる．

3.4 帯行列の場合

係数行列が帯形で半帯幅が対角要素を含んで $m+1$ であるとする．$n>p, n+l>q$ とし各 PE への割当は 3.3 節に示した方式と同一とする．ガウス消去法変形コレスキー法とも 3.3 節に示したステップと同一となる．しかしながら計算過程において帯の外へフィルインが発生する可能性がないので，帯外の部分の計算はすべて省略できるから，計算量は大幅に減少する．計算時間のオーダは $p \geq m, q \geq m+l$ のとき前進消去，後退代入とも $O(n)$ であり，$p \ll m, q \ll m+l$ のとき $O\{nm(l+m)/pq\}$ であり，後退代入は $p \ll m$，$q \ll l$ のとき $O(nml/pq)$ となる．

3.5 疎行列の場合

大形の連立一次方程式がスパース係数行列をもっている場合はたいへん都合よく，ガウス消去法，変形コレスキー法いずれの場合にも計算時間および必要とする記憶容量とも削減できる．

この場合，各 PE には非零要素のみ格納するようにし，割り当てられた非零要素は二つのポインタ要素と値自身をもち，行方向および列方向のリンクリストの形で格納されているとする．消去過程でのフィルインはリンクリストに挿入され，消去された要素の領域は自由リスト領域にもどされ再利用される．この方式を採用することにより必要なメモリ容量は圧縮され，実際の計算も非零要素のみに行われることになる．

4. ブロードキャストメモリの制御と PE 間の同期

提案したシステムを実現するためにはブロードキャストメモリへのアクセスの競合に対する制御方式と，PE 間の同期方式が確立している必要がある．

まずブロードキャストメモリへの書込みの競合に対する制御であるが，本論文で提案したアルゴリズムではブロードキャストメモリへの書込みの競合は発生しない．

書込みと読出しとの競合は書込みオペレーションを優先するように設計しておけば競合によるブロードキャストメモリへの書込みの時間遅れは防止できる．

仮にブロードキャストメモリへの書込みの競合が発生する使い方を想定すると，リングアービタやディジーチェイン，多入力多出力形アービタを用いて制御する必要がある．この場合ブロードキャストメモリのポート数が多い場合には無視できないオーバヘッドが生じる可能性がある．

次に PE 間の同期であるが，全 PE が一つの共通クロックによって駆動されているとすると，PE 間の同期としては次の二つが考えられる．第1の方式は1台の PE（必ずしも固定する必要はない）が計算の主導権をとり同期信号用のデータを他の PE にマトリクスブロードキャストメモリを介して伝送し歩調を合わせながら計算を進めていく方式である．信号を受け取る側の PE は同期信号データを受け取ると事前に定められたステップだけ計算を進め次の同期信号を受けとるまで待ちの状態に置かれる．同期信号間の時間幅は全 PE が定められたステップの計算を十分進められるだけの時間幅を見込んでいるようにする．

第2の方式はより汎用性のある制御方式でセマフォアによる制御を行う．セマフォアは各ブロードキャストメモリ上に設けることができ，列 PE アレイまたは行 PE アレイが共同して同期に使うことができる．同一列または同一行上にない PE 間の同期は両 PE とセマフォアを共有できる PE を中継して行うことになる．第2の方式ではブロードキャストメモリへの書込みの競合が発生する可能性があるのでアービタ等による制御が必要となる．

5. 他の2次元アレイプロセッサとの比較

本提案システムと同様に PE を2次元平面に配置しそれらを相互結合した構成のアレイプロセッサですでに提案されているものがいくつかある．連立一次方程式の $O(n)$ 時間計算を期待できるものとして ILLIAC Ⅳ やシストリックアレイなどが存在する．

両者の構成上の特徴と問題点を上げると，

（i） ILLIAC Ⅳ…PE がマトリクス状に結合され4隣接 PE 間の直交接続となっている．SIMD 形の制御で実行する命令は全 PE 同一で外部からブロードキャストされる．各 PE はデータ格納用のメモリと命令の解読実行を行う演算要素から構成されている．SIMD 形なので柔軟性に欠けること，行列要素の PE への割当方法，非隣接 PE 間のデータのルーティングに問題がある．

（ii） シストリックアレイ…PE を2次元に配列し6隣接要素と接続したアレイで，PE は内部レジスタと簡単な演算機能をもった回路からのみ構成されている．各 PE にはプログラム実行（命令の解読実行）を行う機能はなく，全体としてパイプライン制御された多入力多出力の演算回路とみなすことができる．VLSI 向き回路として注目を集めているが大幅にサイズの異なる行列や疎行列に対して柔軟に対応することが困難である．

ブロードキャストメモリ結合形プロセッサシステムはこれらと比較して以下のような特徴をもつ．

（1） MIMD 形システムで PE として汎用マイクロプロセッサを無理なく使用できるので設計製作が容易である．また MIMD 形なので問題に柔軟に適応できる能力をもっている．

（2） （1）と関連するがサイズの大幅に異なる密行列帯行列，疎行列等の係数行列をもつ連立一次方程式に対しても無理なく対応でき効率的に解法計算が行える．とくに実用上重要な大形疎係数行列

をもつ連立一次方程式の計算に最も柔軟に適応し効率よく疎行列の特徴を生かした計算を進めることができる．

(3) 汎用マイクロプロセッサ数台から数十台を用いたシステムから，シストリックアレイに用いられるような単純な演算機能をもつ PE 数千台のシステムまで想定することが可能であり実現性が高い．

等が上げられる．

6. 結　　論

現在科学技術計算の高速化を目指したコンピュータで実用化されているものはパイプライン制御を導入したプロセッサで，並列プロセッサを用いたシステムで実用化されたものはほとんどない．しかしながらパイプライン制御方式でのスピード向上が限界に近づきつつあること LSI 技術の発展により同一プロセッサを多数用いた並列計算機システムがコストの面からも有利であることなどから今後は並列計算機による専用マシンの製作に研究の方向が向いていくと考えられる．本研究で示した手法は n 元連立一次方程式を $O(n)$ 時間で解く可能性を示したもので，従来の単一プロセッサ方式での $O(n^3)$ に比して飛躍的に短縮できる可能性を示したものである．

また帯係数行列をもつ連立一次方程式においては，半帯幅 m に対して $p \times q$ 台のプロセッサで $p \leq m$, $q \leq m+l$ のとき $O(nm(m+l)/pq)$ 時間で計算できることを示しており実用上きわめて興味深い結果と考えられる．

ブロードキャストメモリはバス構成が単純で，p, q の値が数十程度になっても比較的容易に実現可能である．複数 PE を 1 チップ化すれば PE が数百から数千のシステムを作ることも可能であると考えられる．

今後は実験システムの試作を行い有効性を実証していくことが課題である．

参 考 文 献

1) 戸川：マトリクスの数値計算，オーム社，東京 (1971)．
2) 金田：並列処理システムによる連立一次方程式と楕円形偏微分方程式の数値計算法，情報処理，Vol. 16, No. 2, pp. 122-129 (1975)．
3) 金田：環状結合型超多重プロセッサシステムによる大次元連立一次方程式の並列計算，情報処理学会論文誌，Vol. 21, No. 5, pp. 402-406 (1980)．
4) 小林，相磯他：離散系シミュレータ KDSS-I の設計と試作，情報処理学会第 21 回全国大会講演論文集 (1980)．
5) 小高，川辺：超高速演算の動向，情報処理，Vol. 21, No. 1, pp. 927-937 (1980)．
6) 金田：ブロードキャスト・メモリを持つ並列計算機システムによる大次元連立一次方程式の並列計算，電子通信学会電子計算機研究会，EC80-42, pp. 41-45 (1980)．
7) 小畑，金田，前川他：ブロードキャスト・メモリ結合形並列計算機の試作，電子通信学会電子計算機研究会，EC81-37, pp. 9-17 (1981)．
8) Kung, H. T. and Leiserson, C. E.: Systolic Arrays (for VLSI), Proc. Symp. Sparse Matrix Computations and Their Applications, 2-3, pp 256-282 (1978)．

(昭和 56 年 12 月 10 日受付)
(昭和 57 年 4 月 19 日採録)