マトリクスブロードキャストバス結合形並列プロセッサによる軸選択形ガウス消去の並列計算法†

金田 悠紀夫†† 小畑 正貴††† 角木 裕成††††

2次元のマトリクス状に結合された並列プロセッサシステムを用いてガウス消去により n 元の連立一次方程式を $O(n)$ 時間で解く計算法については報告されているが，消去の過程で軸選択を行っていないという欠点がある．筆者らが以前提案したマトリクスブロードキャストメモリ結合形並列計算機および BC プロセッサアレイを用いたガウス消去の並列計算法に拡張を加え，消去時に軸列上での軸選択を導入した並列計算法を提案する．本計算法では n 元の非帯係数行列をもつ連立一次方程式の場合 $O(n\log_2 n)$ 時間，半帯幅 p の場合 $O(n\log_2 p)$ 時間でガウス消去計算が実現される．

1. はじめに

電子計算機のもつ数値計算能力の飛躍的な向上に伴い，計算科学とでも呼ぶべき数値計算技法を中核とした学問分野が開拓され，今後ますます発展をとげようとしている．計算科学における多くの問題は大型の線形方程式の問題に帰着させることが可能で，大型線形方程式の高速計算法の開発がきわめて重要な課題となっている．

並列計算の技法を導入しないとすると，直接法（ガウス消去法や修正コレスキー法）を用いた場合，係数行列が n 元の連立方程式を解くのに非帯係数行列の場合には $O(n^3)$ 時間，帯幅 m の帯係数行列の場合には $O(nm^2)$ 時間を必要とすることが知られている．また並列計算法を用いた場合には，$n\times n$, $m\times m$ のマトリクス状に配置されたプロセッサアレイを用いた並列計算により，いずれも $O(n)$ 時間で計算可能なことが知られている[1)~4)]．

しかしながら，これらの計算法では消去計算の過程での軸選択（ピボッティング）を行っておらず，軸要素が0となったり，0にきわめて近い値をもったときの対策が講じられていない問題点がある．

本論文は，文献3)，4) で提案した並列プロセッサアレイでの計算方式に若干の拡張を加えることにより軸選択操作を消去計算過程に組み込むことが可能なことを示し，軸列上での軸選択を導入した場合でも，その計算時間を，係数行列が非帯の場合には $O(n\log_2 n)$，対角からの列方向の帯幅が p の帯行列の場合には $O(n\log_2 p)$ とすることが可能なことを示す．

2. マトリクスブロードキャストバス結合形並列計算機

図1に提案するマトリクスブロードキャストバス結合形並列計算機システムの構成例を示す．本システムは文献3) で著者らが提案した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において用いたブロードキャストメモリをブロードキャストバスに置き換えたものでシステム構成を簡略化したものである．

本ブロードキャストバスは図2に示すように p 台のプロセッサを一次元のアレイとして結合する1本の共通バスである．p 台のプロセッサのうち1台しか一時にはデータをバス上に出力することができないように制御されているが，バス上のデータの入力は同時に複数台のプロセッサが行うことが可能となっている．したがってデータの放送機能をもつことになり，離れた位置に存在するプロセッサに対しても短時間でデータを転送することが可能である．

図1に提案したシステムは，$p\times q$ 台のプロセッサをマトリクス状に配置し行方向に p 本，列方向に q 本の計 $p+q$ 本のブロードキャストバスで結合したもので i 行 j 列の位置にあるプロセッサを P_{ij} と呼ぶことにする．

本システム上にサイズ $p\times q$ の行列 $A(\{a_{ij}\})$ を P_{ij} に a_{ij} を割り付けるようにして格納する．

A に対する演算

（i） i 行と j 行の入替え

---

† Parallel Gaussian Elimination with Partial Pivoting on a Matrix Broadcast Bus Connected Parallel Processor System by YUKIO KANEDA (Systems Engineering, Kobe University), MASAKI KOHATA (Graduate School of Science and Technology of Kobe University) and HIRONARI KAKUKI (Sharp Co., Ltd.).

†† 神戸大学工学部システム工学科
††† 神戸大学大学院自然科学研究科
†††† シャープ(株)

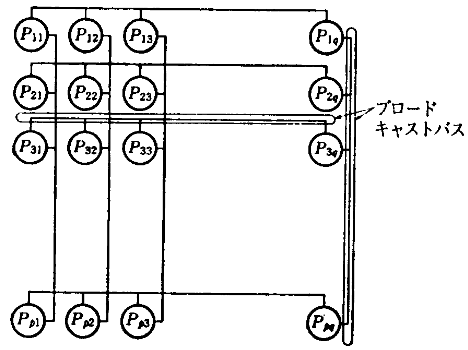

図 1 マトリクスブロードキャストメモリ結合形並列計算機システム
Fig. 1 The structure of the matrix-broadcast memory connected parallel processor system.

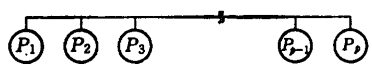

図 2 ブロードキャストバス結合形一次元プロセッサアレイ
Fig. 2 A broadcast-bus connected one dimensional processor array.

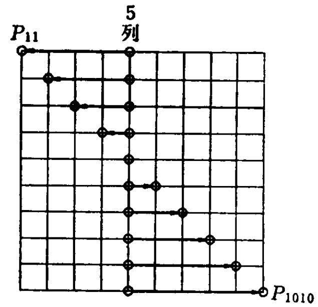

図 3 対角プロセッサへのデータ転送
Fig. 3 Data transfer to the diagonal processors.

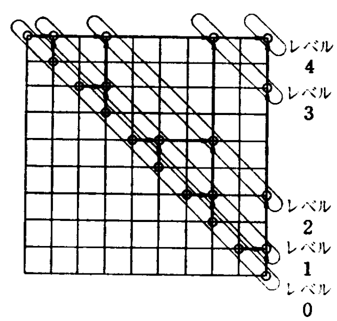

図 4 2分サーチの手順
Fig. 4 Procedure of binary search.

(ⅱ) i 列要素 (a_{1i}〜a_{pi}) 中の絶対値最大のものの探索

は以下の手順で並列計算可能である.

(ⅰ)の場合

P_{i1}〜P_{iq} が並行して列方向のブロードキャストバスを介して a_{i1}〜a_{iq} を P_{j1}〜P_{jq} に転送する. 引き続き P_{j1}〜P_{jq} が a_{j1}〜a_{jq} を列方向のブロードキャストバスを介して P_{i1}〜P_{iq} に転送する. 以上の2回の転送により i 行と j 行の入替えが可能となる.

(ⅱ)の場合

① j 列要素 a_{1j}〜a_{pj} を行ブロードキャストバスを介して行方向に転送し対角上のプロセッサ P_{11}, $P_{22}, \cdots, P_{pp}$) 群へ並行転送する (**図 3**).

② 2分探索法を用いて絶対値最大のものを探す. 対角プロセッサを葉ノードとする2分木を**図 4** のように構成し, 各枝ノードと根ノードに対応するプロセッサを定めそれぞれレベル0のプロセッサ群, レベル1のプロセッサ群, …レベル $\lceil \log_2 p \rceil$ のプロセッサ群と呼ぶことにする.

2分探索の計算は以下の手順となる.

ステップ 1: レベル0のプロセッサ群はそれぞれのもつ要素をレベル1のプロセッサ群に, 行方向または列方向のブロードキャストバスを介して転送する. 二つのデータが転送されてきたレベル1のプロセッサは両データの比較を行い絶対値の大きいほうをレベル2のプロセッサにブロードキャストバスを介して転送する. また一つのデータのみが転送されてきたプロセッサはその値自身をレベル2のプロセッサにブロードキャストバスを介して転送する.

⋮

ステップ i: レベル i のプロセッサ群はそれぞれレベル $i-1$ のプロセッサ群から送られてきたデータを受け取り, 2入力データを受け取ったプロセッサは絶対値の大きいほうのデータを, 1入力データを受け取ったプロセッサはそのデータ自身を $i+1$ レベルのプロセッサ群にブロードキャストバスを介して転送する.

⋮

ステップ $\lceil \log_2 p \rceil$: レベル $\lceil \log_2 p \rceil$ のプロセッサはレベル $\lceil \log_2 p \rceil - 1$ のプロセッサ群から送られてきた二つのデータを受け取り, そのうち絶対値の大きいほうのデータを選択する.

上述の計算手順を実行することにより, 絶対値最大の要素を探索することが可能で, 必要とする計算時間は $O(\lceil \log_2 p \rceil)$ となる.

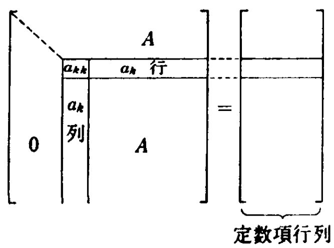

図 5　前進消去の計算手順（k 段の消去）
Fig. 5　The forward elimination (kth step).

3. 軸列上での軸選択形ガウス消去の並列計算

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, 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.

まず A が非帯行列の場合について考える.

ガウス消去には前進消去と後退代入の手順があり, 軸選択が行われるのは前進消去の手順である.

2次元プロセッサアレイとして $n\times(n+l)$ のものを想定し, 各 a_{ij} は P_{ij} に割り当てられているものとする.

ここでは軸選択に着目して前進消去のアルゴリズムの並列化について論ずる.

図 5 に示すように第 $k-1$ 列までの前進消去が終わり第 k 列上の軸選択を行う場合を考える.

軸選択は次の二つのステップで実現される.

(i)　k 列要素 $a_{kk}\sim a_{nk}$ のうちで絶対値最大の要素 a_{jk} を見つける.

(ii)　k 行と j 行を入れ替える.

(i)に示したステップの演算は $n+1-k$ 個の要素から絶対値最大の要素を探索する演算となり, 2章で示した2分探索法を用いると, $\lceil\log_2(n+1-k)\rceil$ 時間で実行できる.

(ii)に示したステップは行の入替えの演算で2章(i)で示した手順によりブロードキャストバスを用いた2回のデータ転送で実現をすることができる.

したがって前進消去計算は

```
for  k := 1  to  n-1  do
  begin
    {(i)  k 列要素 akk〜ank のうち絶対値最大の要
         素 ajk を探索する};
    {(ii)  k 行と j 行の入替えを行う};
    {(iii)  k 行を軸として k 列の消去計算を進める}
  end
```

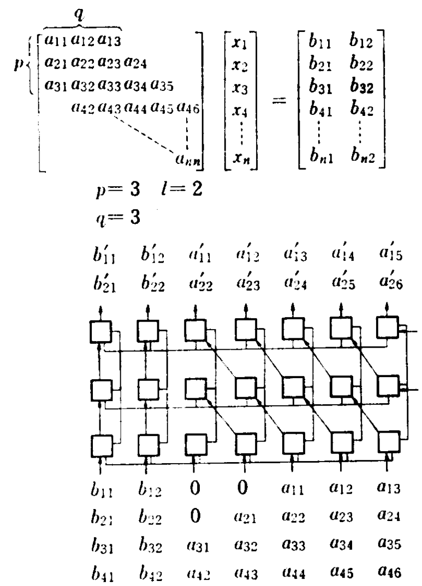

図 6　BC プロセッサアレイ上での軸選択形ガウス消去計算
Fig. 6　Gaussian elimination with partial pivoting on the BC processor array.

となる.

(i)の計算は各 k の値について, $\lceil\log_2(n-k+1)\rceil$ 時間必要とし, (ii)および(iii)は k の値にかかわらず一定時間で, (iii)の計算法については文献3) で論じた. したがって全体としての計算時間は $O(n\log_2 n)$ 時間で抑えられることになる.

4. BC プロセッサアレイ上での帯ガウス消去計算

次に帯係数行列をもつ連立一次方程式に列方向の軸選択を伴うガウス消去計算を適用する場合を考える. 帯行列に対する前進消去の過程で列方向の軸選択を行っても, 帯幅の拡大はたかだか2倍までしか拡がらないので消去の過程で帯行列の性質が失われてしまうということがないというすぐれた性質をもっている.

ここでは文献4) で提案した BC プロセッサアレイ上での軸選択を伴った帯ガウス消去の並列計算法について論じる. 前章までで論じた手法の BC プロセッサアレイへの導入方法について検討してみる.

$Ax=B$ に対して前進消去を行う過程に着目する.

A は**図 6** に示すように $n\times n$ のサイズで帯幅 $m=p+q-1$ の帯行列とし B を $n\times l$ の密行列とする.

図 6 に BC プロセッサアレイによるガウス消去実行の例が示してある. 計算は図 6 に示すように B に関しては $P_{p1}\sim P_{pl}$ の入力端子に下方から1行ずつ入

力され上方に伝播させていく．行列 A のほうは帯部分が $P_{p,l+1}$〜$P_{p,l+m}$ の入力端子に1行ずつ入力され斜上方に伝播されていく．$P_{1,l+m}$〜$P_{p-1,l+m}$ の入力端子には0が入力されていく．

計算は p ステップまで伝播が進んでから $k=1$〜n に対して以下の手順で実行される．

(i) $P_{1,l+1}$〜$P_{p,l+1}$ 上の係数を行ブロードキャストバスを介して対角プロセッサ群 $P_{1,l+1}$, $P_{2,l+2}$, …, $P_{p,l+p}$ に転送する．

(ii) 2章で示した2分探索の手法を用いて $P_{1,l+p}$ 上に絶対値最大要素を求め，その値が存在していたプロセッサ $P_{i,l+1}$ を求める．

(iii) 第1行の要素と第 i 行の要素を列方向のブロードキャストバスを介して入れ替える．

(iv) $P_{1,l+1}$ が a_{kk}^{-1} の値を計算し行 BC 端子に出力し行方向に放送する．

(v) プロセッサアレイの第1行が

$$a_{kj}=a_{kj}/a_{kk}(k+1\leq j\leq k+m)$$

$$b_{kj}=b_{kj}/a_{kk}(1\leq j\leq l)$$

の計算を行い a_{kj}, b_{kj} の値を列 BC 端子に出力して列ブロードキャストバスを介して列方向に放送する．

(vi) 各プロセッサが a_{ij} および b_{ij} の更新を

$$\begin{cases} a_{ij}=a_{ij}-a_{ik}\cdot a_{kj} \\ b_{ij}=b_{ij}-a_{ik}\cdot b_{kj} \end{cases}$$

を行う．a_{ik} は行ブロードキャストバスより受け取る．

(vii) a_{ij} を左斜上方に，b_{ij} を上方に伝播させ，$P_{1,l+m}$〜$P_{p-1,l+m}$ に0を入力する．

(i)，(iii)，(iv)，(v)，(vi)，(vii) の計算時間は p, q の値とは独立で一定時間となり，(ii) の計算時間は p によって定まり $\lceil \log_2 p \rceil$ 時間となる．

したがって全体の計算時間は $O(n\lceil \log_2 p \rceil)$ 時間で抑えられることとなる．

5. 結　論

2次元プロセッサアレイやシストリックアレイを用いて n 元連立一次方程式を $O(n)$ 時間で解く手法については従来から研究が進んでいるが軸選択を消去過程にいかに組み込むかが大きな問題であった．本研究はブロードキャストバスのもつ放送機能と2次元マトリクス状に配置されたプロセッサアレイの構造上の特徴を有効に活用することにより，行選択と行入替えを伴うガウス消去計算を係数行列が非帯の場合で，$O(n \log_2 n)$ 時間，列方向の帯幅が p の帯行列の場合には $O(n \log_2 p)$ 時間で計算できることを示した．

参 考 文 献

1) Kung, H. T.: The Structure of Parallel Algorithm, *Advances in Computers*, Vol. 19, pp. 65-112, Academic Press, New York (1980).
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，小畑，前川：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，情報処理学会論文誌，Vol. 23, No. 5, pp. 570-575 (1982).
4) 金田，小畑，前川：BC プロセッサアレイと高並列マトリクス計算，情報処理学会論文誌，Vol. 24, No. 2, pp. 175-187 (1983).
5) 小畑，金田，前川：ブロードキャストメモリ結合形マルチマイクロプロセッサシステムの試作，情報処理学会論文誌，Vol. 24, No. 3, pp. 351-356 (1983).

(昭和58年9月5日受付)
(昭和58年11月15日採録)