Vol. 42 No. SIG 9(HPS 3) 情報処理学会論文誌：ハイパフォーマンスコンピューティングシステム Aug. 2001

非対称三重対角行列向けの並列連立一次方程式解法

山本有作[†] 猪貝光祥[††] 直野健[†]

本論文では，領域分割法を改良した非対称行列向けの並列三重対角ソルバのアルゴリズムを提案する．従来の領域分割法は，軸選択を行った場合に部分領域間の消去の独立性が失われるため，一般の非対称三重対角行列に適用することが困難であった．本アルゴリズムでは，部分領域の境界に隣接する節点について番号の付け替えを行うことにより，部分軸選択を行った場合でも部分領域間の消去の独立性を保証する．これにより，部分軸選択付きの LU 分解を並列に行うことが可能となる．SR8000/F1 の 1 ノード（8 プロセッサの共有メモリ型並列機）による評価では，8000 元の非対称三重対角行列の LU 分解において，本手法は従来の部分軸選択付き逐次型三重対角ソルバの 5.5 倍の高速化を達成した．

A Parallel Linear Equation Solver for Nonsymmetric Tridiagonal Matrices

Yusaku Yamamoto,[†] Mitsuyoshi Igai[††] and Ken Naono[†]

We propose a new parallel solver for nonsymmetric tridiagonal matrices, which is an improvement over the dissection method. The conventional dissection method is difficult to apply to a general nonsymmetric tridiagonal matrix, because the independence of decomposition operations in each subdomain is lost when pivoting is introduced. In our algorithm, due to the reordering of the nodes adjacent to the boundary nodes, the independence of decomposition operations in each subdomain is guaranteed even when partial pivoting is introduced. Thus, the LU decomposition of the whole matrix with partial pivoting can be done in parallel. We evaluated our algorithm on 1 node of the SR8000/F1 (a shared-memory parallel computer with 8 processors) and obtained speedup of 5.5 times compared with the conventional sequential tridiagonal solver with pivoting, when computing the LU decomposition of a nonsymmetric tridiagonal matrix of order 8000.

1. はじめに

三重対角行列を係数とする連立一次方程式の求解は，対称行列の固有ベクトル計算のための逆反復法[1)]，偏微分方程式を解くための ADI 法[2)]，スプライン補間の計算[3)]などをはじめとして，科学技術計算で広く利用される計算の 1 つである．三重対角行列のサイズが大きい場合，高速に解を求めるには，並列計算機の利用が必要となる．並列計算機を用いて三重対角行列を係数とする連立一次方程式を解くための方法としては，従来，領域分割法[4)]，巡回縮約（Cyclic Reduction）法[7),10)]，ETC （Ends Toward the Center）順序に基づく方法[5)]，QR 分解に基づく方法[12)] などが提案されてきた．

† 株式会社日立製作所中央研究所
Central Research Laboratory, Hitachi Ltd.
†† 株式会社日立超 LSI システムズ
Hitachi ULSI Systems Corp.

このうち巡回縮約法は，方程式の奇数番目の変数を消去して半分の元数の方程式を作るという操作を再帰的に繰り返して方程式を解く方法であり，奇数番目の変数それぞれについて独立に消去を行えるため，きわめて高い並列性を持つ．そのため巡回縮約法は，ベクトル計算機向けライブラリ[9)]をはじめとして多数の実装がなされており[8)]，ブロック三重対角行列[10)]，帯行列[14)]などへも拡張されている．しかしこの手法では，変数の消去の順番があらかじめ決まっており，数値的安定化のための軸選択を取り入れることが困難である．そのため，一般の非対称三重対角行列に対しては適用できず，適用対象は主に対角優位の行列に限定されていた．

領域分割法および ETC 順序に基づく方法は LU 分解あるいはコレスキー分解に基づく解法であるが，これらも軸選択を行わないことを前提として分解演算の並列性を抽出しており，適用対象の行列について巡回縮約法と同様の制限がある．

一方，QR 分解に基づく方法[12]は，係数行列が正則でありさえすれば解を求めることが可能である．しかし行列が特異に近い場合には，解の精度を高めようとするとやはり軸列の選択が必要なことが知られており[13]，軸選択を行わない並列解法は適用範囲に限界がある．

そこで本論文では領域分割法を改良し，部分軸選択を行うことが可能な非対称三重対角行列向けの並列解法を提案する．これにより，一般の非対称三重対角行列を係数とする連立一次方程式を並列に解くことが可能となる．

なお，一般の非対称三重対角行列に対する軸選択付きの並列解法文献は文献 11) でも提案されている．しかしこの方法では，逐次的に分解しなくてはならない最後の小行列のサイズを $2p$，解法全体でのプロセッサ間同期の回数を q とすると，$pq = N$（N は方程式の元数）というトレードオフの関係がある．これに対して本論文で提案する並列解法は，同期は解法全体を通して 1 回で済み，逐次的に分解すべき小行列のサイズは N によらずにプロセッサ台数のみで決まるという利点を持つ．

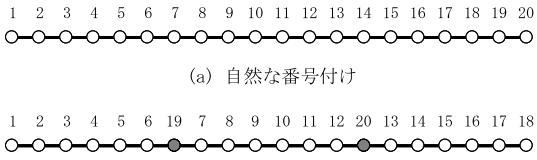
以下では，まず 2 章で領域分割法による三重対角行列の並列求解アルゴリズムを説明し，一般の非対称行列に適用した場合の問題点を述べる．次に，3 章で部分軸選択が可能な新しい並列解法を提案する．4 章では SR8000 の単一ノード（8 プロセッサの共有メモリ型並列）を用いて，その性能と精度の評価を行う．最後に 5 章でまとめと今後の課題を述べる．

2. 領域分割法とその問題点

2.1 領域分割法による三重対角解法

いま，三重対角行列 T を係数とする N 元連立一次方程式 $T\mathbf{x} = \mathbf{b}$ を直接解法で解く場合を考える．領域分割法では，適当な置換行列 P を用いて T を $T' = PTP^t$ に変換することにより並列性を抽出する．そこで，まず置換の記述に便利な行列のグラフを導入する．行列 A が N 次の正方行列で，かつ非零要素の位置が対称であるとする．このとき，A の各列に対応する 1 から N までの節点を持ち，$A_{ij} \neq 0$ のときに限り節点 i と節点 j 節点の間に枝を持つ無向グラフを，行列 A のグラフ G_A と呼ぶ．定義より，T のグラフは N 個の節点を直列に結んだ**図 1** (a) のグラフとなる．また，行列に対する行と列の同時置換 $T' = PTP^t$ は，T のグラフ G_T では節点番号の付け替えに対応する．

領域分割法では，p 台のプロセッサで $T\mathbf{x} = \mathbf{b}$ を並列に解く場合，G_T を $p-1$ 個の境界節点により大きさがほぼ等しい p 個の部分領域に分割する．そして，第 1 の部分領域に属する節点，第 2 の部分領域に属する節点，...，第 p の部分領域に属する節点の順に番号を付け直し，最後に $p-1$ 個の境界節点に $N-p+2$ から N までの番号を付ける．$p = 3$ の場合について，番号の付け替えを行った後のグラフを図 1 (b) に示す．影の付いた節点が境界節点である．



(a) 自然な番号付け

(b) 領域分割法による番号付け

図 1 三重対角行列のグラフ
Fig. 1 A graph associated with a tridiagonal matrix.

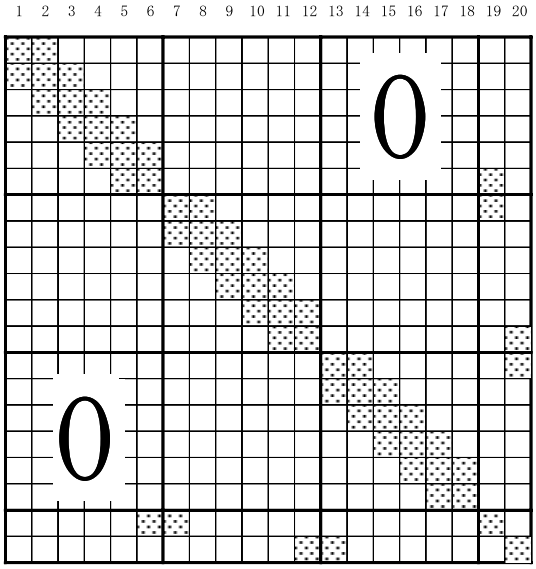


図 2 領域分割法により置換を行った行列
Fig. 2 A tridiagonal matrix permutated by the nested dissection method.

また，対応する行と列の置換を行った行列を**図 2** に示す．影のついた四角が非零要素である．図より，三重対角行列 T が 3 個の対角ブロックを持つ縁付きブロック対角行列へと変形されることが分かる．軸選択を行わない場合，各ブロック内部の分解は独立に行えるため，領域分割法では三重対角行列の求解を p 個のプロセッサを用いて並列に行うことが可能となる．

2.2 非対称行列の場合の問題点

非対称行列の LU 分解を行う場合，一般には数値的安定性を保つため，軸選択を行うことが必要である[6]．

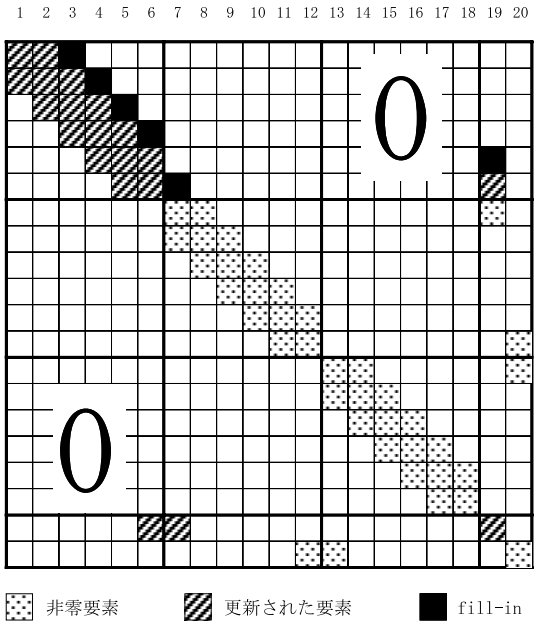


図 3 第 6 列の消去終了時の非零パターン
Fig. 3 Nonzero structure after elimination by the 6-th column.

軸選択の方法としては，ピボット列の中で絶対値最大の要素をピボット要素として選択する部分軸選択が広く使われる．そこで本節では，前節で述べた領域分割法による並列解法に部分軸選択を取り入れた場合の並列性について考察する．

いま，行と列の同時入替えにより図 2 の形に変形した行列に対して部分軸選択をともなう LU 分解を適用することを考える．このとき，第 1 の部分領域に属する節点に対応する列（最初の 6 列）の消去が終わった段階での行列の非零パターンを**図 3** に示す．なお，ここでいう非零パターンとは，第 6 列の消去終了時における非零パターンの，あらゆる可能な部分軸選択を行った場合にわたっての和集合である．また図中では，第 6 列の消去終了時までに更新された要素を斜線を引いた四角で表し，零から非零になった要素（fill-in）を黒い四角で表した．

図より，第 6 列の消去によって，第 7 列の最後から 2 行目の要素が更新されていることが分かる．ところが第 7 列の消去では，この要素も含め，第 7 列の対角以下の要素の間で軸選択を行う．そのため，軸選択を行う場合には，第 6 列の消去でこの要素の値が確定してからでないと第 7 列の軸選択が行えない．したがって，この領域分割法では，軸選択によって第 1 の部分領域（第 1 列〜第 6 列）の消去と第 2 の部分領域（第 7 列〜第 12 列）の消去との間に依存関係が生じ，並列性が失われてしまうことが分かる．

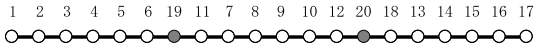


図 4 本研究の手法による番号付け
Fig. 4 Reordering of the nodes by the proposed method.

3. 軸選択が可能な並列解法

3.1 原　　理

そこで本研究では領域分割法を改良し，軸選択を可能にした新しい並列解法を提案する．

前章の例において，第 1 の部分領域の消去と第 2 の部分領域の消去との間に依存関係が生じたのは，第 7 列の最後から 2 行目の非零要素の存在によるが，この要素が非零であるのは，第 1 の部分領域の右端の節点と第 2 の部分領域の左端の節点とが境界節点を通して接続されているためである．そこで本研究で提案するアルゴリズムでは，領域分割法で番号付けを行った後，各部分領域に属する節点の中でさらに番号の付け替えを行う．具体的には，境界節点に隣接しない節点の番号を 1 ずつ繰り上げ，代わりに領域左端の節点の番号を領域右端の節点の番号より 1 だけ小さい値にする．図 1 に示したグラフに対し，この方法による番号付けを行った結果を**図 4** に示す．

この番号付けに対応する行と列の置換を行った行列を**図 5** に示す．領域左端の点に部分領域内で最後から 2 番目の番号を付けたことにより，依存関係が生じる原因となっていた要素が，第 2 の部分領域の最後から 2 番目の列に移動している．

3.2 消去演算における並列性

図 5 において，行列の行および列を各部分領域に対応してブロックに分け，各ブロックごとに非零要素が 1 個でもある部分は影付き，全部が零要素の場合は空白とした結果を**図 6** に示す．ただし列に関しては，境界節点に隣接する節点に対応する列を別ブロックとしてある．この図において列ブロック B に着目すると，そのうちで非零要素を含むのは上から 2 番目のブロックだけであり，かつ，行列中でこのブロックの左側の部分はすべて零となっている．したがって，列ブロック B より左側の列に関する消去によって，列ブロック B に含まれる要素はいっさい影響を受けないことが分かる．同様に，列ブロック C のうちで非零要素を含むのは上から 3 番目のブロックだけであり，かつ，行列中でこのブロックの左側の部分はすべて零である．したがって，列ブロック C より左側の列に関する消去によって，列ブロック C に含まれる要素はいっさい影響を受けない．このことより，列ブロッ

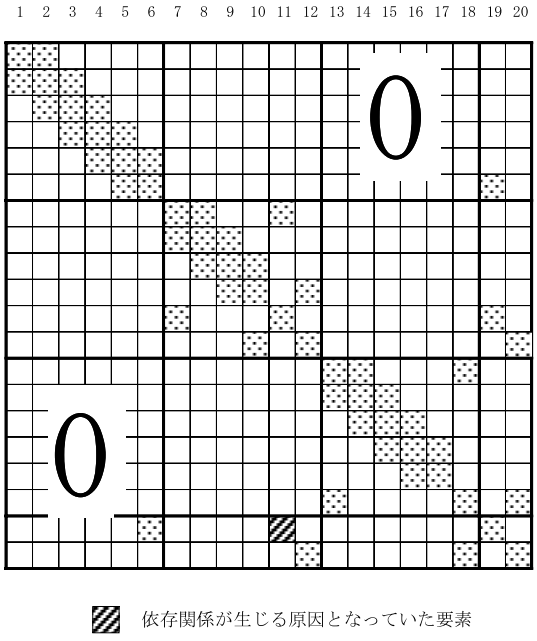


図 5 本研究の手法により置換を行った行列
Fig. 5 A tridiagonal matrix permutated by the proposed method.

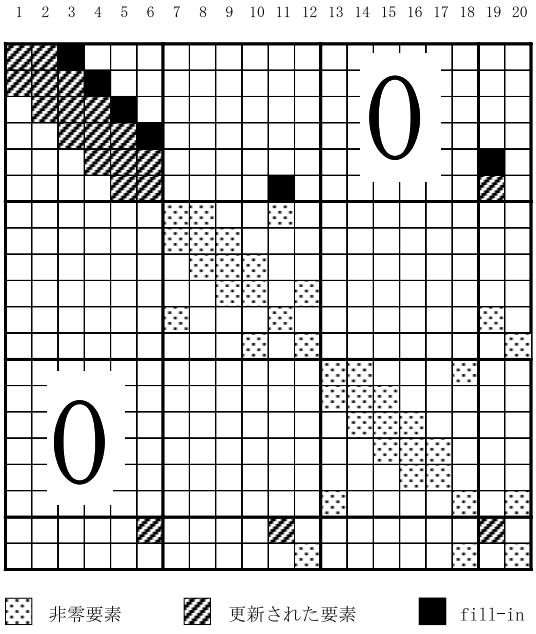


図 7 第 6 列の消去終了時の非零パターン
Fig. 7 Nonzero structure after elimination by the 6-th column.

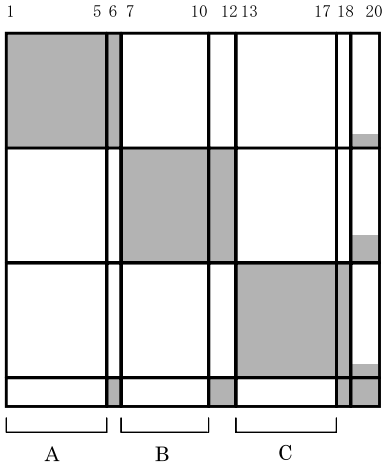


図 6 非零パターンのブロック構造
Fig. 6 Block structure of nonzero elements.

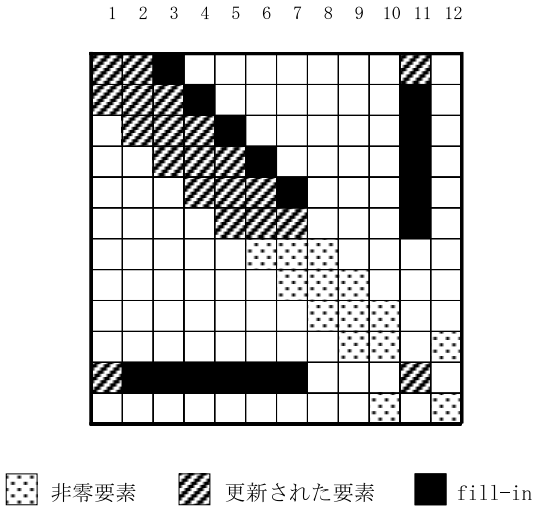


図 8 列ブロック B の消去途中の非零パターン
Fig. 8 Nonzero structure during elimination by columns in block B.

ク B に含まれる列の消去，列ブロック C に含まれる列の消去は，それぞれそれ以前の列の消去と独立に行えることが分かる．

図 5 の行列に対し，第 6 列目まで（B ブロックの直前まで）の消去を行ったときの非零パターンを**図 7** に示す．図より，列ブロック B（第 7 列〜第 10 列）の要素は，この消去により影響を受けないことが分かる．したがって，この順序付けによれば，3 個の列ブロック A，B，C の消去を 3 台のプロセッサで並列に行うことが可能である．

また，列ブロック B（境界節点に隣接する節点も含む）における消去途中の非零パターンを**図 8** に示す．ただし本図では，分かりやすくするため，ブロックに属する列を 12 本に拡大し，ブロック内の 5 番目の要素による消去が終了した時点での非零パターンを示した．一般に軸選択付きの三重対角行列の LU 分解

では，主対角成分の 2 個上に対角線に沿った fill-in が現れるが，本手法ではこれに加えて，ブロック内で下から 2 行目および右から 2 列目にも fill-in が生じることが分かる．これにより，第 K 段で消去演算に関与する要素は，逐次型の三重対角ソルバでは第 K 行と第 $K+1$ 行，および第 K 列から第 $K+2$ 列までの計 6 個であったのが，行・列ともに 1 つずつ増え，12 個となる．したがって，消去のための演算量はその分だけ増加する．また，各段で軸選択を行うにあたっての候補は，対角要素，その 1 個下の要素に加え，前段までの fill-in によって生じるブロック内で下から 2 行目の要素も候補に加わり，3 個となる．

なお，以上ではプロセッサ台数が 3 台の場合について説明を行ったが，本手法による並列化は，任意のプロセッサ台数に対して適用可能である．本手法では，プロセッサ間の同期は，各プロセッサが担当する列ブロックの消去が終了した時点 1 回のみで済む．また，逐次的に分解する必要がある右下部の小行列のサイズは，境界節点の数と同じであり，プロセッサ台数− 1 となる．

3.3 改良の可能性

本解法では，行列のグラフ G_T に対する節点番号の付け方を改良することにより，軸選択を行った場合でも並列性を保てる解法を実現した．これは，係数行列の行と列に対して同じ置換 $T' = PTP^t$ を行うという条件の下で係数行列を変形することに相当する．この方法は，領域分割法のブロック対角構造を保てる点，対角ブロック内部での消去後の非零構造が三重対角行列の分解結果 + 縦横 1 列ずつの fill-in で与えられ，計算量の増加がそれほど大きくない点などのメリットを持つ．しかし，係数行列が非対称行列であることを考慮すると，行と列に対して異なる置換を適用することも原理的には可能であり，これによりさらに並列性を高められる可能性もある．この点についての検討は今後の課題である．

4. 性能評価

4.1 実装方式

本手法の有効性を評価するため，SR8000/F1 の 1 ノード上で，従来の軸選択付き三重対角ソルバとの性能比較を行った．SR8000 の 1 ノードは，8 個の RISC 型プロセッサからなる共有メモリ型並列機であり，各プロセッサは 1.5GFLOPS，1 ノードでは 12GFLOPS のピーク性能を持つ．

本手法で計算を行うにあたっては，三重対角行列のグラフ G_T をほぼ同じ大きさを持つプロセッサと同数個の領域に分割し，各領域での消去をそれぞれ 1 個のプロセッサに担当させた．また，境界節点および境界節点に隣接する節点の消去は，最後にまとめて 1 個のプロセッサで行った．行列は密行列形式で格納し，ピボット選択にあたっては，3.2 節および図 8 で示したように，ピボット候補が 3 個に限定され，かつ位置も規則的に求められることを利用して，各段ごとにプログラム中で位置を計算してアクセスを行った．なお，並列化は SR8000 の並列化ディレクティブである*POPTION を用いて行い，各領域での消去をサブルーチン化してそれぞれ 1 個のプロセッサに実行させた．

一方，従来の三重対角ソルバは並列化可能部分がないため，計算は 1 プロセッサのみで行った．行列は，fill-in の部分も含め 4 本の対角線を，大きさ $N \times 4$ の配列に格納した．

なお，すべての実数演算は倍精度で行った．また実行時間の測定にあたっては，1 ノードの 8 プロセッサを占有してプログラムを実行し，SR8000 の高精度時間計測ルーチンである XCLOCK を用いて経過時間を測定した．測定の対象は LU 分解の実行時間であり，行列の置換に要する時間は含めていない．

4.2 実行時間の評価

実行時間の評価にあたっては，例題として各要素が [-0.5, 0.5) の一様乱数であるランダム三重対角行列（以下 (a) とする）を用い，元数 N を 500，1000，2000，4000，8000，プロセッサ台数を 1，2，4，8 と変えて評価した．ここで，プロセッサ台数が 1 の場合は従来法，2 以上の場合は本手法を用いている．各ケースに対し，乱数の初期値を 5 通りずつ変えて実行したが，実行時間はほぼ一定であった．

LU 分解の実行時間を**表 1** および**図 9** に示す．従来法による 1 プロセッサの実行時間と本手法による 8 プロセッサの実行時間とを比較すると，本手法では節点番号の付け替えにともなって各段での消去に関与する要素数が 2 倍に増え，また最後の逐次実行部分があるため，$N = 500$ の場合は従来法に比べて 1.26 倍の速度向上にとどまっているが，元数 N が増加するに従って従来法との開きは増大し，$N = 8000$ の問題では約 5.5 倍の速度向上を達成している．また，2，4 プロセッサの場合も，元数が増加するにつれ，本手法は従来法に比べて順調に速度向上を達成している．

本手法の実行時間の内訳を**図 10** に示す．元数が小さい場合は逐次実行部分（境界節点および境界節点に隣接する節点の消去）の占める割合が大きいが，この部分の時間は元数によらず一定であり，元数が大きく

表 1 従来法と本手法の実行時間
Table 1 Execution time of the conventional and the proposed method.

元数 N	従来法 (1PU)	本手法 (2PU)	本手法 (4PU)	本手法 (8PU)	速度向上 (8PU)
500	3.26E-4	2.04E-4	1.85E-4	2.58E-4	1.26
1000	6.24E-4	3.49E-4	2.66E-4	3.05E-4	2.04
2000	1.21E-3	6.63E-4	4.32E-4	3.84E-4	3.15
4000	2.44E-3	1.32E-3	7.42E-4	5.48E-4	4.45
8000	4.79E-3	2.59E-3	1.38E-3	8.75E-4	5.47

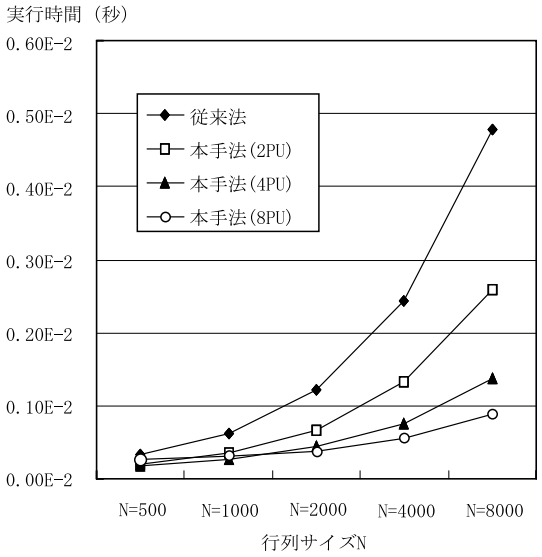


図 9 LU 分解の実行時間
Fig. 9 Execution time of the LU decomposition.

なるにつれ，並列実行部分（各領域での消去）が大部分を占めるため，本手法は大規模問題ほど有利となる．

なお，実行時間は次節で述べる (b)，(c) の行列に対しても測定したが，結果は (a) の行列とほとんど同じであったため省略する．

4.3 精度評価

次に，本手法の精度評価を行った．比較対象は軸選択付きの逐次型三重対角ソルバおよび軸選択なしの領域分割法による三重対角ソルバである．例題としては，(a) 前節で用いたランダム三重対角行列，(b) ランダム三重対角行列において対角要素に 10^{-4} を掛けた行列，(c) フランク行列 $A_{ij} = \min(i, j)$ をハウスホルダー法により三重対角化して得られる行列において，対角成分からその最小固有値を引いた行列，の 3 通りを用いた．なお，(c) のような行列に対する求解は，逆反復法により三重対角行列の固有ベクトルを求める場合に必要となる．元数 N は前節と同様に $N = 500$ から 8000 とし，残差 $\| T\mathbf{x} - \mathbf{b} \|_\infty$ により解の精度

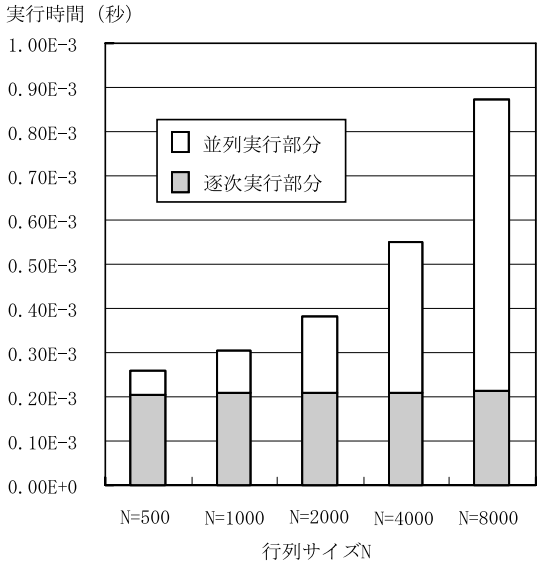


図 10 本手法の実行時間の内訳
Fig. 10 Details of the execution time.

を評価した．

(a)，(b)，(c) の行列に対する残差を，それぞれ**図 11**，**図 12**，**図 13** に示す．図より，(c) の行列の $N = 2000$ の場合を除き，本手法では領域分割法より高い精度を達成できていることが分かる．特に，対角優位性が大きく崩れている (b) の行列では，本手法は領域分割法に比べて最大 2 桁程度の高精度化を達成しており，このような行列では軸選択が必須であることが分かる．一方，従来の逐次型三重対角ソルバと比較した場合，問題によって優劣はあるものの，本手法はほぼ同等の精度を達成できている．なお，図 11，図 12 に図示した残差は乱数の特定の初期値に対する値であるが，初期値を変えた場合でも，3 種類の解法の精度差はほぼ同じ傾向を示した．参考のため，(b) の行列で $N = 2000$ の場合に初期値を 10 通りに変えて残差を比較した結果を**図 14** に示す．図 12 と同様，本手法の精度は逐次型三重対角ソルバと同等であり，領域分割法より 2 桁程度良いことが分かる．

以上より，前節で述べた並列化による性能向上を考慮すると，本手法は軸選択が必要な問題を並列計算機を用いて高速に解きたい場合に有効な解法であると考えられる．

なお，本手法の残差が逐次型三重対角ソルバより大きかった $N = 8000$ での (a)，(b) の行列については，ピボット成長率[6),13)]の調査により，その原因の分析を試みた．ピボット成長率 g は，係数行列 A の要素および LU 分解後の上三角行列 U の要素によって

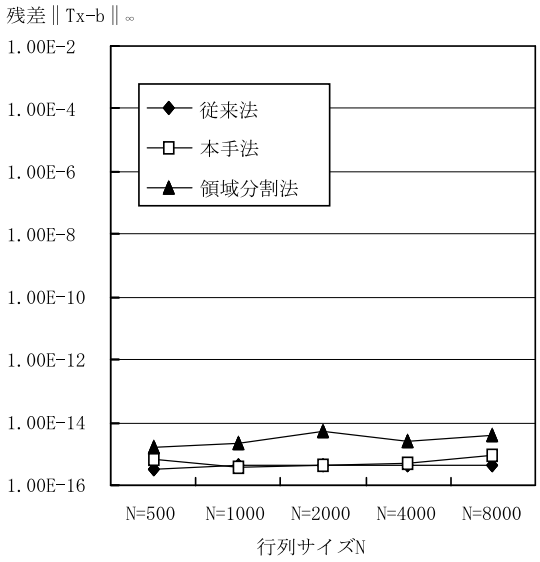


図 11 ランダム行列に対する各手法の残差
Fig. 11 Residual of each method for random matrices.

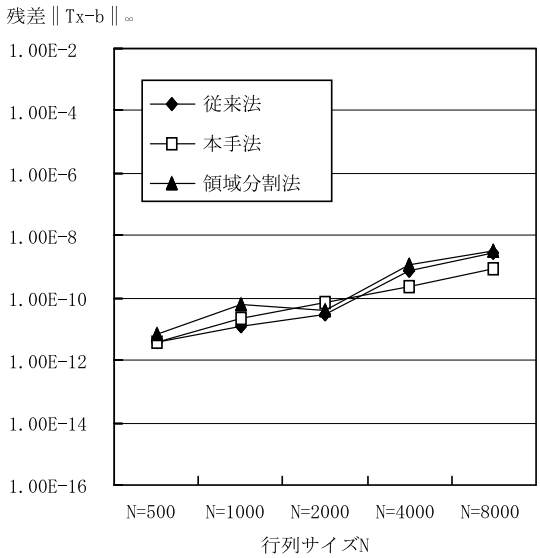


図 13 フランク行列を三重対角化した行列に対する対各手法の残差
Fig. 13 Residual of each method for matrices obtained by tri-diagonalizing the Frank matrices.

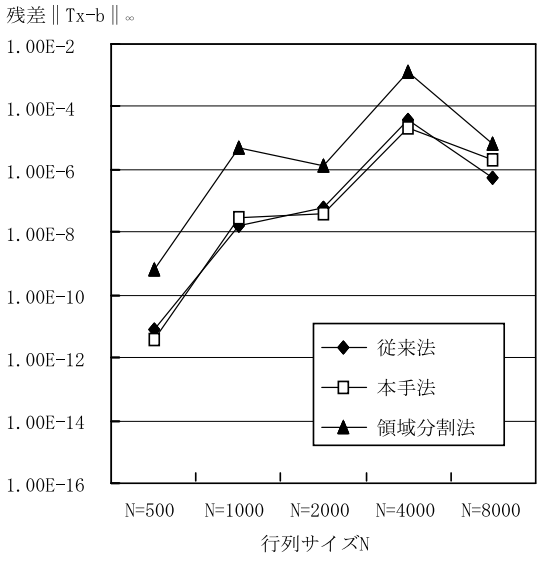


図 12 対角成分に 10^{-4} を掛けたランダム行列に対する各手法の残差
Fig. 12 Residual of each method for random matrices whose diagonal elements are multiplied by 10^{-4}.

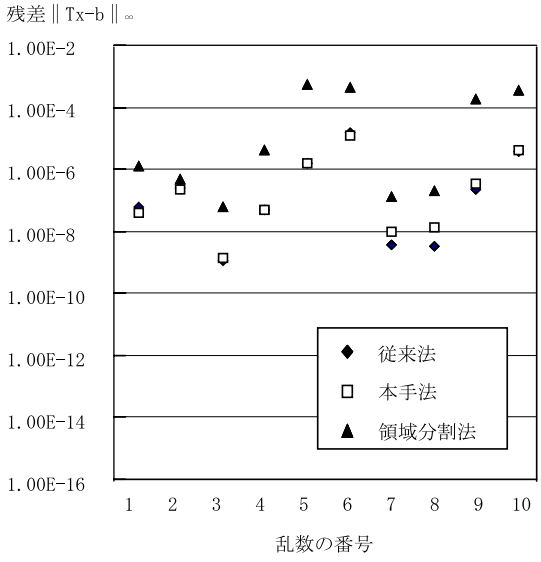


図 14 乱数の初期値を変えた場合の各手法の残差
Fig. 14 Residual of each method when the seed of the random numbers are changed.

$g = \max|U_{ij}|/\max|A_{ij}|$ と定義され，その大きさが軸選択付き LU 分解における誤差の大きさを示す尺度となる．しかし，(a) の場合，従来法では $g = 2.70$ に対して本手法では $g = 2.42$，(b) の場合，従来法では $g = 1.93$ に対して本手法では $g = 1.62$ と，いずれも本手法の方がむしろ g の値が低かった．そのため，本手法の残差が大きい原因は，ピボット成長率によっては説明できなかった．本手法の精度についての理論的な解析は，今後引き続き行う予定である．

4.4 逆反復法への適用

本節では，本手法の応用分野として有望と考えられる固有ベクトル計算のための逆反復法[1),6)]について，本手法を適用した場合の効果を考察する．

まず，固有ベクトルどうしの直交化演算[1),6)]を行わない場合，逐次型三重対角ソルバを用いた逆反復法で

は SR8000/F1 で 8000 元の三重対角行列の全固有ベクトルを求めるために約 55 秒を要し，このうち LU 分解の時間が約 40 秒と 70%を占める．したがって，本手法により LU 分解を 5.5 倍に高速化できれば，逆反復法の全実行時間は $15 + 40/5.5 = 22$（秒）と，2.5 倍に高速化できる．なお，実行時間中で LU 分解が占める割合は，行列の元数，求める固有ベクトルの本数によらず，ほぼ一定である．

一般には，固有ベクトルどうしの直交化演算が必要なため，LU 分解が実行時間中に占める割合は低下するが，求めたい固有ベクトルが属する固有値どうしが十分離れており，直交化演算が少なくて済む場合には，本手法による LU 分解の高速化は大きな効果があると考えられる．

5. おわりに

本研究では，領域分割法を改良し，部分軸選択が可能な非対称行列向けの並列三重対角ソルバのアルゴリズムを提案した．SR8000/F1 の 1 ノードによる評価では，8000 元の非対称三重対角行列の LU 分解において，本手法は従来の部分軸選択付き逐次型三重対角ソルバの 5.5 倍の高速化を達成した．今後の課題としては，分散メモリ型並列計算機上での実装，逆反復法など実際の応用への組み込みと評価があげられる．

謝辞 日頃からご指導いただいている（株）日立製作所中央研究所の稲上泰弘博士，伊藤智博士，および同ソフトウェア事業部の後藤志津雄 HPC 推進部長，五百木伸洋主任技師に感謝申し上げます．また，貴重なコメントを下さった査読者の方々に感謝いたします．

参考文献

1) Wilkinson, J.H. and Reinsch, C.(Eds.): *Linear Algebra*, Springer-Verlag (1971).
2) Varga, R.S.: *Matrix Iterative Analysis*, Prentice-Hall (1962).
3) Reinsch, C.H.: Smoothing by Spline Functions, *Numerische Mathematik*, Vol.10, pp.177–183 (1967).
4) Heath, M.T., et al.: Parallel Algorithms for Sparse Linear Systems, *Parallel Algorithms for Matrix Computations*, SIAM (1990).
5) 寒川 光：ETC 順序による 3 重対角行列の並列ソルバー，JSPP2000 論文集，pp.83–90 (2000).
6) Golub, G.H. and van Loan, C.F.: *Matrix Computations*, The Johns Hopkins University Press (1989).
7) Stone, H.S.: An Efficient Parallel Algorithm for the Solution of a Tridiagonal Linear System of Equations, *J. Assoc. Comput. Mach.*, Vol.20, pp.27–38 (1973).
8) Sumiyoshi, K. and Ebisuzaki, T.: Performance of Parallel Solution of a Block Tridiagonal Linear System on Fujitsu VPP 500, *Parallel Computing*, Vol.24, pp.287–304 (1998).
9) 秋田典伸：ベクトル計算機における三項方程式の解法，情報処理学会第 28 回全国大会予稿集，pp.1305–1306 (1984).
10) Heller, D.: Some Aspects of the Cyclic Reduction Algorithm for Block Tridiagonal Linear Systems, *SIAM J. Numer. Anal.*, Vol.13, No.4, pp.484–496 (1976).
11) Hegland, M.: On the Parallel Solution of Tridiagonal Systems by Wrap-around Partitioning and Incomplete LU Factorization, *Numerische Mathematik*, Vol.59, No.5, pp.453–472 (1991),
12) Amodio, P. and Brugnano, L.: The Parallel QR Factorization Algorithm for Tridiagonal Linear Systems, *Parallel Computing*, Vol.21, pp.1097–1110 (1995).
13) Demmel, J.W.: *Applied Numerical Linear Algebra*, SIAM (1997).
14) Dubois, P. and Rodrigue, G.: An Analysis of the Recursive Doubling Algorithm, *High Speed Computer and Algorithm Organization*, Kuck, D.J. and Sameh, A.H. (Eds.), Academic Press, New York (1977).

(平成 13 年 2 月 5 日受付)
(平成 13 年 5 月 30 日採録)

山本 有作（正会員）

1966 年生．1990 年東京大学工学部計数工学科（数理工学コース）卒業．1992 年同大学院工学系研究科物理工学専攻修士課程修了．同年（株）日立製作所中央研究所入所．以来，並列計算機 SR2001，SR2201，SR8000 向け行列計算ライブラリの研究開発に従事．大規模疎行列に対する固有値解法，連立一次方程式解法，およびその応用に興味を持つ．

猪貝 光祥

1963 年生．1987 年横浜市立大学文理学部物理課程卒業．同年現（株）日立超 LSI システムズ入社．以来，科学技術計算用ソフトウェアおよびその並列化手法に関する研究開発に従事．

直野　健（正会員）

1968 年生．1992 年京都大学理学部数学科卒業．1994 年同大学院理学研究科数理解析専攻修士課程修了．同年（株）日立製作所中央研究所入所．以来，並列計算機 SR2201，SR8000 向け行列計算ライブラリの研究開発に従事．特に並列固有値計算に興味を持つ．日本応用数理学会員．