

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[†]

[†] (株) 日立製作所中央研究所

[§] (株) 日立超 LSI システムズ

概要

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。

A Parallel Linear Equation Solver for Nonsymmetric Tridiagonal Matrices

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

[†] Central Research Laboratory, Hitachi Ltd.

[§] Hitachi ULSI Systems Corp.

Abstract

We propose a new parallel solver for nonsymmetric tridiagonal matrices, which is an improvement of 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.

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一つである。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, ETC (Ends Toward the Center) 順序に基づく方法[5]などが提案されてきた。しかしこれらは基本的に軸選択を行わないコレスキー分解に基づくアルゴリズムであり、対称正定値な行列には適しているものの、一般の非対称行列に対しては適用が困難であった。

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。以下では、まず第 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、非対称行列に適用した場合の問題点を述べる。次に、第 3 章で部分軸選択が

可能な並列解法について説明する。最後に第4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 $Tx = b$ 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 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 $Tx = b$ 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。

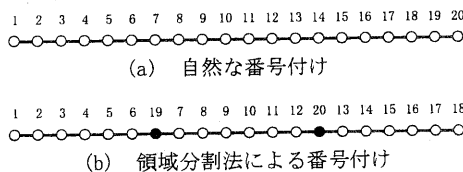


図 1 三重対角行列とそのグラフ

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わかる。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。

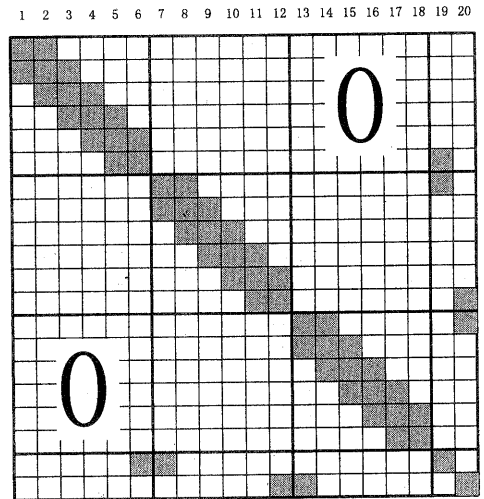


図 2 領域分割法により置換を行った行列

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]。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、行と列の同時入れ替えにより図 2 の形に変形した行列に対して部分軸選択を伴う 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 (第 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、ここで言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渡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。

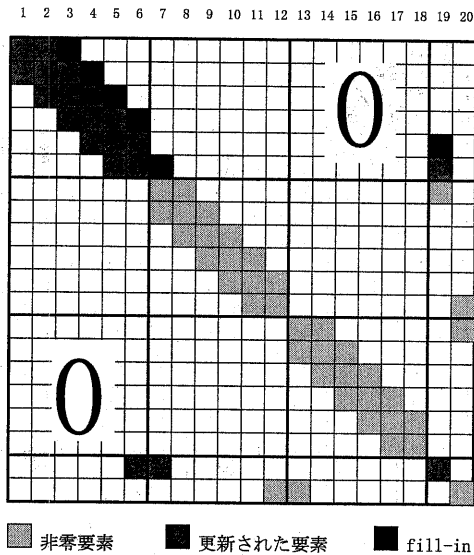


図3 第6列の消去終了時の非零パターン

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わかる。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わかる。

3. 軸選択が可能な並列解法

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更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。

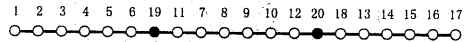


図4 本研究の手法による番号付け

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。

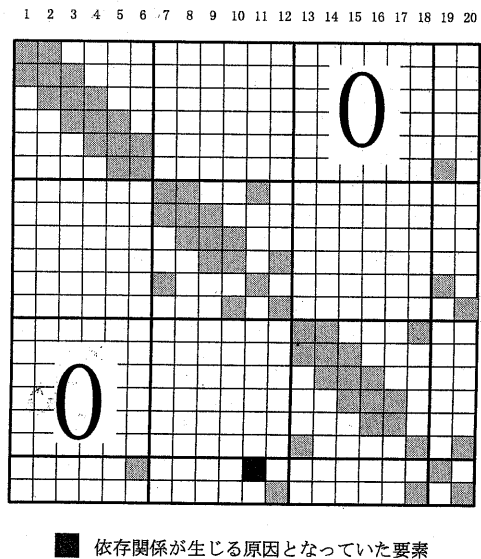


図5 本研究の手法により置換を行った行列

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、各ブロック毎に非零要素が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

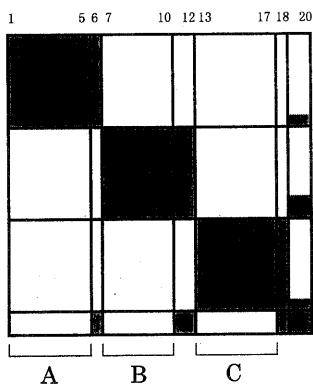


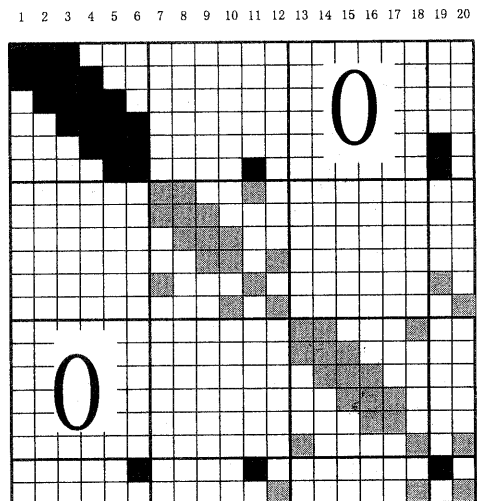
図6 非零パターンのブロック構造

まれる要素は一切影響を受けないことがわかる。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に含まれる要素は一切影響を受けない。このことより、列ブロック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わかる。

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わかる。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。以上ではプロセッサ台数が3台の場合を示したが、本方式による並列化は、任意のプロセッサ台数に対して適用可能である。

4. 終わりに

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。今後の課題としては、従来のソルバとの性能・精度の比較、共有メモリ・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挙げられる。



■ 非零要素 ■ 更新された要素 ■ fill-in

図7 第6列の消去終了時の非零パターン

参考文献

- [1] J. H. Wilkinson and C. Reinsch (eds.): "Linear Algebra", Springer Verlag, 1971.
- [2] R. S. Varga: "Matrix Iterative Analysis", Prentice-Hall, 1962.
- [3] C. H. Reinsch: "Smoothing by Spline Functions", Numerische Mathematik, Vol. 10, pp. 177-183 (1967).
- [4] M. T. Heath et. al.: "Parallel Algorithms for Sparse Linear Systems", in 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", JSPP2000 論文集, pp. 83-90 (2000).
- [6] G. H. Golub and C. F. van Loan: "Matrix Computations", The Johns Hopkins University Press, 1989.