# 並列Hessenbergダブルシフト QR 法のための新しいデータ分割法

須田礼仁[†] 西田晃[††] 小柳義夫[††]

Hessenberg 行列に対するダブルシフト QR 法は最も重要な固有値解法の1つで，並列化においては必要となるメモリ量の問題から特に高い並列化効率が望まれるものである．しかしながら，これまでに提案されてきたデータ分割法では，変換行列の先行計算や通信遅延の隠蔽が不可能であったため，十分な並列化効率を得ることができなかった．本論文では完全なロードバランスと通信待ちのないパイプラインを実現することのできる新しいデータ・計算分割と計算・通信のスケジューリングを提案する．本手法では変換行列を先行して計算することができ，通信遅延も 1/4 ステップ以上の計算で隠蔽することができる．我々は提案手法を用いて QR 法を AP1000+ に並列実装し，並列化効率を評価した．その結果，行列サイズ $n$ とプロセッサ数 $p$ に対し，$n/p = 50$ 程度で 50%，$n/p = 100$ 程度で 80%，$n/p = 150$ 程度で 90%の並列化効率を得た．このような高い並列化効率はこれまでの研究では見られなかったものであり，本手法が解法の並列性を効率良く引き出すことができることが確認された．

# A New Data Mapping Method for Parallel Hessenberg Double Shift QR Algorithm

Reiji Suda,[†] Akira Nishida[††] and Yoshio Oyanagi[††]

The double shift QR method for Hessenberg matrices is one of the most reliable eigensolvers for asymmetric real matrices. However, the parallel efficiency of the double shift QR method has been unsatisfactory, because no data mapping method that enables advance computation of the look-ahead step and communication latency hiding was known. This paper proposes a new data mapping method, where the perfect load balance and the streamlined pipeline of computations are achieved with the advance computation of the look-ahead step and the communication latency hiding. Our implementation of the parallel QR method on AP1000+ achieves 50% parallel efficiency for $n/p \approx 50$, where $n$ and $p$ are the matrix size and the number of processors, respectively. The parallel efficiency reaches 80% for $n/p \approx 100$ and 90% for $n/p \approx 150$. Such a high parallel efficiency was not observed in other researches, and it proves that the proposed method efficiently exploits the parallelism of the double shift QR method.

## 1. はじめに

非対称密行列の全固有値解法として最も重要で信頼できるものは，Householder 変換を用いて行列を Hessenberg 形に変換し，これにダブルシフト QR 法を適用するという2段階の方法である．この方法では両方の段階での計算量が行列サイズ $n$ に対し $O(n^3)$ であるため，大規模問題に対しては並列計算が重要である．前半の Householder 変換による Hessenberg 形への変換については効率的な並列実装が可能であるが，後半のダブルシフト QR 法の効率的な並列実装の実現は現在までなされていないといえる．その根本的な原因は，ダブルシフト QR 法が $O(n)$ の比較的低い並列性しか有していない点にある．しかもその少ない並列性を分散メモリ計算機上で有効に利用しうるデータ分割手法[4]が比較的最近まで知られていなかったことも効率的並列化の研究の遅れの一因である．

本研究の目的は，ダブルシフト QR 法のアルゴリズムをそのままの形で効率的に並列化することにある．他のアプローチとして，マルチシフトなどの手法によって並列性を増加させるというものもある[9]が，このような手法を用いると対象とする行列によっては収

† 名古屋大学工学研究科計算理工学専攻
Department of Computational Science and Engineering, Graduate School of Engineering, Nagoya University
†† 東京大学理学系研究科情報科学専攻
Department of Information Science, Faculty of Science, the University of Tokyo

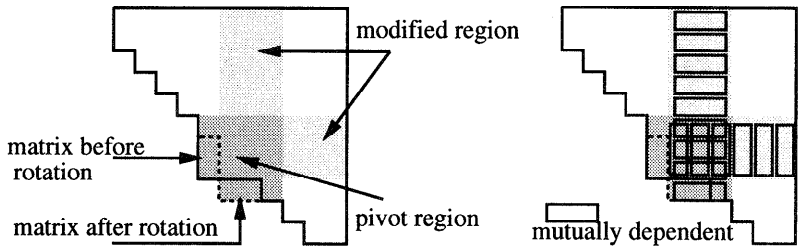



図1 QR法の1ステップにおける行列要素の更新
Fig. 1 Matrix elements updated in a bulge chase.

束性の悪化などの問題が生じる可能性があり，速度向上率や並列化効率が対象の行列に依存することになるため，容易に性能を評価できなくなってしまう．むしろダブルシフトQR法を最高の効率で並列実装したものの方が計算対象の行列の性質によらず堅実な速度向上を与えるため，マルチシフトよりも多少遅い場合があっても安心して使用できると考えることができる．本研究ではこのような立場に立ち，ダブルシフトQR法の並列化に取り組むことにした．

本論文で提案する手法で，我々はデータ分割を改良して完全なロードバランスを実現すると同時に，通信レイテンシを完全に隠蔽して計算がパイプライン状に淀みなく流れることを可能とした．本章の残りではまず基礎として用いたEISPACKのHessenbergダブルシフトQR法の計算の概要を示し，これまでに提案されている並列QR法のためのデータ分割手法を紹介する．

### 1.1 Hessenberg QR法

Hessenberg行列に対するQR法にはいくつかの改良版があるが，ここでは代表的なライブラリの1つであるEISPACKのHQRを基礎に置く．このルーチンはHessenberg行列を入力とし，ダブルシフトQR法によって固有値を求めるもので，我々の並列プログラムはこのルーチンと完全に同一の計算を行うように作成した．Hessenberg行列 $A$ に対するダブルシフトQR法の1反復の変換は

$$H = P_{n-1}P_{n-2}\cdots P_2P_1AP_1^TP_2^T\cdots P_{n-2}^TP_{n-1}^T$$

と書くことができる．ここで $P_i$ はHouseholder変換行列 $P_i = I - 2w_iw_i^T$ であって，$w_i$ は $w_i^T = (0,0,\cdots,0,\alpha_i,\beta_i,\gamma_i,0,\cdots,0,0)$ のような3つの連続した要素以外は0であるようなベクトルである．Francis stepともいわれるこれ全体を本論文では「1反復」，bulge chaseといわれる $P_i$ を左右から掛ける計算のそれぞれを「1ステップ」と呼ぶことにする．

最初の変換 $P_1$ の3つの非零要素は右下隅の $2\times2$ 小行列と最初の $3\times3$ 小行列から決まる．この変換をすると $A_{3,1}$，$A_{4,1}$，$A_{4,2}$ が非零となる．それ以外の変換 $P_i$ では行列の第 $i-1$ 列の対角より下の3つの要素 $A_{i,i-1}$，$A_{i+1,i-1}$，$A_{i+2,i-1}$ から決まる．$P_i$ の変換の後，下の2つの要素である $A_{i+1,i-1}$ と $A_{i+2,i-1}$ は0となるが，そのかわり $A_{i+3,i}$ と $A_{i+3,i+1}$ が非零になる．

1ステップの変換 $P_iBP_i^T$ での行列の更新は図1のようになる．更新前の行列は実線で示すようにHessenberg行列から少しコブが出た形である．1ステップの計算で更新される要素は網をかけた3行3列のL字型の領域である．データの依存関係を四角で囲むと図1の右図のようになり，この図の四角のどれかが複数のプロセッサに跨るときに通信が必要となる．更新される要素のうち，行と列がクロスする網かけの濃い部分を枢軸部分と呼ぶことにする．

### 1.2 これまで用いられてきたデータ分割

効率的に並列計算を行うためにはロードがバランスしており，通信が少ないことが必要である．行列計算の多くは列ごと，行ごと，あるいは二次元にデータを分割することによりこれが実現できる．しかしHessenbergダブルシフトQR法の場合には一度に3行3列しか更新されないため，このような分割ではロードバランスが悪い．これに対し，1991年にvan de Geijn[4]はblock Hankel-wrapped storage schemeを提案した．これは第 $i,j$ ブロックをプロセッサ $(i+j)/m \bmod p$ に割り当てる（$m$ は正整数）．たとえば，$p=4$，$m=2$ の場合図2の左の図のようになる．この方法はロードバランス，通信量ともオーダーとしては最適である．WuとChu[5]はこれとは逆に主対角方向にデータをならべて図2の右の図のようにしたが，これはvan de Geijnのものと比べてロードバランスも通信量も少しずつ悪い．

これらの手法を用いることによってQR法を並列に計算することが可能となるが，得られる速度向上率や並列化効率[5),6)]は必ずしも満足できるほど高くはない．さらに効率的な並列処理を実現するには，ソフトウェ

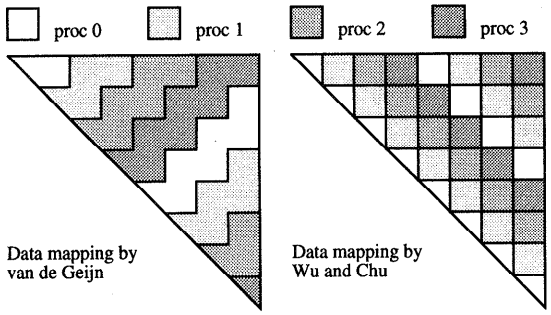



図2 これまでに提案されてきたデータ分割手法
Fig. 2 Some data mapping methods in other researches.

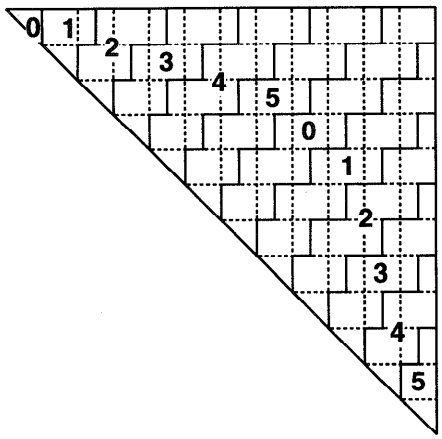



図3 提案するデータ分割
Fig. 3 The data mapping proposed in this paper.

アパイプライニングや通信遅延の隠蔽などの技術を用いる必要がある．ダブルシフトQR法の場合，枢軸部分が計算されていなければそのステップの変換はできないので，枢軸部分の計算を先行させて計算をパイプライン化し，変換情報の通信のレイテンシを隠蔽することが，効率的な並列処理の実現のために必要である．しかし上述のデータ分割手法はこのようなパイプラインスケジュールを想定して設計されていないため，枢軸部分の計算をうまく先行させることができず，通信時間の隠蔽もできない．

## 2. 並列Hessenberg QR法のための新しいデータ分割

### 2.1 データの分割とプロセッサへの割付け

**図3** は本論文が提案するデータ分割を示している．この方法はプロセッサ数 $p$ に対して行列を $2p \times 2p$ のブロックに分割したものを基礎としており，図ではこのブロックを破線で示している．図3は $p = 6$ の場合を示しており，実線はデータ分割の区切りを，数字はそのデータを担当するプロセッサの番号を示している．行列は斜めの帯状に分割され，各プロセッサが $p$ 離れた帯を2つずつ扱うが，これはvan de Geijnの分割とよく似ている．この帯状の構造はロードがバランスする並列ダブルシフトQR法のデータ分割方法において，通信が必要となるプロセッサ境界の要素数を最小オーダーにするためには必須である．我々の手法ではこの斜めの帯が対角付近を除いてvan de Geijnの分割よりも1.5ブロック左にずれており，これにより対角付近のロードが下げられて枢軸部分を先行して計算することが可能となる．

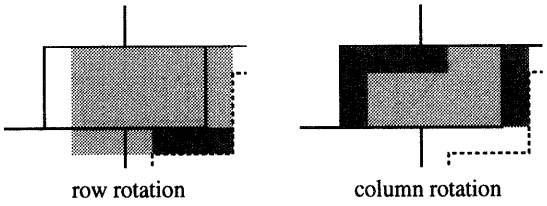



図4 計算の割当て
Fig. 4 Allocation of computations.

### 2.2 計算の分割とプロセッサへの割付け

複数のプロセッサにあるデータを必要とする計算があったとき，その計算をどちらのプロセッサに割り振るかというのが計算分割・割付けの問題である．本論文では**図4** の計算分割・割付けを提案する．図は中央の部分行列の行変換・列変換を行う際に，この部分行列が割り当てられているプロセッサが計算を行い更新する部分に網をかけて示している．網かけの濃い部分は計算後にデータを送る部分である，破線はこのプロセッサが余計に持たなければならない「縁」の範囲で，双方とも2列分の幅がある．この計算分割は，同じプロセッサが同じ要素を1反復で2度以上送ることがないところに特長がある．

### 2.3 計算のスケジューリング

**図5** は図3と同じ設定で第5ステップの変換を行う際のスケジューリングを示している．ここでは1/2ブロックの変換をスケジュールの単位に用いるので，これを「ハーフブロック」と呼ぶことにする．対角ブロックは計算される要素数がおよそ半分しかないからこれだけでハーフブロックとなるが，それ以外ではブロックを縦に2分割してそれぞれをハーフブロックとする．各プロセッサは各ステップで4つのハーフブロックの計算を行う．したがって，ロードは各ステップでバランスしている．図では1ステップの時間を4つの四半期に分け，この4つの四半期にそれぞれのプロセッサが計算するハーフブロックを1から4の数字で示している．矢印は計算結果の送信の概略を示している．こ

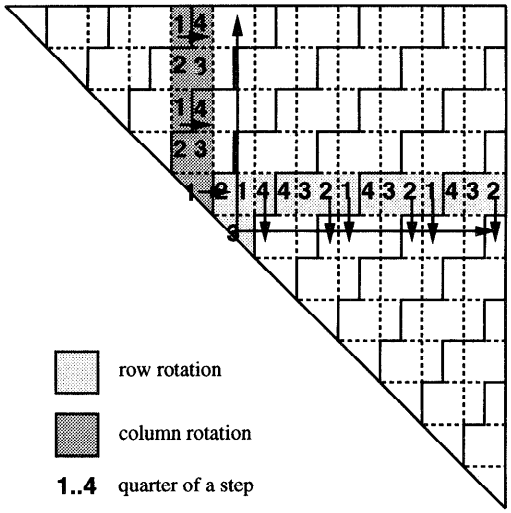



図5 提案スケジューリングと通信の様子
Fig. 5 Proposed scheduling and communication.

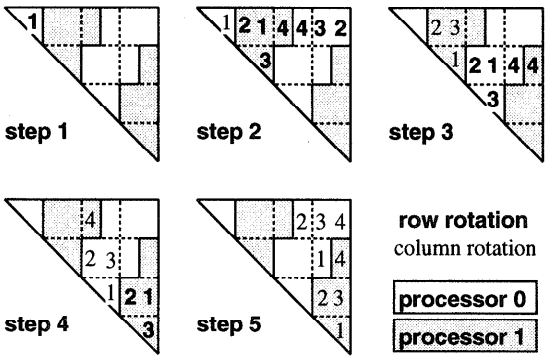



図6 2プロセッサの場合の1反復のスケジューリングの全体
Fig. 6 The computations of a Francis step with 2 processors.

こで，プロセッサ5が次の第6ステップの枢軸部分を第3四半期に先行計算している点が重要である．これが第3四半期にスケジュールされることにより変換行列の計算からそれを用いるまで1四半期分の余裕ができる（第3四半期で計算した結果を次のステップの第1四半期に使用するため．以下，これを $3 \to 1^+$ と表す）ので，この間に通信遅延を隠蔽することが可能となる．残った列変換はここでのプロセッサ4のように，当該のステップの最初に行う．その結果は次のステップで隣のプロセッサが列変換に用いるが，$1 \to 2^+$ の1ステップ分の余裕で通信遅延を隠蔽できる．

行変換は各プロセッサで右から左に実行する．これは右の2つのハーフブロックの計算結果を次のステップで隣のプロセッサが使用するからである．この順序によって $2 \to 3^+$ の1ステップ分の余裕ができ，通信遅延が隠蔽できる．枢軸の付近ではスケジューリングが異なっているが，$4 \to 2^+$ の1四半期分の余裕がある．一方，左の2つのハーフブロックの計算結果は後の列変換の際に必要となるが，これは枢軸付近以外では次々ステップ以降であるし，枢軸付近でも $2 \to 1^+$ のように2四半期分の余裕があり，十分通信遅延を隠蔽できる（図が煩雑になるのを防ぐため，時間的な余裕の大きいこの通信の図示は省略した）．

列変換は各プロセッサで下から上に実行する．これは最も下のハーフブロックの計算結果を同じステップで隣のプロセッサが使用するからである．このスケジューリングでは $1 \to 4$ の2四半期分の余裕がある．このように，すべての通信で送信から受信まで1四半期分以上の余裕があるため，1ハーフブロックの計算時間が通信所要時間より長ければ，受信待ちがまったくない滑らかなパイプラインが実現される．$n/p = b$ とすると，各ステップでの通信時間は $O(b)$ なのに対し計算時間は $O(b^2)$ となるので，$b = n/p$ がある程度大きければこの条件は満たされる．

反復の最初と最後ではこれまでの説明とは状況が若干異なる．**図6** に $p = 2$ の場合の1反復のスケジューリングの全体を示した．最初のステップではプロセッサ0が最初の枢軸ブロックを計算するだけで，この計算結果が放送されるのを他のプロセッサが待つ．また，最後から2番目のステップではプロセッサ0だけは3ハーフブロックしか計算がない．これらスケジューリングの例外はロードインバランスを引き起こすが，そのオーバヘッドは全体の計算量の $O(1/p)$ となり，プロセッサ数が多くなれば相対的に小さくなる．

### 2.4 データの再分配のコストについて

ところで，このマッピング手法は他に見られない独特のものである．したがって，本手法を応用する場合には，QR法を始める前に本手法に従ってデータの再分配を行う必要があるであろう．その際移動するデータ量は行列要素数である $n^2/2 + o(n)$ である．一方，本手法で全固有値を求めるまでに必要な通信量は，1つの固有値を求めるのに平均で $c$ 反復が必要であるとすると，プロセッサあたり $7cn^2$ 程度となる．$c$ は通常3程度であるから，プロセッサあたりの通信量は $20n^2$ 程度となる．したがって，データ再分配のコストはQR法そのもののコストに比べて相当に低く，再分配の必要性が本手法の性能的な有効性に致命的な影響が与えることはないと考えられる．

### 2.5 分割の細分化

以上で説明したデータ分割手法は行列を $2p \times 2p$ のブロックに分割したものを基礎として $2p$ 個の帯を作り，これをプロセッサに割り当てていたが，分割を細

分化することも可能である．この場合細分化された帯のプロセッサへの割当てを，サイクリックにするかブロックにするかという選択肢がある．

いずれの割当て方法にせよ，反復の最初と最後のステップで生じるロードバランスの悪さが改善される．また通信の余裕も相対的に大きくなる．たとえば，半分の幅に細分化すれば各プロセッサは1ステップで8ハーフブロックを計算することになるので，変換情報の通信の余裕は $3 \to 1^+$ で5八半期分となる．しかしブロックサイズが小さくなるためループが短くなって計算性能が相対的に低くなったり，雑多なオーバヘッドで増えるものがあるなど，細分化すれば性能が向上するとは必ずしもいえない．

### 2.6 HQRにおける計算領域の縮小の問題

以上のデータ分割は行列が $n \times n$ であることを仮定している．しかし，我々のプログラムが基礎としているEISPACKのHQRでは3種類の処理により計算領域が縮小される．第1に，固有値が求まるごとに減次が行われ，行列の最後の列と行が計算対象からはずされる．第2に，下対角要素で0にきわめて近いものがある場合，そこで計算領域が2つの小行列に分離される．第3に，下三角要素が0でなくても一定の条件を満たせば，そこから上の部分は行変換を省略することができる．これらの処理により，実際に計算される領域は図7で網かけで示した台形の領域に縮小されてしまう．以下ではEISPACKの変数名に従って，図に示すように減次後の行列サイズを $n$，第2・第3の処理の行われる位置を $l$ および $m$ とする．

これらの処理のうち，第3の処理では計算量が減るだけで並列性に影響はないが，第1・第2の処理が行われると1ステップで計算される行列の領域が図5で示された領域よりも狭くなり，ロードバランスが崩されてしまう．これに対してデータを再分配してロードバランスを回復することが考えられる．しかし第2の処理では $l$ の変化は容易に予測できないので，再分配は慎重に行わなければならない．一方，減次では $n$ は一度にたかだか1ずつしか変化しないため再分配も比較的容易と思われるが，計算範囲が極端に小さくなった場合にはプロセッサ数を減らした方が高速となるなどのため，データ再分配の最適な方法を見いだすことは簡単ではない．一方，前節で説明したようにデータ分割を細分化してサイクリックにプロセッサに割り当てれば，計算領域が小さくなったときのロードバランスの悪化を緩和することが可能である．しかしこの方法では通信のオーバヘッドが増大し，通信の余裕も絶対時間として短くなるなど問題も多い．計算領域の縮小にともなうこれらの問題は現在未解決であって，今回の実装の際にはデータの再分配は行わず，理想的でないデータ分割のまま計算を続行する方法をとった．これに関しては今後研究をさらに進めてゆきたいと考えている．

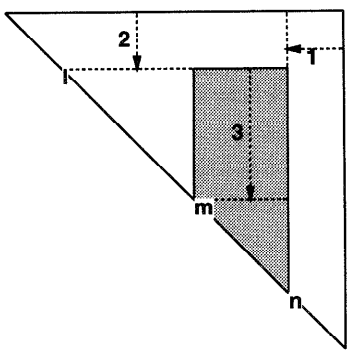



図7 計算領域の縮小
Fig. 7 Matrix size reduction.

## 3. 性能評価

この章では並列計算機AP1000+上に実装された並列HQRの性能の評価結果を報告する．実装に用いたAP1000+は16 MBのメインメモリと36 KBのライトスルーキャッシュを持つ50 MHzのクロックのSuperSPARCをCPUとし，トーラストポロジの25 MB/sの相互結合網を主なプロセッサ間通信に用いるマルチコンピュータシステムである．プログラミングは独自のメッセージ通信ライブラリを用いる．今回はC言語でプログラムを記述した．評価の際にはDMAを用いた高速な遠隔書き込み機能を使用したが，通常のメッセージ通信を用いても極端な性能の違いはなかったので，今回の性能評価でAP1000+を使用したために特殊な結果が得られたとは考えていない．提案手法では，通信は放送以外はリングトポロジの隣接通信のみを必要とするので，トーラスをリングにマッピングして使用した．

並列ダブルシフトQR法の性能指標としては，並列化効率が非常に重要であることが以前から指摘されてきた[6]．これは，並列化効率が低いとプロセッサあたりに必要となるメモリの量が増大するという問題があるからである．並列QR法の並列化効率は行列サイズ $n$ とプロセッサ数 $p$ に対して $n/p$ でほぼ決まるので，一定の並列化効率を維持するために $n/p = b$ と固定すると，1プロセッサあたりのメモリ量は $bn$ となる．並列化効率が低いと必要な $b$ が大きくなってしまい，大きな $n$ に対してメモリが不足するという事態が生

じてしまうのである．このため並列化効率を特徴付ける $b = n/p$ を小さくおさえることが実用上非常に重要となるのである．

これに基づき，以下では並列化効率を性能の指標とするが，今回の評価では以下の2つの点に注意した．まず，性能を全固有値を求める場合のほか，最初に固有値が求まるまでの計算のみに対しても測定した．後者の場合，前章で述べた計算領域の縮小はほとんど生じないため，提案手法が理想的な形で働く．これによって提案手法において設計どおり受信待ちなしにパイプラインが進むことを確認することができるほか，今回行わなかったデータの再分配を今後行った場合の性能を予測するうえで有用な情報を得ることができる．もう1点は並列化効率の計算方法である．並列化効率は1台での性能との相対性能から求めるのが本来の意味にふさわしいと考えるが，メモリや時間の制約のため大規模な問題は1台では実行できないことがある．そこで本論文ではプロセッサあたりのMflops値を計算することで間接的に並列化効率を評価する[6]．すなわち，実行された総浮動小数点数演算数 $V$，プロセッサ数 $p$，並列実行に要した実時間 $T$ から，プロセッサあたりのMflops数 $F$ を $F = 10^{-6}V/(pT)$ で計算する．1台でのダブルシフトQR法の最高性能はほぼ20 Mflopsであったので，並列化効率は $F/20$ で概算評価した．なお，実験で用いた行列は [0,1) の一様乱数を用いて生成している．

**図8** は最初の固有値が求まるまでの，計算領域の縮小がない場合の計算の性能を示している．グラフは縦軸にプロセッサあたりのMflops値，横軸に $n/p$ をとっている．グラフから分かるように，$n/p$ が50程度で効率が50%，100になると80%，150程度で90%に達し，高い並列化効率が実証されている．このような高い並列化効率は，これまでの研究[1)~6)]ではほとんど見られなかったもので，本手法が可能としたパイプライン計算と通信遅延の隠蔽によりこのような並列化効率が実現したものと考えられる．なお，$n/p = 40$ では $p = 2$ の場合だけが性能が高いが，これはデータがすべてキャッシュに乗っており，キャッシュミスのオーバヘッドがかからないためと考えられる．

**図9** は800元の行列を4台で解いたときのおよそ1反復分の実行状況のトレースを各プロセッサの各時刻での仕事内容を色で示すことにより表示している．黒く塗ってあるのが待ち時間を含めた受信の部分であるが，大抵の場合データの到着が受信より早く，すぐに制御が計算に戻るため線状に見える．受信待ちは反復の最初と最後のステップに少しずつみられるほかはほとんどなく，パイプラインがきれいに流れていることが分かる．

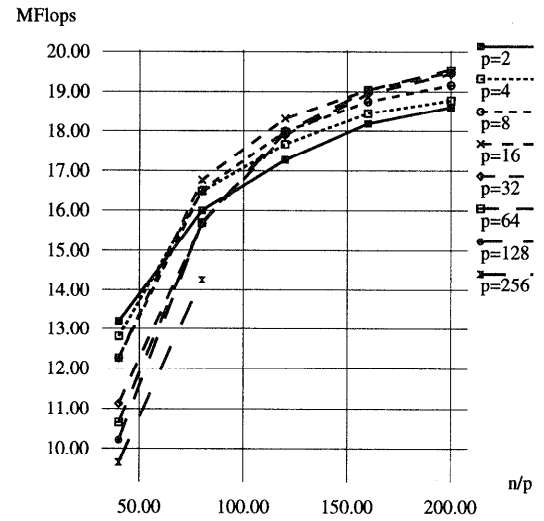



**図8** 計算領域の縮小のない場合の計算性能（Mflops）と $n/p$ との関係
Fig. 8 Mflops vs $n/p$ without matrix size reduction.

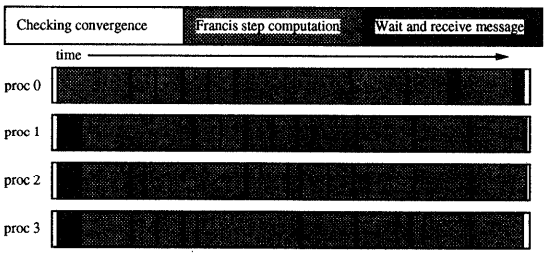



**図9** 第1反復のトレースの結果
Fig. 9 Trace results for the first Francis step.

**図10** はすべての固有値を求めるまでの計算についての性能とプロセッサあたりの行列サイズとの関係を示している．ロードバランスが計算領域の縮小によって壊されているため，性能は計算領域の縮小がない場合の結果を下回っているが，(計算領域の縮小がないときには通信遅延を隠蔽する働きのあった) スケジューリングの余裕時間が，ここではある程度のロードの不均等を吸収する働きをしており，これにより性能の低下が軽減されている．計算領域の縮小が生じる場合の性能の向上については今後いっそうの研究の必要がある．

## 4. ま と め

本論文ではHessenberg行列に対するダブルシフトQR法の並列化のための新しいデータ分割手法を提案

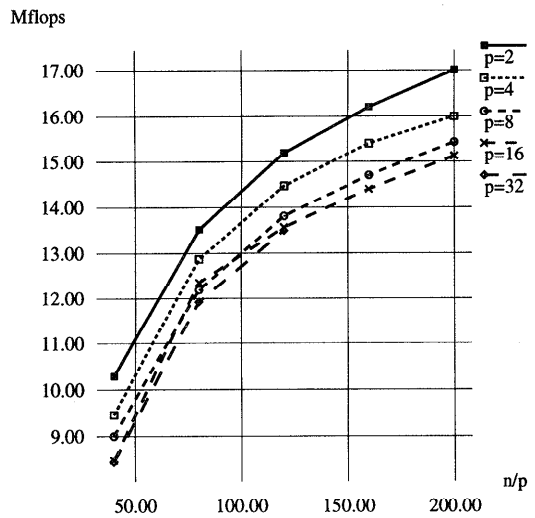



**図 10** 全固有値計算の性能（Mflops）と $n/p$ との関係
Fig. 10 Mflops vs $n/p$ for the total eigensolution.

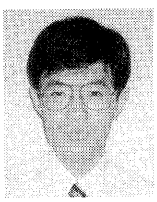
した．このデータ分割手法では各ステップでロードがバランスしており，枢軸部分を先行して計算したり通信遅延を隠蔽したりすることが可能である．我々は提案手法を用いて並列ダブルシフト QR 法を AP1000+ に実装することにより，まったく待ちのないなめらかなパイプラインと，これまでに他の研究で報告されているものよりもはるかに高い並列化効率が実現されることを実証した．これにより並列 QR 法の問題であるプロセッサあたりのメモリ量の問題も格段に軽減されたことになる．計算領域の縮小への対応など，さらに高い並列化効率と性能を目指して今後も研究を進めたい．

**謝辞** 貴重な助言を下さった電総研の建部修見氏と査読者の皆様に感謝申し上げます．なお，本研究の一部は文部省科学研究費（基盤研究 (C) 09680327，奨励研究 (A) 09780246）ならびに日本学術振興会未来開拓学術研究推進事業（地球規模流動現象解明のための計算科学）による．

## 参 考 文 献

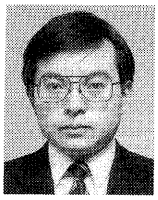
1) Bai, Z. and Demmel, J.: On a block implementation of Hessenberg multishift QR iteration, *International Journal of High Speed Computing*, Vol.1, No.1, pp.97–112 (1989).
2) Boley, D., Maier, R. and Kim, J.: A parallel QR algorithm for the nonsymmetric eigenvalue problem, *Computer Physics Communications*, 53, pp.61–70 (1989).
3) Geist, G.A. and Davis, G.J.: Finding eigenvalues and eigenvectors of unsymmetric matrices using a distributed-memory multiprocessor, *Parallel Computing*, 13, pp.199–209 (1990).
4) van de Geijn, R.A.: Storage schemes for Parallel Eigenvalue Algorithms, *Numerical Linear Algebra, Digital Signal Processing and Parallel Algorihtms*, NATO ASI Series Vol.F70, pp.639–647, Springer-Verlag (1991).
5) Wu, L. and Chu, E.: New distributed-memory parallel algorithms for solving nonsymmetric eigenvalue problems, *Proc. 7th SIAM Conference on Parallel Processing for Scientific Computing*, pp.540–545 (1995).
6) Henry, G. and van de Geijn, R.: Paralllelizing the QR algorithm for the unsymmetric algebraic eigenvalue problem: Myths and reality, *SIAM J. Sci. Comput*, Vol.17, No.4, pp.870–883 (1996).
7) 須田礼仁，西田 晃，小柳義夫：並列 Hessenberg QR 法のための新しいデータ分割法と AP1000+への効率的実装，JSPP'97 論文集，pp.377–384 (1997).
8) Nishida, A., Suda, R. and Oyanagi, Y.: Polynomial Acceleration for Restarted Arnoldi Method and its Parallelization, *Proc. 3rd IMACS International Symposium on Iterative Methods in Scientific Computation* (to appear).
9) Henry, G., Watkins, D. and Dongarra, J.: A parallel implementation of the nonsymmetric QR algorithm for distributed memory architectures, Computer Science Dept., Technical Report, CS-97-352, University of Tenessee (1997).

(平成 9 年 10 月 31 日受付)
(平成 10 年 4 月 3 日採録)



**須田 礼仁**（正会員）

1968 年生．1991 年，東京大学理学部情報科学科卒業．1993 年，同大学院理学系研究科情報科学専攻修士課程修了．同年同大学理学系研究科助手．1996 年，東京大学博士（理学）．1997 年，名古屋大学工学研究科講師．数値計算，並列処理に関する研究に従事．科学技術計算の並列処理技術，ハイパフォーマンスプログラミング，大規模疎行列連立一次方程式の解法等に興味を持つ．日本応用数理学会会員．



**西田　晃**（正会員）

1970年生．1993年東京大学理学部情報科学科卒業．1998年同大学院理学系研究科情報科学専攻博士課程修了．理学博士．同年同大学助手．現在に至る．数値解析，並列処理に関する研究に従事．特に大規模線形計算に興味を持つ．米応用数理学会，日本応用数理学会各会員．

**小柳　義夫**（正会員）

1943年生．1966年，東京大学理学部物理学科卒業．1971年，同大学院理学系研究科物理学専門課程修了，理学博士．同年同大学助手．高エネルギー物理学研究所理論部門助手，筑波大学電子情報工学系講師，助教授，教授を経て，1991年東京大学理学部情報科学科教授．並列処理，数値解析，計算物理学に関する研究に従事．特に，偏微分方程式の高速並列解法，最小二乗法の数値計算，乱数やモンテカルロ法に興味を持つ．物理学会，日本統計学会，応用統計学会，計算機統計学会，応用数理学会等各会員．