並列固有値ソルバーの実現とその並列性の改良

片桐孝洋[†]　金田康正[††]

本論文では，分散メモリ型並列計算機において大規模な実対称密行列の全固有値，全固有ベクトルを求めることが可能な並列固有値ソルバーの実現方法とその性能とについて報告する．本並列ソルバーは大規模演算を考慮し，計算に必要なデータをなるべく均等に各プロセッサが保存するように設計されている．一方，逆反復法を用いた固有ベクトル計算処理における再直交化処理の並列性が従来から問題となっているが，本ソルバーにおいてこの直交化処理を並列化した．その結果，従来の方式と同精度の解が得られる問題において，8.3 倍程度の速度向上が確認された．さらに加えて，重複固有値が甚だしい固有値問題においても，精度と並列実行性能の面から，従来の再直交化方式に比べて優れた結果が得られるような再直交化方式を示す．また，この方式が実際に優れていることを glused Wilkinson 行列による数値実験により確認した．

A Parallel Implementation of Eigensolver and an Improvement of Its Parallelism

Takahiro KATAGIRI[†] and Yasumasa KANADA[††]

In this paper, we show how our parallel implementation of eigensolver can give all eigenvalues and all eigenvectors of dense real symmetric matrices and its performance. We give careful consideration to treat large scale eigenproblem, therefore we designed our parallel eigensolver to distribute the data which is needed to solve eigenproblem equally. On the other hand, it is known that the re-orthogonalization process has poor parallelism on parallel inverse iteration. We show how to parallelize the re-orthogonalization process in this paper. In the result of our parallelization, we can get 8.3 times speedup in our parallel inverse iteration against the conventional ones under the same orthogonal accuracy. In addition, we suggest a re-orthogonalization method which is superior to conventional re-orthogonalization method from accuracy and parallel execution time. The superior cases are observed with a numerical experiment to the glued Wlikinson matrix.

1. はじめに

実対称密行列の標準固有値問題を解く方法で，現在良く知られて実際に用いられているアルゴリズムは以下に示す三つの手順でなされていることが多い．すなわち，1) 実対称密行列を三重対角行列に変換する，2) 三重対角行列の固有値，固有ベクトルを求める，3) 元の実対称密行列の固有ベクトルに変換する，の三つである．この逐次処理で良く用いられている方法を用いて並列処理を行うという研究は，1970 年代前半の Illiac IV での研究に遡る．しかしながら実際の大規模固有値問題を計算できるようなアルゴリズムを設計して評価する研究は現状ではあまりなされていないように思われる．その一例として，世界標準の行列計算ライブラリである ScaLAPACK において，密集固有値の数が多くなるとメモリオーバーで計算が続行できなくなるという問題[1)]が生じることを，直野ら[2)]が指摘している．

そこで本研究ではまず第一に，密集固有値の数が多い大規模な固有値問題でも実行できる並列固有値ソルバーを実現し，そのアルゴリズムの性能評価を行うことを目的とする．次に，並列化の報告がなされていない逆反復法における再直交化処理について，直交化精度を犠牲にすれば並列化出来ることを示し，さらに実際の問題においてその速度および精度を吟味する．さらに加えて，密集固有値の数が甚だしい固有値問題を処理する場合においても，精度と並列実行時間の観点において従来の再直交化処理に比べ優れた方式を示し，その性能を評価する．

† 東京大学大学院理学系研究科情報科学専攻
Department of Information Science, Graduate School of Science, the University of Tokyo
†† 東京大学大型計算機センター
Computer Centre, the University of Tokyo

2. 並列固有値ソルバーの各処理の説明

2.1 全体の処理の流れ

我々の並列固有値ソルバーは，以下のような手順で固有値および固有ベクトルを計算する．ここで，固有

値を計算する対象の行列はあらかじめ各プロセッサエレメント (PE) に分配済と仮定する．

1) 対称密行列を Householder 法を用いて並列に三重対角行列に変換する (**三重対角化ルーチン**)．
2) 各 PE に分散されている三重対角行列の要素を，全 PE が全てその要素を所有するように再分散する (**再分散ルーチン**)．
3) 2) で収集された三重対角行列に対して，二分法で並列に全固有値を求める (**固有値探索ルーチン**)．
4) 3) で各 PE で部分的な範囲で計算された固有値を，全 PE が全ての固有値の範囲を所有するように収集する (**固有値収集ルーチン**)．
5) 逆反復法を用いて，三重対角行列の全固有ベクトルを並列に計算する (**固有ベクトル計算ルーチン**)．
6) 5) で計算された各固有ベクトルを，元の対称密行列の固有ベクトルに並列に変換する (Householder **逆変換ルーチン**)．

ここで，2) で収集される行列の要素は，固有値を計算する行列の要素に対して非常に少なくなっている点に注意する．すなわち最初の行列が $n \times n$ 要素とすると，2) で収集する三重対角行列はその対称性から $2n$ となる．

2.2 三重対角化ルーチン

我々は既に，非対称行列および対称行列用の Householder 法を用いた並列相似変換ルーチンを開発している[3)]．ここでは，三重対角化は対称行列を処理の前提としているので，対称行列用に特化したアルゴリズム[3)]を用いることにする☆．

このアルゴリズムでは，処理する行列をサイクリックサイクリック分割方式で分散する．いま PE 台数を p，二次元的な PE 番号を $P_{ncidx,ncidy}$ $(ncidx, ncidy = 0, 1, ..., \sqrt{p}-1)$ とするとき，$\sqrt{p} \times \sqrt{p}$ のように構成されているとする．この場合，サイクリックサイクリック分割方式とは，行列 A の要素 $a_{ij}(i, j = 1, 2, ..., n)$ を

$$ncidx = (i-1) \bmod \sqrt{p}$$
$$ncidy = (j-1) \bmod \sqrt{p} \qquad (1)$$

の PE に割り当てる分割方式である．このとき，三重対角化の処理中のベクトルリダクション演算に費やす通信時間および放送時間を $1/\sqrt{p}$ のオーダで削減することが可能となる．

このルーチンでは第 k $(k = 1, 2, ..., n-2)$ 反復時，Householder 法による相似変換

$$H^{(k)} A^{(k)} H^{(k)} \qquad (2)$$

を行っている． なお $H^{(k)}$は

$$H^{(k)} \equiv (I - \sigma_k u_k u_k^T) \qquad (3)$$

である．ここで固有ベクトルを求める場合に，行列 $H^{(k)}$ が必要となる．一般に $H^{(k)}$ を計算して個別に記憶するのは記憶領域と計算量から効率が悪いので，ベクトル u_k(以下枢軸ベクトル) とスカラ σ_kとを記憶しておく．この時，各 PE 当たりの記憶領域を減らす目的から，これらのデータは各 PE に分散して所有させる．

この場合，計算の都合で枢軸ベクトル u_k は $ncidx$ が等しいプロセッサ群 (すなわち $\sqrt{p}$ 個) にサイクリック分割方式で分散されている．いま枢軸ベクトル u_k の要素を $v_i(i = k, k+1, ..., n)$ とすると

$$ncidy = (i-1) \bmod \sqrt{p} \qquad (4)$$

の PE に分散して所有されている．一方，三重対角化の処理中に全 PE がスカラ σ_k を計算することから，スカラ σ_k は全て $(k = 1, 2, ..., n-2)$ を各 PE が所有するように実装した．

2.3 再分散ルーチン

このルーチンでは，サイクリックサイクリック分割方式で分散されている三重対角行列の要素を，全 PE がその要素全てを所有するように再び分散を行う．いま自分の PE の番号を $P_{ncidx,ncidy}$ $(ncidx, ncidy = 0, 1, ..., ncelx \equiv \sqrt{p}-1)$，とラベルづけする．このとき，このアルゴリズムは図 1のようになる．

```
⟨1⟩  if (ncidx .eq. ncidy) then
⟨2⟩    ncidx が等しいプロセッサに所有データを
         放送;
⟨3⟩  else
⟨4⟩    データを受信;
⟨5⟩  endif
⟨6⟩  do i=0, ncelx
⟨7⟩    if (i .eq. ncidx) then
⟨8⟩      ⟨1⟩ – ⟨4⟩ の処理で受信したデータを ncidy
           が等しいプロセッサに放送;
⟨9⟩    else
⟨10⟩     データを受信;
⟨11⟩   endif
⟨12⟩ enddo
```

図 1 再分散ルーチンのアルゴリズム

なお副対角要素の収集は，図 1中の ⟨1⟩，⟨2⟩ の処理を，$(ncidx + 1) \bmod \sqrt{p}$ が $ncidy$ と同じになる PE が放送するように変更することで，同様に処理が可能

☆ このルーチンは対称性を利用していないので，n^2 の行列保存領域と $8/3n^3$ の演算回数を必要とする．ここで誤解されないために明記しておくが，行列演算に関する並列処理では，一般に対称性を利用することで $1/2$ の保存領域と演算時間の利点を引き出せる場合が多いと思われる．しかしながら Householder 法を用いた三重対角化に限っては，k回目の反復において，対称性を利用する方法はベクトルリダクション演算のベクトル長が $n-k$，およびそれに従事する PE が p 個 なのに対し，文献 3) で提案した方法では ベクトル長が $(n-k)/\sqrt{p}$，それに従事する PE が$\sqrt{p}$ 個 に減る．よって PE 数が極めて多い場合，もしくはリダクション演算が極めて低速な場合は，対称性を利用しない方法が有効になる場合があると考えて良い．

であることに注意しておく．

2.4 固有値探索ルーチン

固有値の探索には二分法を用いる．二分法の実装方式は文献 4) の BISECT ルーチンと同様なので，ここでは説明を省略する．孤立した固有値における精度向上のための反復回数 nbi は，固有値のみを求める場合には 200 とし，固有ベクトルを求める場合には，固有ベクトル計算ルーチン内で Rayleigh 商による固有値の改良を行うことから 6 としている[4]．さらにこの反復中で，固有値の存在区間がマシンϵ 以下となれば，打ち切る実装となっている[4]．

この二分法の並列化は容易で，初期値に対する自明な並列処理が可能である．いま各 PE の 一次元的なラベルを $P_{cid}, cid = 0, 1, ..., p-1$ とする．このとき各 PE では

$$k = \lceil n/p \rceil \times cid + 1$$
$$ne = \lceil n/p \rceil$$

のように，小さい方から数えて k 番目の固有値から ne 個の固有値を計算するように変更することで実現できる．

2.5 固有値収集ルーチン

各 PE が計算した固有値 λ_k の存在範囲 を $xi(k)$〜$xs(k)$ とする．初期状態では各 PE は $k = \lceil n/p \rceil \times cid+1, ..., \lceil n/p \rceil \times (cid+1)$ の範囲しか $xi(k)$〜$xs(k)$ を所有していないが，このルーチンで全 PE が λ_kの存在範囲を全て $(k = 1, 2, ..., n)$ を所有するようにする．我々の固有値ソルバーではこの処理を，各 PE が順番に 1 対全放送をすることで実現した．

2.6 固有ベクトル計算ルーチン

三重対角行列に特化した部分枢軸選択付きの LU 分解を用いた逆反復法を実現した．詳細な実装方式については文献 4) の TINVS ルーチンを参照されたい．逆反復法の概略を以下に示す．ここで各固有値は，$\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$のようにソートされているとする．また 行列 Tを三重対角行列とする．ここで，各固有ベクトル $z_i (i = 1, 2, ..., n)$ が収束しない場合の逆反復の最大反復回数 l_{max}は 40 回☆とした．

- **do** $i = 1, ..., n$
  1. $\lambda_i - \lambda_{i-1} \leq eps$ ならλ_iを同グループに入れる．
  2. 区間 [-1,1] の乱数を要素に持つベクトル x_i を生成し，$y_i \equiv x_i / ||x_i||_2$.
  3. LU 分解 $T - \lambda_i = L_i U_i$.
  4. 収束するか l_{max}回まで反復 (最低 4 反復).
     4a. $L_i U_i z_i = y_i$ を解く．
     4b. 同クループ内の $z_j (j < i)$ と再直交化．
     4c. 正規化 $z_i = z_i / ||z_i||_2$
     4d. レーリー商 $\bar{\lambda} = z_i^T T z_i$ による λ_iの改良．
     4e. 収束判定 $||T z_i - \lambda_i z_i||_2 \leq \epsilon$ なら収束．

この逆反復法を並列化する場合，逐次アルゴリズムで考慮しなかった問題が生じる．すなわち重複固有値や密集固有値に対する固有ベクトルを計算する場合，固有ベクトルの精度を向上させる為に，再直交化の処理 (上記の 4b.) を反復の度に行わなくてはならない．この処理により，逆反復法自体の並列性が低くなる．

2.6.1 Peters-Wilkinson のグループ分け方式

このグループ分け方式では，現在計算中の固有ベクトルに対応する固有値との最大距離が eps 内の固有値に対応する固有ベクトルと直交化させるのはもちろん，直交化させる固有ベクトルに対応する固有値に対しても，距離 eps より前にある固有値に対応する固有ベクトルとも直交化させるようにグループ分けする方式である．

ここで距離 eps とは，三重対角行列 Tとし，その対角要素を $\alpha_1, ..., \alpha_n$，副対角要素を $\beta_1, ..., \beta_{n-1}$ とすると，

$$eps = 10^{-3} ||T||_1 \tag{5}$$

である．ここで $||T||_1$ は

$$||T||_1 = \sum_{i=1}^{n} |\alpha_i| + \sum_{i=1}^{n-1} |\beta_i| \tag{6}$$

から得られる数値である．

2.6.2 Gram-Schmidt の直交化方式

直交化の方式には Gram-Schmidt の直交化方式を用いる．現在計算中の固有ベクトルの番号を i, そのベクトルを $\hat{v}_i$ とする．このとき，番号を i の固有ベクトルと直交化が必要な固有ベクトルの個数を n_i，固有ベクトルを $v_{i-1}, v_{i-2}, ..., v_{i-n_i}$とすると，Gram-Schmidt の直交化方式は

$$v_i = \hat{v}_i - \sum_{k=1}^{n_i} (\hat{v}_i, v_{i-k}) v_{i-k} \tag{7}$$

の計算を行うことである．ここで，v_i が求める直交化済の固有ベクトルであり，(,) の表記は内積を意味している．この直交化処理を並列化する場合，実装方法により全く並列化できなくなる．それを以下に示す．

[直交化アルゴリズム 1] (MGS 法)

文献 4) などの逐次プログラムでは，直交化ルーチンは以下の様に実装されている．すなわち

$$\begin{aligned} t_1 &= \hat{v}_i - (\hat{v}_i, v_{i-1}) v_{i-1} \\ t_2 &= t_1 - (t_1, v_{i-2}) v_{i-2} \\ &\cdots \\ t_{n_i} &= t_{n_i - 1} - (t_{n_i - 1}, v_{n_i}) v_{n_i} \\ v_i &= t_{n_i} \end{aligned} \tag{8}$$

のような計算をする．この方法は計算誤差の累積を考慮した直交化法として知られており，MGS(Modified Gram-Schmidt) 法と呼ばれる．MGS 法は，t_1で得ら

☆ この最大反復回数の設定値の根拠はない．EISPACK などでは 5 回としている．また文献 4) では 20 回程度として，さらに最新の固有値を参照して逆反復を繰り返す実装になっている．

れたベクトルをすぐに t_2で用いるなどしていることから明らかなように，フロー依存性があり並列化が全くできない．

[**直交化アルゴリズム** 2] (CGS **法**)

本来の Gram-Schmidt の直交化の式 (7) から考えると，v_i と直交するベクトルを求める計算は，直交化すべき固有ベクトル全部で同時に求めても，直交化ベクトルを求めることはできるはずである．すなわち，

$$\begin{aligned} t_1 &= \hat{v}_i - (\hat{v}_i, v_{i-1})v_{i-1} \\ t_2 &= -(\hat{v}_i, v_{i-2})v_{i-2} \\ &\cdots \\ t_{n_i} &= -(\hat{v}_i, v_{i-n_i})v_{i-n_i} \\ v_i &= t_1 + t_2 + \cdots + t_{n_i} \end{aligned} \tag{9}$$

のような計算で直交化処理を行っても問題はないと考えられる．この方法は古典的な直交化法として知られており，CGS(Classical Gram-Schmidt) 法と呼ばれる．CGS 法は明らかに $t_1, t_2, ..., t_{n_i}$ の計算に対して並列性がある．各 PE に直交化すべき固有ベクトルが分散して所有している場合，$\hat{v}_i$ が得られた時点で，各 PE が並列に直交化処理をすることが可能となる．理論的には CGS 法は直交化する各ベクトルそれぞれの直交化の崩れがゼロならば MGS 法と精度は同じになるが，直交化の崩れが大きくなると精度の保証ができなくなることに注意しておく．

[**直交化アルゴリズム** 3] (**混合法**)

直交化アルゴリズム 1 と 2 を混合した再直交化法を考える．すなわち，逆反復中の再直交化処理には直交化アルゴリズム 2(以下 CGS 法) による再直交化処理を行うが，収束したら直交化アルゴリズム 1(以下 MGS 法) による再直交化処理を 1 回だけ行う．この直交化方式は直交性と並列化による高速化のトレードオフを解消できる可能性がある☆.

2.6.3 固有ベクトルの計算順序

ここでは Peters-Wilkinson によるグループ分け方式において，距離 *eps* でグループ分けした固有ベクトルをどのように計算するか述べる．本実装方式では，計算すべき固有ベクトルの数は各 PE に均等になる ($\lceil n/p \rceil$ 個づつ所有する) ように分散する．このとき，グループ分けした固有値ごとに，グループ内で番号をつける (グループ内番号)．そして以下のような順番で固有ベクトルを計算する．

1) PE 内で所有するグループ内番号 1 の固有ベクトルを計算する．
2) (グループ番号が若い方から順に) 固有ベクトルの計算に入る．
   if (固有ベクトルが PE 内にある) **then**
   ・再直交化して固有ベクトル計算;
   else
   ・**(直交化 Alg.1 の場合)**
   – グループ内番号 1 の固有ベクトルを所有する PE に転送して P_{cid-1} から受信待機;
   ・**(直交化 Alg.2 の場合)**
   – グループ内番号 1 の固有ベクトルを所有する PE から P_{cid-1} までの PE に転送;
   – グループ内番号 1 から P_{cid-1}までの PE から受信待機;
   ・再直交化処理;
   ・固有ベクトル計算;
   endif
3) **if** (PE 間をまたぐグループがある) **then**
   ・固有ベクトル受信待機;
   ・再直交化処理;
   ・固有ベクトル送信;
   endif

2.7 Householder 逆変換ルーチン

Householder 逆変換では，得られた三重対角行列 T の固有ベクトル$\hat{w}_i$, $i = 1, 2, ..., n$ を，元の対称行列 A の固有ベクトル w_i, $i = 1, 2, ..., n$ に変換する．この処理は

$$\begin{aligned} w_i &= H^{(1)}H^{(2)}\cdots H^{(n-2)}\hat{w}_i \\ &= (I - \sigma_1 u_1 u_1^T)(I - \sigma_2 u_2 u_2^T)\cdots \\ &\quad \cdots (I - \sigma_{n-2} u_{n-2} u_{n-2}^T)\hat{w}_i, \quad i = 1, 2, .., n \end{aligned} \tag{10}$$

である．このルーチンを実現する際に注意すべき事項は，三重対角化ルーチンで得られる枢軸ベクトル $u_1, u_2, ..., u_{n-2}$， およびスカラ $\sigma_1, \sigma_2, ..., \sigma_{n-2}$が必要となるが，枢軸ベクトル $u_1, u_2, ..., u_{n-2}$ はその要素が各 PE に分散されて所有されている点である．

ここで，各枢軸ベクトル u_k が分散されて所有していることから，各 w_iごとに 何度も u_k を集める式 (10) の方法は無駄が多い．よって，計算順序を入れ換える方法を実装する．すなわち

$$\begin{aligned} \hat{w_i}^1 &= H^{(n-2)}\hat{w}_i, \qquad i = 1, 2, ..., n \\ \hat{w_i}^2 &= H^{(n-3)}\hat{w_i}^1, \qquad i = 1, 2, ..., n \\ &\vdots \\ \hat{w_i}^{n-3} &= H^{(2)}\hat{w_i}^{n-4}, \qquad i = 1, 2, ..., n \\ w_i &= H^{(1)}\hat{w_i}^{n-3}, \qquad i = 1, 2, ..., n \end{aligned} \tag{11}$$

のような計算順序で変換を行う．式 (11) の各行の処理には，各固有ベクトルに対して自明な並列性がある．よって，これら各固有ベクトルの変換処理に対して並列実装を行う．

☆ 逆反復中の再直交化処理は，重複及び密集固有値に対して収束方向を定めるために必須である．ここで，再直交化処理に CGS 法を用いているので，直交化の崩れが大きい場合は解法が破綻すると思われるが，混合法の場合では再直交化すべき固有ベクトルは全て MGS 法で直交化済みである．即ち，直交化の崩れに関して，CGS 法のみを用いて再直交化した場合と比較すると極めて小さい (計算誤差を累積した固有ベクトルを参照しない) と言える．ゆえに混合法での逆反復中の CGS 法が破綻する場合は少ないと推察される．

このときの問題は，どのように分散されて所有している枢軸ベクトル u_k を収集するかであるが，この処理は以下のように容易に実装できる．1) $ncidx = 0$ のプロセッサが所有している部分的な u_k を，$ncidy$が等しい PE に 1 対多放送をする (マルチキャスト)．2) 同様な処理を $ncidx = 1$ の PE が行う．3) 1)，2) の処理を $ncidx = \sqrt{p} - 1$ の PE まで繰り返す．

3. 性能評価

この章では，日立 SR2201 に本稿で示した並列固有値ソルバーを実現し，その性能について評価した結果を記す．ここで，SR2201 の各 PE の理論ピーク性能は 300MFlops, PE 間は三次元クロスバ網で結合されており，その最大転送性能は 300Mbyte/秒である☆．

3.1 Frank 行列による実行性能

ここで性能評価およびデバックのため，以下の行列の固有値を求めた．

$$A_n = (a_{ij}),\ a_{ij} = n - \mathbf{max}(i,j) + 1 \qquad (12)$$

この行列は Frank 行列と呼ばれており，固有値は解析的に求めることができ

$$\lambda_k = \frac{1}{2\left(1 - \cos\frac{(2k-1)}{(2n+1)}\pi\right)}, k = 1, 2, ..., n \quad (13)$$

となる．Frank 行列の固有値分布の特徴として, k が小さい領域では分散しているが，大きくなるにつれ固有値が密集してくることが挙げられる．すなわち，密集固有値に対する固有値問題の典型的な例題である．

3.1.1 全固有値を求める場合

ここでは日立 SR2201 を 4PE〜256PE まで用いて，10000 次元の行列の全固有値計算処理の時間を計算した．その結果を表 1 に示す．表 1からわかるように，

表 1 固有値 10000 個の計算時間 [秒]
($nbi = 200$, 真値との最大相対誤差 0.39×10^{-7})

PE 数	4	16	64	256
三重対角化	4562.9	886.7	216.8	85.2
(占める割合 %)	(96.9%)	(95.1%)	(94.9%)	(95.3%)
再分散	0.049	0.025	0.025	0.034
固有値探索	142.3	45.1	11.4	4.10
(占める割合 %)	(3.02%)	(4.84%)	(5.03%)	(4.5%)
総合時間	4705.3	931.9	228.3	89.3
速度向上率				
(4PE を基準)	1	5.05	20.6	52.6

並列処理においても三重対角化ルーチンが実行時間の大部分 (90%以上) を占めている．一方，再分散ルーチンは高々全体の 5%であり，この場合は改善の必要はないと思われる．また，並列固有値探索ルーチンはベクトル化など高速化技法を施していないが，実行時間全体に占める割合は多くても 5%前後であり，特に早急な改善の必要はないものと考えられる．

一方，$p = 1024$ に固定して，n を変化させて時間を測定した．その結果を表 2に示す．ここで GFLOPS 値の計算には，行列消去部で対称性を利用し下三角部分のみ消去する場合は $4/3n^3$を用いるが，本並列アルゴリズムでは行列全体を更新しなくてはならないことから $8/3n^3$ を用いている．

表 2 $p = 1024$ での固有値の計算時間 [秒]
(Frank 行列, $nbi = 200$)

固有値数	30000	60000
三重対角化	534.6	3308
(占める割合 %)	(98.2 %)	(98.9 %)
GFLOPS	134.6	174.1
再分散	0.067	0.110
固有値探索	9.602	33.9
(占める割合 %)	(1.76 %)	(1.01 %)
総合時間	544.3	3342
真値との最大相対誤差	0.17×10^{-6}	0.71×10^{-6}

表 2から，n が 30000, 60000 と大きくなるにつれて，三重対角化ルーチンの占める割合も大きくなっていく．GFLOPS 性能では $n = 60000$ の時，ピーク性能の 56%余りを達していることから，実装技術が極端に貧弱とは考えられない．よって，さらなるアルゴリズムの改良および通信処理の高速化が必要となろう．

3.1.2 全固有値および全固有ベクトルを求める場合

[各処理の割合] ここでは，PE 台数 4〜64 台まで用いて，$n = 3000$ の Frank 行列の全固有値および全固有ベクトルを，MGS 法で求めた場合の各処理の実行時間と全体の処理時間に対する占める割合を表 3に示す．表 3によると，固有ベクトル計算のための並列逆反復法が 90%以上を占める．さらに PE 数が増加しても，総合時間がほとんど減少しない．これは再直交化処理が並列化できていないことによるものである．

[直交化アルゴリズムと実行時間] 次に直交化処理と実行時間との関係を調べるため，MGS 法，CGS 法および全く直交化処理をしない場合の固有ベクトル計算時間を比較する．その結果を図 2に示す．図 2中の Alg.1 は MGS 法, Alg.2 は CGS 法，Not Orthogonalized は再直交化処理なしの実行時間を示している．図 2から，直交化処理を並列化することで，この行列の場合は MGS 法を用いた直交化処理 (Alg.1) に比べ 2.2 倍 (4PE)，6.7 倍 (16PE)，8.2 倍 (64PE) の速度向上が達されることがわかる．

また，直交処理を行わない場合はリニアな速度向上が得られるばかりか，CGS 法よりも 169 倍〜721 倍

☆ 東京大学大型計算機センターが所有している 1024PE の SR2201 のうち 1024PE 全てを使用した．また，コンパイラとして 日立の最適化 FORTRAN90 V02-03，オプションとして-W0,'PVEC(PVFUNC(1), VERCHK(0), DIAG(1)), opt(o(s), fold(2), prefetch(1), rapidcall(1), ischedule(3), reroll(1), scope(1), split(2), uinline(2)) を指定した．測定日は 1997 年 9 月 15 日から 10 月 3 日である．

表 3　Frank 行列 3000 の全固有値，固有ベクトル計算時間 [秒] ($nbi = 6$, 直交化アルゴリズム 1 (MGS 法))

PE 数	4	16	64
三重対角化	85.6	26.4	10.8
(占める割合 %)	(2.31%)	(0.77%)	(0.32%)
再分散	0.005	0.003	0.006
(占める割合 %)	(0.00%)	(0.00%)	(0.00%)
固有値探索	4.08	1.04	0.28
(占める割合 %)	(0.11%)	(0.03%)	(0.01%)
固有値収集	0.01	0.05	0.27
(占める割合 %)	(0.00%)	(0.00%)	(0.01%)
全固有ベクトル求解	3453	3376	3405
(占める割合 %)	(93.0%)	(97.9%)	(99.2%)
Householder 逆変換	167.7	45.1	14.8
(占める割合 %)	(4.52%)	(1.26%)	(0.43%)
総合時間	3710	3447	3431

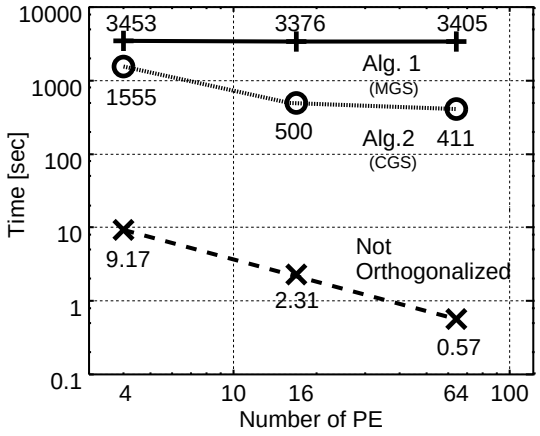

図 2　並列逆反復法における再直交化アルゴリズムと実行時間 (Frank 行列，$n = 3000$)

も速度向上がなされ，全体の計算時間に占める割合も 3%ほどになる．これらのことから，固有ベクトル計算時間の 90%以上は直交化処理およびそれに伴う計算待ち時間であることがわかる．

[**直交化アルゴリズムと直交性**]　CGS 法による速度向上は確認された．しかし，計算方式を変えることで極端に解の精度が悪化するならば実用上問題となる．本節では，固有ベクトルの直交性を $Ortho$ で評価する．

ここで 直交性 $Ortho$ は $V \equiv [w_1, w_2, ..., w_n]$ とし，$A \equiv V^T V - I = (a_{ij}), i, j = 1, ..., n$ とするとき，$Ortho \equiv \sqrt{\sum_{i,j} a_{ij}^2}$とした．

その結果を表 4に示す．表 4から MGS 法と CGS 法では直交性の変化は認められない．また，当然ながら直交化を行わないと直交性は悪化した．ここで PE 台数が 64 台のとき，MGS 法と CGS 法において 4 台と 16 台の直交性に比べ直交性が悪くなっている．この原因は逐次では固有値が大きくなるにつれ，固有値の存在範囲の初期値が充分小さくなっているが，並列化することでその情報が後の固有値探索の時に伝わらなくなる．その結果として，固有値の追い詰めが甘くなり，精度が低下する．すなわち，固有ベクトル計算時の初期固有値の精度の悪さが直交性の悪さに影響していると考えられる．以上の結論として，現状の二分法での固有値追い詰め回数 $nbi = 6$ を大きくすれは，PE 台数が増えても同一の直交性が得られると考えられる．

表 4　Frank 行列 $n = 3000$ の固有ベクトルの直交性 ($nbi = 6$)

PE 数	4	16	64
Alg.1	$.283 \times 10^{-12}$	$.280 \times 10^{-12}$	$.690 \times 10^{-11}$
Alg.2	$.278 \times 10^{-12}$	$.274 \times 10^{-12}$	$.690 \times 10^{-11}$
直交化なし	$.978 \times 10^{-7}$	$.152 \times 10^{-6}$	$.330 \times 10^{-5}$

[**初期固有値の精度と直交性**]　次に，先程の推察が正しいことを示すため，直交性と直交化距離 eps との関係を調べる．PE 台数を 64 に固定して二分法での固有値の追い詰め回数 nbi を 2〜40 まで 2 づつ増加させる場合と，直交化距離を従来の $||T||_1 \times 10^{-3}$ではなく，$||T||_1 \times 10^{3}$ から $||T||_1 \times 10^{-15}$ まで，10 倍 刻みで変化させた場合について，直交性 $Ortho$ の変化を調べた．それを 図 3に示す．図 3の結果から，直交化

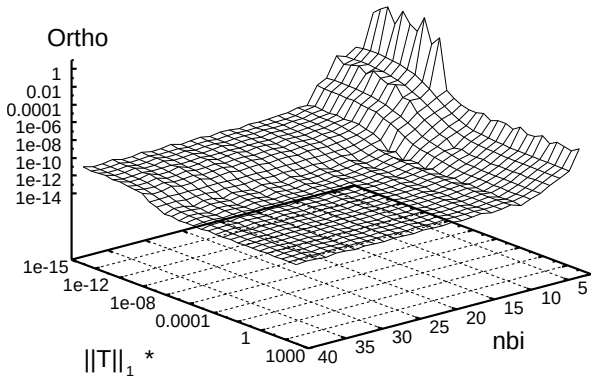

図 3　初期固有値の精度と直交性 (Frank 行列，直交化 Alg.2(CGS 法), $n = 3000, p = 64$)

距離 eps を固定した場合，2 分法の追い詰め回数 nbi を大きくすると，直交性が良くなる傾向にあることがわかる．すなわち固有値を精度よく求めておけば，固有値の精度が悪い場合に比べて直交性の悪化をより少なくできると推察される．

[**直交化距離 eps に関する直交性と実行時間**]　$nbi = 100$ と充分大きくした場合，すなわち固有値を精度良く求めておいた場合に，直交化距離と直交性の関係がどのように変化するか実験を行う．PE 台数を 4，16，64 とした場合の結果を，図 4に示す．ここで $max||Ax - \lambda x||_2$ は残差ベクトルの 2-ノルムの最大値である．図 4から，PE 台数が変化しても直交性は変化しない．直交化距離が $||T||_1 \times 10^{-8}$ の地点から直交性が急に悪くなり．$||T||_1 \times 10^{-11}$ ぐらいから一

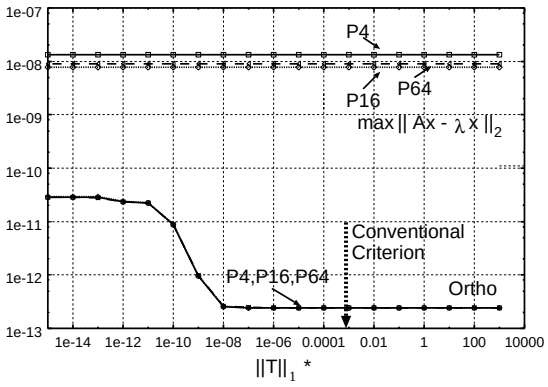

図 4　再直交化距離と精度
(Frank 行列, 直交化 Alg.2(CGS 法),$n = 3000$)

定となる．$||T||_1 \times 10^{-11}$のとき，直交化される固有ベクトルの数は 0 なので直交化なしと同じである．

結果として，この条件の場合は直交化しないと 約 2 桁の直交性の低下になることがわかる，処理速度の観点からいえば，直交化しなければリニアな速度向上がなされ，かつ直交化を行う場合に比べて 169 倍〜721 倍も速度向上がなされることを考慮すると，この 2 桁の直交性の低下が実際の問題でどのように影響するのか調べることが高速ルーチン作成の鍵といえよう．

3.2　glued Wilkinson 行列による実行性能

ここでは重複固有値の数が甚だしい場合に直交化精度と実行速度を評価するため，以下の行列の固有値を求めた．

$$W_{21}^{+} = \begin{bmatrix} 10 & 1 & & & & & \\ 1 & 9 & 1 & & & & \\ & 1 & \ddots & \ddots & & & \\ & & \ddots & 0 & \ddots & & \\ & & & \ddots & \ddots & 1 & \\ & & & & 1 & 10 \end{bmatrix}$$

$$W_{g}^{+} = \begin{bmatrix} W_{21}^{+} & \delta & & & & \\ \delta & W_{21}^{+} & \delta & & & \\ & \delta & \ddots & \ddots & & \\ & & \ddots & W_{21}^{+} & \delta & \\ & & & \delta & W_{21}^{+} \end{bmatrix}$$

行列 W_{21}^{+} は 21 次元の Wilkinson 行列であり，W_{g}^{+} は glued Wilkinson 行列として知られている．この glued Wilkinson 行列の固有値はその多くが重複して群をなしており，重複固有値の問題として最も難しい問題の一つとして知られている．なお，本実験では $\delta = 10^{-14}$ として性能を測定する．

3.2.1　全固有ベクトルを求める場合

glued Wilkinson 行列は三重対角行列であるから，三重対角化処理と逆変換処理は行う必要がない．ここでは直交性が問題となると考えられるので，全固有ベクトルを求める場合において，直交性と実行時間を評価する．ここで，二分法による追い詰め回数 nbi は 100 としている．

図 5に，我々が示した三種の再直交化アルゴリズムを glued Wilkinson 行列の固有ベクトル計算に適用した場合の直交性を示す．ここで，行列の次元は 840 である．この行列の固有値は，-1.1〜10.8 の間に 40 個 ごとに重複した固有値群をなすような分布になっている．

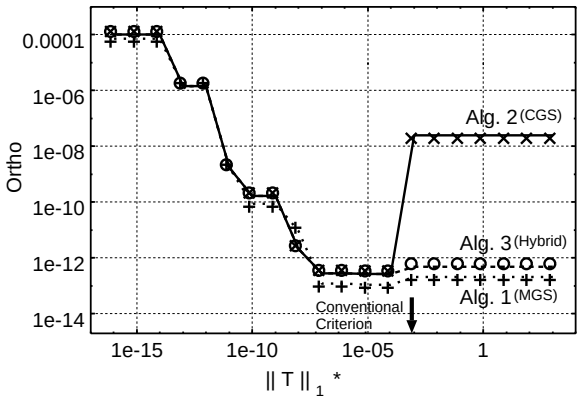

図 5　逆反復法における再直交化アルゴリズムと直交性
(glued Wilkinson 行列, $n = 840, p = 16$)

図 5では，CGS 法による直交性が従来の基準で直交化した場合，4 桁程度 MGS 法に対して悪化している．しかしながら，CGS 法と MGS 法とを混合した直交化アルゴリズム 3(以下，混合法) では，MGS 法に比べても直交性の崩れが少なくなっていることが分かる．

次に，再直交化処理のアルゴリズムの違いにより，実行時間がどのように変化するか示す．図 6は実行時間を示している．ここでプロセッサ台数は 16 台を用いている．

図 6から従来の直交化の基準を用いた場合，MGS 法に対し，CGS 法は 4 倍程度の高速化がなされている．一方，混合法では，CGS 法に比べて 1.6 倍程度高速に処理がなされている．この結果はある意味では驚くべき結果である．なぜならば，直観的に混合法は CGS 法よりも遅くなると考えられるからである．このような現象が観測された理由として，直交性が CGS 法に対して良くなったことにより，収束までの反復回数が減り，その結果として速度向上がなされたのではないかと考えられるが，詳しい解析は今後の課題である．

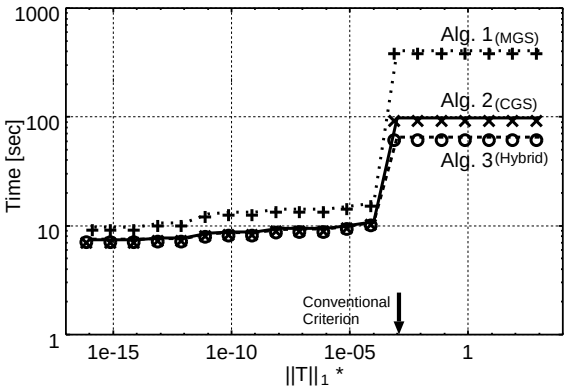

図 6　逆反復法における再直交化アルゴリズムと並列実行速度
(glued Wilkinson 行列, $n = 840, p = 16$)

4. 関 連 研 究

MGS 法を並列化して高速化を図る方針の並列アルゴリズムとして，Lo らの並列アルゴリズム[5]が挙げられる．Lo らのアルゴリズムでは，1) 重複固有値を除く固有値を，全て別のグループとする．2) そのグループ毎に独立に逆反復法を行う．3) 隣接する固有値を MGS 法で直交化する．ここで 3) の処理は，本論文の方式とは異なり 2 重ループとなるので並列化が可能である．

また固有ベクトルの直交化に関する処理の制約を緩和して並列性を増加させる方針のアルゴリズムとして，直野・猪貝・山本らにより提案されたマルチカラー逆反復法[2]が挙げられる．このアルゴリズムは理論上は現在計算中の固有ベクトルに対応する固有値との距離が *eps* 内の固有値に対応する固有ベクトルのみ直交化させれば十分なことを利用し，その距離内で直交化がなされるように計算のスケジューリングを行う方法である．

5. おわりに

本報告では，分散メモリ型並列計算機で全固有値，全固有ベクトルを求めることが可能な並列固有値ソルバーを実現して実行性能を報告した．現状では，最も問題となる処理は逆反復における再直交化処理である．本報告において，この再直交化処理を並列化する実装を試み，従来の再直交化では全く並列処理の効果が得られなかった問題に対して，CGS 法を用いた再直交化処理を行えば同精度の解が得られる条件で約 8.2 倍程度の高速化が得られる場合があることを示した．

一方，今回我々は CGS 法による並列再直交化を行いつつ，収束した時点で MGS 法を用いて直交性を向上させる混合的な再直交化アルゴリズムを示した．glued Wilkinson 行列による実験結果から，甚だしい重複固有値があるような問題を処理する場合においては，このアルゴリズムは精度と実行速度の両方の面においても，その他の再直交化の方式に対して優れている可能性がある．

今後の課題として，直交化アルゴリズムに関する性能評価をもっと多くの例題とアルゴリズムで追試する必要があると考えている．例えば，関連研究で列挙した手法と本論文で提案する手法に関して，性能や精度がどのように違うのか詳細に解析することも今後の課題である．次に，今回提案した混合的な再直交化法中の CGS 法であるが，この直交化処理は最初の数回の反復で固有ベクトルの方向が定まれば事実上は不要な処理といえる．よって CGS 法で収束まで毎回再直交化する必要はないかもしれない．いずれにせよ，実験的もしくは理論的に研究しないと一概にはいえない．最後に，本報告で紹介したソルバーを分子軌道計算等の実際の工学シミュレータに実装して実用性の評価を行うことも今後の課題である．

謝辞　日頃討論頂いた，東京大学大学院理学系研究科情報科学専攻金田研究室の諸氏に感謝致します．また，有益な意見をいただいた査読者の各位に感謝致します．

参 考 文 献

1) Demmel, J. and Stanley, K.: The Performace of Finding Eigenvalues and Eigenvectors of Dense Symmetric Matrices on Distributed Memory Computers, Technical report, University of Tennessee - Knoxville, UT-GS-94-254.
2) 直野健, 猪貝光祥, 山本有作: 並列固有値ソルバーの開発と性能評価, 並列処理シンポジウム JSPP'96 論文集, pp. 9–16 (1996).
3) 片桐孝洋, 金田康正: 分散メモリ型並列計算機による Householder 法の性能評価, 情処研報, 96-HPC-62, pp. 111–116 (1996).
4) 村田健郎, 名取亮, 唐木幸比古: 大型数値シミュレーション, 岩波書店, pp. 157–165 (1990).
5) Lo, S.-S., Philippe, B. and Sameh, A.: A Multiprocessor Algorithm for the Symmetric Tridiagonal Eigenvalue Problem, *SIAM J. Sci. Stat. Comput.*, Vol. 8, No. 2, pp. s155–s165 (1987).