共有メモリ型並列計算機向けの高並列固有ベクトル解法とSR8000での評価

山 本 有 作† 猪 貝 光 祥†† 直 野 健†

分子設計，構造計算などの分野で大きな需要がある対称行列向けの固有ベクトル解法として，共有メモリ型並列計算機に適した新しいアルゴリズムを開発した．本手法では，従来の逆反復法で並列化のネックとなっていた修正グラム・シュミット方式の直交化をとりやめ，すでに求めた固有ベクトルの直交補空間を陽的に保持して，その基底をハウスホルダー変換で更新していくことにより，直交性の高い固有ベクトルを求める．並列性が高いハウスホルダー変換を用いることで，並列粒度を従来の $O(N)$ から $O(N^2)$ に高め，並列起動回数を大幅に削減した．SR8000 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，1000 元の行列の全固有ベクトルを求める場合，本手法は従来法の 2.4 倍の性能を達成した．

A New Algorithm for Accurate Computation of Eigenvectors on Shared-memory Parallel Processors and Its Evaluation on the SR8000

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

We developed a new algorithm for computing the eigenvectors of a real symmetric matrix on shared-memory parallel computer, which will be useful in the area of molecular design and structural analysis. Instead of using the modified Gram-Schmidt orthogonalization, which was the bottleneck in parallelizing the conventional inverse iteration algorithm, we chose to hold the basis of orthogonal complementary subspace of the calculated eigenvectors explicitly, and successively modify it by the Householder transformation. Thus, the parallel granularity was increased from $O(N)$ to $O(N^2)$, and the overhead due to the parallel startup time was drastically reduced. We evaluated our algorithm on 1 node of the SR8000 (a shared-memory parallel computer with 8 processors) and obtained performance 2.4 times higher than that of the conventional method, when computing all the eigenvectors of a matrix of order 1000.

1. はじめに

実対称行列およびエルミート行列の固有値・固有ベクトルの計算は，連立一次方程式の解法と同様，広い分野で使われる基本的な線形計算の 1 つであり，分子計算，構造計算などに幅広い応用を持つ．近年では，シミュレーションの大規模化・精密化にともない，これらの分野でも並列計算機の利用への要求が高まっており，特に計算量の多い固有値・固有ベクトル計算部分の効率的な並列化は重要な課題である．

これらの固有値・固有ベクトル計算を行うための標準的な解法として，入力行列 A をハウスホルダー変換により実対称三重対角行列 T に変換し，二分法により T の固有値を求め，この固有値を用いて逆反復法により T の固有ベクトルを求め，最後にハウスホルダー逆変換により T の固有ベクトルを求める方法がある．このうち，ハウスホルダー変換，二分法，ハウスホルダー逆変換の部分については，効率的な並列化手法が知られており[1),4),5)]，並列計算機上でのライブラリとしての実装も行われている[1),5)]．しかし固有ベクトルを求めるための逆反復法（以下，古典的逆反復法と呼ぶ）については，固有ベクトルの直交性を高めるための直交化計算がネックとなって並列性が低く，逐次アルゴリズムをそのまま並列化したのでは，並列計算機上で十分な性能が得られないことが知られている[1)]．

そのため，古典的逆反復法を改良し，並列性を高めた解法として，Dhillon によるアルゴリズム[8)]やマル

† 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.

チカラー逆反復法[1]などの解法が提案されてきた．このうち Dhillon のアルゴリズムは，古典的逆反復法における LU 分解で Twisted LU 分解と呼ぶ特殊な方法を用いることにより，直交化計算を行うことなく直交性の高い固有ベクトルを求められるようにした手法である．この手法は各固有ベクトルの計算を独立に行えるため，原理的にはきわめて高い並列性を持ち，注目を集めている．しかし，このアルゴリズムでは前提として固有値を今までよりも格段に高い精度で求めておく必要があるため，従来の二分法による固有値の計算がそのままでは利用できない．また，固有値が縮重している場合には，固有ベクトルの直交性が低下するという問題があることが知られている．一方，マルチカラー逆反復法は，古典的逆反復法において，誤差評価に基づき必要度の低い直交化演算を省くことにより，演算量の削減と並列性の向上を実現した手法である．この手法では，グラフ理論における頂点の塗り分けアルゴリズムの利用によって，古典的逆反復法では並列化の効果が得られない問題に対しても並列性を引き出すことを可能にしており，様々な例題で有効性が確認されている[1]．しかしマルチカラー逆反復法では，直交化を削減したことにより，古典的逆反復法に比べて固有ベクトルの直交性が 1 桁程度落ちることが知られている．これは多くの応用においてはほとんど問題がないが，従来法と同等の精度が要求される用途には対応が困難であった．

そこで本論文では，共有メモリ型の並列計算機向けに，従来の逆反復法と同等の精度を保ちつつ，並列性を大幅に向上させた新しいアルゴリズムとして，ハウスホルダー逆反復法と呼ばれる解法を提案する．以下では，まず 2 章で従来の逆反復法とその問題点を述べた後，3 章でハウスホルダー逆反復法のアルゴリズムと特徴を述べる．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，性能と精度の評価を行う．最後に 5 章でまとめを述べる．

2. 従来の逆反復法とその問題点

2.1 古典的逆反復法

2.1.1 古典的逆反復法のアルゴリズム

N 次元の対称三重対角行列 T とその固有値 λ_i ($\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_N$) が与えられたとき，各固有値 λ_i に対応する固有ベクトル $\mathbf{v}_i$ を求める問題を考える．逆反復法では，$\mathbf{v}_i$ を求めるため，λ_i の近似値 λ'_i と適当な初期値 $\mathbf{v}_i^{(0)}$ から出発して，反復計算

$$\mathbf{v}_i^{(m)} := (T - \lambda'_i I)^{-1} \mathbf{v}_i^{(m-1)} \tag{1}$$

を行う．ここで，I は単位行列である．右辺のベクトル $\mathbf{v}_i^{(m-1)}$ を T の固有ベクトルのなす正規直交系 $\{\mathbf{v}_j\}_{j=1}^N$ で展開して考えると，$(T - \lambda'_i I)^{-1}$ をかけることで固有ベクトル $\mathbf{v}_j$ 方向の成分は $(\lambda_j - \lambda'_i)^{-1}$ 倍されるため，λ'_i が λ_i の十分良い近似値ならば，反復により $\mathbf{v}_i$ 方向の成分が増幅され，$\mathbf{v}_i^{(m)}$ は求める固有ベクトル $\mathbf{v}_i$ に収束すると期待される．

しかし実際の計算では，数値誤差のため，計算結果のベクトルに他の固有ベクトル成分 $\mathbf{v}_j$ の混入が残る．このため，実対称行列の固有ベクトルが持つべき直交性，すなわち $\{\mathbf{v}_j\}_{j=1}^N$ が正規直交系をなすという性質が十分に保証されないという問題が生じる．そこで逆反復法では，式 (1) による反復計算に加え，計算結果のベクトルから，それ以前に計算した固有ベクトルの成分を取り除く直交化という処理を行う．直交化には，修正グラム・シュミット法を用いる．誤差評価によれば，$\mathbf{v}_i$ への $\mathbf{v}_j$ の混入の大きさは $(\lambda_j - \lambda_i)^{-1}$ にほぼ比例するため[1),3)]，直交化は固有値どうしが近い固有ベクトルに対してのみ行うのが普通である．

直交化の処理を加えた逆反復法（古典的逆反復法）のアルゴリズムを**図 1** に示す[3),6)]．ここで・は内積，$||*||$ はベクトルのノルムを表す．なお，固有値 λ_i に縮重または縮重に近い状況がある場合には，初期値の設定方法を変える，固有値を微小にずらすなどの処理が必要となるが，その点については省略してある[3)]．

アルゴリズム 1 において，最内側の k に関するループが直交化のための修正グラム・シュミット法であり，求めた固有ベクトル $\mathbf{v}_i^{(m)}$ を，グループ $G(i)$ 内の以前に求めた固有ベクトル $\mathbf{v}_k$ に対して直交化する．本アルゴリズムにおける固有値のグループ化の例を**図 2** に示す[1)]．

2.1.2 古典的逆反復法の問題点

古典的逆反復法では，直交化演算が固有値グループ $G(i)$ 内に限られるため，異なるグループに属する固有ベクトルの計算は独立に行える．したがって，並列化する場合には各グループをそれぞれ 1 台のプロセッサに担当させる方式が自然であり，分散メモリ型並列計算機向けの行列計算ライブラリ ScaLAPACK[5)]では，この方式による並列化を行っている．

しかし，行列の次数 N が大きくなるにつれ，隣り合う固有値の間隔は狭くなるため，1 つのグループに属する固有値の数は増大する．特に，従来から広く使われている $\varepsilon = 10^{-3}||T||_1$ という基準値[3)]を用いた場合は，$N = 1000$ 以上になると，ほとんどの固有値が 1 つのグループに属してしまうことが知られている[1)]．そのため，この並列化方式では，1 台のプロセッサが

[アルゴリズム１： 古典的逆反復法]

固有値のグループ化：２個の固有値の距離が基準値 ε 以内の場合，それらを同じグループに属すると定義し，固有値のグループ分けを行う。第 i 番目の固有値が属するグループを G(i)と表す。

```
do i=1, N
| 固有ベクトル v_i の初期値 v_i^(0) を設定。
| m=1
| v_i^(m) が収束するまで以下を繰り返す。
| | v_i^(m) := ( T- λ_i I )^-1 v_i^(m-1)
| | すべての k∈ G(i)（ただし k<i）について以下を繰り返す。
| |     v_i^(m) := v_i^(m) – (v_i^(m) ・ v_k) v_k
| | v_i^(m) := v_i^(m) / ‖ v_i^(m) ‖
| v_i := v_i^(m)
end do
```

図 1　古典的逆反復法のアルゴリズム
Fig. 1　Algorithm of the classical inverse iteration method.

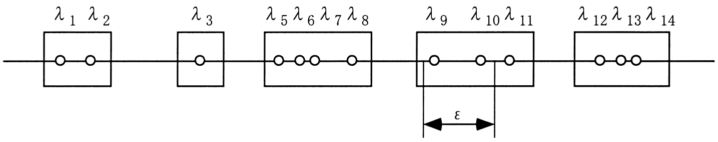

図 2　古典的逆反復法における固有値のグループ化
Fig. 2　Grouping of the eigenvalues in the classical inverse iteration method.

ほとんどの計算を行うことになり，並列化の効果はほとんど得られない．

グループに関する並列化がうまくいかない場合，より内側のループにおいて並列化を行うことになるが，修正グラム・シュミット法は k に関して逐次性があるため，残された並列化方式は，直交化の中心演算 $\mathbf{v}_i^{(m)} := \mathbf{v}_i^{(m)} - (\mathbf{v}_i^{(m)} \cdot \mathbf{v}_k)\mathbf{v}_k$ において，ベクトルの内積演算 $\alpha = \mathbf{v}_i^{(m)} \cdot \mathbf{v}_k$，および AXPY 演算 $\mathbf{v}_i^{(m)} := \mathbf{v}_i^{(m)} - \alpha\mathbf{v}_k$ をそれぞれ並列化することである．しかしこの方式では，ほとんどすべての固有値が 1 つのグループに属する場合，全固有ベクトルを求めるための並列起動回数が $O(N^2)$ となる．ハウスホルダー変換，ハウスホルダー逆変換など，固有値計算の他の部分では並列起動回数がいずれも $O(N)$ であるのに比べると，これは並列化のための大きなオーバヘッドになる．

以上より，古典的逆反復法では，ほとんどの固有値が 1 つのグループに属してしまう場合，効果的な並列化の方法がなかった．

2.2 マルチカラー逆反復法

2.2.1 マルチカラー逆反復法のアルゴリズム

古典的逆反復法において直交化の部分を改良し，計算の並列性を大きく向上させたアルゴリズムがマルチカラー逆反復法[1)]である．古典的逆反復法では，まず固有値をグループ化し，同じグループ内で直交化を行っていたが，$\mathbf{v}_i$ への $\mathbf{v}_j$ の混入の大きさが $(\lambda_j - \lambda_i)^{-1}$ にほぼ比例することを考えると，同じグループ内であっても，十分離れた固有値に属する 2 本の固有ベクトルについては，成分の混じり合いは少ないはずである．そこでマルチカラー逆反復法では，グループ化を廃止し，2 個の固有値の距離が基準値 ε 以下の場合に限って直交化を行う．マルチカラー逆反復法で直交化を行う関係にある固有値ペアの例を**図 3** に示す[1)]．互いに線で結ばれた固有値が，直交化を行うべきペアである．なお，この図の固有値分布では，隣接する固有値間の距離はすべて ε 以下であり，古典的逆反復法では全固有値が 1 つのグループになってしまうことに注意する．こうしてマルチカラー逆反復法では，直交化すべき固有ベクトルの数を古典的逆反復法に比べて大きく削減している．

さらに，直交化すべき固有ベクトルの削減により，直交化計算の並列性も増大する．なぜなら，図 3 において固有値 λ_1 と λ_4 とに属する固有ベクトルは互いに直交化の関係にないため，これらを独立に計算することができるからである．一方，固有値 λ_1 と λ_2 とに属する固有ベクトルは互いに直交化の関係にあるた

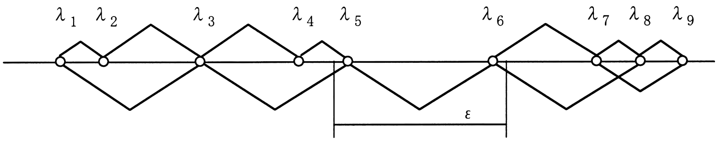

図 3 マルチカラー逆反復法で直交化を行う固有値ペア
Fig. 3 Eigenvalue pairs to be orthogonalized with each other in the multicolor method.

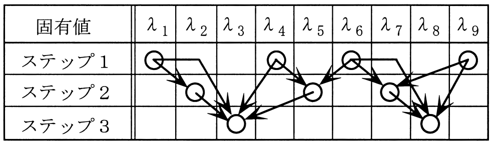

図 4 図 3 の固有値分布に対する固有ベクトルの計算順序
Fig. 4 The order of computation of the eigenvectors corresponding to the eigenvalue distribution of Fig. 3.

め，これらを同時に計算することはできない．いま，十分な数のプロセッサが利用できると仮定して，このような直交性による制約条件の下で，最小何ステップですべての固有ベクトルが計算できるかを考える．すると，図 3 において互いに線で結ばれた固有ベクトルは異なるステップで計算しなくてはならないから，最小のステップ数は，図 3 において互いに線で結ばれた頂点を異なる色で塗るという条件の下で，頂点を塗り分けるのに必要な最小の色の数に等しい．この色の数，および塗り分け方は，グラフ理論の近似アルゴリズムを用いて求めることができる．マルチカラー逆反復法では，このようにして固有ベクトルの最適な計算順序を求め，直交化演算における並列性を最大限に引き出している．図 3 の固有値分布に対する固有ベクトルの計算順序の例を**図 4** に示す．図中の矢印は，先に求めた固有ベクトルを用いて，後の固有ベクトルを直交化することを表す．

2.2.2 マルチカラー逆反復法の問題点

マルチカラー逆反復法は，古典的逆反復法に比べて演算量が小さく，並列性も高いため，数多くの例題に対して高い並列化効率を達成している[1)]．

しかし，マルチカラー逆反復法では，直交化を削減した影響で，古典的逆反復法に比べると固有ベクトルの直交性が 1 桁程度落ちることが知られている[1)]．これは多くの応用においてはほとんど問題がないが，古典的逆反復法と同等の精度が要求される用途に対しては，対応が困難であった．このような問題に対応するには，古典的逆反復法と同等の精度を保ちつつ，並列性を大幅に高めた新しいアルゴリズムの開発が必要である．

3. ハウスホルダー逆反復法

本章では，共有メモリ型並列計算機に適した新しい固有ベクトル計算アルゴリズムであるハウスホルダー逆反復法について，その原理とアルゴリズムとを述べ，演算量を評価する．

3.1 原　　理

古典的逆反復法では，固有ベクトル $\mathbf{v}_i$ を求めた後，すでに求めた固有ベクトル $\{\mathbf{v}_k\}_{k\in G(i)}$ の成分を修正グラム・シュミット法によってそこから取り除くという方式をとっていた．しかし修正グラム・シュミット法は k に関して逐次的であるため，固有値グループに関する並列性が利用できない場合には，1 回の直交化演算 $\mathbf{v}_i := \mathbf{v}_i - (\mathbf{v}_i \cdot \mathbf{v}_k)\mathbf{v}_k$ 自体を並列化せざるをえず，並列粒度が $O(N)$ と小さくなってしまう．

そこで本アルゴリズムでは，すでに求めた固有ベクトルの成分を取り除くという方式をとりやめた．代わりに，すでに求めた固有ベクトルの張る空間 $V_{i-1} = \mathrm{span}\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{i-1}\}$ の直交補空間 $V_{i-1}^{\perp}$ の正規直交基底 Q_{i-1}（Q_{i-1} は $N \times (N-i+1)$ の行列）をつねに陽的に保持し，$\mathbf{v}_i$ を $V_{i-1}^{\perp}$ に射影することにより，$\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{i-1}$ に直交する新しい固有ベクトルが求められるようにした．射影により $\mathbf{v}_i$ を求めたら，今度はハウスホルダー変換により Q_{i-1} を直交変換し，$\mathbf{v}_i$ を第 1 列ベクトルとする新しい正規直交基底 Q'_{i-1} を作る．そして Q'_{i-1} の第 1 列を取り除くことにより，$V_i = \mathrm{span}\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_i\}$ の直交補空間 $V_i^{\perp}$ の正規直交基底 Q_i を作る．こうして直交補空間の正規直交基底 Q_i を更新しつつ，固有ベクトル $\mathbf{v}_i$ を 1 本ずつ求めていく．なお，Q_i の初期値としては $Q_0 = I_N$（$N \times N$ の単位行列）をとる．

本アルゴリズムでは，単位行列 $Q_0 = I_N$ を 1 ステップずつハウスホルダー変換によって変換しながら，最終的に N 本の固有ベクトルを列ベクトルとする行列へと変換する．ハウスホルダー変換では，直交性がきわめて精度良く保たれることが知られているので[7)]，本アルゴリズムでは最終的に得られた固有ベクトルの直交性は高いと期待される．また，主要な演算は $\mathbf{v}_i$ の $V_{i-1}^{\perp}$ への射影（行列ベクトル積）とハウスホルダー

[アルゴリズム２： ハウスホルダー逆反復法]

$Q_0 = I_N$

$V_0 = \phi$ （N×0 の行列）

do i=1, N

逆反復の初期値 $\mathbf{v}_i^{(0)}$ を設定する。

m=1

固有ベクトル $\mathbf{v}_i^{(m)}$が収束するまで以下を繰り返す。

$\mathbf{v}_i := (T-\lambda_i I)^{-1}\mathbf{v}_i^{(m-1)}$

$\mathbf{p}_i = Q_{i-1}{}^t\mathbf{v}_i$ （ $\mathbf{p}_i$は長さ N-i+1 のベクトル） (3.1)

$\mathbf{p}_i$の第２成分以下を０にするハウスホルダー変換 $H_i = I_{N-i+1} - \alpha_i \mathbf{w}_i \mathbf{w}_i^t$を求める。

（ $\mathbf{w}_i$は長さ N-i+1 のベクトル）

$\mathbf{q}_i = \alpha_i Q_{i-1}\mathbf{w}_i$ （ $\mathbf{q}_i$は長さ N のベクトル） (3.2)

$\mathbf{v}_i^{(m)} = (Q_{i-1})_1 - \mathbf{q}_i(\mathbf{w}_i^t)_1$ （ $(A)_i$は行列 A の第 i 列目のみからなるベクトルを表す。）

m := m+1

$Q_{i-1}' = Q_{i-1} - \mathbf{q}_i \mathbf{w}_i^t$ （ハウスホルダー変換による Q_{i-1}の更新） (3.3)

$V_i = [V_{i-1} \mid (Q_{i-1}')_1]$

Q_{i-1}'の第１列を取り去ってできる行列を Q_iとする。

end do

図 5 ハウスホルダー逆反復法のアルゴリズム
Fig. 5 Algorithm of the Householder inverse iteration method.

変換であり，アルゴリズム全体を通じた並列起動回数は $O(N)$ である．

3.2 アルゴリズム

本アルゴリズム（ハウスホルダー逆反復法）の詳細を**図 5** に示す．なお，逆反復の初期値設定の部分は，縮重または縮重に近い状況がある場合も含め，古典的逆反復法と同様である．

3.3 演 算 量

ハウスホルダー逆反復法の中心演算は上記アルゴリズム中の (3.1)，(3.2)，(3.3) であり，(3.1) が $\mathbf{v}_i$ の直交補空間 $V_{i-1}^{\perp}$ への射影，(3.2)，(3.3) がハウスホルダー変換である．逆反復が 1 回で収束すると仮定すれば，第 i 番目の固有ベクトルを計算するときの各式の演算量はそれぞれ $2N \times (N-i+1)$ である．したがってアルゴリズム全体での演算量はそれぞれ約 N^2 であり，合計では $3N^3$ となる．一方，古典的逆反復法の演算量は，全固有値が 1 つのグループに属する場合，$2N^3$ であるから，ハウスホルダー逆反復法は古典的逆反復法に比べて 1.5 倍の演算を必要とすることになる．

しかし，古典的逆反復法では中心演算が内積や AXPY（$\mathbf{v}_i := \mathbf{v}_i + c\mathbf{v}_j$ という形のベクトル演算）などの BLAS1 演算で，ほとんど最適化の余地がないのに対し，本アルゴリズムの中心演算は行列ベクトル積と行列の rank-1 更新などの BLAS2 演算であり，ループ展開でロード/ストアを削減することにより最適化を行う余地がある．したがって，並列化を行わない場合でも，本アルゴリズムは最適化により古典的逆反復法と同等の実行時間を達成できる可能性があり，これと並列起動回数を $O(N^2)$ から $O(N)$ へ削減した効果を合わせると，並列時には古典的逆反復法よりも高速化できる可能性があると考えられる．

4. 性 能 評 価

4.1 性 能 評 価

4.1.1 中心演算の性能

ハウスホルダー逆反復法の性能評価に先立ち，3.3 節で述べた 3 種類の中心演算について，SR8000 の 1 ノード上での性能評価を行った．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 GFLOPS，1 ノードでは 8 GFLOPS のピーク性能を持つ．並列化は自動並列化コンパイラを用い，並列化対象のループを指定することにより行った．各中心演算の基本的なコードを**図 6** (a)〜(c) に示す．ここで，行列 Q は列方向が連続アドレスになるよう格納し，do ループの順番はいずれも最内側ループでの Q に対するアクセスが連続になるようにとってある．

各中心演算に対する最適化方式，並列化方式，性能を**表 1** に示す．ただし，性能は行列サイズが $N = 4000$ のときの結果である．最適化としては，各演算について j 方向にループ展開を行うことにより，Byte/Flop 値（演算 1 回あたりに対して必要なロード/ストアのバイト数）を削減した．なお，展開数は実験的に最適

表 1 各中心演算の性能
Table 1 Performance of each of the kernel loops.

アルゴリズム中の番号		(3.1)	(3.2)	(3.3)
ループの種類		行列ベクトル積（内積形式）	行列ベクトル積（AXPY 形式）	行列の Rank-1 更新
最適化方式	ループ展開	j に関して 16 倍展開	j に関して 8 倍展開	j に関して 12 倍展開
	Byte/Flop	4.25	5.0	8.33
並列化方式		j に関してブロック分割	j に関してブロック分割	j に関してブロック分割
性能（GFLOPS）		3.96	3.41	1.81

```
do j=1, N-i+1
    s = 0.0d0
    do k=1, N
        s = s+Q(k, j)*v(k)
    end do
    p(j) = s
end do
```

(a) 式（3.1）のコード

```
do j=1, N-i+1
    do k=1, N
        q(k) = 0.0d0
    end do
    do k=1, N
        q(k) = q(k)+Q(k, j)*w(j)
    end do
end do
```

(b) 式（3.2）のコード

```
do j=1, N-i+1
    do k=1, N
        Q(k, j) = Q(k, j) – q(k)*w(j)
    end do
end do
```

(c) 式（3.3）のコード

図 6 各中心演算の FORTRAN コード
Fig. 6 FORTRAN codes corresponding to each of the kernel loops.

な値を決定した．また，並列化方式としては，いずれも j 方向にブロック分割を行った．この結果，各中心演算では約 1.8 GFLOPS〜約 4 GFLOPS の性能が得られた．

一方，古典的逆反復法の中心演算である修正グラム・シュミット直交化のループの性能は $N = 4000$ のとき 1.4 GFLOPS であった．したがって，ハウスホルダー逆反復法の中心演算では，古典的逆反復法の 1.3〜2.8 倍の性能が達成できていることが分かる．これより，1.5 倍という演算量の差を考慮しても，ハウスホルダー逆反復法は古典的逆反復法より高い性能を達成できる見込みがあると考えられる．

4.1.2 アルゴリズム全体の性能

次に，実対称三重対角行列の全固有値・固有ベクトルを求める場合において，ハウスホルダー逆反復法と古典的逆反復法との実行時間を比較した結果を**表 2** に示す．行列はフランク行列 $A_{ij} = \min(i, j)$ をハウスホルダー変換により三重対角化した行列であり，括弧内が逆反復法による固有ベクトル求解のみの時間，括弧外が二分法により固有値を求める部分も含めた時間である．なお，参考のため，SR8000 の 1 ノードと同じピーク性能を持つベクトル型スーパーコンピュータ S3800 上での古典的逆反復法による時間も同時に示した．ただし，こちらはライブラリであり，逆反復法のみの時間を測定できなかったため，二分法との合計時間のみを示した．

表 2 より，ハウスホルダー逆反復法は N が小さいところで特に効果的であり，固有ベクトル求解の時間は古典的逆反復法に比べて最大 2.4 倍程度高速化されていることが分かる．また，二分法と合わせた実行時間で見ると，古典的逆反復法を用いた場合，並列起動オーバヘッドにより並列計算機 SR8000 での実行性能はベクトル機 S3800 での性能に劣るが，ハウスホルダー逆反復法を用いることにより，並列機でもベクトル機と同等の性能が得られることが分かる．

同じ問題に対し，SR8000 の高速版である SR8000 モデル F1（各プロセッサの性能は 1.5 GFLOPS，ノードのピーク性能は 12 GFLOPS）で求解を行ったときの逆反復法の実行時間を**表 3** に示す．この場合も，ハウスホルダー逆反復法では，古典的逆反復法に比べ，最大 2.8 倍の高速化が達成できている．なお，モデル F1 の方が高速化の効果が大きいのは，中心演算が BLAS2 でメモリのスループットに対する要求が小さく，プロセッサの性能を引き出しやすいというハウス

表 2 従来法との性能比較 1（二分法+逆反復法，括弧内は逆反復法のみの時間）
Table 2 Performance comparison of the new and the conventional method (1).

手法	従来法（SR8000）	本手法（SR8000）	従来法（S3800）
N=1000	4.21s（3.92s）	2.06s（1.64s）	2.15s
N=2000	18.84s（17.61s）	12.05s（10.68s）	12.40s
N=4000	98.46s（94.11s）	83.37s（78.68s）	80.65s

表 3 従来法との性能比較 2（SR8000 モデル F1，逆反復法のみの時間）
Table 3 Performance comparison of the new and the conventional method (2).

手法	従来法	本手法
N=1000	3.20s	1.13s
N=2000	14.06s	7.87s
N=4000	67.76s	57.60s

表 4 従来法との精度比較 1（フランク行列）
Table 4 Comparison of the accuracy of the new and the conventional method (1).

	残差		直交性	
	従来法	本手法	従来法	本手法
N=1000	0.164×10^{-7}	0.164×10^{-7}	0.135×10^{-12}	0.266×10^{-14}
N=2000	0.111×10^{-6}	0.111×10^{-6}	0.909×10^{-13}	0.333×10^{-14}
N=4000	0.571×10^{-6}	0.571×10^{-6}	0.767×10^{-13}	0.954×10^{-14}

表 5 従来法との精度比較 2（一般固有値問題）
Table 5 Comparison of the accuracy of the new and the conventional method (2).

	残差		直交性	
	従来法	本手法	従来法	本手法
N=1000	0.110×10^{-10}	0.110×10^{-10}	0.182×10^{-10}	0.127×10^{-10}
N=2000	0.256×10^{-10}	0.251×10^{-10}	0.155×10^{-10}	0.281×10^{-10}
N=4000	0.544×10^{-10}	0.545×10^{-10}	0.171×10^{-10}	0.198×10^{-10}

ホルダー逆反復法の特長が，プロセッサが高速になるほど顕著に現れるためと考えられる．

4.2 精度評価

固有ベクトルの残差（$T\mathbf{v}_i - \lambda_i \mathbf{v}_i$ の L_2 ノルムの最大値），および固有ベクトルの直交性（$V^tV - I_N$ の要素の最大値）について，従来法との比較を行った結果を表 3 に示す．行列としては，フランク行列，および一般固有値解法に組み込んだ場合の 2 通りを評価した．一般固有値解法では，固有方程式 $A\mathbf{v} = \lambda B\mathbf{v}$ において，左辺の行列 A は区間 $[0,1)$ の乱数行列，右辺の行列 B は対角成分がすべて 10^4，それ以外の成分が $[0,1)$ の乱数からなる行列とした．フランク行列の結果を**表 4**，一般固有値問題の結果を**表 5** に示す．表より，本手法の残差は従来法と同等であり，直交性はフランク行列の場合は従来法より 1 桁程度良く，一般固有値問題の場合はほぼ同等であることが分かる．

5. おわりに

本論文では，対称行列の固有ベクトル計算を並列計算機で行うための新しいアルゴリズムとしてハウスホルダー逆反復法を提案した．ハウスホルダー逆反復法では，すでに求めた固有ベクトルの直交補空間を陽的に保持し，その基底をハウスホルダー変換で更新していくことにより，並列化のネックであった修正グラム・シュミット法による直交化なしに固有ベクトルを求めることが可能であり，これにより並列起動回数を $O(N^2)$ から $O(N)$ に削減できる．また，ハウスホルダー変換では直交性がきわめて精度良く保たれるため，最終的に得られる固有ベクトルの直交性も高い．SR8000 の 1 ノード（8 プロセッサの共有メモリ並列）で評価した結果，ハウスホルダー逆反復法は，1000 元から 4000 元までの行列の全固有ベクトルを求める問題において，古典的逆反復法の最大 2.4 倍の性能を達成した．また，固有ベクトルの直交性も，古典的逆反復法と同等の精度が得られた．

今後の課題としては，本アルゴリズムの分散メモリ型並列計算機への適用があげられる．

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，JSPP2000での発表の際に貴重なコメントをくださった図書館情報大学の長谷川秀彦先生，筑波大学の朴泰祐先生，産業技術融合領域研究所の長嶋雲兵先生に感謝いたします．

参 考 文 献

1) 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).
2) Berry, M.W. and Browne, M.: *Understanding Search Engines*, SIAM (1999).
3) Wilkinson, J.H. and Reinsch, C. (Eds.): *Linear Algebra*, Springer-Verlag (1971).
4) Dongarra, J.J. and van de Geijn, R.A.: Reduction to Condensed Form for the Eigenvalue Problem on Distributed Architectures, *Parallel Computing*, Vol.18, pp.973–982 (1992).
5) Choi, J., et al.: ScaLAPACK: A Portable Linear Algebra Library for Distributed Memory Computers – Design Issues and Performance, *LAPACK Working Notes 95* (1995).
6) 村田健郎ほか：スーパーコンピュータ：科学技術計算への適用，丸善 (1985).
7) Golub, G.H. and van Loan, C.F.: *Matrix Computations*, The Johns Hopkins University Press (1989).
8) Dhillon, I.: A New $O(n^2)$ Algorithm for the Symmetric Tridiagonal Eigenvalue/Eigenvector Problem, Ph.D. Thesis, Computer Science Division, University of California, Berkeley, California (1997).

(平成 12 年 8 月 29 日受付)
(平成 13 年 1 月 11 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修了．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．