ブロックハウスホルダ変換について

村上 弘

首都大学東京 都市教養学部 都市教養学科 数理科学コース

大規模な行列問題を効率良く解くには、算法をブロック化して記憶参照の局所性を向上させる必要がある。本論文では WY 表現 [7, 12] とは異なるブロック鏡映変換 [10] に基づいた block Householder 変換の構成法を紹介する。もしもある算法が通常の「数要素の行列」に対するもので Householder 変換から構成されているならば、その算法の自然な拡張を「行列を要素とする行列」すなわちブロック行列に対するものとして容易に構成できる。例として大規模対称密行列のブロック三重対角化による固有値解法を掲げる。

On the Block Householder Transformation

Murakami Hiroshi

Faculty of Urban Liberal Arts, Tokyo Metropolitan University

To solve large matrix problems efficiently, block algorithms are to be used to enhance the locality of storage reference. In this paper, a realization of the block Householder transformation based on the block reflector [10] is introduced rather than the WY representations [7, 12]. If an algorithm for an usual matrix of numbers which is constructed by the Householder transformations is given, a natural extension of the algorithm for a block matrix (matrix consists of matrices) using the Block Householder transformations can be easily constructed also. As an example, solution of the eigenproblem of a large dense symmetric matrix by the block tridiagonalization is demonstrated.

1　導入

本論文では block Householder 変換のある構成法を示し、応用例としてブロック三重対角化を経由して大規模な対称密行列の比較的少数の固有値と固有ベクトルを求める計算を行う。

block householder 変換については参考文献に挙げたレポートや論文 [7, 11, 10, 12, 16, 17, 18, 20, 6, 14] などの研究報告がある。

2　block Householder 変換

いま $N > b$ とし、与えられた $N{\times}b$ 行列 C の任意の直交分解を $C{\Rightarrow}XZ$ とする。I_b を b 次単位行列として、X は $N{\times}b$ 行列で $X^TX = I_b$ を満たす。Z は b 次の正方行列。注意: C が階数不足の場合には X は拡張された正規直交基底の組で、例えば Householder-QR 法により自然に構成される。

いま E_b をその最初の b 個の行が単位行列 I_b でそれ以外は 0 の $N{\times}b$ の拡張単位行列とする。X の最初の b 個の行を並べて作った正方小行列 X_1 の特異値分解を $X_1{\Rightarrow}Q\Lambda V$ とする。Q と V は b 次の直交行列、Λ は b 次の非負対角行列。$I_b = X^TX = X_1^TX_1 + (X - E_bX_1)^T(X - E_bX_1) = \Lambda^2 + (X - E_bX_1)^T(X - E_bX_1)$ より Λ の対角要素は 1 以下である。

X の最初の b 個の行に QV を加えた行列 $Y = X + E_bQV$ を定義すると、関係 $X^TX = I_b$, $X^TE_b = X_1^T$, $X_1 = Q\Lambda V$ を用いて $Y^TY = 2V^T(I_b + \Lambda)V$ が示せる。これから Y^TY の固有値の下限が 2 で上限が 4 だから Y の条件数は $\sqrt{2}$ 以下である。

Y の任意の直交分解を $Y{\Rightarrow}US$ とする。$N{\times}b$ 行列 U は $U^TU = I_b$ を満たし、b 次正方行列 S の条件数は $\sqrt{2}$ 以下で可逆行列。Y の直交分解 $Y{\Rightarrow}US$ には任意性があるが、関係 $U^TU = I_b$ や UU^T は U の選択にはよらない。(選択の 1 つは $U{\Leftarrow}Y[V^T\{2(I_b + \Lambda)\}^{(-1/2)}]$ で、効率的に計算できる。)

U(但し $U^TU = I_b$) を軸とする一般化鏡映変換 $H = I - 2UU^T$ は対称な N 次の直交行列である。$HX = -E_b(QV)$ が成り立つ。

[証明] $Y = X + E_bQV$ で $X_1 = Q\Lambda V$ ゆえ $E_b^TY = X_1 + QV = Q(I_b + \Lambda)V$。すると $(E_bQV)^TY = V^TQ^T(E_b^TY) = V^T(I_b + \Lambda)V =$

$(1/2)Y^TY$ よりその転置は $Y^TE_bQV = (1/2)Y^TY$ となる。$E_bQV = Y - X$ だから $Y^T(Y - 2X) = 0$ であり $Y^T = S^TU^T$ と S が可逆行列より $U^T(Y - 2X) = 0$ すなわち $2U^TX = U^TY$、さらに $U^TY = S, US = Y$ より $UU^TY = Y$ だから $2UU^TX = U(2U^TX) = UU^TY = Y$ がわかる。よって $HX = (I-2UU^T)X = X-2UU^TX = X-Y = -E_b(QV)$ となる。[**証明終**]

よって HX の最初の b 個の行を並べて作った正方小行列は $-QV$ でそれ以外の行は 0 になる。同様に $HC = H(XZ) = (HX)Z = -E_b(QV)Z$ だから HC の最初の b 個の行を並べて作った正方小行列は $-QVZ$ でそれ以外の行は 0 になる。

与えられた $N{\times}b$ 行列 C に対し、上記の手順で行列 U を求めて $H = (I-2UU^T)$ とすれば HC は最初の b 個の行以外が 0 になる。つまり列を b 個持つ任意の行列に対し最初の b 個以外の行を消去する対称な直交変換が常に構成できる。これは $b = 1$ の場合の通常の Householder 変換の拡張になっていて、ブロックサイズ b のブロック行列に対する Householder 変換とみなせる。このことから要素が数の行列に対して通常の Householder 変換を使用する算法があれば、その算法に於ける行列要素を数からブロック小行列に置き換えて Householder 変換も block Householder 変換で置き換えて模倣すれば自然なブロック化算法を導けることになる。

注意: C の直交分解として QR-分解を採用し、P を (列ピボット操作の) 置換行列、R を右三角行列とする QR-分解を $CP{\Rightarrow}XR$ ならば $HC = (I-2UU^T)C = -E_b(QVRP^{-1})$ である。HC は最初の b 個の行以外は全て 0 になる。HC の最初の b 個の行を並べて作った正方小行列を β と置くと $\beta = -QVRP^{-1}$ だから β の QR-分解は $\beta P = (QV)(-R)$ となる。

まとめ: C から U を構成する手順:
$N > b$ とする。以下では C,X,Y,U は重ね書きにより記憶場所を共有できる。

- **ステップ 1:** $N{\times}b$ 行列 C の任意の直交分解を $C{\Rightarrow}XZ$ とする。但し X は b 個の (拡張された) 正規直交基底の組で $X^TX = I_b$. 直交分解の構成には Householder-QR 法を用いるのがよい。
- **ステップ 2:** X の最初の b 個の行を並べて作った小行列 X_1 を $X_1{\Rightarrow}Q\Lambda V$ と特異値分解する。
- **ステップ 3:** X の最初の b 個の行を変更した行列 $Y = X + E_bQV$ を作る。
- **ステップ 4:** U を良条件の行列 Y の任意の正規直交化とする。$U{\Leftarrow}Y[V^T\{2(I_b + \Lambda)\}^{(-1/2)}]$ にとれる。

$H = (I - 2UU^T)$ を $N{\times}b$ 行列 C に作用させると最初の b 個の行を並べて作った小行列 β は $-QVZ$ に等しく、それ以外の行は 0 になる。

注意: 一般には行列の階数不足は稀れな状況だが、上記の構成法中で C が階数不足の場合には X として b 個の拡張正規直交基底の組を用いることでそれ以降のステップは C の階数に依らない統一的記述となっている。しかし以下のように行列の寸法を b から r に下げて計算すれば、本質的な r 個の正規直交基底の組を用いて U が構成できる。

まとめ: C から U を構成する手順 (その 2):
$N > b$ とする。以下の C,X,Y,U は重ね書きにより記憶場所を共有できる。

- **ステップ 1':** $N{\times}b$ 行列 C の任意の直交分解を $C{\Rightarrow}XZ$ とする。$r({\le}b)$ を C の階数として、X は $N{\times}r$ 行列で $X^TX = I_r$。Z は $r{\times}b$ 行列とする。
- **ステップ 2':** X の最初の r 個の行を並べて作った r 次正方小行列の特異値分解 $X_1{\Rightarrow}Q\Lambda V$ を作る。Q と V は r 次の直交行列で、Λ は r 次の正値対角行列。
- **ステップ 3':** X の最初の r 個の行を変更した $N{\times}r$ 行列 $Y = X + E_rQV$ を作る。ここで E_r は $N{\times}r$ の拡張された単位行列。
- **ステップ 4':** Y の任意の正規直交化を $N{\times}r$ 行列 U とする。$U{\Leftarrow}Y[V^T\{2(I_r + \Lambda)\}^{(-1/2)}]$ にとれる。

$H = (I - 2UU^T)$ を $N{\times}b$ 行列 C に作用させると最初の r 個の行を並べて作った $r{\times}b$ 行列は $-QVZ$ に等しく、それ以外の行は 0 になる。

3　ブロック変換の応用例

密行列のブロック三重対角化: 与えられた対称密行列 A の固有値問題の解法を考察する。A は次数 b のブロック小行列に分割されているとする。A の最初のブロック列ベクトルから第一ブロック行を除いたブロック縦ベクトルを C と見なす。(下図参照)

$$
\begin{array}{cc}
 & \leftarrow\ b\ \rightarrow \\
\begin{array}{c} \uparrow \\ b \\ \downarrow \\ \\ A = \\ \\ \end{array} &
\left(\begin{array}{c|c}
\alpha & \\
\hline
\begin{array}{c}\text{この}\\ \text{部分を}\\ \text{C}\\ \text{とする}\end{array} & \widetilde{A}
\end{array}\right)
\end{array}
$$

上述の構成に従い、C に対応する変換行列 H を求めれば、C の (実効的な) 階数を $r({\le}b)$ とすると、HC

は最初の r 個の行以外は (実効的に)0 になる。

$$
\begin{array}{c}
\leftarrow\ b\ \rightarrow \\
\begin{pmatrix} \\ \\ C \\ \\ \\ \end{pmatrix}
\end{array}
\overset{\text{鏡映変換 } H \text{ を作用}}{\Longrightarrow}
\begin{array}{cc}
 & \leftarrow\ b\ \rightarrow \\
\begin{array}{c} \uparrow \\ r \\ \downarrow \\ \\ \\ \\ \end{array} &
\begin{pmatrix} \beta \\ \hline 0 \\ \vdots \\ \vdots \\ 0 \end{pmatrix}
\end{array}
$$

ブロック行列 HAH は最初のブロック列あるいはブロック行がブロック三重対角形になる。(下図参照)

$$
\begin{array}{cc}
 & \begin{array}{cc} \leftarrow\ b\ \rightarrow & \leftarrow r \rightarrow \end{array} \\
\begin{array}{c} \uparrow \\ b \\ \downarrow \\ \uparrow \\ r \\ \downarrow \\ \\ \\ \\ \end{array} &
\left(\begin{array}{c|cccc}
\alpha & \beta^T & 0 & \cdots & \\ \hline
\beta & & & & \\
0 & & H\widetilde{A}H & & \\
\vdots & & & & \\
 & & & &
\end{array}\right)
\end{array}
$$

通常の Householder 三重対角化と同様に、各段階ごとにブロック 1 個分ずつ寸法を縮めて変換の適用を繰り返すとブロック行列はブロック三重対角形に変換される。各段階の変換ごとのブロック縦ベクトルのブロックのサイズ (列の幅 b) は必ずしも毎回同じでなくとも良い。(下図参照)

$$
H^{(n-2)}\cdots H^{(3)}\,H^{(2)}\,H^{(1)}\,A\,H^{(1)}\,H^{(2)}\,H^{(3)}\cdots H^{(n-2)}
$$

$$
= \begin{pmatrix}
\alpha_1 & \beta_1^T & 0 & \cdots & & & & \\
\beta_1 & \alpha_2 & \beta_2^T & 0 & \cdots & & & \\
0 & \beta_2 & \alpha_3 & \beta_3^T & 0 & & & \\
\vdots & 0 & \beta_3 & \alpha_4 & \beta_4^T & & & \\
 & \vdots & & & & & & \\
 & & 0 & \ddots & \ddots & & & \vdots \\
 & & & & & \ddots & \ddots & 0 \\
 & & & & & \beta_{n-2} & \alpha_{n-1} & \beta_{n-1}^T \\
 & & & & \cdots & 0 & \beta_{n-1} & \alpha_n
\end{pmatrix}
$$

両側変換 HAH の計算法: 通常の Householder 法同様に両側変換 HAH $(H = I - 2UU^T)$ は補助のブロックベクトル p,q を用いて $p{\leftarrow}AU,\ q{\leftarrow}2(p - U(U^Tp)),\ A{\leftarrow}A - Uq^T - qU^T$ として要素のブロック小行列を単位とする計算ができる。実装上はブロック行列 A の対称性を利用するほか、Fortran 向きならブロック縦ベクトルを基本とし A の対称下半分のブロック要素だけを走査が列方向優先で連続となるよう格納する。(下図参照)

$$
\begin{pmatrix}
* & & & & \\
\downarrow & & & & \\
* & * & & & \\
\downarrow & \downarrow & & & \\
* & * & * & & \\
\downarrow & \downarrow & \downarrow & & \\
* & * & * & * & \\
\downarrow & \downarrow & \downarrow & \downarrow & \\
* & * & * & * & *
\end{pmatrix}
$$

両側 block Householder 変換の p,q を作る計算や A を更新する計算は殆どブロック小行列同士の乗算や加減算なので BLAS3 ライブラリを用いれば極めて効率良く計算ができる。並列処理を採り入れる場合もブロック小行列を単位とする演算なので粒度を容易に大きくできる。

記憶格納上の考察: 固有ベクトルを求める場合には、後の過程で逆変換を行う際に各段階のブロックベクトル U とその (実効的な) 階数を完全に再現する必要があるので、変換の各段階で変換により消去される A の列の記憶場所に U を格納する。補助のブロックベクトル p,q は段が進むごとに必要なサイズが縮むので、p_i,q_i の場所にブロック三重対角行列 T の主対角要素と副対角要素のブロック小行列 α_i,β_i を上書きして格納できる。

逆変換による固有ベクトルの計算: ブロック三重対角行列 T の固有ベクトル y を変換して元の行列 A に対応する固有ベクトル x とするには、以前に格納しておいた各段階のブロックベクトル $\{U^{(i)}\}$ を格納時とは逆順に読み出す。$\{U^{(i)}\}$ から鏡映変換 $\{H^{(i)}\}$ の作用を逆順に構成して y に作用させる: $x{\leftarrow}H^{(1)}\,H^{(2)}\cdots H^{(n-2)}\,y$.

実際には上の鏡映変換を x と y をブロックベクトルとして必要な個数の固有ベクトルを束ねて適用すれば $\{U^{(i)}\}$ を読み出す記憶参照量が減らせ、BLAS3 も利用可能となり効率が向上する。

計算量: 真の次数 N の対称密行列のブロック三重対角化の演算量は (古典的行列乗算法では) 約 $(4/3)N^3$ flops でブロック幅 $b(\ll N)$ にはあまりよらず、ブロック化しない三重対角化とほぼ同じである。

記憶転送量: ブロック幅 b によるブロック三重対角化では、変換を受ける行列 A の記憶への走査が $N-2$ 回から $(N/b)-2$ 回に減らせるので、全体の記憶転送量は約 b 分の 1 となる。逆変換も同様。演算量はブロック化であまり変化しないので記憶参照の局所性は約 b 倍に向上し、b を適切にとると演算速度と記憶供給速度を均衡させうる。しかし、後の過程で帯 Householder 法によりブロック三重対角行列 T の固有値問題を解く部分の演算量は N^2b あるいは $Nb^2\ell$ に比例する項を含み、また記憶参照量も増加するので b を大きくとり過ぎると逆効果になる。もちろん最適な b の値の選択は、N と ℓ 以外にも計算機システムの構成に依存する。

ブロック三重対角行列の帯行列への還元: 以下では簡単化のためブロック幅は全て同じとする。
ブロック幅 b のブロック三重対角行列 T は真の帯幅が $2b-1$ の帯行列だから、T に対して対称帯行列の固有値問題の解法を直接適用してもよい。しかし少ない手間で行列 T の帯幅 $2b-1$ を縮小し帯幅 b の対称帯行列 B に変形できる [15][21]。
いま T を次のように書く:

$$T = \begin{pmatrix} \alpha_1 & \beta_1^T & & & & \\ \beta_1 & \alpha_2 & \beta_2^T & & & \\ & \beta_2 & \alpha_3 & \beta_3^T & & \\ & & \beta_3 & \ddots & \ddots & \\ & & & \ddots & \ddots & \beta_{n-1}^T \\ & & & & \beta_{n-1} & \alpha_n \end{pmatrix}.$$

まず最初は第 2 行に注目し、β_1 の QR-分解を $\beta_1 \rightarrow Q_1 R_1$ と行う。ここで Q は直交行列で R は右三角行列。この QR-分解はピボット操作なしの Householder-QR 分解により構成できる。実際の計算では、Q_1 は陽的には構成せず、積形式 (factored form) として保持する。この QR-分解を構成した後で直交変換を適用する: $diag[I, Q_1^T, I, \cdots, I]\ \ T\ \ diag[I, Q_1, I, \cdots, I]$

$$= \begin{pmatrix} \alpha_1 & R_1^T & & & & \\ R_1 & Q_1^T \alpha_2 Q_1 & (\beta_2 Q_1)^T & & & \\ & \beta_2 Q_1 & \alpha_3 & \beta_3^T & & \\ & & \beta_3 & \ddots & \ddots & \\ & & & \ddots & \ddots & \beta_{n-1}^T \\ & & & & \beta_{n-1} & \alpha_n \end{pmatrix}.$$

同様に k-段目に於いて、$(k+1)$-番目の行にある副対角要素を (簡単のためピボット操作なしの)Householder-QR 法で QR-分解し、Q_k と R_k を決める。

QR-分解の積形式の Q_k と R_k の両方の記憶は β_k が格納されていた場所に上書きすれば格納できる。そうして第 $(k+1)$ 行には左から Q_k の転置を、第 $(k+1)$ 列には右から Q_k を乗じる。

対称性から主対角の下半分と左副対角要素の記憶場所だけが計算に必要である。第 $(n-1)$-段目が終わると、ブロック三重対角行列 T は変換されて帯幅 b の対称帯行列 B になる。$B = diag[I, Q_1^T, \cdots, Q_{n-1}^T]\ T\ diag[I, Q_1, \cdots, Q_{n-1}]$

$$= \begin{pmatrix} \gamma_1 & R_1^T & & & & \\ R_1 & \gamma_2 & R_2^T & & & \\ & R_2 & \gamma_3 & R_3^T & & \\ & & R_3 & \ddots & \ddots & \\ & & & \ddots & \ddots & R_{n-1}^T \\ & & & & R_{n-1} & \gamma_n \end{pmatrix}.$$

固有ベクトルの逆変換: ブロック対称三重対角行列 T の固有ベクトル y を、B の固有ベクトル z から得るには、逆変換 $y \leftarrow diag[I, Q_1^T, \cdots, Q_{n-1}^T]\ z$ を用いる。y と z を一度に変換する個数分の列ベクトルを束ねたブロックベクトルにすると、複数の固有ベクトルをまとめて変換でき参照の局所性が高まる。

帯幅縮小の計算量: b 次の小行列の QR-分解の手間は 1 個あたり $O(b^3)$ で、Q_k を乗じる手間も $O(b^3)$。$n = N/b$ だから、前進変換の全計算量は $O(Nb^2)$ で、逆変換する固有ベクトルを ℓ 個とすると固有ベクトルの後退変換の全計算量は $O(Nb\ell)$。

対称帯行列の固有値問題の解法: 大次元の対称帯行列の固有値問題を解く場合、求めたい固有解の個数が行列の次元に比べ極めて少ない場合には

- 対称帯行列を (真の) 三重対角行列に変換する。
- 三重対角行列の求めたい範囲の固有値をスツルム二分法により得る。
- 固有値に対応した固有ベクトルを帯行列の LU 分解を利用した逆反復法で解く。

により求められる。この方法は必要とする固有ベクトルの個数が極めて多い場合、たとえば行列次数の全部あるいは相当の割合などでは LU 分解を行う逆反復法の手間から実施が困難である。帯行列やブロッ

ク三重対角行列に対する直接的な分割統治法 [19][24] が研究されている。必要な固有ベクトルの割合が多い場合には有望であろう。

対称帯行列の三重対角化には Rutishauser-Schwarz 法 [1] も利用可能だが、性能の点から帯行列に対する村田の帯 Householder 法 [3] を用いた。そうして対称帯行列の固有値問題を次の組み合わせて解いた:「村田の帯 Householder 三重対角化」、「固有値のスツルム二分法」、「帯行列に対する逆反復法」。
これらの処理には [3] に掲載されているコードを用いた。

m を帯行列の帯幅、N を (ブロックとしてではない) 真の行列の次数とし、スツルム二分法による固有値計算の分離限界の許容値を ε とするとき、固有値と固有ベクトルを ℓ 個求めるときの計算量は:

帯行列三重対角化+スツルム二分法+帯逆反復法
$$= O(N^2 m) + O(N\ell \log_2(1/\varepsilon)) + O(Nm^2\ell)\,.$$

帯幅の縮小操作で帯幅 m は $2b-1$ から b に下げている。縮小操作に演算量 $O(Nb^2) + O(Nb\ell)$ を投じても、帯行列の固有値問題が帯幅の半減により速く解けるので得である。

記憶量の考察: 村田の帯 Householder 法が要求する記憶量は:

- 帯行列 B に対しての実数 $N\times(m+1)$ 語、
- 作業用領域として実数 $4N + (3m+1)\times N = (3m+5)\times N$ 語と整数 N 語、
- 固有ベクトル ℓ 個の格納用の実数 $N\times\ell$ 語。

固有ベクトルと実数型作業ベクトルは重ね書きで記憶領域を共用できる。

4 実験結果の例

実験の結果の (紙数の制約から) 一部を示す。CPU は Pentium 4 2.6C、主記憶は dual channel DDR-3200 512MB×4 本の 2Gbyte で、全ての場合で参照は主記憶に納まっている。コンパイラは intel Fortran 8.1、ブロック小行列の BLAS3 乗算には GOTO-BLAS を用いている。

図 1 は横軸はブロックサイズ b を 4 の倍数で 8 から 100 まで、縦軸には (ブロック三重対角化の演算量を $(4/3)N^3$ と換算した) 演算速度を GFLOPS 単位でとり、行列次数 1000, 2000, 4000, 8000, 16000 に対しプロットしたものである。行列次数 N が大きい程グラフの曲線は上側にある。性能ピークに対応するブロックサイズは N が大きいほど増加する傾向にある。$N = 1000$ では b=40〜50 付近、$N = 2000$ では b=40〜60 付近、$N = 4000$ では b=50〜80 付近、$N = 8000$ では b=60〜90 付近、$N = 16000$ では b=70〜90 付近に緩やかなピークがある。

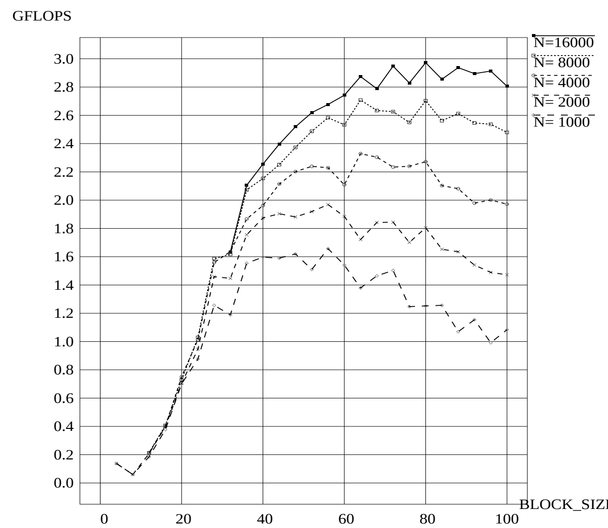

図 1: ブロック三重対角化の換算速度 Pentium4 2.6C

図 2 は同上のデータを横軸には行列次数を 20000 まで、縦軸には換算演算速度を GFLOPS 単位でとり、ブロックサイズ b を 12, 16, 20, 24, 28, 40, 52, 80 に対しプロットしたものである。プロットされた曲線は b が大きい程上側になっている。$b = 80$ で $N = 20000$ のとき性能値は 3.1GFLOPS である。

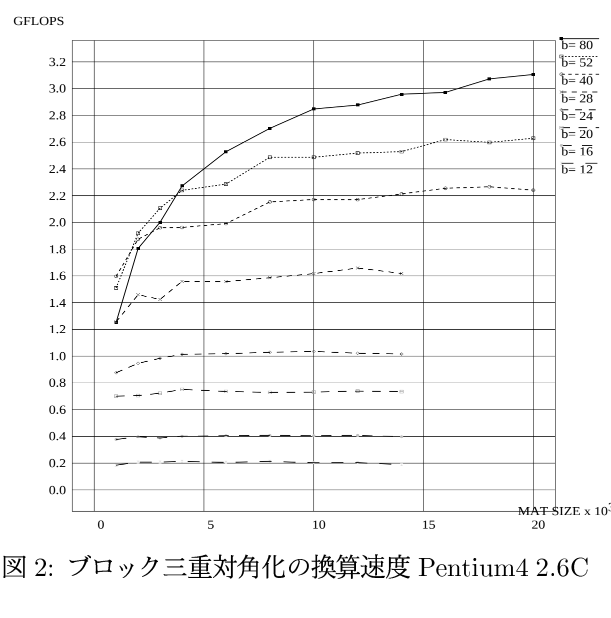

図 2: ブロック三重対角化の換算速度 Pentium4 2.6C

5 結論

ブロック鏡映変換に基づく block householder 変換の構成法を示した。それを用いて大規模対称密行列の固有値問題の解をブロック三重対角化を経由して少数求める例を示した。大規模な行列問題では算法のブロック化の効果は高い。高い参照の局所性により大部分の記憶参照が平均的に小容量だが高速の上位の記憶階層内に留まり階層間の記憶転送量が減らせるので記憶を待つ時間が少なくなり演算装置が有効に使え、計算全体での経過時間を短縮できる。

参考文献

[1] Wilkinson J.H. and Reinsch C., eds.: *Handbook for Automatic Computation, Vol.2, Linear Algebra*, Springer-Verlag, New York, (1971).

[2] Golub G.H. and Van Loan C.F.: *Matrix Computations*, Chap.8 : 'The Symmetric Eigenvalue Problem', The Johns Hopkins Univ. Press, Baltimore, Maryland, (1983).

[3] 村田健郎, 小国力, 唐木幸比古, 「スーパーコンピュータ 科学技術計算への適用」, 丸善, 1985.

[4] 村田健郎,「線形代数と線形計算法序説」, サイエンス社,(1986 年 4 月).

[5] 名取亮, 野寺隆 共編著,「スーパーコンピュータと大型数値計算」, 共立出版,bit 別冊,(1987 年 11 月).

[6] Jack J. Dongarra, Sven J. Hammarling and Danny C. Sorensen, 'LAPACK Working Note #2; Block Reduction of Matrices to Condensed Forms for Eigenvalue Computations', ANL/MCS-TM-99, September, 1987.

[7] Bischof C. and Van Loan C.: 'The WY representation for products of Householder matrices', *SIAM J. Sci. Stat. Comput.*, Vol.8, No.1, pp.2-13(1987).

[8] 村田健郎, 小国力, 三好俊郎, 小柳義夫 共編著, 「工学における数値シミュレーション」, 丸善, 1988.

[9] 村上 弘,「対称密行列に対するアウトコア・ハウスホルダー法- 大型対称密行列の固有値問題 - 」, 日本シミュレーション学会, 第 9 回計算電気・電子工学シンポジウム論文集,255-258 頁,1988 年 3 月 31 日.

[10] Schreiber R. and Perlett B.: 'Block Reflectors: Theory and Computation', *SIAM J. Numer. Anal.* Vol.25, No.1. pp.189-205(1988).

[11] Dongarra J.J. and Sorensen D.C.: 'Block reduction of matrices to condensed forms for eigenvalue computations', *J. Comput. Appl. Math.* Vol.27, pp.215-227(1989).

[12] Schreiber R. and Van Loan C.: 'A storage efficient YW representation for products of Householder transformations', *SIAM J. Sci. Stat. Comput.* Vol.10, No.1 pp.53-57(1989).

[13] 小国力 編, 村田健郎, 三好俊郎, ドンガラ J.J., 長谷川秀彦共著, 「行列計算ソフトウェア - WS, スーパーコン, 並列計算機 - 」, 丸善,1991.

[14] Christian H. Bischof, 'A Summary of Block Schemes for Reducing a General Matrix to Hessenberg Form', Argonne National Laboratory, ANL/MCS-TM-175, February 1993.

[15] Christian Bischof, Bruno Lang and Xiaobai Sun, 'Parallel Tridiagonalization through Two-Step Band Reduction', Argonne Preprint ANL-MCS-P412-0194, to appear in Proc. of the 1994 Scalable High-Performance Computing Conference, Knoxville, May, 1994.

[16] Wilfried N. Gansterer, Dieter F. Kvasnicka and Christoph W. Ueberhuber, 'Optimizing Locality of Reference in Symmetric Eigensolvers', AURORA TR1997-13, November, 1997.

[17] Wilfried N. Gansterer, Dieter F. Kvasnicka and Christoph W. Ueberhuber, 'Efficiency Enhancement by Blocking', AURORA TR1998-10, August, 1998.

[18] Wilfried N. Gansterer, Dieter F. Kvasnicka and Christoph W. Ueberhuber, 'High Performance Computing in Material Sciences; Higher Level BLAS in Symmetric Eigensolvers', AURORA TR1998-13, November, 1998.

[19] Wilfried N. Gansterer, Josef Schneid, and Christoph W. Ueberhuber, 'A Divide-and-Conquer Method for Symmetric Banded Eigenproblems; Part-I: Theoretical Results', AURORA TR1999-12, August, 1998.

[20] Wilfried N. Gansterer, Dieter F. Kvasnicka and Christoph W. Ueberhuber, 'High Performance Computing in Material Sciences — Numerical Experiments with Symmetric Eigensolvers', AURORA TR1998-19, November, 1998.

[21] Christian H. Bischof, Bruno Lang and Xiaobai Sun, 'A Framework for Symmetric Band Reduction', Also Preprint Argonne Preprint ANL/MCS-P586-0496, October 1996.

[22] Murakami Hiroshi, 'An Implementation of the Block Reflector Method for the Dense Eigenproblem', 東京都立短期大学 経営情報学科研究論叢 ISSN1343-3202, pp.41-67, No.5 (2001 年 3 月).

[23] L. Wei, X. Li, 'A New Divide and Conquer Algorithm for Real Symmetric Band Generalized Eigenvalue Problem', Fifth International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP'02), p.0105, October, 2002.

[24] Wilfried N. Gansterer, Robert C. Ward. and Richard P. Muller, 'An extension of the divide-and-conquer method for a class of symmetric block-tridiagonal eigenproblems', ACM Transactions on Mathematical Software (TOMS), Vol.28, Issue 1, pp.45-58, March, 2002.

[25] Stathopoulos A., and Wu K., 'A Block Orthogonalization Procedure with Constant Synchronization Requirements', SIAM J. Sci. Comput., pp.2165-2182, vol.23, No.6 (2002).

[26] Wilfried N. Gansterer, Yihua Bai, M. Day and Robert C. Ward., 'A Framework for Approximating Eigenpairs in Electronic Structure Computations', Computing in Science & Engineering, pp.50-59, Sep/Oct issue (2004).