ベクトル計算機と汎用計算機のための対称帯行列固有値解法[†]

長谷川 秀 彦[††]

対称帯行列の固有値・固有ベクトルを求める算法から三重対角化を経由する方法とスツルム逆反復法をとりあげ，汎用的で高速なプログラムに改良する方法とその評価について述べる．三重対角化を経由する方法では，三重対角化にハウスホルダー変換を用い，2分法と逆反復法を組み合わせて用いるのがよい．三重対角化に対するギブンス法や2分法に代わる多分法は汎用性に欠ける．スツルム逆反復法では，軸選択なしの対称ガウスを併用して2分法の負担を軽減するとよい．いずれの場合も帯ガウスや対称ガウスを2段2行同時として用いるのがよい．結果として，精度がよく，安定で汎用的なプログラムを作ることができた．これらは「同一のプログラムがベクトル計算機と汎用計算機の両方で高速に実行できる」という本論文の立場での必須条件を満足する．本論文では算法の見直し・改良と，繰り返して使われる部分の高速化（2段2行同時など）によって高速化を実現した．今まではベクトル計算機向けの高速化手法として汎用計算機にとって好ましくない手法がしばしば推奨されていたが，本論文ではベクトル計算機と汎用計算機の両方にとって好ましい手法だけを用いて，高速化ができることを示した．

1. はじめに

最近までCPUタイムの制限から大規模な行列の固有値・固有ベクトルの解析は困難であったが，ベクトル計算機はこの事情を大幅に改善するものとして期待されている．本論文では，精度がよく（マシンイプシロンから数桁），安定で汎用的な固有値解析の算法を対象とする．また現時点では汎用計算機を使うことも少なくないため，同一のプログラムがベクトル計算機と汎用計算機で高速に実行できることも必須条件とする．計算機ごとに異なるプログラムを使うと，アルゴリズムの違いによる誤差の評価が必要になるほか，プログラムを保守するための負担がはなはだ大きくなる．また実際の運用でトラブルを起こしやすいので運用上も負担が大きい．

ここでは，対称帯行列の固有値・固有ベクトルを小さい方から帯半幅の5倍に相当する組数だけ求める場合について各種の算法を実測し，その結果に基づいて議論を行う．固有値だけを求める場合には評価が多少異なる．

対称行列の固有値解析に用いられる算法には

（1） ヤコビ法
（2） 三重対角化，*QL* 法
（3） 三重対角化，2分法，逆反復法
（4） スツルム逆反復法
（5） スツルム同時逆反復法
（6） ランチョス法

などがある．(1)は面内回転を用いて行列を対角化する．この算法は帯行列の場合でも fill-in によって密行列化するため，メモリ容量が問題である．さらにCPUタイムも膨大である．

(2)と(3)は面内回転やハウスホルダー変換を用いて対称行列を三重対角化し，三重対角行列の固有値 λ，固有ベクトル $\boldsymbol{u}$ を求める．帯行列に対して一般の三重対角化法を適用すると fill-in によって密行列化するが，特別な三重対角化法を使えば帯行列のまま三重対角化ができる．帯行列の固有ベクトル $\boldsymbol{v}$ は λ を用いて逆反復法で求める．逆変換で $\boldsymbol{v}$ を求める方法は，逆変換を記憶するためのメモリ容量が大きい．2分法は一部の固有値・固有ベクトルを求めるための算法とされているが，高速化を図ると大規模行列では全固有値・固有ベクトルを求める場合でも *QL* 法より高速になる[1]．*QL* 法は全固有値・固有ベクトルを求める算法で，特定の区間に属する固有値・固有ベクトルを求めるのは困難である．

(4)，(5)，(6)は密行列に対しては使われない方法である．(4)スツルム逆反復法は2分法で固有値の存在区間を決定し，逆反復法で固有値・固有ベクトルを求める．三重対角化を行わないため，少数の固有値・固有ベクトルを求める場合は特に高速である．精度も良く，一般固有値問題に拡張が容易だが，2分法に要するCPUタイムが多い[2]．

(5)スツルム同時逆反復法は2分法で固有値の存在

† Symmetric Band Eigenvalue Solvers for Vector Computers and Conventional Computers by HIDEHIKO HASEGAWA (University of Library and Information Science).
†† 図書館情報大学

区間を決定し，同時逆反復法と逆反復法を組み合わせて固有値・固有ベクトルを求める．文献2)によれば，固有値のグループ分けが難しく，特に複数の密集固有値が存在する場合が問題である．

(6)ランチョス法は再直交化の範囲・程度などデータに依存する部分や，経験に基づく部分が多く汎用的なアルゴリズムが作りにくい．Parlett と Scott の選択的再直交化が推奨されているが，それでも汎用性に疑問があるとされている[3),4)]．

本論文では(3)三重対角化，2分法，逆反復法と(4)スツルム逆反復法についての比較を行う．

実測には，ベクトル計算機として東京大学大型計算機センターの S-810/20，汎用計算機として図書館情報大学の M-260 H を用いた．

2. 三重対角化を経由する方法

帯行列の三重対角化プログラムにはハウスホルダー変換を用いた TRIDIF[5),6)] と，*FGT* (Fast Givens Transformation) を用いた BANDR[7)~9)] がある．どちらも k 列の消去で生じた fill-in によって帯幅が広がらないよう付加的な変換で fill-in の消去を繰り返して帯半幅を元の帯半幅の2倍に保つ．

得られた三重対角行列に対して，スツルムの定理を利用した2分法で固有値の分布を調べる．スツルムの定理を利用した固有値カウントは一次巡回演算となるため，パイプライン方式のベクトル計算機に向かない．ベクトル計算機では複数の値についての固有値カウントを同時に行う多分法を用いてベクトル化を図るのがよいとされている[2),10)]．

2分法は固有値の存在区間を希望の精度で決定できるが，ここでは区間幅が

$$0.88\times10^{-15}\times\|A\|_1$$

以下になるまで2分法を繰り返す．ただし途中で孤立した固有値であることが判明した場合には，そのあと10回の繰り返しで2分法を打ち切ることで（区間幅は約 10^{-3} になる）*QL* 法よりも高速になる[1)]．

固有値の存在区間の情報を利用して三重対角行列に対する逆反復法を行い，レーリー商を用いて正確な固有値 λ・固有ベクトル $\boldsymbol{u}$ を決定する．逆反復法のシフト点は固有値の存在区間を下から 1：3 に内分する点を選ぶ．直交化の範囲は固有値の密集の程度によって決め，逆反復の繰り返し回数は最大で 40 回，密集と判定された部分では最低4回の逆反復を行うものとする．

固有値問題 $A\boldsymbol{v}=\lambda\boldsymbol{v}$ に直交変換を用いて三重対角化を施すと $T\boldsymbol{u}=PAP^{\mathrm{T}}\boldsymbol{u}=\lambda\boldsymbol{u}$，$\boldsymbol{v}=P^{\mathrm{T}}\boldsymbol{u}$ となり，固有値 λ だけが必要な場合，固有値 λ と三重対角行列の固有ベクトル $\boldsymbol{u}$ が必要な場合はここまでで計算をやめてよい．帯行列の固有ベクトル $\boldsymbol{v}$ は求められた固有値 λ をシフト点にして帯行列に対する逆反復法で求める．以下の計算で $\boldsymbol{u}$ は不用である．

帯行列に対する逆反復法の過程でもレーリー商を用いて，固有値 λ・固有ベクトル $\boldsymbol{v}$ の精度を改善する．逆反復法の過程で $A-\lambda I$ の *LU* 分解が ε で特異と判定されたときは，$\varepsilon/2$ として再度 *LU* 分解を行う．$\varepsilon/2$ でも特異になる場合には，対角要素に強制的に ε' を詰めて計算を続行する．帯行列に対する逆反復法では

$$\varepsilon=0.11\times10^{-15}\times\|A\|_1,$$

$$\varepsilon'=0.55\times10^{-16}\times\|A\|_1$$

とし，Wilkinson の初期ベクトルを用いて最大 40 回の反復とした．

帯行列に対する逆反復法で最も CPU タイムを消費するのは，固有値 λ ごとに $A-\lambda I$ を *LU* 分解する部分である．そのためベクトル計算機向けに2段2行同時の帯ガウスを用いて高速化を図った[11)]．*LU* 分解を2段2行同時で行うと逆反復も2段同時に実行できる．2段2行同時の帯ガウスは S-810/20 で 50%，M-260 H で 30% 程度の高速化になる．

収束判定は残差ノルムで

$$4\times m\times0.22\times10^{-15}\times\|A\|_1$$

とした．m は帯半幅である．必要なメモリ容量は両算法とも帯行列の三重対角化が $2mn$ 語（n は元数），帯行列に対する逆反復法が $3mn$ 語，係数行列を保持するためが mn 語である．三重対角化を経由したときに必要なメモリ容量は，およそ

$$4mn+\text{固有値・固有ベクトル}+\text{作業領域}$$

語である．

2.1 ハウスホルダー三重対角化，2分法，逆反復法

応用を意識した固有値解析の問題として，図1のような2次元の場に対する一般化ポアソン方程式を差分法で離散化して得られる帯行列を用いて実測を行った．この例では分割 $m1$ が非対角帯半幅を決め，帯半幅 $m=m1+1$，元数 $n=m1\,(m1+1)$ である．拡散係数 *DF* が小さくなると0の近くに多数の固有値が現れ，また対称に近い場となるため全体にわたって2重複に近い固有値の組が多数現れる．このため2分法の反復回数が増える．実行に要した CPU タイムと反復

表 1 ハウスホルダー三重対角化を用いた結果 (単位: 秒)
Table 1 Results of Householder's tridiagonalization.

(a) S-810/20

$m1$	n	DF	ハウスホルダー三重対角化	2 分 法	逆反復法 (三重対角行列)	逆反復法 (帯行列)	合 計
20	420	1	1.17	0.53 (1130 回)	0.48 (362 回)	2.42 (100 回)	4.61
		10^{-6}	1.17	0.69 (1462 回)	0.43 (300 回)	2.40 (100 回)	4.70
40	1640	1	17.35	4.14 (2286 回)	3.52 (691 回)	29.00 (200 回)	54.03
		10^{-6}	17.58	5.33 (2935 回)	3.15 (569 回)	30.40 (200 回)	56.48
80	6480	1	292.88	32.72 (4574 回)	27.07 (1288 回)	436.75 (400 回)	789.43
		10^{-6}	292.64	42.74 (5928 回)	24.39 (1101 回)	440.24 (400 回)	800.03

(b) M-260H

$m1$	n	DF	ハウスホルダー三重対角化	2 分 法	逆反復法 (三重対角行列)	逆反復法 (帯行列)	合 計
20	420	1	13.64	1.42 (1130 回)	3.69 (362 回)	47.32 (100 回)	66.08
		10^{-6}	13.63	1.91 (1462 回)	3.27 (301 回)	47.07 (100 回)	65.89
40	1640	1	400.95	11.42 (2286 回)	28.48 (691 回)	1179.22 (200 回)	1620.09
		10^{-6}	403.10	15.39 (2935 回)	24.87 (569 回)	1187.24 (200 回)	1630.61

( )内は反復回数

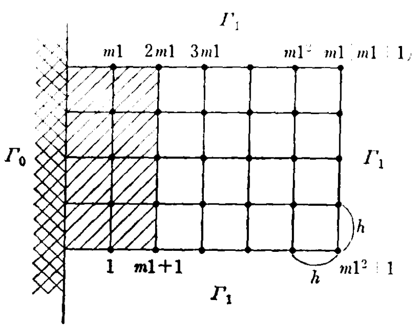

図 1 テスト問題
Fig. 1 Testing problems.

一般化ポアソン方程式 $\mathrm{div}(-k(x,y)\nabla u)=f$ を
境界条件: Γ_0 ディリクレ条件 $u=0$
Γ_1 ノイマン条件 $\nabla u\cdot n=0$
で離散化する. ■部の $k(x,y)$ を DF, □部は $k(x,y)=1$ とする. 非対角帯半幅が $m1$, 元数 $n=m1(m1+1)$ の帯行列ができる.

回数の結果を**表 1** に示す. 精度の評価は 4 章で述べる.

表 2 2 分法と多分法の結果 (単位: 秒)
Table 2 Results of bisection and multisection.

$m1=20$, $n=420$, $DF=1$

		2 分 法	多 分 法
$p=5m1$	S-810/20	0.53 (1130 回)	0.21 (4513 回)
	M-260H	1.42	5.34
$p=n$	S-810/20	2.25 (4812 回)	0.64 (17871 回)
	M-260H	5.94	13.61

( )内は反復回数

2.2 多 分 法

文献 2), 10) によればベクトル計算機では多分法を用いるのがよいとされている. 2 分法は一次巡回演算が現れるため, ベクトル計算機では高速化がされにくい. 複数の値に対する固有値カウントを同時に実行して高速化を図るのが多分法である. 汎用計算機で多分法を用いると無駄な演算のために 2 分法に対応する部分が数倍遅くなる. 多分法のプログラム MSECT[2] と

2分法のプログラム[1]を比較し，結果を表2に示す．

これから，多分法はベクトル計算機では2分法に対応する部分の CPU タイムを 30% にするが，汎用計算機では2～4倍にする．一概にどちらがよいとは言えないが，$m1=80$ のときベクトル計算機での2分法の実行時間はハウスホルダー三重対角化の約 10～15%にすぎない．帯行列に対する逆反復法までを含めるとその割合はさらに小さくなり，$5m1$ 組の固有値・固有ベクトルを求めるときには4%となる．汎用計算機の場合も $m1=40$ のときハウスホルダー三重対角化の約 5% なので，2～4 倍になってもそれほど問題ではない．

ベクトル計算機と汎用計算機で同一のプログラムを使用するという立場では，固有値の分布によらず特定の区間の固有値が簡単に求められる2分法がよい．

2.3 ギブンス法

ハウスホルダー変換の代わりに面内回転を用いても三重対角化ができる．ギブンス変換では変換ごとに開平演算が必要なため，演算量が約半分の *FGT* (Fast Givens Transformation, または Scaled Givens Transformation) が使われる[4),8)]．*FGT* の基本演算は

$$x'=ax+y,$$
$$y'=x+by$$

なのでベクトル計算機向きではない．またハウスホルダー変換の分母はベクトルのノルムなのに対し，*FGT* の分母は2つの値の二乗和なので係数 a, b を求めるときに0による除算が起こりやすく，スケーリングが必要である．

EISPACK の BANDR[7),9)] は仮想メモリ方式を意識していないので，BANDR から変換行列 P を作る部分を取り去り，メモリ参照が連続的になるよう改良した．ハウスホルダー変換を用いた TRIDIF と *FGT* を用いた BANDR の三重対角化に要する CPU タイムを表3に示す．

これから TRIDIF と BANDR の比は $m1=40$ のとき汎用計算機で 0.72 (1/1.38)，ベクトル計算機で 1.55 となる．$m1$ が大きくなると，汎用計算機では差がせばまるが，ベクトル計算機では差がひらく．小規模問題でこの程度の比だと実際の値に大きな差はなく，むしろ大規模問題に対して高速な手法が望ましい．したがってベクトル計算機と汎用計算機で同一のプログラムを使用するという立場では，ハウスホルダー変換を用いて三重対角化するのがよい．

表3 ハウスホルダー変換とギブンス法の比較 (単位：秒)
Table 3 Results of Householder's tridiagonalization and Givens rotation.

$m1$	n	S-810/20		M-260H	
		Householder (TRIDIF)	Givens (BANDR)	Householder (TRIDIF)	Givens (BANDR)
20	420	1.17	1.43	13.64	8.95
40	1640	17.35	26.93	400.95	290.19
60	3660			3002.96	2149.14
80	6480	292.88	558.49		

ギブンス法をベクトル計算機で高速化する手法については文献 12) があるが，文献 12) は特定のベクトル計算機についてだけの高速化を行っており，汎用計算機での実行が考慮されていない．そのため本論文での立場とは異なるので比較は行わなかった．

3. スツルム逆反復法

スツルム逆反復法では，帯行列に対して直接2分法を行って固有値の存在区間を決定し，逆反復法で固有値 λ・固有ベクトル $\boldsymbol{v}$ を求める．この算法は標準固有値問題

$$A\boldsymbol{v}=\lambda\boldsymbol{v}$$

が解ければ，帯に保ったままで一般固有値問題

$$A\boldsymbol{v}=\lambda M\boldsymbol{v}$$

に拡張できる点が特徴である．三重対角化を経由する方法では帯行列のままで一般固有値問題に拡張することが難しい．

帯行列に対する固有値カウントでは，固有値カウント1回につき帯行列の LU 分解もしくは U^TDU 分解程度の演算量が必要である．多数の固有値・固有ベクトルが必要な場合には，固有値カウントの容易な三重対角行列を経由する方法が効率的だが，三重対角化には多くの CPU タイムがかかるので，少数の固有値・固有ベクトルが必要な場合にはスツルム逆反復法がよい．帯幅が特に小さい場合にはスツルム逆反復法もかなりの個数の固有値・固有ベクトルまで効率的になる．

2分法では Martin-Wilkinson の特殊ガウス[7)]と軸選択なしの対称ガウスを併用して固有値カウントを行う[13)]．Martin-Wilkinson の特殊ガウスは精度はよいが，演算量は軸選択なしの対称ガウスの約4倍になる．2分法で軸選択なしの対称ガウスを使用すると，対称正定値性が保証されないため固有値カウントを間違えることがある．原則として固有値カウントを軸選

択なしの対称ガウスで行い，ピボットの絶対値が

$3.52\times10^{-15}\times\|A\|_1$

以下になったときだけ Martin-Wilkinson の特殊ガウスでやり直す．Martin-Wilkinson の特殊ガウスだけの場合と比べて固有値の分離が不十分な可能性があるが，それに対しては逆反復法の過程で手当てをすることで補う．軸選択なしの対称ガウスについても２段２行同時として高速化を図った[14]．Martin-Wilkinson の特殊ガウスは場所によってコンパイラによるベクトル化が行われないのでベクトル化オプション行を用いた．

２分法は固有値の区間幅が

$0.88\times10^{-15}\times\|A\|_1$

以下になるまで繰り返す．ただし途中で孤立した固有値であることが判明した場合には，そのあと６回の繰り返しで２分法を打ち切る．

三重対角行列に対する２分法では多分法を考慮する必要があったが，帯行列に対する２分法では固有値カウント１回についての演算量が U^TDU 分解と同程度なので多分法を考慮する必要はない．

逆反復法は三重対角化を経由する方法における帯行列に対する逆反復法とほぼ同じである．シフト点は固有値の存在区間を下から 1：3 に内分する点を選ぶ．$A-\lambda I$ の LU 分解が ε で特異と判定されたときは，$\varepsilon/2$ として再度 LU 分解を行う．$\varepsilon/2$ でも特異になったら対角要素に ε' を詰めて続行する．

$\varepsilon=0.11\times10^{-15}\times\|A\|_1$,

$\varepsilon'=0.55\times10^{-16}\times\|A\|_1$

とし，Wilkinson の初期ベクトルを用いて最大で 40 回の反復とする．密集と判定された部分では最低４回の反復を行う．LU 分解には２段２行同時の帯ガウス[11]を用いて高速化を行い，レーリー商を用いて固有値 λ・固有ベクトル $\boldsymbol{v}$ の精度を改善する．

収束判定は残差ノルムで

$4\times m\times0.22\times10^{-15}\times\|A\|_1$

とした．必要なメモリ容量は対称ガウスが mn 語，Martin-Wilkinson の特殊ガウスが $3mn$ 語，逆反復法が $3mn$ 語である（m は帯半幅，n は元数）．係数行列を保持するために mn 語必要なのでスツルム逆反復法で必要なメモリ容量は，およそ

$4mn$＋固有値・固有ベクトル＋作業領域

語で，三重対角化を経由する方法と同じである．

表４にスツルム逆反復法の CPU タイムと反復回数を示す．２分法の CPU タイムはベクトル計算機で約 65〜85%，汎用計算機で約 50〜70% を占め，n が大きくなるとその割合も増加する．２分法の反復回数は三重対角行列の場合よりも少ないが，CPU タイムは大幅に上回っている．帯行列に対する２分法では１回の固有値カウントに要する CPU タイムが大きいので，２分法の反復回数を減らすことが有効である．２分法の回数を減らすことで逆反復の回数が増えるが，逆反復法では１つの固有値について１回の LU 分解が必要なだけで，逆反復の回数が増えても CPU タイムの増加はわずかである．

表４ スツルム逆反復法を用いた結果（単位：秒）
Table 4 Results of Sturm inverse-iteration.

(a) S-810/20

ml	n	DF	２ 分 法	逆反復法	合 計
20	420	1	9.05 (734＋2回)	3.98 (487回)	13.03
		10^{-6}	12.97 (1062回)	3.46 (365回)	16.46
40	1640	1	128.94 (1494＋2回)	40.45 (847回)	169.40
		10^{-6}	183.67 (2146回)	37.56 (707回)	221.24
80	6480	1	1904.72 (2980＋1回)	545.54 (1649回)	2450.20
		10^{-6}	2764.78 (4352回)	519.90 (1337回)	3284.68

(b) M-260H

ml	n	DF	２ 分 法	逆反復法	合 計
20	420	1	91.99 (734＋2回)	77.57 (487回)	169.56
		10^{-6}	131.84 (1062回)	66.47 (365回)	198.32
40	1640	1	2286.85 (1494＋2回)	1505.77 (847回)	3792.63
		10^{-6}	3241.81 (2146回)	1405.77 (707回)	4647.58

（ ）内は反復回数
２分法の欄，＋以降は Martin-Wilkinson の特殊ガウスの回数

4. 精度の比較

三重対角化を経由する方法，スツルム逆反復法とも残差ノルムが

$4\times m\times0.22\times10^{-15}\times\|A\|_1$

以下を収束とみなした．このとき

（1） 残差ノルムの最大値

$\|\boldsymbol{r}_i\|_2=\|A\boldsymbol{v}_i-\lambda_i\boldsymbol{v}_i\|_2$,

表 5 誤差解析の結果
Table 5 Error analysis.

$m1$	n	DF	Sturm		Householder		Givens	
			残差ノルム	混じり分	残差ノルム	混じり分	残差ノルム	混じり分
20	420	1	0.15×10^{-12}	0	0.24×10^{-14}	0	0.18×10^{-14}	0
		10^{-6}	0.84×10^{-13}	0	0.25×10^{-14}	0	0.22×10^{-14}	0
		10^{-15}	0.87×10^{-13}	454	0.13×10^{-13}	449	0.13×10^{-13}	449
40	1640	1	0.29×10^{-12}	0	0.27×10^{-14}	0	0.60×10^{-14}	0
		10^{-6}	0.26×10^{-12}	0	0.43×10^{-14}	0	0.44×10^{-14}	0
		10^{-15}	0.68×10^{-11}	1646	0.36×10^{-13}	1687	0.42×10^{-13}	1691
80	6480	1	0.56×10^{-12}	0	0.86×10^{-14}	0	0.98×10^{-14}	0
		10^{-6}	0.50×10^{-12}	0	0.10×10^{-13}	0	0.10×10^{-13}	0

（2） 固有ベクトルの直交性

$$\boldsymbol{v}_i^{\mathrm{T}}\boldsymbol{v}_j=\boldsymbol{v}_j^{\mathrm{T}}\boldsymbol{v}_i,$$

（3） 固有ベクトルの混じり分[1),7)]

$$m_{ij}=\boldsymbol{v}_j^{\mathrm{T}}\boldsymbol{r}_i/(\lambda_j-\lambda_i)$$

を精度の評価項目とした．結果を**表 5** に示す．

（1） 残差ノルムの最大値は収束判定値以下だった．

（2） 固有ベクトルの直交性が 10^{-5} を越えた組み合わせはない．仮に直交性が悪い場合でも直交化の範囲を広げれば改善される．

（3） $DF=1$, 10^{-6} では固有ベクトルの混じり分が 10^{-4} を越える組み合わせはない．$DF=10^{-15}$ として極端な密集を生じさせると混じり分が 10^{-4} を越える組み合わせが現れるが，$m1=40$ のときに求めた固有ベクトル $5m1=200$ の組み合わせ中の 1600 なので割合は約 4% である．値は小さいものばかりである．

これから本論文で述べた算法の精度はいずれも満足な水準に達しているといえる．

5. 算法の有効領域

三重対角化を経由する方法では求めたい固有値・固有ベクトルの組数と関係なく三重対角化を 1 回行う．スツルム逆反復法などは求めたい固有値・固有ベクトルの組数に比例して反復回数が変化する．そこでハウスホルダー三重対角化を経由する方法とスツルム逆反復法の CPU タイムが同じになる平衡点 p_0 が求められる．平衡点 p_0（固有値・固有ベクトルの組数）は式

p_0（スツルム 2 分法＋逆反復法）
＝三重対角化＋p_0（三重対角行列に対する 2 分法＋三重対角行列に対する逆反復法＋逆反復法）

を満足する．計算で求めた p_0 の値を**表 6** に示す．

表 6 ハウスホルダー三重対角化を用いた方法とスツルム逆反復法の平衡点 p_0（組数）
Table 6 The equilibrium of tridiagonalization method and Sturm inverse-iteration.

$m1$	n	DF	S-810/20	M-260H
20	420	1	12.2	11.6
		10^{-6}	9.0	9.3
40	1640	1	26.1	31.1
		10^{-6}	19.2	23.5
80	6480	1	59.9	
		10^{-6}	42.1	

$n \fallingdotseq m^2$ の帯行列のときは帯半幅 m の約半分以下の組数の固有値・固有ベクトルを求めるならスツルム逆反復法が有利である．m, n が大きくなるとスツルム逆反復法の有効領域が広がるが，帯半幅 m を越えることはないだろう．

固有値だけが必要な場合にはハウスホルダー三重対角化を経由する方法が圧倒的に有利である．

$n \fallingdotseq km^2$ の帯行列のとき，p 組の固有値・固有ベクトルを求める場合の演算量を積和演算を基準として概算する．三重対角化を経由する方法の演算量はハウスホルダー三重対角化が $3mn^2$, 三重対角行列に対する 2 分法が αpn, 三重対角行列に対する逆反復法が $p(2n+\beta\cdot 2n)$, 帯行列に対する逆反復法が $p(2m^2n+2mn)$ である．スツルム逆反復法の演算量は対称ガウスを用いた 2 分法が $\alpha' p(m^2n/2)$, 帯行列に対する逆反復法が $p(2m^2n+\beta'\cdot 2mn)$ である．

α, β, α', β' は実測から得られた値を用いる（表 1,

表4参照). α は三重対角行列に対する2分法の平均的な反復回数で, $\alpha=10\sim15$ になる. β は三重対角行列に対する平均的な逆反復回数で, $\beta=2\sim3$ になる. α' は対称ガウスを用いた2分法の平均的な反復回数で, $\alpha'=7\sim12$ になる. β' はスツルム逆反復法の帯行列に対する平均的な逆反復回数で, $\beta'=3\sim5$ になる. 三重対角化を経由する方法の帯行列に対する逆反復では, 固有値が正確に求められているため1回の逆反復で収束する. α と α', β と β' を比べると, スツルム逆反復法では2分法の反復回数を減らして逆反復で補っていることがわかる. 以下では多くの CPU タイムがかかる場合に注目し, $\alpha=15$, $\beta=3$, $\alpha'=10$, $\beta'=5$ として概算を行う.

三重対角化を経由する方法の総演算量は

$$3mn^2+p\{15n+2n+6n+2m^2n+2mn\},$$

$n=km^2$, $p=5m$ とおいて m の関数に直すと

$$(3k^2+10k)m^5+10km^4,$$

スツルム逆反復法の総演算量は

$$p\{10m^2n/2+2m^2n+10mn\},$$

同様に m の関数に直すと

$$35km^5+50km^4$$

となる. m^5 の係数だけをとりだすと, 三重対角化を経由する方法が $3k^2+10k$, スツルム逆反復法が $35k$ となる. $k=1$ の行列ではスツルム逆反復法が約 2.7 倍の演算量となり, 汎用計算機での実測結果とよく合う. 元数に比べて帯幅が狭い $k=2$ の行列ではスツルム逆反復法が約 2.2 倍の演算量となる. これからスツルム逆反復法は帯幅が狭いときに有効なことがわかる. ベクトル計算機では演算量が直接 CPU タイムに反映しないので何ともいえないが, 帯幅が狭いときにはベクトル長が短くなりベクトル計算機での加速率が下がるだろう.

6. まとめ

本論文では対称帯行列の固有値・固有ベクトルを求めるための算法をベクトル計算機, 汎用計算機の双方で評価を行い, 両方の計算機に対して同一のプログラムでサービスをするという立場から優劣を考えた.

三重対角化を経由する方法では, 三重対角化にハウスホルダー変換を用い, 2分法と逆反復法を組み合わせて用いるのがよいことが明らかになった. *FGT* を用いた三重対角化は汎用計算機では若干速いがベクトル計算機向きでないため, ハウスホルダー三重対角化を用いるのがよい. 多分法はベクトル計算機では高速だが, 汎用計算機では遅く汎用性に欠ける.

スツルム逆反復法では, 2分法の固有値カウントに原則として軸選択なしの対称ガウスを用い, カウントを間違えた可能性があるときだけ Martin-Wilkinson の特殊ガウスを使うのがよい. 特に2分法の負担が重いので2分法の回数を減らすのがよい.

反復1回当たりの高速化を図るため, 帯ガウスや対称ガウスは2段2行同時とした. どの算法も精度についての問題はなかった.

その結果, ベクトル計算機 S-810/20 で帯半幅 $m=81$, 元数 $n=6480$ の対称帯行列の小さいほうから 400 組の固有値・固有ベクトルが, 三重対角化を経由する方法では

約 800 秒 (2秒/組),

スツルム逆反復法では

約 3200 秒 (8秒/組)

で求められた. スツルム逆反復法はわずかの変更で帯を保ったまま一般固有値問題にも適用できる. 以前報告したデータ[15]では $m=21$, $n=840$ の問題に Martin-Wilkinson の特殊ガウスだけを使ったスツルム逆反復法は約 900 秒かかっていたのが, 軸選択なしの対称ガウスを併用すると約 420 秒になり, さらに2段2行同時にすると約 340 秒になった. 元のプログラムから約3倍の高速化が図られているが, これらの高速化は算法の見直し・改良と, 繰り返して使われる部分の高速化 (2段2行同時など) で実現された. 今まではベクトル計算機向けの高速化手法としてしばしば汎用計算機にとって好ましくない手法が推奨されていたが, 本論文ではベクトル計算機と汎用計算機の両方に好ましい手法だけを用いて, 高速な固有値解析プログラムが作成できることを示した.

謝辞 本論文のテーマ全体にわたって指導と討論をしてくださった村田健郎先生に深く感謝します.

参考文献

1) 村田健郎: 標準形対称行列固有値解法の見直し (I), 図書館情報大学研究報告, Vol. 5, No. 1, pp. 1-24 (1986).
2) 村田健郎, 小国 力, 唐木幸比古: スーパーコンピュータ, p. 304, 丸善, 東京 (1985).
3) 名取 亮, 野寺 隆: ランチョス法その後, 数理解析研究所講究録, No. 585, pp. 275-293 (1986).
4) Parlett, B. N.: *The Symmetric Eigenvalue Problem*, p. 348, Prentice-Hall, Englewood-Cliffs (1980).
5) 村田健郎, 堀越清視: 対称帯行列を三重対角化

するための新アルゴリズム，情報処理，Vol. 16, No. 2, pp. 93-101 (1975).
6) 長谷川秀彦：拡張ハウスホルダー法による対称帯行列の三重対角化プログラム，図書館情報大学研究報告，Vol. 5, No. 1, pp. 47-70 (1986).
7) Wilkinson, J. H., Reinsch, C.: *Handbook for Automatic Computation, Vol. II—Linear Algebra*, p. 439, Springer-Verlag, Berlin (1971).
8) Gentleman, W. M.: Least Squares Computation by Givens Transformations without Square Roots, *J. Inst. Maths Applics*, Vol. 12, pp. 329-336 (1973).
9) Garbow, B. S. et al.: *Matrix Eigensystem Routines—EISPACK Guide Extension, Lecture Notes in Computer Science*, Vol. 51, p. 343, Springer-Verlag, Berlin (1977).
10) 島崎真昭：対称三項行列の固有値の並列探索，数理解析研究所講究録，No. 514, pp. 219-230 (1984).
11) 長谷川秀彦：帯行列を係数とする連立一次方程式の解法(I)，図書館情報大学研究報告，Vol. 6, No. 1, pp. 45-59 (1987).
12) Kaufman, L.: Banded Eigenvalue Solvers on Vector Machines, *ACM Trans. Math. Softw.*, Vol. 10, No. 1, pp. 73-86 (1984).
13) 村田健郎：標準形対称行列固有値解法の見直し(IV)，図書館情報大学研究報告，Vol. 6, No. 2, pp. 67-78 (1987).
14) 長谷川秀彦：帯行列を係数とする連立一次方程式の解法(III)，図書館情報大学研究報告，Vol. 7, No. 1, pp. 61-73 (1988).
15) 長谷川秀彦，村田健郎：スーパーコンピュータ向けの帯行列固有値解析プログラムについて，第15回数値解析シンポジウム論文集，pp. 15-18 (1986).

(昭和63年5月30日受付)
(平成元年1月17日採録)

長谷川秀彦（正会員）
1958年横浜生．1980年筑波大学第1学群自然学類(数学専攻)卒業．1983年同大学院社会工学研究科（経営工学専攻）中退．同年図書館情報大学助手．学術修士．現在，連立一次方程式の解法，固有値解析などの線形数値計算法に興味を持つ．ACM 会員．