

時間変化する大規模対称行列の固有値計算の速度比較

平 博行[†] 金谷 健一[†]

[†] 岡山大学大学院自然科学研究科

あらまし コンピュータビジョンを含む多くの問題に現れる、時間変化がある大規模対称行列の最大固有値およびその固有ベクトルを計算するための反復解法として、最急降下法、共役勾配法、べき乗法、およびその加速を考察する。まず各手法をまとめ、次にシミュレーションデータを用いて実行時間を比較する。そして、各手法の計算効率と対象とする行列の固有値の分布との関係を明らかにする。

キーワード 大規模対称行列、固有値問題、最急降下法、共役勾配法、べき乗法、収束の加速

Comparison of Eigenvalue Computation Speed for Time-Varying Large-Size Symmetric Matrices

Hiroyuki TAIRA[†] and Kenichi KANATANI[†]

[†] Department of Computer Science, Okayama University, Okayama, 700-8530 Japan

Abstract We study iterative schemes for computing the largest eigenvalue and the corresponding eigenvector of a time-varying large-scale symmetric matrix, which frequently arises in many problems including computer vision. Here, we focus on the steepest descent, the conjugate gradient, the power method, and its acceleration. We first summarize their algorithms and then compare their computation time, using simulated data. We show how the computational efficiency of individual methods is related to the eigenvalue distribution of the matrix in question.

Key words large-scale matrix, eigenvalue problem, steepest descent, conjugate gradient method, power method, acceleration of convergence

1. まえがき

最適化はコンピュータビジョンのみならず、工学のあらゆる問題の中心的な処理の一つである。このとき、最大または最小にする目的関数がパラメータの2次形式であれば、解は対称な係数行列の最大または最小固有値に対する固有ベクトルで与えられる[5]。このため、対称行列の固有値計算は最適化の最も重要な要素である。

対称行列の固有値計算の代表的な数値解法にヤコビ法とハウスホルダー法がある[11]。特に後者は精度と計算効率に優れていることが知られ、広く利用されている。しかし、これらは全部の固有値と固有ベクトルを計算するものである。それに対して最適化問題では最大または最小固有値とその固有ベクトルのみが得られれば十分である。本論文ではそのような場合の計算法を考察する。

以下では、最大化を想定し、最大固有値に対する固有ベクトルの計算を考える。これは、利得を最大にするような始めから最大化問題として定式化される場合とともに、モデルとデータの食い違いを最小にするような最小化問題も最大化問題に帰着することが多いからである。

代表的なものは多数の多変量データを高次元空間の点群とみ

なし、それに直線や平面や超平面を当てはめる問題である。これはその点群の重心を座標原点にとれば、点群に部分空間を当てはめる問題となる。このとき、各点からその部分空間までの距離の二乗和を最小にする問題は、各点をその部分空間に射影した垂線の足の原点からの距離の二乗和を最大にする問題と等価となる。

さらに本論文では固有値を計算する行列が大規模であり、かつ時間的に連続変化する場合を考える。例えば、新しいデータが次々の観測され、既に計算した最適解を逐次的にオンライン更新する場合がある。あるいは固有値計算が最適化の一部をなし、計算した固有値、固有ベクトルによって行列を変化させ、再び固有値、固有ベクトルを計算し、これを収束するまで反復するような処理がある。例えば未校正カメラ画像から3次元復元を行う自己校正法においてこのような処理が現れる[6], [8], [9]。固有値計算の効率が問題になるのは主としてこのような場合であり、行列が小規模であったり、一度だけ計算すればよい場合は、どのような計算法を用いても実質的な差は生じない。

行列が連続的に変化すると、その固有値とその固有ベクトルは変化する前の値とほぼ等しいと期待される。したがって、固有値と固有ベクトルを始めから計算し直すのではなく、前回の値を少ない計算量で補正することによって新しい解に更新でき

れば都合がよい．

このことから，本論文では与えられた初期値から固有値，固有ベクトルを計算する反復解法を考察する．このような方法では初期値が十分解に近ければ少ない反復で効率よく解が計算できるからである．

固有値計算の反復解法は物理学や工学の多くの問題に用いられているが，その主目的は大規模疎行列に対する効率化である．ほとんどの行列要素が 0 であれば，反復公式は項数の少ない簡単な式になる．自然現象に由来する問題（相互作用が局所的）では疎行列がよく現れる．しかし，本論文では画像からの 3 次元復元 [6], [8], [9] などへの応用を想定し，行列は一般に密であるとする．

以下，本論文では最急降下法，共役勾配法，ベキ乗法，ベキ乗法の加速を比較検討する．最急降下法は最大固有値の計算を 2 次形式の最大化に帰着させて最大化の探索を行うものであり，共役勾配法はその改良である [3], [5]．共役勾配法にもいくつかのバリエーションがあり，ここでは原形 [13] とフレッチャー・リーブスの方法およびボラック・リビエールの方法 [2], [10], [11] を比較する．

2. 最急降下法

よく知られているように，対称行列 A の最大固有値に対する固有ベクトルはレイリー商

$$J = \frac{(x, Ax)}{(x, x)} \quad (1)$$

を最大にするベクトル x である [4]．以下，本論文ではベクトル a, b の内積を (a, b) と書く． x を何倍しても J の値は同じなので， x としては単位ベクトルを選べば，制約条件 $\|x\| = 1$ のもとでの 2 次形式 (x, Ax) の最大化になるが [4]，式 (1) のままのほうが制約条件がないので都合がよい．

式 (1) の勾配 $g = (\partial J / \partial x_i)$ は次のようになる．

$$g = \frac{2(x, x)Ax - 2(x, Ax)x}{(x, x)^2} \quad (2)$$

この勾配方向に J を極大にする直線探索を行うのが「最急降下法」^(注1)（あるいは「勾配法」）である．そこで探索方向を

$$p = g \quad (3)$$

と置く．式 (1) に

$$x' = x + \alpha p \quad (4)$$

を代入し， α で微分して 0 とおくと次式を得る．

$$(a_{pp}b_{xp} - a_{xp}b_{pp})\alpha^2 + (a_{pp}b_{xx} - a_{xx}b_{pp})\alpha + (a_{xp}b_{xx} - a_{xx}b_{xp}) = 0 \quad (5)$$

ただし，次のように置いた．

$$\begin{aligned} (x, Ax) &= a_{xx}, & (x, Ap) &= a_{xp}, & (p, Ap) &= a_{pp}, \\ (x, x) &= b_{xx}, & (x, p) &= b_{xp}, & (p, p) &= b_{pp} \end{aligned} \quad (6)$$

(注1): J を最大化するのであるから，正しくは「最急上昇法」と言うべきであるが，慣用により「最急降下法」と呼ぶ．

式 (3) より p は J が増加する方向を指しているので， α は正である．式 (5) に正の解が二つあるなら，他の解は極大値を通り越した極小値である．そこで正の解が二つあれば小さいほうを選ぶ．以上より，次の手順が得られる．

- (1) 初期値 x を与える．
- (2) 次のベクトル y を計算する．

$$y = Ax \quad (7)$$

- (3) 次のように a_{xx}, b_{xx} を計算する．

$$a_{xx} = (x, y), \quad b_{xx} = (x, x) \quad (8)$$

- (4) 次のスカラー J とベクトル p, z を計算する．

$$J = \frac{a_{xx}}{b_{xx}}, \quad p = 2 \frac{y - Jx}{b_{xx}}, \quad z = Ap \quad (9)$$

- (5) 次の $a_{xp}, a_{pp}, b_{xp}, b_{pp}$ を計算する．

$$a_{xp} = (x, z), \quad a_{pp} = (p, z), \quad b_{pp} = (p, p) \quad (10)$$

- (6) 次の α_1, α_2 を計算する．

$$\alpha_{1,2} = \frac{a_{xx}b_{pp} - a_{pp}b_{xx} \pm \sqrt{(a_{xx}b_{pp} - a_{pp}b_{xx})^2 + 4a_{xp}^2b_{xx}b_{pp}}}{-2a_{xp}b_{pp}} \quad (11)$$

- (7) α_1, α_2 の小さいほうを α_- ，大きいほうを α_+ とし，次のように α を定める．

$$\alpha = \begin{cases} \alpha_+ & \alpha_- < 0 \\ \alpha_- & \alpha_- \geq 0 \end{cases} \quad (12)$$

- (8) 次の x' を計算する

$$x' = x + \alpha p \quad (13)$$

- (9) $|\alpha b_{pp}/b_{xx}| \approx 0$ なら x' を返して終了．そうでなければ次のように更新してステップ (3) に戻る．

$$x \leftarrow x', \quad y \leftarrow y + \alpha z \quad (14)$$

【計算量】毎回の反復において，行列とベクトルの乗算（計算量が $O(n^2)$ ）はステップ (4) の $z = Ap$ の 1 回のみであり，他はすべてベクトルのスカラー倍，和，差，内積（計算量は $O(n)$ ）である．式 (7) の $y = Ax$ の値は現在の y の値から z を用いて式 (14) の第 2 式によって次回の値に更新している．

3. 共役勾配法

最急降下法では式 (3) のように，探索方向 p を勾配 g にとつたが，必ずしもその必要はない．そこで探索方向 p を次のように定める．

$$p = g + \beta p_0 \quad (15)$$

ただし， p_0 は直前の探索方向である．「共役勾配法」とは直前の探索方向 p_0 と新しい探索方向 p が“共役”，すなわち

$$(p_0, Hp) = 0 \quad (16)$$

となるように定めるものである [3], [5]．ただし， J のヘッセ行

列を $H = (\partial^2 J / \partial x_i \partial x_j)$ と置いた．式 (15) を代入すると

$$(p_0, Hg + \beta H p_0) = 0 \quad (17)$$

となる．これから β が次のように定まる．

$$\beta = -\frac{(p_0, Hg)}{(p_0, H p_0)} \quad (18)$$

直前の直線探索では J が x で極大値をとるから， p_0 は x における J の等高線の接線方向でなければならない[5]．したがって， p_0 は x における J の勾配 g に直交する．

$$(p_0, g) = 0 \quad (19)$$

ヘッセ行列 H が単位行列 I の定数倍であれば式 (18) は $\beta = 0$ となり，探索は最急降下法に一致する．

式 (2) を微分すると，ヘッセ行列は次のようになる（途中計算省略）．

$$H = 2 \frac{A - JI - gx^T - xg^T}{(x, x)} \quad (20)$$

これを式 (18) に代入して，式 (19) を用いると，

$$\beta = -\frac{(p_0, Ag) - (p_0, x)(g, g)}{(p_0, Ap_0) - J(p_0, p_0)} \quad (21)$$

となる．すなわち，式 (20) のヘッセ行列 H そのものは計算する必要がない[13]．探索方向 p が定まれば，式 (4) による次の値 x' を定める α は最急降下法の場合と同じように求まる．したがって，次の手順が得られる．

(1) 初期値 x を与え， $p_0 = 0$ ， $z_0 = 0$ ， $a_{pp}^0 = 0$ ， $b_{pp}^0 = 0$ ， $k = 0$ と置く．

(2) 次のベクトル y を計算する．

$$y = Ax \quad (22)$$

(3) 次のように a_{xx} ， b_{xx} を計算する．

$$a_{xx} = (x, y), \quad b_{xx} = (x, x) \quad (23)$$

(4) 次のスカラー J ，ベクトル g ， w を計算する．

$$J = \frac{a_{xx}}{b_{xx}}, \quad g = 2 \frac{y - Jx}{b_{xx}}, \quad w = Ag \quad (24)$$

(5) 次の β を計算する．

$$\beta = \begin{cases} 0 & k = 0 \\ -\frac{(p_0, w) - (p_0, x)(g, g)}{a_{pp}^0 - Jb_{pp}^0} & \text{それ以外} \end{cases} \quad (25)$$

(6) 次のベクトル p ， z を計算する．

$$p = g + \beta p_0, \quad z = w + \beta z_0 \quad (26)$$

(7) 次の a_{xp} ， a_{pp} ， b_{xp} ， b_{xp} ， b_{pp} を計算する．

$$a_{xp} = (x, z), \quad a_{pp} = (p, z), \quad b_{xp} = (x, p), \quad b_{pp} = (p, p) \quad (27)$$

(8) 次の α_1 ， α_2 を計算する．

$$\alpha_{1,2} = \frac{a_{xx}b_{pp} - a_{pp}b_{xx} \pm \sqrt{(a_{xx}b_{pp} - a_{pp}b_{xx})^2 + 4a_{xp}^2b_{xx}b_{pp}}}{-2a_{xp}b_{pp}} \quad (28)$$

(9) α_1 ， α_2 の小さいほうを α_- ，大きいほうを α_+ とし，次のように α を定める．

$$\alpha = \begin{cases} \alpha_+ & \alpha_- < 0 \\ \alpha_- & \alpha_- \geq 0 \end{cases} \quad (29)$$

(10) 次の x' を計算する

$$x' = x + \alpha p \quad (30)$$

(11) $|\alpha b_{pp}/b_{xx}| \approx 0$ なら x' を返して終了する．そうでなければ次のように更新してステップ (3) に戻る．

$$\begin{aligned} x &\leftarrow x', & y &\leftarrow y + \alpha z, & p_0 &\leftarrow p, & z_0 &\leftarrow z, \\ a_{pp}^0 &\leftarrow a_{pp}, & b_{pp}^0 &\leftarrow b_{pp}, & k &\leftarrow k + 1 \end{aligned} \quad (31)$$

【計算量】上記の手順でも，毎回の反復で行列とベクトルの乗算が現れるのはステップ (4) の $w = Ag$ の 1 回のみである．上記の計算には $z (= Ap)$ と $y (= Ax)$ の値も必要であるが， z は w を用いて式 (26) の第 2 式で計算し， y は現在の y の値から z を用いて式 (31) の第 2 式によって次回の値に更新している．

最大化する関数が式 (1) のレイリー商 J であれば，ヘッセ行列 H は式 (20) の形に求まり，それを用いた β が式 (21) で与えられるので， H 自身を評価する必要がない．しかし一般の非線形関数ではヘッセ行列を計算するのが困難なことが多く，また変数が多いと H の格納に多くのメモリを要する^(注2)．そのような非線形最適化の共役勾配法でよく用いられているのは式 (18) の β を次の形に近似することである．

$$\beta = \frac{(g, g)}{(g_0, g_0)}, \quad \beta = \frac{(g, g - g_0)}{(g_0, g_0)} \quad (32)$$

ただし， g_0 は直前の探索における勾配 g の値である．第 1 式は「フレッチャー・リーブスの式」，第 2 式は「ポラック・リビエールの式」と呼ばれている [2], [10], [11]．共役勾配法のライブラリプログラムには普通これらが用いられている^(注3)．目的関数が 2 次式であれば式 (32) は式 (18) に一致する（証明は [3] 参照）．

4. べき乗法

4.1 原 型

対称行列 A の最大固有値に対する単位固有ベクトルを求める「べき乗法」は次の反復を行うものである [1], [7], [12]．ただし，以下 $N[\cdot]$ は単位ベクトルへの正規化作用素を表す．

(1) 初期値（単位ベクトル） x^0 を任意に与える．

(2) 次の単位ベクトル x^1 を計算する．

$$x^1 = N[Ax^0] \quad (33)$$

(3) $x^1 \approx x^0$ であれば x^1 を返して終了する．そうでなければ $x^0 \leftarrow x^1$ としてステップ (2) に戻る．

【原理】行列 A の固有値を絶対値の大きい順に並べたものを $\lambda_1, \lambda_2, \dots, \lambda_n$ とし，対応する単位固有ベクトルの直交系を $u_1,$

(注2)：本論文のすべての手法は，入力データ A 以外に行列変数を用いてない．

(注3)：文献 [11] のプログラムではポラック・リビエールの式が採用されている．

u_2, \dots, u_n とする．これらは正規直交系をなすから，任意のベクトル x は次のようにこれらの線形結合で表せる．

$$x = c_1 u_1 + c_2 u_2 + \dots + c_n u_n \quad (34)$$

$Au_1 = \lambda_1 u_1, \dots, Au_n = \lambda_n u_n$ であるから，上式の両辺に A^k を掛けると次のようになる．

$$\begin{aligned} A^k x &= c_1 \lambda_1^k u_1 + c_2 \lambda_2^k u_2 + \dots + c_n \lambda_n^k u_n \\ &= c_1 \lambda_1^k \left(u_1 + \frac{c_2}{c_1} \left(\frac{\lambda_2}{\lambda_1} \right)^k u_2 + \dots + \frac{c_n}{c_1} \left(\frac{\lambda_n}{\lambda_1} \right)^k u_n \right) \end{aligned} \quad (35)$$

λ_1 は (絶対値) 最大の固有値であるから， $|\lambda_2/\lambda_1| < 1, \dots, |\lambda_n/\lambda_1| < 1$ であり，いずれも k 乗すると $k \rightarrow \infty$ で 0 に収束する．したがって，十分大きい k に対しては

$$A^k x \approx c_1 \lambda_1^k u_1 \quad (36)$$

となり，これを単位ベクトルに正規化すると u_1 が得られる．

4.2 加速

数値解析 [7], [12] でよく知られているように，誤差が指数的に減少する反復は反復の前後の値からその減衰定数を推定して収束を速めることができる．これは「エイトケンの δ^2 」, 「リチャードソン補外」などと呼ばれ，その代表的な例は数値積分の分点を増やしながらか加速する「ロンバーグ積分法」である．同じ原理はべき乗法の収束の加速に利用できるが，べき乗法はベクトルの反復なので，そのままの形では適用できない．そこで以下のような変形を考える^(注4)．

- 単位ベクトル x^0 を任意に与え， $x^1 = x^0$ と置く．
- 次の計算を $k = 1, 2, \dots$ に対して計算する．

(1) 次の x^2 を計算する．

$$x^2 = N[Ax^1] \quad (37)$$

(2) $x^2 \approx x^1$ であれば x^2 を返して終了する．

(3) k が奇数なら次のように x^0, x^1 を更新してループを終了し，次の k の計算に進む．

$$x^0 \leftarrow x^1, \quad x^1 \leftarrow x^2 \quad (38)$$

(4) k が偶数なら，次の計算を行う．

$$\Delta x^0 = x^1 - x^0, \quad \Delta x^1 = x^2 - x^1, \quad \gamma = \frac{(\Delta x^1, \Delta x^0)}{\|\Delta x^0\|^2} \quad (39)$$

(5) 次のように x^0, x^1 を更新してループを終了し，次の k の計算に進む．

$$x^0 \leftarrow x^1, \quad x^1 \leftarrow N\left[\frac{\gamma x^1 - x^2}{\gamma - 1}\right] \quad (40)$$

【導出】 $x^k = N[A^k x]$ と置く． $|\lambda_2/\lambda_1| > |\lambda_3/\lambda_1| > \dots$ であるから，式 (35) より次の関係を得る．

$$x^k \approx u_1 + C \gamma^k u_2, \quad C \equiv \frac{c_2}{c_1}, \quad \gamma \equiv \frac{\lambda_2}{\lambda_1} \quad (41)$$

式 (41) の k を $k+1$ とすると次の関係を得る．

$$x^{k+1} \approx u_1 + C \gamma^{k+1} u_2 \quad (42)$$

式 (41), (42) をまとめると次のように書ける．

$$\begin{pmatrix} u_1 & C \gamma^k u_2 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & \gamma \end{pmatrix} \approx \begin{pmatrix} x^k & x^{k+1} \end{pmatrix} \quad (43)$$

これは次のように変形できる．

$$\begin{aligned} \begin{pmatrix} u_1 & C \gamma^k u_2 \end{pmatrix} &\approx \begin{pmatrix} x^k & x^{k+1} \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & \gamma \end{pmatrix}^{-1} \\ &= \frac{1}{\gamma - 1} \begin{pmatrix} x^k & x^{k+1} \end{pmatrix} \begin{pmatrix} \gamma & -1 \\ -1 & 1 \end{pmatrix} \end{aligned} \quad (44)$$

両辺の第 1 列を取り出すと次のようになる．

$$u_1 \approx \frac{\gamma x^k - x^{k+1}}{\gamma - 1} \quad (45)$$

次に γ を定める．上式から次の関係が成り立つ．

$$u_1 \approx \frac{\gamma x^k - x^{k+1}}{\gamma - 1} \approx \frac{\gamma x^{k+1} - x^{k+2}}{\gamma - 1} \quad (46)$$

これから次の関係を得る．

$$\gamma \Delta x^k - \Delta x^{k+1} \approx 0 \quad (47)$$

ただし，次のように置いた．

$$\Delta x^k = x^{k+1} - x^k, \quad \Delta x^{k+1} = x^{k+2} - x^{k+1} \quad (48)$$

式 (47) の両辺と Δx^k との内積をとると次のようになる．

$$\gamma \|\Delta x^k\|^2 - (\Delta x^k, \Delta x^{k+1}) \approx 0 \quad (49)$$

ゆえに γ が次のように推定できる．

$$\gamma = \frac{(\Delta x^k, \Delta x^{k+1})}{\|\Delta x^k\|^2} \quad (50)$$

以上より次のような加速ができる．まず初期値 x^0 から x^1, x^2 を計算し，これらから式 (50) によって γ を推定する．そして，その γ を用いて式 (45) によって x^1, x^2 から u_1 を推定したものを改めて x^2 と置く．この x^2 から x^3, x^4 を計算し，以下同様にして，偶数番目に値を式 (45) によって更新する．これらの手順をまとめたものが前述のアルゴリズムである．

4.3 2 次の加速

より長い系列をとれば，さらに加速することができる．

- 単位ベクトル x^0 を任意に与え， $x^2 = x^1 = x^0$ と置く．
- 次の計算を $k = 1, 2, \dots$ に対して計算する．

(1) 次の x^3 を計算する．

$$x^3 = N[Ax^2] \quad (51)$$

(2) $x^3 \approx x^2$ であれば x^3 を返して終了する．

(3) k が 3 の倍数でなければ次のように x^0, x^1, x^2 を更新してループを終了し，次の k の計算に進む．

$$x^0 \leftarrow x^1, \quad x^1 \leftarrow x^2, \quad x^2 \leftarrow x^3 \quad (52)$$

(4) k が 3 の倍数なら次の計算を行う．

$$\begin{aligned} \Delta x^0 &= x^1 - x^0, \quad \Delta x^1 = x^2 - x^1, \quad \Delta x^2 = x^3 - x^2, \\ A &= \|\Delta x^0\|^2, \quad B = \|\Delta x^1\|^2, \quad C = (\Delta x^0, \Delta x^1), \\ D &= (\Delta x^0, \Delta x^2), \quad E = (\Delta x^1, \Delta x^2), \quad \Delta = C^2 - AB, \\ P &= \frac{BD - CE}{\Delta}, \quad Q = \frac{CD - AE}{\Delta} \end{aligned} \quad (53)$$

(注4): 著者が調べた限りでは文献には見当たらなかった．

(5) 次のように x^0, x^1, x^2 を更新してループを終了し、次の k に計算に進む。

$$x^0 \leftarrow x^1, \quad x^1 \leftarrow x^2, \quad x^2 \leftarrow N \left[\frac{Px^1 - Qx^2 + x^3}{P - Q + 1} \right] \quad (54)$$

【導出】式 (35) からより精密な次の関係を得る。

$$x^k \approx u_1 + C_1 \gamma_1^k u_2 + C_2 \gamma_2^k u_3 \quad (55)$$

ただし、次のように置いた。

$$C_1 \equiv \frac{c_2}{c_1}, \quad C_2 \equiv \frac{c_3}{c_1}, \quad \gamma_1 \equiv \frac{\lambda_2}{\lambda_1}, \quad \gamma_2 \equiv \frac{\lambda_3}{\lambda_1} \quad (56)$$

$k+1, k+2$ に対して次の関係を得る。

$$\begin{aligned} x^{k+1} &\approx u_1 + C_1 \gamma_1^{k+1} u_2 + C_2 \gamma_2^{k+1} u_3, \\ x^{k+2} &\approx u_1 + C_1 \gamma_1^{k+2} u_2 + C_2 \gamma_2^{k+2} u_3 \end{aligned} \quad (57)$$

式 (55), (57) をまとめると次のように書ける。

$$\begin{aligned} \begin{pmatrix} u_1 & C_1 \gamma_1^k u_2 & C_2 \gamma_2^k u_3 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 \\ 1 & \gamma_1 & \gamma_1^2 \\ 1 & \gamma_2 & \gamma_2^2 \end{pmatrix} \\ \approx \begin{pmatrix} x^k & x^{k+1} & x^{k+2} \end{pmatrix} \end{aligned} \quad (58)$$

これは次のように変形できる。

$$\begin{aligned} \begin{pmatrix} u_1 & C_1 \gamma_1^k u_2 & C_2 \gamma_2^k u_3 \end{pmatrix} \\ \approx \begin{pmatrix} x^k & x^{k+1} & x^{k+2} \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 \\ 1 & \gamma_1 & \gamma_1^2 \\ 1 & \gamma_2 & \gamma_2^2 \end{pmatrix}^{-1} \\ = \frac{\begin{pmatrix} x^k & x^{k+1} & x^{k+2} \end{pmatrix}}{(\gamma_1 - 1)(\gamma_2 - 1)(\gamma_2 - \gamma_1)} \begin{pmatrix} -(\gamma_2 - \gamma_1)(\gamma_2 + \gamma_1) & & \\ -(\gamma_2 - \gamma_1)(\gamma_2 + \gamma_1) & & \\ \gamma_2 - \gamma_1 & & \\ -\gamma_2(\gamma_2 - 1) & \gamma_1(\gamma_1 - 1) & \\ (\gamma_2 - 1)(\gamma_2 + 1) & -(\gamma_1 - 1)(\gamma_1 + 1) & \\ -(\gamma_2 - 1) & \gamma_1 - 1 & \end{pmatrix} \end{aligned} \quad (59)$$

両辺の第 1 列を取り出すと次のようになる。

$$u_1 \approx \frac{\gamma_1 \gamma_2 x^k - (\gamma_1 + \gamma_2) x^{k+1} + x^{k+2}}{\gamma_1 \gamma_2 - (\gamma_1 + \gamma_2) + 1} \quad (60)$$

次に、この式中に現れる $\gamma_1 \gamma_2, \gamma_1 + \gamma_2$ を定める。式 (60) から次の関係が成り立つ。

$$\begin{aligned} u_1 &\approx \frac{\gamma_1 \gamma_2 x^k - (\gamma_1 + \gamma_2) x^{k+1} + x^{k+2}}{\gamma_1 \gamma_2 - (\gamma_1 + \gamma_2) + 1} \\ &\approx \frac{\gamma_1 \gamma_2 x^{k+1} - (\gamma_1 + \gamma_2) x^{k+2} + x^{k+3}}{\gamma_1 \gamma_2 - (\gamma_1 + \gamma_2) + 1} \end{aligned} \quad (61)$$

これから次の関係を得る。

$$\gamma_1 \gamma_2 \Delta x^k - (\gamma_1 + \gamma_2) \Delta x^{k+1} + \Delta x^{k+2} = 0 \quad (62)$$

ただし、次のように置いた。

$$\Delta x^k = x^{k+1} - x^k, \quad \Delta x^{k+1} = x^{k+2} - x^{k+1},$$

表 1 実験に用いた 256×256 対称行列 A, B の大きい 4 個の固有値と実行時間 (秒)。

	A	B
固有値	1.000000	1.000000
	0.255404	0.992142
	0.002364	0.989402
	0.000001	0.988023
実行時間	1.221357	1.452753

$$\Delta x^{k+1} = x^{k+3} - x^{k+2} \quad (63)$$

式 (62) の両辺と Δx^k および Δx^{k+1} との内積をとると

$$\begin{aligned} \gamma_1 \gamma_2 \|\Delta x^k\|^2 - (\gamma_1 + \gamma_2) (\Delta x^k, \Delta x^{k+1}) \\ + (\Delta x^k, \Delta x^{k+2}) = 0 \\ \gamma_1 \gamma_2 (\Delta x^k, \Delta x^{k+1}) - (\gamma_1 + \gamma_2) \|\Delta x^{k+1}\|^2 \\ + (\Delta x^{k+1}, \Delta x^{k+2}) = 0 \end{aligned} \quad (64)$$

とになる。これから $\gamma_1 \gamma_2, \gamma_1 + \gamma_2$ が次のように求まる。

$$\begin{aligned} \Delta &\equiv (\Delta x^k, \Delta x^{k+1})^2 - \|\Delta x^k\|^2 \|\Delta x^{k+1}\|^2 \\ \gamma_1 \gamma_2 &= \left(\|\Delta x^{k+1}\|^2 (\Delta x^k, \Delta x^{k+2}) \right. \\ &\quad \left. - (\Delta x^k, \Delta x^{k+1}) (\Delta x^{k+1}, \Delta x^{k+2}) \right) / \Delta \\ \gamma_1 + \gamma_2 &= \left((\Delta x^k, \Delta x^{k+1}) (\Delta x^k, \Delta x^{k+2}) \right. \\ &\quad \left. - \|\Delta x^k\|^2 (\Delta x^{k+1}, \Delta x^{k+2}) \right) / \Delta \end{aligned} \quad (65)$$

以上より次のような加速ができる。まず初期値 x^0 から x^1, x^2, x^3 を計算し、これらから式 (65) によって $\gamma_1 \gamma_2, \gamma_1 + \gamma_2$ を推定する (γ_1, γ_2 を計算する必要はない)。そして式 (60) によって x^1, x^2, x_3 から u_1 を推定したものを改めて x^3 と置く。次にその x^3 から x^4, x^5, x^6 を計算し、以下同様にして、3 の倍数番目に値を式 (60) によって更新する。これらの手順をまとめたものが前述のアルゴリズムである。

5. 実験

画像からの 3 次元復元問題 [6], [8], [9] に現れた 256×256 対称行列 A, B を用いて実行時間を比較する。 A はランク 4, B はランク 12 であり、表 1 に大きい 4 個の固有値を示す。このように A は最大固有値が突出して大きく、 B はほぼ一樣な固有値を持つ。表 1 にはこれらに対してハウスホルダー法のライブラリプログラム [11] を用いたときの実行時間も示す。ただし、CPU には Pentium 4 3.GHz, 主メモリ 2GB, OS には Linux を用いた。

次に、時間変化のある場合や固有値計算を反復において、直前で計算した固有ベクトルが解のよい近似になっている場合を想定し、真値 \bar{u} (単位ベクトル) から角度 $\theta = 0.1, 0.2, \dots, 1.0$ 度をなす単位ベクトルをそれぞれの角度ごとに 1,000 回ランダムに生成して固有値計算を行い、全体の 10,000 回の試行の実行時間の平均、最小値、最大値、および標準偏差を計測した。これを表 2 に示す。ただし、得られた固有ベクトルはすべて単位ベクトルに正規化し、真の解 \bar{u} との差が $\|\bar{u} \pm u\| < 10^{-6}$ となるまで反復した (固有ベクトルは符号が不定であり、反対向きでもよいため、差が小さいほうの向きで判定する)。

表 2 ランダムおよび 1° 近傍の初期値に対する実行時間 (秒) の平均 (上段) と標準偏差 (下段) .

手法	行列	最小値	最大値	平均値	標準偏差
最急降下法	A	.002251	.009247	.002339	.000123
	B	.002233	.049602	.006651	.007412
共役勾配法 (原型)	A	.002289	.005620	.002379	.000095
	B	.002290	.006877	.002927	.000706
共役勾配法 (フレッチャー・リーブス)	A	.002280	.009342	.002377	.000117
	B	.002288	.014383	.002940	.000797
共役勾配法 (ポラック・リビエール)	A	.002294	.005407	.002383	.000101
	B	.002299	.115828	.004185	.003064
べき乗法 (原型)	A	.001486	.004660	.001567	.000086
	B	.001486	.209656	.014752	.022081
べき乗法 (1 次の加速)	A	.001544	.008790	.001633	.000106
	B	.001553	.128133	.003603	.004463
べき乗法 (2 次の加速)	A	.001536	.007139	.001620	.000128
	B	.001554	.117343	.003495	.003209

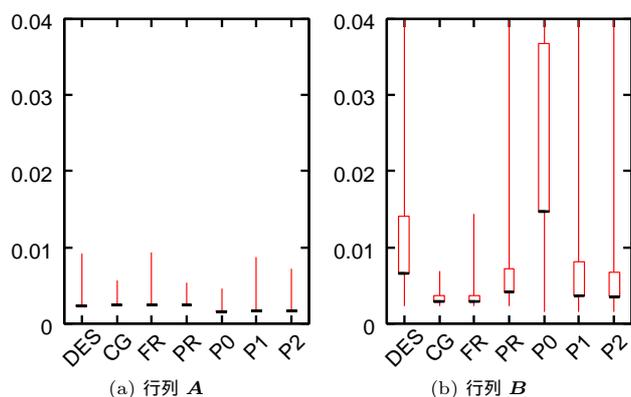


図 1 各手法の収束時間 (秒) の最小値と最大値を線分で結び、平均値を底辺として標準偏差を高さとする箱を示した図 . DES: 最急降下法, CG: 共役勾配法 (原型), FR: 共役勾配法 (フレッチャー・リーブス), PR: 共役勾配法 (ポラック・リビエール), P0: べき乗法 (原型), P1: べき乗法 (1 次の加速), P2: べき乗法 (2 次の加速) .

図 1 は表 2 の数値を図示したものである . 詳細に検討すると、どの方法でも実行時間はごく小さい値に集中し、平均値が非常に小さいにもかかわらず、分布が非常に大きい値まで指数的に裾野を引いている . このため、平均値から変動は値が大きいほうに著しく片寄っている . これを考慮して、図 1 では、最小値と最大値を端点とする線分を描き、平均値を底辺として標準偏差を高さとする箱を図示している .

6. 考 察

まず表 1 と 2 を比較して分かることは、最大固有値とその固有ベクトルのみが必要な場合は一般に反復解法を用いたほうが圧倒的に計算時間が少ないということである . これは各反復ステップで行列とベクトルの積 (計算量 $O(n^2)$) を一度しか計算しないためである (n は行列のサイズ) . すべての固有値、固有ベクトルを求めるには通常は少なくとも計算量 $O(n^3)$ (= n 次元連立 1 次方程式を解く計算量) が必要である .

次にわかることは、べき乗法の収束が固有値の分布に大きく依存することである . 表 2 と図 1 からわかるように、突出した固有値を持つ行列 A に対してはべき乗法は非常に有効であり、わずかではあるが、最急降下法や共役勾配法よりも効率的

である . このような場合はべき乗法は加速してもほとんど効果がない .

一方、ほぼ一様な固有値を持つ行列 B に対してはべき乗法の原型は非常に効率が悪い . これは式 (35) から容易に推察できる . しかし、これを加速すると共役勾配法とほぼ同等の速度となり、しかも初期値にほとんど依存しない . 加速は 1 次までで十分である . 2 次まで加速しても平均的にはほぼ同じ時間であり、かえって時間がかかる場合も生じる .

最急降下法と共役勾配法 (原型) は突出した固有値を持つ行列 A に対してはほとんど差がないが、ほぼ一様な固有値を持つ行列 B に対しては共役勾配法 (原型) のほうが効率的になっている . ただし、初期値によっては収束に時間がかかることがある . 共役勾配法に広く用いられているフレッチャー・リーブスの式やポラック・リビエールの式を用いても平均的には原型と大差ないが、行列 B ではポラック・リビエールの式を用いると収束が遅い場合も生じる .

以上をまとめると、固有値の分布にも初期値にも依存せずに良好な収束性能を示すのはべき乗法の 1 次の加速であると結論される . 画像からの 3 次元復元の自己校正法ではこれが用いられている [9] .

7. ま と め

本論文ではコンピュータビジョンを含む多くの問題に現れる、時間変化がある大規模行列の最大固有値とその固有ベクトルを計算するための反復解法をまとめ、その実行時間を比較した .

その結果、べき乗法の 1 次の加速が固有値の分布にも初期値にも依存せずに良好な収束性能を示すことがわかった . 共役勾配法はそれに次いで有効であるが、原型とフレッチャー・リーブスの式のどちらがよいとも言えない . それに比べるとポラック・リビエールの式はやや劣るようである .

文 献

- [1] 甘利俊一, 金谷健一, 「線形代数」, 講談社, 1987 .
- [2] R. Fletcher, *Practical Methods of Optimization*, John Wiley, New York, NY, U.S.A., 1980, 1987, 2000 .
- [3] 金谷健一, 「空間データの数理 - 3 次元コンピューティングに向けて」, 朝倉書店, 1995 .
- [4] 金谷健一, 「これなら分かる応用数学教室 - 最小二乗法からウェブレットまで」, 共立出版, 2003 .
- [5] 金谷健一, 「これなら分かる最適化数学 - 基礎原理から計算手法まで」, 共立出版, 2005 .
- [6] 金谷 健一, 森 昭延, 菅谷 保之, 自己校正法の最新レシピ, 情報処理学会研究報告, 2006-CVIM-153-31 (2006-3), 199-206 .
- [7] 森正武, 「数値解析」, 共立出版, 1973 .
- [8] 森 昭延, ハノ・アッカーマン, 金谷健一, 高速射影復元: 徹底的な効率化を目指して, 情報処理学会研究報告, 2007-CVIM-157-15 (2007-1), 109-116 .
- [9] 森 昭延, 金谷健一, 菅谷保之, 最新の自己校正法の性能評価, 情報処理学会研究報告, 2006-CVIM-154-37 (2006-5), 347-354 .
- [10] E. Polak, *Computational Methods in Optimization*, Academic Press, New York, NY, U.S.A., 1971 .
- [11] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*, Second Edition, Cambridge University Press, Cambridge, U.K., 1992 .
- [12] A. Ralston, *A First Course in Numerical Analysis*, McGraw-Hill, New York, NY, U.S.A., 1965 .
- [13] X. Yang, T. K. Sarkar, and E. Arvas, A survey of conjugate gradient algorithms for solution of extreme eigen-problems of a symmetric matrix, *IEEE Trans. Acoust. Speech, Signal Process.*, **37**-10 (1989-10), 1550-1556 .