数理解析研究所講究録
1505巻 2006年 1-13



# 固有値/特異値計算ライブラリの性能評価のための数式処理のアルゴリズム

JST・立教大学 木村欣司 (Kinji Kimura)
Japan Science and Technology Agency, Faculty of Science, Rikkyo University

## 1 はじめに

新しい固有値あるいは特異値計算, 固有ベクトルや特異ベクトルのための計算アルゴリズムを開発することはそれ自体が容易なことではないが, 既存のライブラリとの比較評価をおこなうこともまた容易なことではない. 著者は, 京都大学中村研究室の新しい特異値分解アルゴリズム設計実装ならびにその評価の研究に関わっているためその必要があった. その研究は現在も進行中であるが, これまで利用してきた比較評価のためのさまざま手法について報告する.

既存のライブラリとの比較評価をおこなうことは, あらかじめ固有値/特異値, 固有ベクトルや特異ベクトルのわかっている行列を構成することと同値である. それにより, 個々のライブラリが真値とどのくらい違った値を計算するかを測定することができる. あらかじめ固有値/特異値, 固有ベクトルや特異ベクトルのわかっている行列を構成する手法は, 以下の6つに分類される.

1. 精度保証付き数値計算を利用する
2. 与えられた固有値や特異値をもつ行列を構成する
3. 三角関数などで真値を書くことのできる行列を利用する
4. あらかじめ固有多項式のわかっている行列を利用する
5. Kronecker 積を利用する
6. 数式処理を利用して十分に高精度な値を計算する

これらを順に紹介する.

## 2 精度保証付き数値計算を利用する方法

### 2.1 概略

1. 計算精度が double のプログラム P を用意する.
2. P の double の演算をすべて多倍長数の演算に置きなおす. そのプログラムを Q とする.
3. 整数の行列 R を用意する.
4. Q に R を入力し, 多倍長数で固有値と固有ベクトルの近似値を求める.

5. 近似固有値/特異値に対して, 多倍長数で精度保証付き数値計算をおこない近似固有値/特異値の精度を調べる.
6. 近似値の精度が 10 進で 16 桁をはるかに超えていれば真値とみなす.
7. P に R を入力し近似計算をおこなう.
8. 7. でえられた double の近似値を多倍長数に変換し, 4. で構成された真値との差を多倍長数で計算する.

### 2.2 精度保証付き数値計算のための公式

現在, 固有値や特異値のための精度保証付き数値計算のための公式は多数存在する. 固有ベクトル/特異値ベクトルについても公式が存在する. 詳しくは, [10][14][15] を参照されたい.

### 2.3 実装についての注意

P の double の演算をすべて多倍長数の演算に置きなおすことは容易に可能である. そのような目的で

1. MPFUN/ARPREC
2. Omni OpenMP Compiler Project(GMP)

は, 大変優れた道具である.

## 3 与えられた固有値や特異値をもつ行列を構成する方法

### 3.1 密行列の場合

1. 適当な対角行列 $D$ を用意する.
2. $n \times n$ の乱数行列を用意し, それを $X$ とする.
3. あらかじめ固有値のわかっている対称行列を構成する場合, $X$ に対して多倍長数でグラムシュミットの直交化法をおこない直交行列 $Y$ を用意する. 非対称行列の場合, $X$ に対して多倍長数で逆行列 $Z$ をつくる.
4. 対称行列の場合は, $B = YDY^T$ を多倍長数で計算することにより得られる. 非対称行列の場合は, $C = XDZ$ を多倍長数で計算することにより得られる.
5. 多倍長数の行列 $B, C$ から double の行列 $B', C'$ に変換する.

特異値の場合も, これらの方法を変形することで対応できる. たとえ **4.** の後に精度保証をしたとしても, **5.** で変換誤差が発生するためこの方法は数学的に厳密ではない. しかし, これこれの条件をみたす行列で実験をおこないたいという場合には便利な方法といえよう.

### 3.2 対称行列の場合の改良

内積は誤算を含みやすいため, 対称行列の場合には次のような方法も知られている.

1. 適当な対角行列 $D$ を用意する
2. 多倍長数を利用して適当な Jacobi 回転行列を生成し, それを $D$ の左右からかける
3. 非対角成分が大きくなるまで, 2. を繰り返す.

### 3.3 与えられた固有値や特異値をもつ3重対角行列/2重対角行列を構成する方法

多倍長数を利用した Lanczos 法を用いて対角行列を3重対角行列に変換することにより実現できる.[16] 非対称三重対角行列の問題を構成する場合には, 双 Lanczos 法を用いればよい.[8] 与えられた特異値をもつ2重対角行列をつくるには, Golub-Kahan-Lanczos 法を用いることができる.[4] **適当な Jacobi 回転行列を生成しそれを $D$ の左右からかけるという操作を繰り返すと次第に非0成分が外側に広がっていくため, 直交行列を左右からかける操作では与えられた固有値や特異値をもつ3重対角行列を構成できないことを注意する.**

## 4 三角関数などで真値を書くことのできる行列を利用する方法

そのような行列で有名なものとして,

1. Helmholtz 方程式の5点差分からつくられる行列
2. Poisson 方程式の5点差分からつくられる行列

を挙げる. これらの行列は, 固有値と固有ベクトルの厳密解を三角関数で書き下せる.

### 4.1 3重対角行列 (I)

#### 4.1.1 定理

$n$ 次の3重対角行列 [17]

$$A = \begin{pmatrix} b & c & & & \\ a & b & c & & \\ & a & b & \ddots & \\ & & \ddots & \ddots & c \\ & & & a & b \end{pmatrix}$$

の固有値 $\lambda_j$, 固有ベクトル $x^{(j)}$ は次式で与えられる.

$$\lambda_j = b + 2\sqrt{ac}\cos\frac{j\pi}{n+1} \quad (1 \le j \le n),$$

$$x^{(j)} = \left[\sin\frac{j\pi}{n+1}, \sqrt{\frac{a}{c}}\sin\frac{2j\pi}{n+1}, \cdots, \left(\sqrt{\frac{a}{c}}\right)^{n-1}\sin\frac{nj\pi}{n+1}\right]^{\mathrm{T}}.$$

Helmholtz 方程式や Poisson 方程式の 5 点差分行列の固有値問題は, 適当な行列を左右からかける ことによりこの形式の 3 重対角行列に変換できるため厳密解が三角関数で書き下せる.[5] 3 重対角行列のテスト行列という意味でもこの定理は重要である.

#### 4.1.2 密行列への変換

$$A = \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & -1 & 2 & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 2 \end{pmatrix},\ A^{-1} = \frac{1}{n+1}\begin{pmatrix} n & n-1 & \cdots & 2 & 1 \\ n-1 & 2(n-1) & \cdots & 4 & 2 \\ n-2 & 2(n-2) & \cdots & 6 & 3 \\ n-3 & 2(n-3) & \cdots & 8 & 4 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ 2 & 4 & \cdots & 2(n-1) & n-1 \\ 1 & 2 & \cdots & n-1 & n \end{pmatrix}.$$

この関係を利用する.

## 4.2 3 重対角行列 (II)

#### 4.2.1 定理

$n$ 次の 3 重対角行列 [16]

$$\tilde{A}_n = \begin{pmatrix} 1 & -1 & & & \\ -1 & 2 & -1 & & \\ & -1 & 2 & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 2 \end{pmatrix}, \quad \tilde{A}_n^{-1} = \begin{pmatrix} n & n-1 & \cdots & 2 & 1 \\ n-1 & n-1 & \cdots & 2 & 1 \\ \vdots & \vdots & & \vdots & \vdots \\ 2 & 2 & \cdots & 2 & 1 \\ 1 & 1 & \cdots & 1 & 1 \end{pmatrix}$$

$\tilde{A}_n$ の固有多項式を $\Gamma_n(\lambda)$ としたとき $\Gamma_n(\lambda) = (2-\lambda)\Gamma_{n-1}(\lambda) - \Gamma_{n-2}(\lambda)$ であり,

$$\lambda_j = 4\sin^2\left(\frac{2j-1}{2(2n+1)}\pi\right), \quad j = 1, 2, ..., n$$

となり, 固有ベクトルは

$$x^{(j)} = \left[\sin\left(\frac{n(2j-1)}{2n+1}\pi\right), \sin\left(\frac{(n-1)(2j-1)}{2n+1}\pi\right), \cdots, \sin\left(\frac{2j-1}{2n+1}\pi\right)\right]^{\mathrm{T}}$$

となる.

### 4.3 2重対角行列 (I)

$A$ のコレスキー分解は,

$$A = LL^{\mathrm{T}} = \begin{pmatrix} \sqrt{\frac{2}{1}} & & & & \\ -\sqrt{\frac{1}{2}} & \sqrt{\frac{3}{2}} & & & \\ & -\sqrt{\frac{2}{3}} & \sqrt{\frac{4}{3}} & & \\ & & \ddots & \ddots & \\ & & & -\sqrt{\frac{n-1}{n}} & \sqrt{\frac{n+1}{n}} \end{pmatrix} \begin{pmatrix} \sqrt{\frac{2}{1}} & -\sqrt{\frac{1}{2}} & & & \\ & \sqrt{\frac{3}{2}} & -\sqrt{\frac{2}{3}} & & \\ & & \sqrt{\frac{4}{3}} & -\sqrt{\frac{3}{4}} & \\ & & & \ddots & \ddots \\ & & & & \sqrt{\frac{n+1}{n}} \end{pmatrix}$$

となり, 行列 $L^{\mathrm{T}}$ の特異値と右特異ベクトルは,

$$\sigma_j = 2\sin\left(\frac{j\pi}{2(n+1)}\right), \quad x^{(j)} = \left[\sin\frac{j\pi}{n+1}, \sin\frac{2j\pi}{n+1}, \cdots, \sin\frac{nj\pi}{n+1}\right]^{\mathrm{T}}, j = 1, 2, \ldots, n$$

となる.

#### 4.3.1 密行列への変換

$L^{\mathrm{T}}$ の逆行列は

$$L^{-\mathrm{T}} = \begin{pmatrix} \frac{1}{\sqrt{1\cdot 2}} & \frac{1}{\sqrt{2\cdot 3}} & \frac{1}{\sqrt{3\cdot 4}} & \cdots & \frac{1}{\sqrt{n(n+1)}} \\ & \frac{2}{\sqrt{2\cdot 3}} & \frac{2}{\sqrt{3\cdot 4}} & \cdots & \frac{2}{\sqrt{n(n+1)}} \\ & & \frac{3}{\sqrt{3\cdot 4}} & \cdots & \vdots \\ & & & \cdots & \frac{n-1}{\sqrt{n(n+1)}} \\ & & & & \frac{n}{\sqrt{n(n+1)}} \end{pmatrix}$$

となる. $L^{-\mathrm{T}}$ の特異値と左特異ベクトルは

$$\sigma_j = \frac{1}{2\sin\left(\frac{j\pi}{2(n+1)}\right)}, \quad x^{(j)} = \left[\sin\frac{j\pi}{n+1}, \sin\frac{2j\pi}{n+1}, \cdots, \sin\frac{nj\pi}{n+1}\right]^{\mathrm{T}}, j = 1, 2, \ldots, n$$

となる.

### 4.4 2重対角行列 (II)

$\tilde{A}_n$ のコレスキー分解は,

$$\tilde{A}_n = \tilde{L}\tilde{L}^{\mathrm{T}} = \begin{pmatrix} 1 & & & & \\ -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \end{pmatrix} \begin{pmatrix} 1 & -1 & & & \\ & 1 & -1 & & \\ & & 1 & \ddots & \\ & & & \ddots & -1 \\ & & & & 1 \end{pmatrix},$$

より $\tilde{L}^T$ の特異値は,

$$\sigma_j = 2\sin\left(\frac{2j-1}{2(2n+1)}\pi\right), \quad j = 1, 2, ..., n$$

となり, 右特異ベクトルは

$$x^{(j)} = \left[\sin\left(\frac{n(2j-1)}{2n+1}\pi\right), \sin\left(\frac{(n-1)(2j-1)}{2n+1}\pi\right), \cdots, \sin\left(\frac{2j-1}{2n+1}\pi\right)\right]^{\mathrm{T}}$$

となる.

#### 4.4.1 密行列への変換

$\tilde{L}^{\mathrm{T}}$ の逆行列は

$$\tilde{L}^{-\mathrm{T}} = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ & 1 & 1 & \cdots & 1 \\ & & 1 & \cdots & \vdots \\ & & & \ddots & 1 \\ & & & & 1 \end{pmatrix}$$

となる. $\tilde{L}^{-\mathrm{T}}$ の特異値は,

$$\sigma_j = \frac{1}{2\sin\left(\frac{2j-1}{2(2n+1)}\pi\right)}, \quad j = 1, 2, ..., n$$

となり, 左特異ベクトルは

$$x^{(j)} = \left[\sin\left(\frac{n(2j-1)}{2n+1}\pi\right), \sin\left(\frac{(n-1)(2j-1)}{2n+1}\pi\right), \cdots, \sin\left(\frac{2j-1}{2n+1}\pi\right)\right]^{\mathrm{T}}$$

となる.

### 4.5 有名な参考文献の紹介

Robert T Gregory, David L Karney, Collection of Matrices for Testing Computational Algorithms Interscience, (1969) は, このように三角関数などで厳密解を書くことのできる行列が多数記載されている優れた本である. なお, ここでは代表的な例のみを紹介したがこのような議論はさらに一般化できかなり大きなクラスの行列の厳密解が三角関数などで記述できる. それらの詳細は, 今後論文などで紹介するつもりである.

## 5 あらかじめ固有多項式のわかっている行列を利用する方法

1 変数代数方程式

$$f(x) = x^n + a_{n-1}x^{n-1} + \cdots + a_0$$

の根を固有値とするコンパニオン行列 $C_n$ は

$$C_n = \begin{pmatrix} 0 & \cdots & \cdots & 0 & -a_0 \\ 1 & \ddots & \ddots & 0 & -a_1 \\ 0 & \ddots & \ddots & \vdots & \vdots \\ \vdots & & \ddots & 0 & -a_{n-2} \\ 0 & \cdots & & 1 & -a_{n-1} \end{pmatrix}$$

である.

よって, $\tilde{A}_n C_n (\tilde{A}_n)^{-1}$ などにより与えられた固有多項式の根を固有値とする整数の非対称行列を構成することができる. あらかじめ根のよくわかっている多項式を利用するとよいであろう. もちろん, 整数を要素とする対角行列 $D$ に対して $\tilde{A}_n D (\tilde{A}_n)^{-1}$ などにより整数の非対称行列を構成することもできる.

# 6 Kronecker 積を利用する方法

## 6.1 基本性質

$m$ 次正方行列 $A$ と $n$ 次正方行列 $B$ の Kronecker 積 $A \otimes B$ は $mn$ 次正方行列であるが, $A$ の固有値を $\lambda_1, ..., \lambda_m$, $B$ の固有値を $\mu_1, ..., \mu_n$ とすると $A \otimes B$ の固有値は $\lambda_i \mu_j (i = 1, ..., m : j = 1, ..., n)$ である. さらに, $m$ 次正方行列 $A$ の固有ベクトルを $v_1, ..., v_m$ として $n$ 次正方行列 $A$ の任意の固有ベクトルを $w_1, ..., w_n$ とすれば Kronecker 積 $A \otimes B$ の固有ベクトルは $v_i \otimes w_j (i = 1, ..., m : j = 1, ..., n)$ である. すなわち

$$(A \otimes B) \times (v_i \otimes w_j) = (A \times v_i) \otimes (B \times w_j) = (\lambda_i v_i) \otimes (\mu_j w_j) = (\lambda_i \mu_j)(v_i \otimes w_j)$$

ここで, $\times$ は通常使う行列とベクトルの積である. $(A \times v_i) \otimes (B \times w_j) = (\lambda_i v_i) \otimes (\mu_j w_j)$ を示すことは容易でないので, [9] を参照されたい. また, $A \otimes B$ の逆行列は, $A^{-1} \otimes B^{-1}$ である.

## 6.2 終結式の応用

Kronecker 積 $\otimes$ により構成される行列の固有多項式を計算する,

$$U = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}, V = \begin{pmatrix} j & k \\ l & m \end{pmatrix}, W = U \otimes V = \begin{pmatrix} aj & ak & bj & bk & cj & ck \\ al & am & bl & bm & cl & cm \\ dj & dk & ej & ek & fj & fk \\ dl & dm & el & em & fl & fm \\ gj & gk & hj & hk & ij & ik \\ gl & gm & hl & hm & il & im \end{pmatrix}.$$

$W$ の固有多項式を計算する手順は以下のように与えられる.

1. $U$ と $V$ の固有多項式を個別に計算する,

$$\begin{aligned} f(x) = \det(U - xI) &= -x^3 + (a+e+i)x^2 + (-ea - ia + db + gc - ie + hf)x \\ &\quad + (ie - hf)a + (gf - id)b + (hd - ge)c, \\ g(x) = \det(V - xI) &= x^2 + (-j - m)x + mj - lk. \end{aligned}$$

2. 次の多項式を計算する,

$$h(x,y) = \text{numerator}\left(g\left(\frac{x}{y}\right)\right).$$

3. $W$ の多項式は, 次の終結式を計算すればよい,

$$\det(W - xI) = \text{res}_y(f(y), h(x,y)) \quad \text{or} \quad -\text{res}_y(f(y), h(x,y)).$$

この例の場合には, $\text{res}_y(f(y), h(x,y))$ を選ぶ. $\text{res}_y$ とは $y$ について終結式を計算することを意味する.

数式処理を利用することにより固有多項式を高速に計算できるが, 個々の行列の固有値や固有ベクトルを高精度に計算することにより大きな行列の固有値や固有ベクトルを構成するほうが効率がよい. ここでは, 数学的興味として紹介する.

## 7 数式処理を利用して十分に高精度な値を計算する方法

数値計算ライブラリの性能を評価することを目的としているため, 整数を要素とする行列についてのみ議論をする. その理由は, 入力した時点で数値誤差を含まないようにするためである.

### 7.1 整数を要素とする非特異な密行列の連立一次方程式 $Ax = v$ を解く方法

Hensel 構成により計算する方法を紹介する.[2] この方法は, 数値計算の連立一次方程式の反復改良法 (残差修正法) に対応する.

$$x = x_0 + px_1 + p^2x_2 + \cdots$$

$p$ 進数で展開すると,

$$A(x_0 + px_1 + p^2x_2 + \cdots) = v \mod p^\alpha.$$

$A$ を $\bmod p$ で一度 LU 分解すれば

$$\begin{aligned} Ax_0 &= v \mod p \\ Ax_1 &= \frac{v - Ax_0}{p} \mod p \\ Ax_2 &= \frac{(v - Ax_0)/p - Ax_1}{p} \mod p \\ &\vdots \end{aligned}$$

は高速に解くことができる. さらに, $p$ 進整数から有理数への変換法が存在する.

### 7.2 整数を要素とする密行列の行列式の評価公式

$n \times n$ の行列 $A$ に対して $\det(A)$ の評価は以下のようにおこなう.[11]

$$A = \begin{pmatrix} a_{1,1} & \cdots & a_{1,n} \\ \vdots & & \vdots \\ a_{n,1} & \cdots & a_{n,n} \end{pmatrix},$$

より

$$u_1 = (a_{1,1}, \ldots, a_{1,n}), \ldots, u_n = (a_{n,1}, \ldots, a_{n,n}), \quad v_1 = (a_{1,1}, \ldots, a_{n,1}), \ldots, v_n = (a_{1,n}, \ldots, a_{n,n}),$$

を定義すると, Hadamard の公式より

$$\det(A) \text{ の絶対値} \quad \leq \quad \min(\|u_1\|_2 \|u_2\|_2 \cdots \|u_{n-1}\|_2 \|u_n\|_2, \|v_1\|_2 \|v_2\|_2 \cdots \|v_{n-1}\|_2 \|v_n\|_2) \equiv H.$$

### 7.3 多変数多項式を要素とする行列の行列式の評価公式

多変数多項式の 1 ノルムは, 係数の絶対値の総和と定義する.

$$\begin{pmatrix} ||a_{1,1}||_1 & \ldots & ||a_{1,n}||_1 \\ \vdots & & \vdots \\ ||a_{n,1}||_1 & \ldots & ||a_{n,n}||_1 \end{pmatrix}$$

として Hadamard の公式を適用する. そのときの値を $H_1$ とすると

$$\text{多変数多項式を要素とする行列の行列式の係数の絶対値最大} \leq H_1$$

が成立する. 詳しくは, [3] を参照されたい.

#### 7.3.1 系:固有多項式の係数の絶対値最大の見積もり公式

$$\begin{pmatrix} |a_{1,1}| + |1| & \ldots & |a_{1,n}| \\ \vdots & \ddots & \vdots \\ |a_{n,1}| & \ldots & |a_{n,n}| + |1| \end{pmatrix}$$

として Hadamard の公式を適用する. そのときの値を $H_2$ とする $H_2$ は固有多項式の係数の絶対値の上界を与える.

### 7.4 整数を要素とする行列の固有多項式の計算

#### 7.4.1 方法 I

最小多項式の次数が行列サイズに満たない場合にも利用できる算法を紹介する. まず, 固有多項式の係数の上界 $H_2$ がわかっていることを注意する. 有限体 $\mathbb{Z}/p\mathbb{Z}$ 上で"ガウスの消去法の拡張"をおこなうと, いかなる整数を要素とする行列も上 Hessenberg 行列に変形できる. ここで, "ガウスの消去法の拡張"による上 Hessenberg 行列への変形は数値計算ではもはや過去の算法であるため老婆心ながらここで述べる.

$n$ を行列サイズとし pivot の成分を $a(k,k)$ とすると

$$\alpha = \frac{a(i,k)}{a(k,k)}$$
$$a(i,j) \leftarrow a(i,j) - \alpha a(k,j) \quad j = k+1, ..., n$$

として消去法をおこなうのがガウスの消去法であるが, この行消去を行ったあと

$$a(m,k) \leftarrow a(m,k) + \alpha a(m,i) \quad m = 1, ..., n$$

として列に対して行を消去したときに用いた量を足しこみ固有値を不変にする算法である. さらに, 上 Hessenberg 行列から有限体 $\mathbb{Z}/p\mathbb{Z}$ 上の固有多項式が計算できる. $p_i$ を変化させて何度もガウスの消去法の拡張を繰り返す. 中国剰余定理をもちいて固有多項式の係数を合成する. 中国剰余定理の解の正規化条件が $\left[-\frac{\prod p_i - 1}{2}, \frac{\prod p_i - 1}{2}\right]$ であるから $\frac{\prod p_i - 1}{2}$ が $H_2$ 以上になると固有多項式を計算できたことになる.[7]

"ガウスの消去法の拡張"を利用して固有多項式を計算する代わりに Danilevsky 法を用いて有限体 $\mathbb{Z}/p\mathbb{Z}$ 上の固有多項式を計算することもできる.[8]

#### 7.4.2 方法 II

最小多項式の次数が行列サイズに一致する場合のみに利用できる算法を紹介する. 正確には, 一致しない場合にもこの算法を利用することができるがそのときに方法 I と比較するとこの方法のほうが非効率であるためこの方法を利用すべきでない. 数値計算の GMRES 法に対応する. $v$ を乱数ベクトルとして $A^k v$ を有限体 $\mathbb{Z}/p\mathbb{Z}$ ではなく整数で計算し

$$\left( \begin{array}{c|c|c|c} v & Av & \cdots & A^{n-1}v \end{array} \right) \left( \begin{array}{c} c_0 \\ \vdots \\ c_{n-1} \end{array} \right) = -A^n v$$

を連立一次方程式の解法で解く. $c_{n-1}, \cdots, c_0$ は固有多項式の係数となる. 解があらかじめ整数であるとわかっているとき Hensel 構成の実装は工夫できる. 整数から有理数に変換する必要がないのはあきらかであるが, 数値計算の残差反復法のアナロジーであることからも明らかなように右辺の残差ベクトルが 0 になった時点で計算を終了してよい. $x_j$ を $[-\frac{p-1}{2}, \frac{p-1}{2}]$ に規格化することにより残差が 0 になる $j$ が存在することが保証されるのである.

## 8 1変数代数方程式の実根を求める方法

非対称行列の固有値までも扱うには複素根までもとめる方法を考えなければならないが, 複素根までもとめるアルゴリズムは数多く存在しどのような算法が高速であるかすべてを実装し確認することは困難である. 対称行列の固有値や特異値の固有多項式は全根が実根であるが, 非対称行列では実根と複素根が混在するためその扱いも本質的に異なると思われる. よって, ここでは議論を明確にするという目的と京都大学中村研究室の新しい特異値分解のアルゴリズムの性能評価を主な目的としているという立場から実根のみを求める方法についてのみ紹介することとする.

現在知られている代表的な計算法を 3 つあげる.

1. Strum の方法
2. 平野の方法
3. Uspensky の方法

整数を係数とする 1 変数の多項式の実根を求める場合, 2. または 3. のほうが 1. よりも優れていることが計算量解析の結果によりすでにわかっている.[1] パラメータを係数とする 1 変数の多項式の場合, 1. を使わないかぎり実根を持つ条件を得ることはできない. また, 1. を利用することで指定した数の実根を持つための条件なども導出できる. Pdearson らの方法 [13] も知られているが, 明らかに計算量が多いためここでは比較しない.

### 8.1 Strum の方法

Strum の方法は有名であるため, ここでは解説しない. [11] を参照されたい. しかし, 係数にたくさんのパラメータの持った場合にはその効率化は容易でないことを注意する. その話題は, あまりに深い内容であるためここでは省略する.

### 8.2 平野の方法

#### 8.2.1 数学的事実

「与えられた関数 $f(x)$ の導関数 $f'(x)$ の隣り合う実数解の間では $f(x)$ は単調な関数であり, その間には高々1 つの解しか持たない」この事実をもちいて real root finding をおこなうのが平野の方法である.[12]

#### 8.2.2 定理

係数が実数の多項式 $f(x)$ に対して, 次のような多項式列

$$f_1(x), f_2(x), \ldots, f_k(x)$$

をつくる.

1. $f_1(x) = f(x)/\mathrm{gcd}(f(x), f'(x))$ とおく.
2. $f_j(x)$ まで定義されていて $f_j(x)$ の次数が 1 より大きいときは
   $f_{j+1}(x) = f_j'(x)/\mathrm{gcd}(f_j'(x), f_j''(x))$ とおく.
3. $f_{j+1}(x)$ の次数が 1 より大きいときは前の操作をくり返す.

この多項式列は次の条件を満たす.

- すべての $f_j(x)$ は重複因子を持たない. $f_1(x)$ と $f(x)$ は同じ解を持つ.
- $f_j(x)$ と $f_{j+1}(x)$ は共通解を持たない.
- $j = 1, 2, \ldots, k-1$ に対して, $\alpha, \beta$ を隣り合う $f_{j+1}(x)$ の解とする. $f_j(x)$ は区間 $(\alpha,\beta)$ に高々1 つの解しか持たない. もしこの区間に $f_j(x)$ が解を持てば $f_j(\alpha)$ と $f_j(\beta)$ は符号が異なる.

この定理により, 実根を探索できるわけである. 詳しくは, [12] を参照されたい.

### 8.3 Uspensky の方法

(Descarte の符号律) 実係数をもつ多項式

$$f(x) = a_0x^n + a_1x^{n-1} + \cdots + a_n$$

の係数列 $a_0, a_1, \cdots, a_n$ の符号の変りの数 (0 になるものは飛ばして数える) を $W$ とすれば, $f(x)$ の正根 (0 を入れない) の数は $W$ またはそれよりも偶数だけすくない. よって, $W$ が 0 ならば根がないことは確定する. $W$ が 1 ならば, 1 根あることが確定する. $W$ が 2 以上の場合, 領域を分割してそれぞれの領域を調べればよい. 具体的には, 以下の操作をおこなう.

$x \to 1/(x+1)$ と変数変換すると, 新しい $x$ の $(0,\infty)$ の実根の数を調べるということは, 元の世界では $(0,1)$ を調べることに相当する. また, $x \to x+1$ と変数変換すると, 新しい $x$ の $(0,\infty)$ の実根の数を調べるということは, 元の世界では $(1,\infty)$ を調べることに相当する. しかし, この二つの操作だけでは $x=1$ の根を逃してしまうため $x \to x+1$ の後定数項が消えていないかを check することも必要である. 以上の操作を調べている領域の実根の数が 0 または 1 になるまで繰り返す.[1]

しかし, このような操作だけでは $x = 10^{10000000}$ のような根には対応できない. そのために, 次の定理を用意する.

#### 8.3.1 定理

実係数の方程式

$$x^n + a_1x^{n-1} + \cdots + a_n = 0$$

において, 負の係数を $a_\alpha, a_\beta, a_\gamma, \cdots$ とすれば

$$G_4 = 2\max\left[(|a_\alpha|)^{1/\alpha}, (|a_\beta|)^{1/\beta}, (|a_\gamma|)^{1/\gamma}, \cdots\right]$$

が正根の限界である.[6] よって, $x \to 1/x$ により最小の正根を下から抑えられる. この定理によって原点移動がおこなえるため $x = 10^{10000000}$ のような根にも対応できる.

## 9 タイミングデータ

以下の計算は, Intel Xeon 2.8GHz, Memory 2GByte によっておこなった. $1000 \times 1000$ の整数を要素とする対称行列を用意した. 各行列の要素には 1 から 10 の整数の乱数を格納した. はじめに、この行列の固有多項式を計算することを試みる. 方法 II を用いた場合, 固有多項式を計算するのに 24 分 45 秒を必要とした. 方法 I を用いた場合, 1 時間以上計算したが終わらなかった. さらに, すべての実根を求めるために平野の方法では 15 分 36 秒を必要とした. Uspensky の方法では 18 分 45 秒を必要とした. 十分現実的な時間で数式処理により数値計算ライブラリの評価のためのテスト行列を作成できる.

## 10 まとめ

数値計算ライブラリの性能を評価するためにあらかじめ真値のわかっている行列を如何に構成するかという問題に対して十分に多くの手法を紹介できたと考える. 新しいライブラリを設計実装した際に, その評価のためここでの議論が役立つことを期待する.

## 参考文献

[1] G.E.Collins and Alkiviadis G.Akritas, Polynomial Real Root Isolation Using Decarte's Rule of Signs, Proceedings of the 1976 ACM Symposium on Symbolic and Algebraic Computation

[2] K. O. Geddes, Stephen R. Czapor, George Labahn, Keith O. Geddes, S. R. Czapor, G. Labahn, Algorithms for Computer Algebra, Kluwer Academic Pub., United States, (1992).

[3] A.J. Goldstein, R.L. Graham, A Hadamard-type bound on the coefficient of a determinant of polynomials, SIAM Review 16, 394-395, (1974).

[4] G. Golub, W. Kahan, Calculating the singular values and pseude-inverse of a matrix, SIAM J. Numeri. Anal., Vol. 2, pp.205-224, (1965).

[5] E. Ishiwata, Y. Muroya, K. Isogai, Adaptive improved block SOR method with orderings, J.JIAM, vol.16, No.3, pp.443-466 (1999).

[6] J.R.Johnson, Algorithms for Polynomial Real Root Isolation, Quantifier Elimination and Cylindrical Algebraic Decomposition, SpringerWienNewYork, Austria, (1998).

[7] S. Lo, M. Monagan, A. Wittkopf, A Modular Algorithm for Computing the Characteristic Polynomial of an Integer Matrix in Maple, http://www.cecm.sfu.ca/CAG/papers/CPpaper.pdf.

[8] 有本卓, 数値解析 (I), コロナ社, 東京, 1997.

[9] 伊理正夫, 一般線形代数, 岩波書店, 東京, 2003.

[10] 大石進一, 非線形解析入門, コロナ社, 東京, 2000.

[11] 高木貞治, 代数学講義 (改訂新版), 共立出版, 東京, 1965.

[12] 齋藤友克, 竹島卓, 平野照比古, グレブナー基底の計算 実践篇, 東京大学出版会, 東京, 2003.

[13] 野呂正行, 横山和弘, グレブナー基底の計算 基礎篇, 東京大学出版会, 東京, 2003.

[14] 宮島信也, 荻田武史, 大石進一, 実対称行列の各固有値に対する精度保証付き数値計算法, 日本応用数理学会論文誌, **15(3)**, pp. 253-268, (2005).

[15] 宮島信也, 荻田武史, 大石進一, 実対称行列の各固有対の精度保証付き計算法, 日本応用数理学会, 平成 18 年研究部会連合発表会.

[16] 森正武, 数値解析第 2 版, 共立出版株式会社, 東京, 2002.

[17] 山本哲朗, 数値解析入門 [増訂版], サイエンス社, 東京, 2003.