

高精度な行列分解アルゴリズムとその応用

荻田 武史

東京女子大学 現代教養学部 数理科学科

アルゴリズムによる計算科学の融合と発展

筑波大学 計算科学研究センター

2009年4月

概要

目的 LU分解，QR分解，Cholesky分解，特異値分解などの行列分解のための**高精度なアルゴリズム**を提案する．

A : $n \times n$ 実行列．

提案方式は A が非常に**悪条件**な場合も取り扱える．

悪条件: A の**条件数**が非常に大きい．

⇒ 例えば，IEEE 754倍精度の範囲で条件数が 10^{100} を超えるような問題も取り扱える．

⇒ 直接的な応用: 連立一次方程式，行列式．その他，様々な分野に応用可能．

条件数

正方行列 A に対して, A の条件数を下記のように定義する :

$$\kappa(A) := \|A\| \cdot \|A^{-1}\|.$$

⇒ $\kappa(A)$ は問題の難しさの指標となる

⇒ 例えば, 連立一次方程式 $Ax = b$ を考える ($x^* := A^{-1}b$).

右辺に摂動: $Ay = b + \Delta b$ ($y^* := A^{-1}(b + \Delta b)$).

このとき, 解の変化量 $\Delta x = y^* - x^*$ は下記をみたとす :

$$\frac{\|\Delta x\|}{\|x^*\|} \leq \kappa(A) \frac{\|\Delta b\|}{\|b\|}$$

IEEE 754規格の倍精度演算

- ⇒ 1回の演算の相対精度: $u \approx 1.11 \times 10^{-16}$
- ⇒ 条件数が 10^{16} より大きい ($\kappa(A) > u^{-1}$) 場合, 通常の倍精度演算によって得られた計算結果は**1桁も合っていない**場合がある.
- ⇒ **基本精度の限界** (基本精度: 単精度や倍精度)
- ⇒ このような問題を**“悪条件問題”**と呼ぶことにする.
- ⇒ 悪条件問題に対し, **“結果精度の高い”**行列分解アルゴリズムを提案する.

Notation

$$A = (a_{ij}), C = (c_{ij}) \in \mathbb{R}^{n \times n}$$

- $|A|$: A のすべての要素に絶対値を付けた行列, $|A| = (|a_{ij}|)$.
- $A \leq C$: すべての (i, j) に対して, $a_{ij} \leq c_{ij}$

ベクトルについても同様の notation を利用する .

LU分解（部分軸交換付）

$$A \in \mathbb{R}^{n \times n}, PA \approx LU :$$

$$\begin{cases} P \in \mathbb{R}^{n \times n} & : \text{置換行列} \\ L \in \mathbb{R}^{n \times n} & : \text{単位下三角行列} \\ U \in \mathbb{R}^{n \times n} & : \text{上三角行列} \end{cases}$$

例) 連立一次方程式 $Ax = b$ を解く .

$$\rightarrow PAx = Pb \quad \rightarrow LU\tilde{x} = b$$

$$\rightarrow Ly = Pb \text{ を解く .} \quad \rightarrow U\tilde{x} = y \text{ を解く .}$$

LU分解の精度をどのように推定するか？

残差 $\|PA - LU\|$ で推定？

例えば，LU分解の残差 $\|PA - LU\|$ あるいは相対残差は？

$$\frac{\|PA - LU\|}{\|A\|} < \varepsilon$$

⇒

これでは**不十分** (Matlab demo.)

例) 標準的なガウスの消去法を用いた場合：

$$\|PA - LU\| \leq \mathcal{O}(\mathbf{u}) \| |L| |U| \| \sim \mathcal{O}(\mathbf{u}) \|A\|$$

メインアイデア

正確なLU分解 $PA = \hat{L}\hat{U}$ を考えると，通常， A の悪条件性は **上三角行列 \hat{U}** に反映される．

$$\kappa(A) \sim \kappa(\hat{U})$$

もし，(1) を満たすような \hat{U}^{-1} の良い近似 X_U が得られれば， $\kappa(A)$ と比べて， $\kappa(AX_U)$ は相対的に小さくなる．例えば， $\kappa(AX_U) \ll \mathbf{u}^{-1}$ ．

$$\|\hat{U}X_U - I\| < 1 \quad (1)$$

⇒

A に対する **前処理** となる (Crout 型でも同様)

A を X_U によって前処理した後は

$$\|X_L A X_U - I\| < 1 \quad (2)$$

あるいは

$$\|A X_U X_L - I\| < 1 \quad (3)$$

を満たすような \hat{L} の近似逆行列 X_L を計算するのは難しくない。

⇒ 重要かつ難しい点：

どのように悪条件な A の前処理 X_U を計算するか？

仮定 $\kappa(\hat{U}) \gg \mathbf{u}^{-1}$, $\hat{X} := \hat{U}^{-1}$.

$X_1 = fl(\hat{X})$ とする (基本精度における最良近似) .

$$\implies X_1 = \hat{U}^{-1} + \Delta , \quad |\Delta_{ij}| \sim \mathbf{u} |\hat{U}_{ij}^{-1}| \text{ (丸め誤差)}$$

このような場合でも, $\|\hat{U}X_1 - I\| < 1$ は必ずしも成立するわけではない:

$$\begin{aligned} \|\hat{U}X_1 - I\| &= \|\hat{U}(\hat{U}^{-1} + \Delta) - I\| = \|\hat{U}\Delta\| \\ &\approx \|\hat{U}\| \|\Delta\| \sim \mathbf{u} \cdot \kappa(\hat{U}) > 1 \end{aligned}$$

\implies \hat{U}^{-1} の近似には, なんらかの**高精度な形式**が必要

前処理としての近似逆行列

丸め誤差によって, X_1 は \hat{U} の逆行列としては機能しない. しかし, A の前処理としては機能する (\hat{U} の条件数とは無関係に).

$$\implies \kappa(\hat{U}X_1) = \mathcal{O}(\mathbf{u}) \cdot \kappa(\hat{U}), \quad \mathbf{u} \approx 10^{-16}$$

\implies 条件数を小さくできる!

\implies これを利用した積型の反復アルゴリズムを開発:

$$PAX_U = \underline{\underline{\hat{L}\hat{U}X_1X_2\dots X_k}} \approx \hat{L} \approx L$$

(反復停止条件: $\|PAX_U - L\| < \varepsilon_{\text{tol}}$)

“高精度”な逆LU分解

$A \in \mathbb{R}^{n \times n}$, $PA \approx LU$:

$$\left\{ \begin{array}{ll} P \in \mathbb{R}^{n \times n} & : \text{置換行列} \\ L \in \mathbb{R}^{n \times n} & : \text{単位下三角行列} \\ U \in \mathbb{R}^{n \times n} & : \text{上三角行列} \end{array} \right.$$

許容誤差 $\varepsilon_{\text{tol}} < 1$ に対して

$$\|L^{-1}PAU^{-1} - I\| \leq \varepsilon_{\text{tol}} \quad \text{あるいは} \quad \|PAU^{-1} - L\| \leq \varepsilon_{\text{tol}}$$

($R \approx A^{-1}$: $\|RA - I\| \leq \varepsilon_{\text{tol}}$ と同様のコンセプト)

提案方式の方針

悪条件問題を取り扱うためには、なんらかの**高精度演算**が必要．

⇒ すべての計算を多倍長精度で計算するのは**効率が良い**くない．

⇒ **内積計算や行列乗算に特定すれば**，比較的高速なアルゴリズムが利用可能．

⇒ **できる限り**，高速な**通常の浮動小数点演算**を利用したい．

高精度な内積計算

$x, y \in \mathbb{F}^n$ に対し, **内積**

$$x^T y = \sum_{i=1}^n x_i y_i$$

の計算は, 数値線形代数の基礎なので, たくさんの高精度なアルゴリズムが提案されている.

高精度な総和計算 / 内積計算の文献 (一部)

- 1965 Møller (BIT)
- 1970 Nickel (ZAMM) (Kahan-Babuška's algorithm)
- 1971 Malcolm (Comm. ACM)
- 1972 Pichat (Numer. Math.)
- 1973 Kiełbasziński (Math. Stos.)
- 1974 Neumaier (ZAMM) (improved Kahan-Babuška's algorithm)
- 1977 Bohlander (IEEE Trans. Comput.)
- 1982 Leuprecht, Oberaigner (Computing)
- 1986 Kulisch, Miranker (SIAM Review, originally 1970's in Karlsruhe)
- 1991 Priest (Proc. IEEE Symposium)
- 1999 Anderson (SIAM J. Sci. Comput.)
- 2002 Li et al. (ACM Trans. Math. Softw., XBLAS)
- 2003 Demmel, Hida (SIAM J. Sci. Comput)
- ⋮

高精度な行列乗算

仮定 任意に高精度な内積の計算結果が得られる。

\mathbb{F} : (基本精度における) 浮動小数点数の集合

u : 浮動小数点演算の相対精度 (IEEE 754 倍精度: $u \approx 10^{-16}$)

できる限り, 通常の浮動小数点数 / 浮動小数点演算を用いたい。

$\implies A_{1:p} := \sum_{i=1}^p A_i, B_{1:q} := \sum_{i=1}^q B_i$ のように行列を表現する。ただし, $A_i, B_i \in \mathbb{F}^{n \times n}$ かつ

$$|A_i| \leq u |A_{i-1}|, \quad |B_i| \leq u |B_{i-1}|$$

のような関係が成り立っているものとする。

仮定 行列乗算の汎用的な関数として下記のようなものが利用可能とする：

$$C_{1:L} = [A_{1:p}B_{1:q}]_K^L, \quad C_{1:L} := \sum_{i=1}^L C_i, \quad C_i \in \mathbb{F}^{n \times n}$$

は，下記をみたすとする ($c_1, c_2 : \mathcal{O}(1)$ の定数)：

$$|C_{1:L} - A_{1:p}B_{1:q}| \leq c_1 \mathbf{u}^L |A_{1:p}B_{1:q}| + c_2 \mathbf{u}^K |A_{1:p}| |B_{1:q}|.$$

これは， $A_{1:p}B_{1:q}$ を K 倍長精度で計算し，その結果を L 倍長精度に丸めたものに対応する ($K \geq L$ でないと意味がない)。

関連研究

Rumpは、悪条件な行列の逆行列を高精度に計算するアルゴリズムを開発した [unpublished, 1980's], [JJIAM, to appear] .

```
function R = AccInv(A,m)           % right inverse version
    n = dim(A);  I = eye(n);
    R = A \ I;  % pure fl-pt (Solve AR = I for R)
    for k=2:m
        C = AccMM(A,R,1);          % accurate dot product
        T = C \ I;  % pure fl-pt (Solve CT = I for T)
        R = AccMM(R,T,k+1);        % accurate dot product
    end
```

アルゴリズム：高精度な逆LU分解

[高精度演算が必要な箇所のみ記載．それ以外は，基本精度]

- 0: $X_{1:0} = I$, $k = 1$ とする .
- 1: $B_k \leftarrow [A \cdot X_{1:k-1}]_k^1$. [高精度に計算 / 基本精度に丸める]
- 2: B_k のLU分解 ($P_k B_k \approx L_k U_k$) .
- 3: $U_k^{-1} \approx T_k$ を計算 .
- 4: $X_{1:k} \leftarrow [X_{1:k-1} \cdot T_k]_k^k$. [高精度に計算 / 高精度に格納]
- 5: もし , $\|U_k\| \|T_k\| \leq \varepsilon_{\text{tol}} \cdot u^{-1}$ ならば
 $L := L_k$, $X_U := X_{1:k}$ および $P := P_k$ を返す .
- 6: $k \leftarrow k + 1$ として , 1に戻る .

アルゴリズム：高精度な逆QR分解

[高精度演算が必要な箇所のみ記載．それ以外は，基本精度]

- 0: $X_{1:0} = I$, $k = 1$ とする .
- 1: $B_k \leftarrow [A \cdot X_{1:k-1}]_k^1$. [高精度に計算 / 基本精度に丸める]
- 2: B_k のQR分解 ($B_k \approx Q_k R_k$) .
- 3: $R_k^{-1} \approx T_k$ を計算 .
- 4: $X_{1:k} \leftarrow [X_{1:k-1} \cdot T_k]_k^k$. [高精度に計算 / 高精度に格納]
- 5: もし , $\|R_k\| \|T_k\| \leq \varepsilon_{\text{tol}} \cdot u^{-1}$ ならば
 $Q := Q_k$, $X_R := X_{1:k}$ を返す .
- 6: $k \leftarrow k + 1$ として , 1に戻る .

難しい点

- アルゴリズムは簡潔だが，解析が難しい
- 実際，**厳密な解析は不可能**（期待値や確率的な話になる）
- ある程度の自然な仮定の下で，解析可能

Proposition 1. $A \in \mathbb{F}^{n \times n}$ とする . $X_{1:k}$ を逆 LU 分解 (逆 QR 分解) アルゴリズムによって得られた $n \times n$ 上三角行列とする . このとき , ある仮定の下で , $k \geq 1$ に対して下記が成立する :

$$\kappa(A X_{1:k}) = 1 + \mathcal{O}(\mathbf{u}^k) \cdot \kappa(A) \quad (4)$$

Remark 1. $A = \sum_{i=1}^m A_i$, $A_i \in \mathbb{F}^{n \times n}$ のような入力でも取り扱えるようにアルゴリズムを拡張するのは難しくない (もし , 高精度な内積計算 / 行列乗算のアルゴリズムが利用可能なら) .

$\|\hat{R}X_{1:k} - I\| = \|AX_{1:k} - \hat{Q}^T\| < 1$ を達成するためには，下記のような k で十分．

$$k = \left\lceil \frac{\log[\varepsilon_{\text{tol}} \cdot \kappa(A)^{-1}]}{\log \mathbf{u}} \right\rceil .$$

しかしながら，一般的に $\kappa(A)$ は事前にはわからないため， k をあらかじめ決めることはできない．

そこで，下記のような停止条件を利用：

$$\|R_k\| \|T_k\| < \varepsilon_{\text{tol}} \cdot \mathbf{u}^{-1} \quad (5)$$

$$(5) \text{ が成立} \implies \|AX_{1:k} - \hat{Q}^T\| < \varepsilon_{\text{tol}}$$

提案アルゴリズムの特徴

1. **問題の難しさに応じて**，必要なだけ反復的に計算精度を増やすことができる（**アダプティブ性**）。
2. 高精度演算は，行列乗算（つまり**内積計算**）のみ。
3. それ以外の処理は，**BLAS**，**LAPACK**などを利用可能で，すべて通常の浮動小数点演算で実行（**高速性**）。
4. 高精度な内積計算 / 行列乗算が高速化されると，提案アルゴリズムも高速化される。

2については，これも**Level-3 BLAS**を利用した高速な方式を開発中（早大・尾崎氏との共同研究）。

数値実験

逆LU分解 / 逆QR分解のふるまいについて数値実験する .

- 倍精度を基本精度とする ($u = 2^{-53} \approx 1.1 \cdot 10^{-16}$)
- テスト行列 : Rumpの行列 (INTLABの `randmat(n, cnd)`)
- $n = 100$, $cnd = 10^{100}$ ($A \in \mathbb{F}^{100 \times 100}$, $\kappa(A) \approx 1.75 \cdot 10^{107}$)

Table 1: Rumpの行列 ($n = 100$ and $\kappa(A) \approx 1.75 \cdot 10^{107}$) に対する逆LU分解の結果

k	$\kappa(U_k)$	$\kappa(T_k)$	$\kappa(AX_{1:k})$	$\mathbf{u}^k \kappa(A)$
1	$3.50 \cdot 10^{18}$	$3.50 \cdot 10^{18}$	$2.37 \cdot 10^{93}$	$1.94 \cdot 10^{91}$
2	$5.28 \cdot 10^{18}$	$5.28 \cdot 10^{18}$	$2.18 \cdot 10^{78}$	$2.16 \cdot 10^{75}$
3	$4.01 \cdot 10^{18}$	$4.01 \cdot 10^{18}$	$1.79 \cdot 10^{64}$	$2.39 \cdot 10^{59}$
4	$4.85 \cdot 10^{18}$	$4.85 \cdot 10^{18}$	$3.45 \cdot 10^{48}$	$2.66 \cdot 10^{43}$
5	$1.99 \cdot 10^{18}$	$1.99 \cdot 10^{18}$	$6.77 \cdot 10^{33}$	$2.95 \cdot 10^{27}$
6	$1.16 \cdot 10^{18}$	$1.16 \cdot 10^{18}$	$1.30 \cdot 10^{18}$	$3.27 \cdot 10^{11}$
7	$2.73 \cdot 10^{17}$	$2.73 \cdot 10^{17}$	$3.96 \cdot 10^2$	< 1
8	$1.91 \cdot 10^2$	$1.91 \cdot 10^2$	$8.68 \cdot 10^1$	< 1

Table 2: Rumpの行列 ($n = 100$ and $\kappa(A) \approx 1.75 \cdot 10^{107}$) に対する逆QR分解の結果

k	$\kappa(R_k)$	$\kappa(T_k)$	$\kappa(AX_{1:k})$	$\mathbf{u}^k \kappa(A)$
1	$3.27 \cdot 10^{19}$	$3.27 \cdot 10^{19}$	$2.00 \cdot 10^{93}$	$1.94 \cdot 10^{91}$
2	$1.86 \cdot 10^{19}$	$1.86 \cdot 10^{19}$	$1.06 \cdot 10^{77}$	$2.16 \cdot 10^{75}$
3	$7.97 \cdot 10^{17}$	$7.97 \cdot 10^{17}$	$1.61 \cdot 10^{62}$	$2.39 \cdot 10^{59}$
4	$2.20 \cdot 10^{17}$	$2.20 \cdot 10^{17}$	$4.23 \cdot 10^{46}$	$2.66 \cdot 10^{43}$
5	$2.31 \cdot 10^{17}$	$2.31 \cdot 10^{17}$	$2.00 \cdot 10^{32}$	$2.95 \cdot 10^{27}$
6	$4.04 \cdot 10^{17}$	$4.04 \cdot 10^{17}$	$6.69 \cdot 10^{16}$	$3.27 \cdot 10^{11}$
7	$2.18 \cdot 10^{18}$	$2.18 \cdot 10^{18}$	$1.39 \cdot 10^3$	< 1
8	$1.39 \cdot 10^3$	$1.39 \cdot 10^3$	$1.00 \cdot 10^0$	< 1

応用(1): 連立一次方程式

連立一次方程式の近似解の計算に提案アルゴリズムを適用:

1. A^T の逆LU分解 (Crout型なら A でよい)

$$PA^T X_U \approx L \quad \Leftrightarrow \quad X_U^T AP \approx L^T$$

2. $\tilde{y} = [X_U^T \cdot b]_m^1$ を計算
3. 三角方程式 $L^T z = \tilde{y}$ を解く (得られた近似解を \tilde{z} とする)
4. $\tilde{x} = P\tilde{z}$ を計算

数値実験(1): (スケール化)Hilbert行列 H_n

H_n は要素が整数の行列 ($n \leq 21$ のとき, 倍精度で正確に表現可能)

- $n = 15$ ($\kappa(H_{15}) \approx 6.12 \times 10^{20}$)
- 右辺: $b = H_{15}e \in \mathbb{F}^{15}$, $e := (1, \dots, 1)^T$
- $H_{15}x = b$ の真の解: $H_{15}^{-1}b = e = (1, \dots, 1)^T$

(Matlab demo)

数値実験 (2): Rump の行列

様々な条件数に対し，提案アルゴリズムを用いて得られた連立一次方程式の近似解の相対精度（ノルム評価）をみる．

- Rump の行列 `randmat` ($n = 500$, 条件数は 10^{20} から 10^{100})
- $b = (1, \dots, 1)^T$
- 提案アルゴリズムの許容誤差 : $\varepsilon_{\text{tol}} = 10^{-6}$
- 多倍長精度との速度比較のため，GMP ベースのガウスの消去法も実行（**真の解は既知とする**トライアンドエラー方式）

```
function [p,rel_err] = test_gmp_lin(A,b,xt,tol)
% xt: given exact solution of Ax = b
% tol: tolerance for relative error

d = 53; norm_xt = norm(double(xt));
while 1
    xv = gmp_lin(A,b,d); % solve Ax=b using GMP
    % normwise relative error
    rel_err = norm(double(xt-xv))/norm_xt;
    if rel_err < tol, break, end
    d = 2*d; % d = 53, 106, 212, ...
end
```

Table 3: Rumpの行列の結果 ($n = 500$, $\varepsilon_{\text{tol}} = 10^{-6}$)

$\kappa(A)$	提案アルゴリズム			GMP-based GEPP		
	ε_1	t_1	k	ε_2	t_2	$d/53$
1.31^{20}	$2.84 \cdot 10^{-13}$	1.28	2	$5.83 \cdot 10^{-14}$	23.35	2
1.73^{33}	$2.44 \cdot 10^{-15}$	3.95	3	$1.11 \cdot 10^{-16}$	65.76	4
2.70^{43}	$4.17 \cdot 10^{-15}$	8.87	4	$1.11 \cdot 10^{-16}$	69.23	4
1.13^{49}	$1.14 \cdot 10^{-13}$	9.23	4	$1.11 \cdot 10^{-16}$	71.47	4
4.97^{59}	$3.05 \cdot 10^{-15}$	16.27	5	$1.04 \cdot 10^{-7}$	70.98	4
7.05^{73}	$5.01 \cdot 10^{-15}$	25.94	6	$1.11 \cdot 10^{-16}$	154.70	8
3.85^{81}	$7.54 \cdot 10^{-11}$	26.39	6	$1.11 \cdot 10^{-16}$	157.12	8
2.74^{92}	$5.38 \cdot 10^{-13}$	39.88	7	$1.11 \cdot 10^{-16}$	158.12	8
8.02^{103}	$4.17 \cdot 10^{-15}$	56.44	8	$1.11 \cdot 10^{-16}$	156.94	8

応用例(2): 行列式の計算

良い方法 (Higham in ASNA) として知られているのは, LU分解して U_{ii} の積を計算:

$$\det(A) \approx \det(P^T LU) = \det(P) \det(U).$$

(もちろん, これは近似計算なので, 結果の保証がない)

行列式は, その符号判定の情報が重要な例が多い.

行列式の精度保証

X_L, X_U : L, U の近似逆行列 .

$$B := X_L P A X_U \Rightarrow \det(B) = \det(P) \det(A) \det(X_U) \Rightarrow$$

$$\det(A) = \det(P) \det(B) / \det(X_U) \quad (6)$$

$\Rightarrow B$: 単位行列 (対角行列) に近い, $\det(B) \approx 1$

\Rightarrow **Gershgorin の定理** により, B の固有値の範囲がわかる

$\Rightarrow \det(B)$ の包含が可能

\Rightarrow **(6)** により, $\det(A)$ の包含も可能

しかしながら， A が**悪条件**な場合， L も U も良いLU分解因子とならない．

⇒ その場合， $B = X_L P A X_U$ は単位行列（対角行列）とかけ離れてしまう．

⇒ Gershgorinの定理が意味をなさない．

この問題も，提案アルゴリズムで解決できる！

(Matlab demo)

今後の課題

今回のコンセプトを用いると

- 高精度な逆 Cholesky 分解
- 高精度な逆 LDL^T 分解
- 高精度な特異値分解

も構成できる（多少の修正は必要．解析も別）．

（固有値分解は少し難しく，現在，検討中）

ありがとうございました。