丸めモードを使用しない行列積の精度保証

2010SE040 服部亮輝
指導教員：杉浦洋

1　はじめに

行列積は線形代数の最も基本的な演算の一つであり，その精度保証は数値計算における重要な課題である． 行列積の精度保証には，丸めモードの切替えを用いた有用なアルゴリズムがあるが，計算環境によっては，丸めモードの切替えが使用できず，丸めモードは丸め込み (四捨五入) に固定されている．ゆえに，丸め込みモードに固定した環境で働く精度保証アルゴリズムが強く望まれる． 本研究では，行列 A と B の積 AB を数値計算により包含する手法について考える． ここでの「包含」とは，行列積 AB が含まれる区間の下端と上端となる行列や，その中心と半径となる行列を求め，結果を数値計算を用いて厳密に包み込む「区間」を得ることである． 尾崎らの計算手法 [2] について学び，理解を深め，Mathematica 用のアルゴリズムの構成を目指す．

2　精度保証付き計算について

2.1　丸めモードについて

一般に実数は機械実数 $\mathbb{F}$ に丸められ，メモリに格納される．IEEE754 では以下の 4 つの丸めモードがある．$\mathrm{c} \in \mathbb{R}$ とする．

・丸め上げ（*round upward*）c 以上の一番小さい浮動小数点数に丸める．$\triangle : \mathbb{R} \longrightarrow \mathbb{F}$ と表す．

・丸め下げ（*round downward*） c 以下の一番大きい浮動小数点数に丸める．$\triangledown : \mathbb{R} \longrightarrow \mathbb{F}$ と表す．

・丸め込み（*round to nearest*）c に一番近い $\mathbb{F}$ の浮動少数点数に丸める．$\square : \mathbb{R} \longrightarrow \mathbb{F}$ と表す．

・丸め捨て（*round toward 0*）絶対値 $|\ \mathrm{c}\ |$ が小さい $\mathbb{F}$ の浮動小数点数で最も近いものに丸める．

3　丸め込みの誤差

$\mathbb{R}$ は実数の集合，$\mathbb{F}$ は浮動小数の集合とする．以下では，IEEE754 規格の $\mathbb{F}$ を浮動小数点に含み， $\mathbb{F}$ 上の浮動小数演算が IEEE754 算術規格をみたすことを仮定する．この仮定は，ほとんどのコンピュータ，ワークステーション，メインフレームに用いられている．計算表現のためにある計算式について $\mathrm{fl}(\cdot)$ は，丸め込みモードの浮動小数演算で計算された値を表す．この場合 IEEE754 では．

$$\mathrm{fl}(x\ op\ y) = (x\ op\ y)(1+\delta), \qquad |\delta| \le u \tag{1}$$

が成立する．ここで $op \in \{+, -, \times, /\}$ であり，$u = 2^{-53}$ は丸め誤差単位である．

3.1　丸めモード切り替えによる精度保証とその問題点

大石，Rump[1] は IEEE754 の丸めモード切り替えを用いた効果的な行列積の精度保証のアルゴリズムを提案した．しかし，FORTRAN77 や Java，Mathematica などの一部の言語では、精度保証に有用な IEEE754 が定める有向丸めの機能がないことが知られている． 現在多くの計算機で，丸め込みモードがデフォルトのモードになっており． このモードの演算のみで機能するアルゴリズムが必要である．

4　簡易法

簡易のため，$A, B \in \mathbb{F}^{n\times n}$ は正方行列とする．$n\mathrm{u} < 1$ のとき，次の不等式が知られている．

$$|AB - \mathrm{fl}(AB)| \ \le \gamma_n |A||B|,\ \gamma_n = \frac{n\mathrm{u}}{1-n\mathrm{u}}$$

また，この右辺は

$$|A|\cdot|B| \le \frac{\mathrm{fl}(|A|\cdot|B|)}{(1-\mathrm{u})^n} \tag{2}$$

で評価できる．ゆえに

$$|AB - \mathrm{fl}(AB)| \le \frac{\gamma_n}{(1-\mathrm{u})^n}\mathrm{fl}(|A|\cdot|B|)$$

で評価できる．さらに $1 \le n \le \mathrm{u}^{-1} - 3$ のとき，

$$\frac{|a|}{1-n\mathrm{u}} \le \mathrm{fl}\Big(\frac{|a|}{1-(n+1)\mathrm{u})}\Big), \tag{3}$$

$$(1+\mathrm{u})^n \le \frac{1}{(1-\mathrm{u})^n} \le \frac{1}{1-n\mathrm{u}}, \tag{4}$$

$$|A||B| \le (1+\mathrm{u})^{n-1}\mathrm{fl}(|A||B|) \tag{5}$$

を用いて，

$$|AB - \mathrm{fl}(AB)| \le \mathrm{fl}\Big(\frac{\overline{\gamma}_n(|A||B|)}{(1-(n+2)\mathrm{u})}\Big) \tag{6}$$

で評価できる．これを簡易法と呼ぶ．

5　尾崎らの精密法

定数 $\lambda \in \mathbb{F}$ を

$$\lambda = \Big\lceil \frac{\log_2(n+1) - \log_2 \mathrm{u}}{2} \Big\rceil$$

と定める．2 つのベクトル $v \in \mathbb{F}^n$ と $w \in \mathbb{F}^n$ をそれぞれ

$$v_i = 2^{\lambda}\cdot 2^{\lceil \log_2 max_{1\le j\le n}|a_{ij}|\rceil}, w_j = 2^{\lambda}\cdot 2^{\lceil \log_2 max_{1\le i\le m}|b_{ij}|\rceil}$$

と定める．ここで $e = (1, 1, \cdots, 1)^T \in \mathbb{F}^n$ を定義し、

$$A^{(1)} = \mathrm{fl}((A + ve^T) - ve^T), \quad A^{(2)} = \mathrm{fl}(A - A^{(1)})$$
$$B^{(1)} = \mathrm{fl}((B + ew^T) - ew^T), \quad B^{(2)} = \mathrm{fl}(B - B^{(1)})$$

を実行する．以上の結果

$$A = A^{(1)} + A^{(2)}, \quad B = B^{(1)} + B^{(2)}$$

が成立する．ここで行列乗算 AB は

$$\begin{aligned} AB &= (A^{(1)} + A^{(2)})(B^{(1)} + B^{(2)}) \\ &= A^{(1)}B^{(1)} + A^{(1)}B^{(2)} + A^{(2)}B \end{aligned}$$

と、三つの行列積により変換され、$A^{(1)}$と $B^{(1)}$ に対して

$$\mathrm{fl}(A^{(1)}B^{(1)}) = A^{(1)}B^{(1)}$$

が成立する．

$$\begin{aligned} \gamma_n|A||B| &\le \overline{\gamma}_n \frac{\mathrm{fl}(|A||B|)}{(1-\mathrm{u})^n} \le (1+\mathrm{u})\frac{\mathrm{fl}(\overline{\gamma}_n(|A||B|))}{(1-\mathrm{u})^n} \\ &\le (\frac{1}{1-\mathrm{u}}) \cdot \frac{\mathrm{fl}(\overline{\gamma}_n(|A||B|))}{(1-\mathrm{u})^n} = \frac{\mathrm{fl}(\overline{\gamma}_n(|A||B|))}{(1-\mathrm{u})^{n+1}} \\ &\le \frac{\mathrm{fl}(\overline{\gamma}_n(|A||B|))}{(1-(n+1)\mathrm{u})} \le \mathrm{fl}\Big(\frac{\overline{\gamma}_n(|A||B|)}{(1-(n+2)\mathrm{u})}\Big) \quad (7) \end{aligned}$$

を得る．よって，$A^{(1)}B^{(2)}$と $A^{(2)}B$ の包含を

$$\begin{aligned} A^{(1)}B^{(2)} &\in \Big\langle \mathrm{fl}(A^{(1)}B^{(2)}), \mathrm{fl}\Big(\frac{\overline{\gamma}_n(|A^{(1)}||B^{(2)}|)}{(1-(n+2)\mathrm{u})}\Big)\Big\rangle \\ &=: \langle M^{(1)}, R^{(1)}\rangle \\ A^{(2)}B &\in \Big\langle \mathrm{fl}(A^{(2)}B), \mathrm{fl}\Big(\frac{\overline{\gamma}_n(|A^{(2)}||B|)}{(1-(n+2)\mathrm{u})}\Big)\Big\rangle \\ &=: \langle M^{(2)}, R^{(2)}\rangle \end{aligned}$$

と得る．ここで $M^{(0)}$を$\mathrm{fl}(A^{(1)}B^{(1)})$ とすれば AB は下記のように包含される．

$$\begin{aligned} AB &\in M^{(0)} + \langle M^{(1)}, R^{(1)}\rangle + \langle M^{(2)}, R(2)\rangle \\ &= \langle M^{(0)} + M^{(1)} + M^{(2)}, R^{(1)} + R^{(2)}\rangle \end{aligned}$$

上式の中心と半径を最近点への丸めで計算を行い，最終的にはすべての成分が浮動小数点数である中心と半径の行列をそれぞれ求めなければならない．
ここで，中心の計算を高精度に行うために，浮動小数点の和に関するアルゴリズムを紹介する． Knuth は $a, b \in \mathbb{F}$ に対して

$$a + b = x + y, x = \mathrm{fl}(a + b)$$

となるような $x, y \in \mathbb{F}$ を求めるアルゴリズムを開発した．すなわち，$|a| \ge |b|$ となるように並べかえてから

$$\begin{aligned} x &= \mathrm{fl}(a + b), \\ y &= \mathrm{fl}(a - x) + b. \end{aligned}$$

このアルゴリズム (尾崎) を $\big[x, y\big] = \mathrm{TS}(a, b)$ とし，行列の和に拡張して次のように適用する．

$$\big[H_1, H_2\big] = \mathrm{TS}(M^{(0)}, M^{(1)}), \big[H_3, T_1\big] = \mathrm{TS}(H_2, M^{(2)}),$$
$$\big[M, T_2\big] = \mathrm{TS}(H_3, H_1).$$

この結果

$$M^{(0)} + M^{(1)} + M^{(2)} = M + T_1 + T_2$$

が成立する．

$$AB \in \langle M + T_1 + T_2, R^{(1)} + R^{(2)}\rangle \subseteq \langle M, |T_1| + |T_2| + R^{(1)} + R^{(2)}\rangle$$

を得る．さらに

$$\begin{aligned} |T_1| + |T_2| + R^{(1)} + R^{(2)} &\le \frac{\mathrm{fl}(|T_1| + |T_2| + R^{(1)} + R^{(2)})}{1 - 3\mathrm{u}} \\ &\le \mathrm{fl}\Big(\frac{|T_1| + |T_2| + R^{(1)} + R^{(2)}}{1 - 4\mathrm{u}}\Big) \\ &:= R \end{aligned}$$

と計算することにより，最終的な包含

$$AB \in \langle M, R\rangle, M, R \in \mathbb{F}^{m\times p}$$

を得る．

6 数値実験

前章のアルゴリズムに従って，数値実験を行った．計算機を FUJITSU のノートパソコン FMV-S8390，CPU は intel Core2 Duo P8700 2.53GHz メモリ 2.00GB である．OS は Windows7 で Mathematica ver.8.0.1.0 上でプログラムを作成した．今回は，前章で述べたアルゴリズム (尾崎) を Mathematica に実現し、簡易法の製作した精度保証付きアルゴリズムとで Mathematica 上での数値計算の精度と計算速度の比較を行った．

時間について精密法との結果を比較すると $n = 256$ の時，精密法の結果は簡易法に比べ，約 3000 の時間がかかった． また n の値が上がるにつれ，計算速度も比例して大きくなることが分かる..

精度において，n の値が小さいときは大きな精度の差は見られないが $n = 256$ において精密法の計算結果は簡易法の計算結果よりも約 3 桁数値が正確に求められていることが分かる．また n の値が大きくなるにつれ，精密法の数値計算は簡易法の数値計算よりもより精度の高い計算結果を得ることが可能となる．

7 終わりに

本研究では，尾崎ら [2] により，成分がすべて浮動小数点数で表現される行列 A と B の積 AB を包含する方法を学び，Mathematica 上でのアルゴリズムの構築を実現した．計算速度において簡易法の方が尾崎よりも高速に数値計算の実行ができた．しかし，精度保証において n の値が大きくなるにつれ，尾崎の数値計算は簡易法より精密な精度保証ができると分かった．

参考文献

[1] Oishi,S.and Rump,S.M. :「Fast venefication of solutions of matrix equations,Numerische Mathematik」, 90[4]. (2002).

[2] 尾崎克久，荻田武史，大石進一 :「有向丸めの変更を使用しないタイトな行列積の包含方法」．応用数理,Vol21,No3,PP186〜196(2011).

[3] 杉浦洋 :「数値計算の基礎と応用-数値解析学への入門」．サイエンス社 (1997).