数理解析研究所講究録
第1927巻 2014年 122-127

Risa/Asirでの行列演算高速化の試み

兵頭礼子
NORIKO HYODO
サレジオ工業高等専門学校
SALESIAN POLYTECHNIC

北村竜之介
RYUNOSUKE KITAMURA
サレジオ工業高等専門学校
SALESIAN POLYTECHNIC

近藤 祐史
YUJI KONDOH
香川高等専門学校
NATIONAL INSTITUTE OF TECHNOLOGY, KAGAWA COLLEGE

村尾裕一
HIROKAZU MURAO
電気通信大学
THE UNIVERSITY OF ELECTRO-COMMUNICATIONS

齋藤友克
TOMOKATSU SAITO
(株)アルファオメガ
ALPHAOMEGA INC.

1 はじめに

本研究では，数式処理システムRisa/Asirの行列演算、特に逆行列及び行列式の演算について高速化を試みる．すでに実装されているアルゴリズムと，新たに実装したアルゴリズムを実行し，演算速度を測り，結果をもとに考察を行う．考察から，アルゴリズムの特徴と行列の特徴がどのように関連しているのかを導き出すことを目的としている．

2 実装

2.1 アルゴリズム

数式処理システムRisa/Asirには，逆行列の関数として，ガウスの消去法が実装されている．本研究では，行列式演算に余因子展開，逆行列演算に余因子行列を用いた逆行列の計算アルゴリズムを実装する．

ここでアルゴリズムの計算量を考えると，ガウスの消去法の計算量は$O(n^3)$であるが，余因子展開を用いたアルゴリズムでは$O(n!)$となり、演算時間に大きな差が生じることが考えられる．しかし，消去法と余因子アルゴリズムの演算の特徴を比較すると，消去法は計算に除算を含んでおり，余因子アルゴリズムは除算を含んでいない．数式処理において，除算のコストは大きいため，アルゴリズム内の除算の存在は演算速度が遅くなる原因となりうる．両アルゴリズムには大きな計算量の差があるが，除算の有無により演算速度が大きく変わるのではないかと予測できる．

2.2 使用する行列データ

逆行列と行列式の演算時間を測定する際，データとして使用する行列を以下に示す．本研究では逆行列を求めるため，使用する行列は，正則であるとする．また，計算に用いる行列のサイズを 2 から 20 までとして，8 種類の要素を格納させる．整数を要素に用いる場合は，すべてランダム関数で生成している．

1. 整数 (1)
   1 から 9 までの整数を生成し，要素に格納する．以下に行列の例を示す．

$$\begin{pmatrix} 5 & 6 & 3 \\ 3 & 8 & 1 \\ 3 & 3 & 4 \end{pmatrix}$$

2. 整数 (2)
   整数を生成し，要素に格納する．以下に行列の例を示す．

$$\begin{pmatrix} 578848526 & 4146913070 & 1452005258 \\ 4212814465 & 950422373 & 3029421110 \\ 3596577449 & 1908844540 & 142578355 \end{pmatrix}$$

3. 有理数 (1)
   分子を 1 として，文武に 1 から 9 の整数を入れた有理数を生成し，要素に格納する．$\frac{1}{1}$ となった場合、有理数ではなく 1 となる．以下に行列の例を示す．

$$\begin{pmatrix} \frac{1}{8} & 1 & \frac{1}{3} \\ \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{6} \end{pmatrix}$$

4. 有理数 (2)
   分母と分子に 1 から 9 の整数を入れた有理数を生成し，要素に格納する．$\frac{nz}{z}$ となった場合、有理数ではなく整数となる．以下に行列の例を示す．

$$\begin{pmatrix} \frac{3}{8} & \frac{3}{4} & \frac{1}{6} \\ \frac{8}{5} & \frac{2}{3} & 9 \\ \frac{1}{4} & \frac{5}{9} & 5 \end{pmatrix}$$

5. 有理数 (3)
   分子を 1 とし，分母に整数を入れた有理数を生成し，要素に格納する．$\frac{1}{1}$ となった場合、有理数ではなく 1 となる．以下に行列の例を示す．

$$\begin{pmatrix} \frac{1}{2465694618} & \frac{1}{112252926} & \frac{1}{334691897} \\ \frac{1}{1249751105} & \frac{1}{1333718913} & \frac{1}{3337628481} \\ \frac{1}{30084166} & \frac{1}{880414402} & \frac{1}{17084333} \end{pmatrix}$$

6. 有理数 (4)
   分母と分子に整数を入れた有理数を生成し，要素に格納する．$\frac{nz}{z}$ となった場合、有理数ではなく整数となる．以下に行列の例を示す．

$$\begin{pmatrix} \frac{1070118630}{2111543209} & \frac{3372116884}{667945179} & \frac{1361507145}{3419810344} \\ \frac{564514868}{1384858297} & \frac{1235233343}{1413475797} & \frac{3753862504}{2643520359} \\ \frac{5520829}{58992789} & \frac{1177564811}{1565944657} & \frac{854777807}{2823672895} \end{pmatrix}$$

7. 一変数多項式
   最大次数 5 次の一変数多項式を要素に格納する．一変数多項式生成の擬似コードを以下に示す．

```
for(K=0;K<5;K++){
     a_{ij} += random( )%10)*x^k;
}
```

8. ファンデルモンド行列

$$\begin{pmatrix} 1 & x & x^2 & x^3 \\ 1 & y & y^2 & y^3 \\ 1 & z & z^2 & z^3 \\ 1 & u & u^2 & u^3 \end{pmatrix}$$

9. 多変数多項式

$$\begin{pmatrix} x^3 & y^4 & z & q^2 \\ x^4 & y & z^2 & q^3 \\ x & y^2 & z^3 & q^4 \\ x^2 & y^3 & z^4 & q \end{pmatrix}$$

これらの行列データを用いて計算速度を計測する．実行環境は，CPU　AMD Phenom II X4 945(3008.42MHz)，Memory 4GB，OS FreeBSD 10.0-Release 64bit 搭載のマシンとする．

3　演算速度の比較

3.1　行列式の計測時間

2.2 に示した行列データを用いて，行列式及び逆行列の演算の速度を計測した．表 1, 2, 3 に行列式の計測時間をまとめた表を示す．ただし，整数 (1)，整数 (2)，有理数 (1)，有理数 (2)，有理数 (3)，有理数 (4)，1 変数多項式では，既に実装されていた消去法アルゴリズムが圧倒的に高速であったため省略し，1 変数多項式の時とファンデルモンド行列、多変数多項式の時の結果のみ示す．

表 1: 行列式計算時間 (1 変数多項式)

サイズ	消去法 (sec)	余因子展開 (sec)
4	0.000304	0.000356
5	0.002513	0.001716
6	0.001931	0.009952
7	0.004596	0.07221
8	0.01974	0.5805
9	0.0336	5.39
10	0.0336	47.4
11	0.05615	565.7
12	0.08379	
13	0.1345	
14	0.1922	
15	0.284	
16	0.4161	

表 2: 行列式計算時間 (ファンデルモンド行列)

サイズ	消去法 (sec)	余因子展開 (sec)
4	8.798e-05	2.2e-05
5	2e-06	7e-05
6	0.02578	0.000482
7	1.667	0.004227
8		0.04197
9		0.4607
10		5.387

表 3: 行列式計算時間 (多変数多項式)

サイズ	消去法 (sec)	余因子展開 (sec)
4	0.000202	4.1e-05
5	0.001855	0.000191
6	0.04741	0.001208
7	2.081	0.01517
8		0.07579
9		0.8009
10		9.297

3.2 逆行列の計算時間

表 4, 5, 6 に逆行列の計算時間をまとめた表を，行列の種類ごとに記載する．ただし，整数 (1)，整数 (2)，有理数 (1)，有理数 (2)，有理数 (3)，有理数 (4)，1 変数多項式では同じような特徴が出たため省略し，1 変数多項式の時とファンデルモンド行列、多変数多項式の時の結果のみ記載する．

4 考察

4.1 まとめと今後の発展

3.1 と 3.2 の結果より，行列式と逆行列の計測時間には同じ特徴が見える．どちらも，整数，有理数，1 変数多項式を要素に持つ行列では消去法のほうが高速であり，ファンデルモンド行列，多変数多項式では余因子アルゴリズムで高速に演算が可能である．実験結果より，行列演算の高速化を図るには次の要因が大きく関係していると考えられる．

- 行列の要素の種類
- アルゴリズム内の演算の種類 (除算の有無)

ただ単純に計算量は考慮しなくてもよいわけでなく，行列の要素に格納されている式の種類によって，それに適したアルゴリズムを選択する場合においては，計算量をあまり考慮する必要がない場合もある．要素内の式が高次元多項式になるほど，余因子アルゴリズムのような，計算量が大きくとも計算に除算を含まないアルゴリズムの方が，行列演算を得意 (行列演算において合理的) とするのではないかと考えられる．高次元多項式だけでなく，有利式の分母と分子に高次元多項式が入るような有利多項式などの場合にも，余因子アルゴリズムは消去法より演算速度が速いと予測できる．

本研究で作成した余因子アルゴリズムのプログラムは余因子行列を求めるため，行列の各要素に対しての小行列を行列式を求めている．この実装ではかなり演算速度に影響を及ぼすと考えられる．そこで，余因子アルゴリズムにハッシュテーブルのデータ構造を用いて，要素同士の計算を行うことで，さらなる演算の高速化を図れるのではないかと考えている．ハッシュテーブルを採用することで，計算量が $O(2^n)$ 程度に計算量を抑えることが可能であると考えている．また，$det()$ 関数の実装にも改善の余地がある．今後の研究では，プログラムを改良し，さらなる行列演算の高速化を図る．また，行列の要素の種類とアルゴリズムの関係を調査する．さらに，演算と使用メモリの関係と，コンピュータ内の動作の特徴，疎行列での挙動および，大きなキャッシュを搭載しているマシンでの挙動なども調査する予定である．これらをを調査することで，行列演算の高速化の詳細な要因を突き止め，行列に対して最適なアルゴリズム選択をするための指標にできると考えている．

表 4: 逆行列計算時間 (1 変数多項式)

サイズ	消去法 (sec)	余因子 (sec)
4	0.002551	0.001775
5	0.005442	0.01082
6	0.01362	0.08296
7	0.02777	0.6395
8	0.05754	5.406
9	0.1077	55.28
10	0.1767	591.4
11	0.3022	
12	0.4937	
13	0.7553	
14	1.148	
15	1.617	
16	2.241	

表 5: 逆行列計算時間 (ファンデルモンド行列)

サイズ	消去法 (sec)	余因子展開 (sec)
4	0.000899	6.4e-05
5	0.01436	0.000394
6	0.5697	0.003339
7	29.01	0.03298
8		0.3912
9		4.677

表 6: 逆行列計算時間 (多変数多項式)

サイズ	消去法 (sec)	余因子 (sec)
4	0.003441	0.000146
5	0.01492	0.000796
6	0.7399	0.008253
7	38.79	0.05344
8		0.6138
9		7.441