科学・技術の世界
数式処理システムによる新時代の数学

行木 孝夫、松本 圭司 (北大 理 数学)

第５回, 2017 November 06

ベクトルと行列の演算，逆行列，行列の基本変形

ベクトルの Maxima への入力および計算の実行

- ２次元の行ベクトル:

  ```
  x:[1,2]; y:[3,4];
  ```

- ３次元の行ベクトル:

  ```
  z:[1,2,3]; w:[4,5,6];
  ```

- ベクトルの和，差，スカラー倍:

  ```
  x+y; z-w; 5*x; 6*y; 7*z-8*w;
  ```

- ベクトルの内積 $(x, y) = x \cdot y$, $(z, w) = z \cdot w$:

```
x.y;  z.w;
```

- 列ベクトルの入力

```
transpose([1,2]); matrix([1],[2]);
transpose([1,2,3]); matrix([1],[2],[3]);
```

- 列ベクトルの演算

  行ベクトルの場合と同様.

ベクトルの Maxima への入力および計算の実行

- 行列の入力:

```
A:matrix([1,2],[3,4]);
B:matrix([4,3],[2,1]);
C:matrix([1,1,1],[1,2,2]);
D:matrix([1,1,1],[1,2,2],[1,2,3]);
```

- 行列の和, 差, スカラー倍

```
A+B; B-A; 2*A;
```

- 行列の積, べき乗

```
A.B; B.A; A.C; C.D; A^^3; D^^2;
```

```
A*B; C^2;
```

と入力すると，各成分の積を計算した行列となる．間違いやすいので，十分に注意すること．行列の計算が正しくないように思った場合は，この入力ミスがないか確認をする．

- 行列式，逆行列，転置行列

  ```
  determinant(A); invert(D); D^^(-1); transpose(C);
  ```

- 行列の基本変形; i行とj行の交換，i列とj列の交換，i行からj行のc倍をひく，i列からj列のc倍をひく

  ```
  rowswap(C,1,2); columnswap(D,2,3);
  rowop(A,1,2,c); columnop(A,1,2,c);
  ```

rowop (columnop) は，c倍した他の行 (列) をひく操作である．後で加える操作を紹介する．

行列の基本変形には，前記の他に i行を c 倍する操作とj列を c 倍する操作がある．Maxima ではこれらの操作を直接実行するコマンドは存在しない．そこで後でそのコマンドを自身で作成してもらう (レポート問題)．

演習問題: 1 以下のベクトル $\mathbf{x} = \begin{pmatrix} 3 \\ 4 \end{pmatrix}, \mathbf{y} = (2, 5, -1)$ と行列

$$A = \begin{pmatrix} 2 & 5 \\ 1 & 4 \end{pmatrix},\ B = \begin{pmatrix} 4 & 2 & 1 \\ 3 & 6 & 2 \end{pmatrix},\ C = \begin{pmatrix} -2 & 3 \\ 5 & 1 \\ -1 & 4 \end{pmatrix}$$

に対して，以下を計算せよ．

(1) ${}^t\mathbf{x}\, B$ (2) ${}^t A\, \mathbf{x}$ (3) $\mathbf{y}\, C\, \mathbf{x}$ (4) $\mathbf{y}\, C + 2\, {}^t\mathbf{x}$

(5) $B\, {}^t y + A\, \mathbf{x}$ (6) $-3A^{-1}$ (7) $C\, A$ (8) $C\, B$

(9) ${}^t B\, A$ (10) $B\, C + 3A$

2 以下の行列式を計算せよ．変数があるものは因数分解せよ．

(1) $\begin{vmatrix} y^2+x & xy+y \\ xy+x & x^2+y \end{vmatrix}$.　(2) $\begin{vmatrix} 5 & 6 & 0 \\ 9 & -8 & 7 \\ -3 & -5 & 0 \end{vmatrix}$.　(3) $\begin{vmatrix} 3 & -12 & 6 \\ -13 & -39 & 78 \\ 2 & -2 & -2 \end{vmatrix}$.

(4) $\begin{vmatrix} a & b & c \\ b & a & b \\ c & b & a \end{vmatrix}$.　(5) $\begin{vmatrix} 1 & 2 & 3 & 4 \\ 3 & 4 & 5 & 6 \\ 1 & 2 & 6 & 7 \\ 3 & 4 & 6 & 9 \end{vmatrix}$.　(6) $\begin{vmatrix} a & b & 0 & 0 \\ 0 & a & b & 0 \\ 0 & 0 & a & b \\ b & 0 & 0 & a \end{vmatrix}$.

3 自然数 n に対して A_n を以下で定める．$\det(A_n)$ の一般形を予想せよ．

$$A_n = \begin{pmatrix} 1 & 2 & 3 & \cdots & n \\ n & 1 & 2 & \cdots & n-1 \\ n-1 & n & 1 & \cdots & n-2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 2 & 3 & 4 & \cdots & 1 \end{pmatrix}$$

この講義の初回で説明したように, 行列の簡約化を求めるコマンドは Maxima には実装されていない. 次週はそのコマンドを作成する. その作成ために, 行列から種々のデータを取り出す方法を説明する.

- 第i行, 第j列, 行の個数, 列の個数

```
row(D,2); D[3];
col(B,1);
length(C);
length(C[1]);
```

行 (列) の個数についてに関数として定義しておくと便利

rowdim(A):=length(A); coldim(A):length(A[1]);

さらに行列の型を取り出す関数を以下のように定めておくことを推奨

size(A):=matrix([length(A),length(A[1])]);

rowop, columnop の操作を c 倍を加える操作に変更した関数を紹介.

```
gyouop(x,i,j,c):=block([xx],xx:copymatrix(x),
                        xx[i]:x[i]+c*x[j],return(xx));
retuop(x,i,j,c):=block([xx],xx:transpose
                       (gyouop(transpose(x),i,j,c)),return(xx));
```

gyouop(x,i,j,c):= は, gyouop という関数を右辺で定義するという意味.
x,i,j,c はその関数の変数.
block( ) はカッコ内をひとまとまりの操作をするという意味.
［xx］, は gyouop を定義するために xx を変数として使うことを宣言. 他でこの変数が使われても影響しないようにするために行う.
xx:copymatrix(x), は基本変形する行列 x のコピーを作成し xx とする.
xx[i]:x[i]+c*x[j], xx の 第i行を x の第i行に x のj 行のc倍を加えたものに変更. もとの行列 x は変更したくないのでコピーに操作する.
return(xx) は xx の値を答えとして返す.

retuop(x,i,j,c) は x を転置し gyouop(x,i,j,c) を施し, それを転置したものを返す操作.

xx:copymatrix(x); xxx:x との相違点.

どちらの場合も x と同じ行列となるが, あとで x を変化させた場合にxx はその変化には関係せずそのままであるが, xxx はその変化に応じて同じように変化する. 以下の例を見るとわかる.

```
x:matrix([1,2],[3,4]); xx:copymatrix(x); xxx:x;
x[1]:x[1]-x[2]; x; xx; xxx;
```

レポート問題 1 行列 A のi行を c 倍する関数と行列 A の j列を c 倍する関数 *rowmul(A,i,c)*, *colmul(A,i,c)* を作成し, その関数の動作確認をせよ.

正方行列 A に対して, そのすべての対角成分の和は A の固有和 (the trace of A) と呼ばれる. A の固有和を求めるコマンドは, 普通に立ち上げた Maxima には存在していない.

レポート問題 2 正方行列 A の固有和を求める関数 *tr(A)* を作成し, その関数の動作確認をせよ.

バッチファイルの作成

バッチファイルとはそれを読み込ませると，自動的にそこに記載されている
内容が実行される形式のファイル.

Maxima 用のバッチファイルの作成方法:

Maxima を立ち上げて，実行したい内容を記載する.

```
rowdim(A):=length(A)$
coldim(A):=length(A[1])$
size(A):=matrix([length(A),length(A[1])])$
gyouop(x,i,j,c):=block([xx],xx:copymatrix(x),
                        xx[i]:x[i]+c*x[j],return(xx))$
retuop(x,i,j,c):=block([xx],xx:transpose
                       (gyouop(transpose(x),i,j,c)),return(xx))$
```

Maxima ワークシートの左上にある "ファイル(F)" をクリックし, 名前を付けて保存（編集の時は "上書き保存" を利用）を選び, フォルダーとファイル名を指定する. 拡張子が wxmx のファイルとなる.

そのあと, Maxima ワークシートの左上にある "エクスポート(E)" を選ぶ. ファイルの種類で mac を選ぶと, 内容がバッチファイルに変換されて保存されるので, それをセーブするフォルダーとファイル名を指定する.

それを呼び出すときは, Maxima ワークシートの左上にある "ファイル(F)" をクリックし, バッチファイル(B) を選び, 目的のファイル名とそのファイルがセーブされているフォルダーを指定する.

バッチファイルは編集しにくいので, 編集は最初に保存した Maxima ワークシートを開く. Maxima ワークシートの左上にある "ファイル(F)" をクリック, さらに "開く(O)" をクリックし, 以前にセーブしたファイルを開く. それを編集した後, "エクスポート(E)" を実行.

References

[1] 上見練太郎, 勝股脩, 加藤重雄, 久保田幸次, 神保秀一, 山口佳三 共著, 微分 改訂版, 共立出版, 東京, 2014.

[2] 上見練太郎, 勝股脩, 加藤重雄, 久保田幸次, 神保秀一, 山口佳三 共著, 積分 改訂版, 共立出版, 東京, 2014.

[3] 泉屋周一, 石川剛郎, 陳蘊剛, 上見練太郎, 三波篤郎, 西森敏之 共著, 行列と連立一次方程式, 共立出版, 東京, 1996.

[4] 泉屋周一, 石川剛郎, 陳蘊剛, 上見練太郎, 三波篤郎, 西森敏之 共著, 線形写像と固有値, 共立出版, 東京, 1996.

[5] 三宅 敏恒 著,
入門線形代数, 培風館, 東京, 1991.

[6] 三宅 敏恒 著,
入門微分積分, 培風館, 東京, 1992.