

# 線形代数(I, 入門)

—数値計算(09/10/23)—

—数式処理ソフトMaple入門(II)を改訂—

関西学院大学理工学部 西谷滋人

Copyright ©2007-2009 by Shigeto R. Nishitani

## 行列と連立方程式と写像

大学でも理系で必修なのは微積分と線形代数です。線形代数というと逆行列と固有値の計算がすぐに思い浮かぶでしょう。計算がややこしくてそれだけでいやになります。でも、行列の計算法は一連の手順で記述できるので、Mapleでは微積分とおなじように一個のコマンドで片が付きます。それが3 x 3以上でも同じです。問題はその意味です。ここでは、線形代数の計算がMapleを使えばどれほど簡単にできるかを示すと共に、線形代数の基本となる概念についてスクリプトと描画を使って、直観的に理解することを目的とします。

まずは連立方程式から入っていきます。中学の時に

$$4x = 2$$

というのを解きますよね。一般的には

$$ax = b$$
$$x = a^{-1}b$$

と書けるというのは皆さんご存知のはず。これと同じようにして連立方程式を書こうというのが逆行列の基本。つまり

$$2x + 5y = 7$$
$$4x + y = 5$$

という連立方程式は、係数から作られる2 x 2行列を係数行列A、左辺の値で作るベクトルをbとして、

$$Ax = b$$
$$x = A^{-1}b$$

としたいわけです。実際にMapleでやってみましょう。

```
> restart;
A:=Matrix([[2,5],[4,1]]);
```

$$A := \begin{bmatrix} 2 & 5 \\ 4 & 1 \end{bmatrix}$$

(1)

### ▼ スクリプトの中身

こちらの欄では、Mapleスクリプトの中身を解説していきます。最初読むときは

```
> b:=Vector([7,5]);
b := \begin{bmatrix} 7 \\ 5 \end{bmatrix}
> with(LinearAlgebra):
> x0:=MatrixInverse(A).b;
x0 := \begin{bmatrix} 1 \\ 1 \end{bmatrix}
```

(2)

(3)

と簡単に求めることができます。

無視してもらって結構です。

(1)行列は英語でMatrixです。こうして行列を作ります。  
(2)ベクトルは英語でVectorです。これで縦ベクトルができます。  
代数はlinear algebraと言います。withでLinearAlgebraというライブラリーパッケージを読み込んでおきます。  
(3)逆行列はmatrix inverseと言います。行列AのMatrixInverseを求めて、ベクトルbに掛けています。

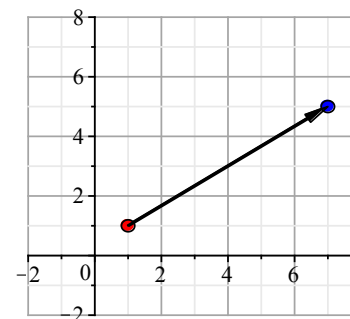
次に、これを2次元上のグラフで見たいと思います。

```
> with(plots):with(plottools)
:
> p0:=convert(x0,list);
p1:=convert(b,list);
p0 := [1, 1]
p1 := [7, 5]
> point1:=[disk(p0,0.2,color=red),
disk(p1,0.2,color=blue)]:
line1:=arrow(p0,p1,.05,.3,
.1):
```

(4)

これ以降は描画関連の操作です。まず描画に必要なライブラリーパッケージ(plotsおよびplottools)をwithで読み込んでおきます。  
(4)ベクトルを位置座標を意味するようにlistへ変換(convert)しておきます。位置p0に円(disk)を半径0.2,赤色で描きます。同じように位置p1に半径0.2,青色でdiskを描きます。もう一つ、p0からp1に向かう矢印(arrow)を適当な大きさで描きます。後ろの数字をいじると線の幅や矢印の大きさが変わります。  
これらをまとめて表示(display)します。このとき、表示範囲を-8..8,-8..8とします。

```
> display(point1,line1,view=[-2..8,-2..8],gridlines=true);
```



逆行列は

```
> MatrixInverse(A);
```

$$\begin{bmatrix} 1 & 5 \\ -\frac{1}{18} & \frac{5}{18} \\ 2 & 1 \\ \frac{9}{9} & -\frac{9}{9} \end{bmatrix}$$

(5)

で求められます。先ほどの矢印を逆に青から赤へたどる変換になっています。これが、連立方程式を解く様子をグラフで示しています。つまり、行列Aで示される変換によって求まる青点で示したベクトルb(7,5)を指す元の赤点を探すというものです。答えは(1,1)となります。

では、元の赤点をもう少しいろいろ取って、行列Aでどのような点へ写されるかを見てみましょう。

```
> N:=30:point2:=[]:line2:=[]:
for k from 0 to N-1 do
  x0:=Vector([sin(2*Pi*k/N),
cos(2*Pi*k/N)]);
  x1:=A.x0;
  p0:=convert(x0,list);
  p1:=convert(x1,list);
  point2:=[op(point2),disk
(p0,0.05,color=red)];
  point2:=[op(point2),disk
(p1,0.05,color=blue)];
  line2:=[op(line2),line(p0,
p1)];
end do;
>
```

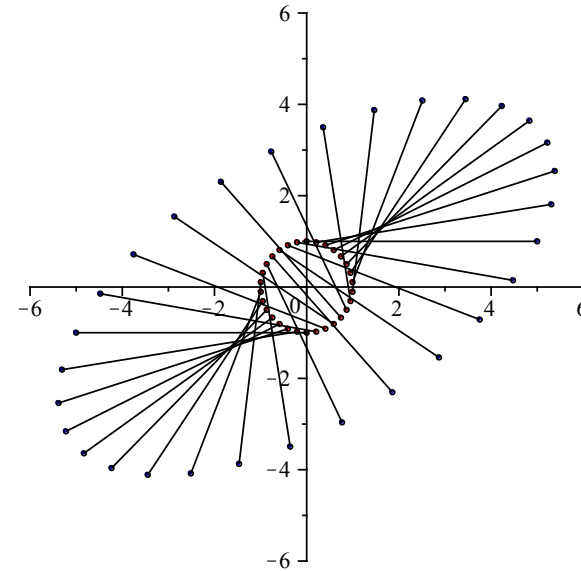
先ほどの描画と使っているコマンドはほぼ同じです。ただし、スクリプトに特有の2つの新しいテクを使っています。この基本形をとり出すと、

```
> list1:=[];
for k from 0 to 2 do
  list1:=[op(list1),k];
end do;
list1;
list1 := [ ]
list1 := [0]
list1 := [0, 1]
list1 := [0, 1, 2]
(6)
```

となります。for-loopでkを0から4まで回し、list1に次々と値を追加していくというテクです。N:=30で分割した円周上の点をx0で求めて、point2にその円とそれのA.x0を、line2にはその2点を結ぶline(線)を足しています。

次の図を見てください。

```
> d:=6;
display(point2,line2,view=[-d..d,-d..d]);
d:=6
```



何やっているか分かりますか？赤点で示される円が、青点で示される楕円へ写されていることが分かるでしょう。

### ▼【課題1】固有ベクトル

では、ここでクイズです。固有ベクトルは上のグラフの何処に対応するか？

ヒントは、「行列Aの固有値、固有ベクトルを $\lambda, x_0$ とすると、

$$A x_0 = \lambda x_0$$

が成立する」です。

固有値と固有ベクトルはMapleでは以下のコマンドで求められます。

```
> lambda,P:=Eigenvectors(A);
```

$$\lambda, P := \left[ \begin{array}{c} -3 \\ 6 \end{array} \right], \left[ \begin{array}{cc} 5 & \\ -1 & 4 \\ 1 & 1 \end{array} \right]$$

(3.1)

ここではMapleコマンドのEigenvectorsで戻り値を $\lambda$ (lambdaと書きます)、Pに代入しています。この後ろ側にある行列Pの1列目で構成されるベクトルが固有値-3に対応する固有ベクトル、2列目のベクトルが固有値6に対応する固有ベクトルです。

▼解答

固有値 $\lambda$ , 固有ベクトル $x_0$ の関係式

$$A x_0 = \lambda x_0$$

を言葉で言い直すと,

「固有ベクトル $x_0$ は変換行列Aによって, 自分の固有値倍のベクトル $x_0$ に写されるベクトル」

となります. つまり変換の図で言うと,

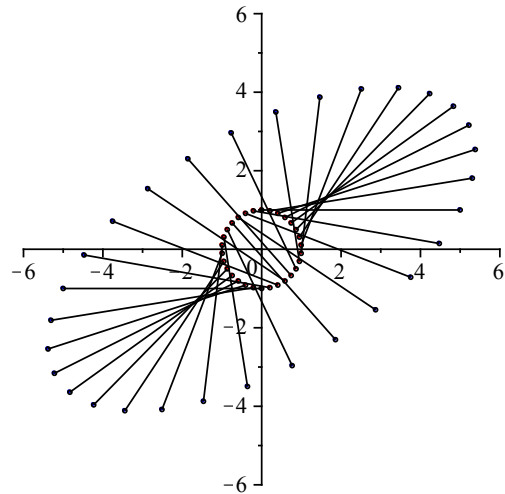
「変換しても方向が変わらない赤点」

となります. これは図で書くと

```
> vv1:=Column(P,1):
  vv2:=Column(P,2):
  a1:=vv1[2]/vv1[1]:
  a2:=vv2[2]/vv2[1]:
  pp1:=plot({a2*x,a1*x},x=-
d..d):
=>
```

Columnによって行列の第列目をとりだし, その比によって直線の傾きを求めています. そうして引いた2本の直線をpp1としてため込んで, 先ほど描いた変換の図に加えて表示(display)させます.

```
> #display(point2,line2,pp1,view=[-d..d,-d..d]);
  display(point2,line2,view=[-d..d,-d..d]);
```



上記出力は, 解答をかくためにpp1を外しています. pp1も入れて描いた直線が引かれた方向ではたしかに変換によっても方向が変わらなさそうに見えるでしょう.

ついでに行列の対角化は次のようにしてできます.

```
> MatrixInverse(P).A.P;
```

$$\begin{bmatrix} -3 & 0 \\ 0 & 6 \end{bmatrix}$$

(3.1.1)

### 【課題2】好きな行列を表示させて

では, 好きな行列をつくって同じようにして図や固有ベクトルを表示させてください. ただし, ここでは次のDeterminantで求まる行列Aの行列式が0とならないようお願いします.

```
> Determinant(A);
```

-18

(4.1)

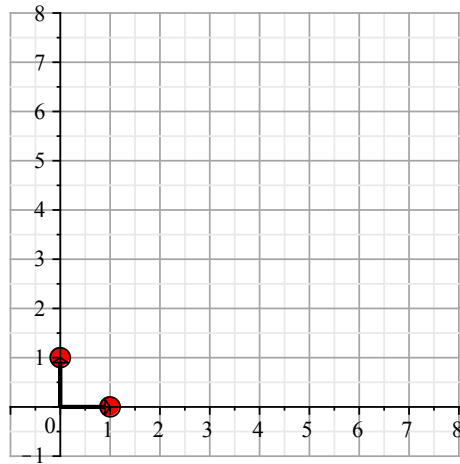
### 【課題3】行列式の意味

では次のクイズ. 先ほど求めた, 行列Aの行列式は, どこに対応するでしょうか? 以下の(1,0),(0,1)の点を変換した点に原点からベクトルを結んでその意味を説明してください. さらに, そのマイナスの意味は?.

### ▼ 解答

変換によって[1,0]と[0,1]のベクトルがどの点へ移るかを表示したのが次の図です.

```
> point3:=[]:line3:=[]:
  XX:=Matrix([[1,0],[0,1]]):
  for i from 1 to 2 do
    x0:=Column(XX,i):
    x1:=A.x0:
    p0:=convert(x0,list):
    p1:=convert(x1,list):
    #point3:=[op(point3),disk(p0,0.2,color=red),disk(p1,0.2,
color=blue)]:
    point3:=[op(point3),disk(p0,0.2,color=red)]:
    #line3:=[op(line3),arrow([0,0],p0,.05,-.3,.1),arrow([0,0],
p1,.05,.3,.1)]:
    line3:=[op(line3),arrow([0,0],p0,.05,-.3,.1)]:
  end do:
  display(point3,line3,view=[-1..8,-1..8],gridlines=true);
```



(上のスクリプトの行頭に#がある2行をその直下の式と入れ替えると意図する解答が出ます。お試しあれ。)

### 行列式が0の写像

では、行列式が0になるというのはどういう状態でしょうか？ 次のような行列を考えてみましょう。

```
> A:=Matrix([[2,1],[4,2]]);
```

$$A = \begin{bmatrix} 2 & 1 \\ 4 & 2 \end{bmatrix} \quad (6.1)$$

この行列式は

```
> Determinant(A);
```

です。これで上と同じように表示させてみましょう。

$$0 \quad (6.2)$$

```
> N:=30;point2:=[]:line2:=[]:
```

```
for k from 0 to N-1 do
```

```
x0:=Vector([sin(2*Pi*k/N),cos(2*Pi*k/N)]);
```

```
x1:=A.x0;
```

```
p0:=convert(x0,list);
```

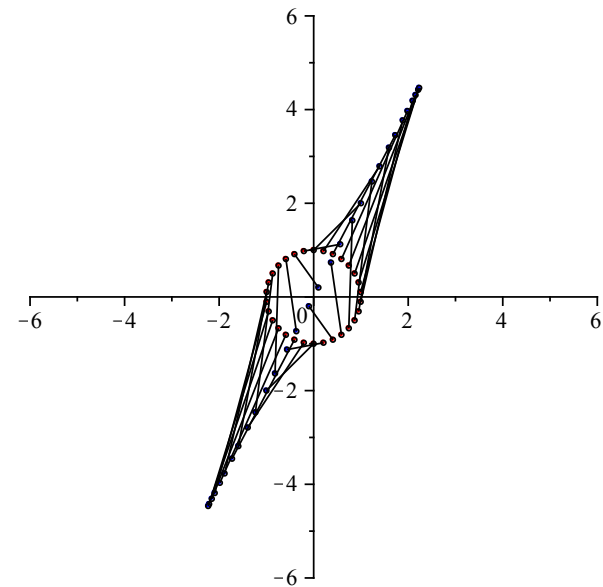
```
p1:=convert(x1,list);
```

```
point2:=[op(point2),disk(p0,0.05,color=red),disk(p1,0.05,color=blue)];
```

```
line2:=[op(line2),line(p0,p1)];
```

```
end:
```

```
> d:=6;
display(point2,line2,view=[-d..d,-d..d]);
d:=6
```



わかります？ 今回の移動先の青点は直線となっています。つまり、determinantが0ということは、変換すると面積がつぶれるという事を意味しています。平面がひとつ次元を落として線になるということです。では、この行列の表わす写像によって原点(0,0)に写される元の座標を求めてみます。連立方程式に戻してみると

```
> A.Vector([x,y])=Vector([0,0]);
```

$$\begin{bmatrix} 2x+y \\ 4x+2y \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (6.3)$$

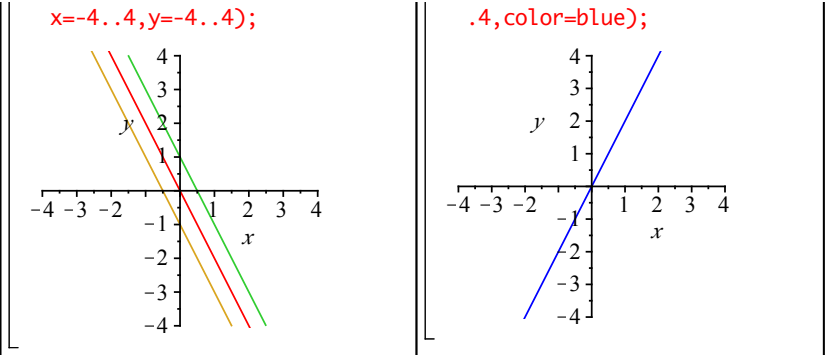
となります。とよく見ると、1行目も2行目もおなじ式になっています。2次元正方行列で、行列式が0の時には必ずこういう形になり、直線の式となります。これを表示すると左図の赤線となります。この直線上の全ての点が[0,0]へ写されることを確認してください。また、緑の線上の点は全て[1,2]へ写されることが確認できます。

```
> A.Vector([-1,2]);
```

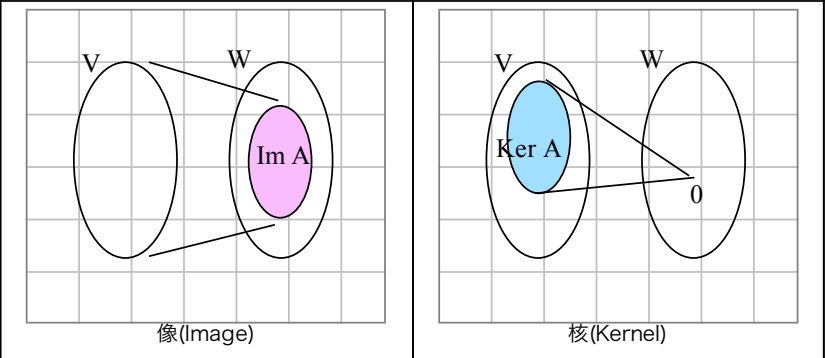
$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (6.4)$$

```
> plot([-2*x,-2*x+1,-2*x-1],
```

```
> plot([2*x],x=-4..4,y=-4.
```



こうしてすべて調べていけば、左の平面上のすべて点は右の青の直線上へ写されることが分かります。今まで見てきた円と楕円とはまったく違った写像が、行列式が0の行列では起こっていることが分かると思います。右の青線を行列Aによる像(Image, Im Aと表記)、左の赤線、つまり写像によって[0,0]へ写される集合を核(Kernel, Ker Aと表記)と呼びます。



行列Aによる写像をfとして、赤点に限らず元の点の集合をV, 移った先の点の集合をWとして、

$$f: V \rightarrow W$$

と表記されます。v,wをV,Wの要素としたとき、違うvが違うwに写されることを単射、全てのwに対応するvがある写像を全射と言います。全単射、つまり全射かつ単射、だと要素は一対一に対応します。

先ほどの  
 $> A;$   
 は全射でもなく、単射でもない例です。

$$\begin{bmatrix} 2 & 1 \\ 4 & 2 \end{bmatrix} \quad (6.5)$$

行列式が0の場合の写像は単射ではありません。このとき、逆写像が作れそうにありません。これを連立方程式に戻して考えましょう。もともと、

$$v = A^{-1} w$$

の解vは点wが写像Aによってどこから写されてきたかという意味を持ちます。逆写像が作れない場合は、連立方程式の解はパラメータをひとつ持った複数の解(直線)となります。これが係数行列の行列式が0の場合に、連立方程式の解が不定となる、あるいは像がつぶれるという関係です。

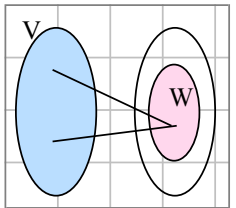
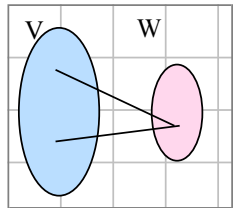
行列の次元が高い場合には、いろいろなつぶれかたをします。行列の階数と次元は  
 $> \text{Rank}(A);$   
 $\text{Dimension}(A);$   
 で求められます。

$$\begin{bmatrix} 1 \\ 2, 2 \end{bmatrix} \quad (6.6)$$

Aをm行n列の行列とすると、  
 $\text{Rank}(A) = \text{Dimension}(\text{Im } A)$   
 $\text{Dimension}(\text{Ker } A) = n - \text{Rank}(A)$

が成立し、これを次元定理と言います。  
 全射と単射の関係は、下の表のような一変数の方程式での解の性質の拡張と捉えることができます。

$a x = b$	一意	$a > 0$ $x = b/a$
	不定	$a = 0, b = 0$ 解は無数
	不能	$a = 0, b > 0$ 解は存在しない

$A x = b$	$m \times n$ 行列 A	全射でない ( $\text{Im } A < m$ ) 値域上にあるときのみ解が 存在	全射 ( $\text{Im } A = m$ ) 解は必ず存在
	単射でない $\text{Ker } A > 0$ 解は複数		
	単射 $\text{Ker } A = 0$ 解はひとつ	