

行列・ベクトル(LinearAlgebra)I 生成

Copyright @2010 by Shigeto R. Nishitani

解説

線形代数の計算にはあらかじめ関数パッケージ(LinearAlgebra)を呼び出しておく。

```
> with(LinearAlgebra):
```

ベクトルの生成(Vector)

```
> v1 := Vector([x, y]);
```

$$v1 := \begin{bmatrix} x \\ y \end{bmatrix} \quad (1.1.1.1)$$

通常の方法では、縦(列)ベクトル(column)ができることに注意。横(行)ベクトル(row)を作るには、明示する必要あり。

```
> v2 := (Vector[row])([x, y, z]);
```

$$v2 := [x \ y \ z] \quad (1.1.1.2)$$

新聞の囲み記事がcolumn、劇場の座席はrow。

行列の生成(Matrix)

```
> A0 := Matrix([[1, 2, 3], [4, 5, 6]]);
```

リストリストからの変換

```
> LL1 := [[1, 2], [3, 4]]:
```

```
> A1 := Matrix(LL1);
```

$$A1 := \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad (1.1.2.1)$$

単位行列の生成。

```
> E := IdentityMatrix(2);
```

$$E := \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (1.1.2.2)$$

対角行列を生成するDiagonalMatrixもある。

同じことは以下のようにしても生成が可能。

```
> Matrix(2,2,shape=identity);
```

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (1.1.2.3)$$

縦横ベクトル・行列の簡易作成法(MVShortcut)

かぎカッコ(<...>)を使って、ベクトルあるいは行列を直感的に作る事が可能。

カンマで区切ると縦に積み、縦棒で区切ると横に積む。セミコロンで区切るとそこで次の行へ。

```
> v1:=<x,y>; #縦ベクトル, 列  
> v2:=<x|y|z>; #横ベクトル, 行  
> A1:=<1,2;3,4>; #2x2行列  
> <A1|v1>; #2x3行列(拡大係数行列などの作成)  
> ?MVShortcut; #参照
```

$$v1 := \begin{bmatrix} x \\ y \end{bmatrix} \\ v2 := [x \ y \ z]$$

$$A1 := \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 & x \\ 3 & 4 & y \end{bmatrix}$$

(1.1.3.1)

行列、ベクトルの成分の抽出(?MVextraction)

行列A1の1行2列の成分を取り出すには、

```
> A1[1,2]; #res: 2
```

行列の一部を行列として取り出すには

```
> A1[1..2,1..2];
```

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

(1.1.4.1)

2x2行列の2列目(行の長さに関係なく)でつくるベクトルは

```
> A1[..,2..2];
```

$$\begin{bmatrix} 2 \\ 4 \end{bmatrix}$$

(1.1.4.2)

同じことが、行(Row)あるいは列(Column)抽出関数でもできる。使い方は次の通り。

```
> Column(A1,2);
```

$$\begin{bmatrix} 2 \\ 4 \end{bmatrix}$$

(1.1.4.3)

課題

1. 次の行列、ベクトルを作れ。

$$\begin{bmatrix} \text{i)} \begin{bmatrix} 3 & 3 & 3 \\ 3 & 3 & 3 \end{bmatrix} \text{ ii)} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \text{ iii)} [1 \ 2 \ 3] \text{ iv)} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix} \text{ v)} \begin{bmatrix} x & x^2 \\ x^2 & x^3 \end{bmatrix} \text{ vi)} \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \end{bmatrix}$$

▼ 解答例

I.

i)

```
> Matrix([[3,3,3],[3,3,3]]);
```

そのまま直打ちしても良いが,

$$\begin{bmatrix} 3 & 3 & 3 \\ 3 & 3 & 3 \end{bmatrix}$$

(1.3.1.1)

少し賢い生成法も記しておく。詳しくはヘルプ参照。

```
> Matrix(<3,3|3,3|3,3>);
```

```
Matrix(2,3,3):
```

$$\begin{bmatrix} 3 & 3 & 3 \\ 3 & 3 & 3 \end{bmatrix}$$

(1.3.1.2)

ii)

```
> Matrix(2,3,shape=identity);
```

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

(1.3.1.3)

iii)

```
> Vector[row]([1,2,3]);
```

```
Vector(<1|2|3>):
```

$$[1 \ 2 \ 3]$$

(1.3.1.4)

iv)

```
> with(LinearAlgebra):
```

```
V:=Vector[row]([1,2,3]);
```

```
DiagonalMatrix(V);
```

$$V := [1 \ 2 \ 3]$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix}$$

(1.3.1.5)

v)

```
> f:=(i,j) -> x^(i+j-1);
```

```
Matrix(2,f);
```

$$\begin{bmatrix} x & x^2 \\ x^2 & x^3 \end{bmatrix}$$

(1.3.1.6)

vi)

```
> Matrix(3,[seq(i,i=1..9)]);
```

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

(1.3.1.7)

vi)

```
> n:=3;
```

```
f:=(i,j)->(i-1)*n+j;
```

```
Matrix(3,3,f);
```

$$f := (i,j) \rightarrow (i-1)n + j$$

$$\begin{bmatrix} | \\ | \\ | \\ | \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

(1.3.1.8)

行列・ベクトル(LinearAlgebra) II スカラー積, 内積, 外積

Copyright ©2010 by Shigeto R. Nishitani

解説

線形代数の計算にはあらかじめ関数パッケージ(LinearAlgebra)を呼び出しておく.

```
> with(LinearAlgebra):
```

簡単な演算

スカラーとのかけ算

```
> v1:=Vector([x, y]):
3*v1;
```

$$\begin{bmatrix} 3x \\ 3y \end{bmatrix}$$

(2.1.1.1)

行列,ベクトルの足し算, 引き算

```
> LL1 := [[1, 2], [3, 4]]:
A1 := Matrix(LL1):
A2 := Matrix([[x, x], [y, y]]):
3*A1-4*A2;
```

$$\begin{bmatrix} 3-4x & 6-4x \\ 9-4y & 12-4y \end{bmatrix}$$

(2.1.1.2)

内積(DotProduct, `.`)

```
> v1:=Vector([1,1,3]): v2:=Vector([1,2,-1]):
v1.v2;
```

0

(2.1.1.3)

外積(CrossProduct, `&x`)

```
> CrossProduct(v1, v2):
v1 &x v2:
```

$$\begin{bmatrix} -7 \\ 4 \\ 1 \end{bmatrix}$$

(2.1.1.4)

スカラー三重積

```
> v3 := Vector([-1,2,1]):
CrossProduct(v1,v2).v3;
```

$$v^3 := \begin{bmatrix} -1 \\ 2 \\ 1 \end{bmatrix}$$

16

(2.1.1.5)

行列の基本操作あるいは, 行列の掃き出しに必要な行列の基本操作は RowOperation, ColumnOperationを参照, ついでに階数(Rank)

```
> Rank(A1);
```

2

(2.1.1.6)

転置(Transpose, `&T`)

行列Aのij成分a[i,j]を転置a[j,i]するには,

```
> Transpose(A1);
```

$$\begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}$$

(2.1.2.1)

また横ベクトルを縦ベクトル(あるいはその逆)にするのも同じ.

```
> Transpose(v1);
```

$$\begin{bmatrix} 1 & 1 & 3 \end{bmatrix}$$

(2.1.2.2)

課題

1. 行列 $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$, $B = \begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix}$, およびベクトル $v = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$ を作り, 以下の計算を行い結果を観察せよ.

i) $A + 3B$, ii) $A - B$, iii) $A + E$, iv) $A \cdot B$, v) $A \cdot v$, vi) $v \cdot A$

vii) v の転置(Transpose)を A に左側から掛けよ, viii) A^3

2. 2次元平面上で原点の周りの角度 t の回転行列は

$A_r := \text{Matrix}([\cos(t), -\sin(t)], [\sin(t), \cos(t)])$;

で定義できる.

i) $\pi/6$ 回転させる行列を作り, 単位ベクトル $(1,0), (0,1)$ がどの点に移動するか確認せよ.

ii) $\pi/6$ 回転させた後, 続けて $\pi/4$ 回転させる操作を続けて行う回転行列を求めよ. また, 角度を直接入力して要素を比較せよ.

3. 行列 $A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$ について $A + A^t$, $A - A^t$ を求めて交代行列, 対称行列を作れ.

解答例

```
1. > with(LinearAlgebra):
A:=Matrix([[1,2],[3,4]]);
B:=Matrix([[2,3],[4,5]]);
v:=Vector([1,2]);
E:=IdentityMatrix(2);
```

$$A := \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

$$B := \begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix}$$

$$v := \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

$$E := \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

(2.3.1.1)

```
> A+3*B;
A-B;
A+E;
A.v;
```

`v.A;`

$$\begin{bmatrix} 7 & 11 \\ 15 & 19 \\ -1 & -1 \\ -1 & -1 \\ 2 & 2 \\ 3 & 5 \\ 5 \\ 11 \end{bmatrix}$$

`Error, (in LinearAlgebra:-VectorMatrixMultiply) invalid input: LinearAlgebra:-VectorMatrixMultiply expects its 1st argument, v, to be of type Vector[row] but received Vector(2, {(1) = 1, (2) = 2})`

v.Aは次元が合わないので計算できない。次元を合わせるためには、vに転置(Transpose)をかけて横ベクトルしておく必要がある。

`> Transpose(v).A;`

$$\begin{bmatrix} 7 & 10 \end{bmatrix}$$

(2.3.1.2)

`> A^3;`

2.

i)

`> with(LinearAlgebra):
e1:=Vector([1,0]); e2:=Vector([0,1]);
Ar:=t->Matrix([[cos(t),-sin(t)],[sin(t),cos(t)]]);
Ar(Pi/6).e1;
Ar(Pi/6).e2;`

$$e1 := \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$e2 := \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

`Ar := t -> Matrix([[cos(t), -sin(t)], [sin(t), cos(t)]])`

$$\begin{bmatrix} \frac{1}{2}\sqrt{3} \\ \frac{1}{2} \\ -\frac{1}{2} \\ \frac{1}{2}\sqrt{3} \end{bmatrix}$$

(2.3.2.1)

ii.) 2つの関数を同時にプロット。

`> Ar(Pi/4).Ar(Pi/6);
Ar(Pi/6+Pi/4);`

$$\begin{bmatrix} \frac{1}{4}\sqrt{2}\sqrt{3} - \frac{1}{4}\sqrt{2} & -\frac{1}{4}\sqrt{2} - \frac{1}{4}\sqrt{2}\sqrt{3} \\ \frac{1}{4}\sqrt{2}\sqrt{3} + \frac{1}{4}\sqrt{2} & \frac{1}{4}\sqrt{2}\sqrt{3} - \frac{1}{4}\sqrt{2} \\ \cos\left(\frac{5}{12}\pi\right) & -\sin\left(\frac{5}{12}\pi\right) \\ \sin\left(\frac{5}{12}\pi\right) & \cos\left(\frac{5}{12}\pi\right) \end{bmatrix}$$

(2.3.2.2)

2つの操作の差のevalfをとるとほぼ0,つまり一致していることが確認できる。

`> evalf(Ar(Pi/6+Pi/4)-Ar(Pi/6).Ar(Pi/4));`

$$\begin{bmatrix} -2.10^{-10} & 0. \\ 0. & -2.10^{-10} \end{bmatrix}$$

(2.3.2.3)

3.

`> A:=Matrix([[1,2,3],[4,5,6],[7,8,9]]);
As:=A+Transpose(A);
Aa:=A-Transpose(A);`

$$A := \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

$$As := \begin{bmatrix} 2 & 6 & 10 \\ 6 & 10 & 14 \\ 10 & 14 & 18 \end{bmatrix}$$

$$Aa := \begin{bmatrix} 0 & -2 & -4 \\ 2 & 0 & -2 \\ 4 & 2 & 0 \end{bmatrix}$$

(2.3.3.1)

行列・ベクトル(LinearAlgebra) II 行列の基本操作, 吐き出し

Copyright ©2010 by Shigeto R. Nishitani

解説

線形代数の計算にはあらかじめ関数パッケージ(LinearAlgebra)を呼び出しておく.

```
> with(LinearAlgebra):
```

行列の基本操作

行列の基本操作あるいは, 行列の掃き出しに必要な行列の基本操作は RowOperation, ColumnOperationを参照.

吐き出し法, LU分解

吐き出し法の計算は, LUdecompositionでおこなう. まず拡大係数行列を作る.

```
> A1:=<1,2;3,4>; b:=<2,3>;
<A1|b>;
```

$$A1 := \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad b := \begin{bmatrix} 2 \\ 3 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 & 2 \\ 3 & 4 & 3 \end{bmatrix} \quad (3.1.2.1)$$

これにLU分解をかける. それぞれP(pivot, 置換), L(Lower triangle, 下三角), U(Upper triangle, 上三角)行列に代入している.

```
> P,L,U:=LUdecomposition(<A1|b>);
```

$$P, L, U := \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 3 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 2 & 2 \\ 0 & -2 & -3 \end{bmatrix} \quad (3.1.2.2)$$

さらに被約階段行列(row reduced echelon matrix; 各行の0でない最初の数が1でかつその数以外の列の数がすべて0の行列)は以下の通りで求まる.

```
> LUdecomposition(<A1|b>, output='R');
```

$$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & \frac{3}{2} \end{bmatrix} \quad (3.1.2.3)$$

階数(Rank)

階数(Rank)

```
> Rank(A1);
```

$$2 \quad (3.1.3.1)$$

課題

I. 行列 $A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$ について, RowOperationのヘルプを参照して, 次の行基本操作を行い階数を求め, コマンドRankの結果と比べよ.

- i) 2行目から1行目の4倍を引く.
- ii) 3行目から1行目の7倍を引く.
- iii) 2行目を-1/3倍する.
- iv) 3行目に2行目の6倍を足す.

RowOperationのコツは, 最初はinplace=trueを入れずに

2. 次の連立方程式の解を掃き出し法で求めよ. GenerateMatrixを使えば連立方程式から拡大係数行列を直接生成することも可能.

$$(i) \begin{cases} x+y-z=2 \\ 2x-3y+z=4 \\ 4x-y+3z=1 \end{cases} \quad (ii) \begin{cases} 2x+4y-3z=1 \\ 3x-8y+6z=5 \\ 8x-2y-9z=23 \end{cases} \quad (iii)$$

$$(iv) \begin{cases} 1x-10y-3z-7u=2 \\ 2x-4y+3z+4u=-3 \\ 3x-2y+6z+5u=-1 \\ x+8y+9z+3u=5 \end{cases} \quad \begin{cases} x+y+z=a+b+c \\ ax+by+cz=ab+bc+ca \\ bcx+cay+abz=3abc \end{cases}$$

3. 次の連立方程式をsolveを使って,

$$\begin{cases} x_1 + 2x_2 - x_3 = 0 \\ x_1 + x_2 + 3x_4 = 0 \\ x_1 + 5x_2 - 2x_3 + 3x_4 = 0 \\ x_1 + 3x_2 - 2x_3 - 3x_4 = 0 \end{cases}$$

x_1, x_2, x_3, x_4 について解け.

次にGenerateMatrixを使って, 拡大係数行列にした後, LUdecompositionを用いて吐き出しを行い結果を比較せよ.

解答例

1.

```
> A:=Matrix([[1,2,3],[4,5,6],[7,8,9]]);
with(LinearAlgebra):
?RowOperation;
```

$$A := \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \quad (3.3.1.1)$$

ヘルプに書かれてある例を見本にして、コマンドを記述。

```
> RowOperation(A,[2,1],-4);
```

$$\begin{bmatrix} 1 & 2 & 3 \\ 0 & -3 & -6 \\ 7 & 8 & 9 \end{bmatrix} \quad (3.3.1.2)$$

結果を最初の引数(A)に上書きするoptionをつける。最初からではなく、うまくいったのを確認してからつけるのがコツ。

```
> RowOperation(A,[2,1],-4,inplace=true);
```

```
> RowOperation(A,[3,1],-7,inplace=true);
```

$$\begin{bmatrix} 1 & 2 & 3 \\ 0 & -3 & -6 \\ 0 & -6 & -12 \end{bmatrix} \quad (3.3.1.3)$$

```
> RowOperation(A,[2,-1/3],inplace=true);
```

$$\begin{bmatrix} 1 & 2 & 3 \\ 0 & 1 & 2 \\ 0 & -6 & -12 \end{bmatrix} \quad (3.3.1.4)$$

```
> RowOperation(A,[3,2],6);
```

$$\begin{bmatrix} 1 & 2 & 3 \\ 0 & 1 & 2 \\ 0 & 0 & 0 \end{bmatrix} \quad (3.3.1.5)$$

最後の行がすべて0になっているので、階数は2となる。Rankにより確認。

```
> Rank(A);
```

$$2 \quad (3.3.1.6)$$

2.

i) GenerateMatrixによる係数行列と右辺のベクトルを生成する方法は以下のとおり。

```
> eqs:={x+y-z=2,2*x-3*y+z=4,4*x-y+3*z=1};
GenerateMatrix(eqs,{x,y,z});
eqs:={x+y-z=2,2*x-3*y+z=4,4*x-y+3*z=1}
```

$$\begin{bmatrix} 1 & 1 & -1 \\ 2 & -3 & 1 \\ 4 & -1 & 3 \end{bmatrix} \begin{bmatrix} 2 \\ 4 \\ 1 \end{bmatrix} \quad (3.3.2.1)$$

i)

```
> A:=Matrix([[1,1,-1],[2,-3,1],[4,-1,3]]);
b:=<2,4,1>;
LUdecomposition(<A|b>,output='R');
```

$$A := \begin{bmatrix} 1 & 1 & -1 \\ 2 & -3 & 1 \\ 4 & -1 & 3 \end{bmatrix}$$

$$b := \begin{bmatrix} 2 \\ 4 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & \frac{13}{10} \\ 0 & 1 & 0 & -\frac{21}{20} \\ 0 & 0 & 1 & -\frac{7}{4} \end{bmatrix} \quad (3.3.2.2)$$

ii)

```
> A:=Matrix([[2,4,-3],[3,-8,6],[8,-2,-9]]);
b:=<1,5,-23>;
LUdecomposition(<A|b>,output='R');
```

$$A := \begin{bmatrix} 2 & 4 & -3 \\ 3 & -8 & 6 \\ 8 & -2 & -9 \end{bmatrix}$$

$$b := \begin{bmatrix} 1 \\ 5 \\ -23 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 3 \end{bmatrix} \quad (3.3.2.3)$$

iii)

```
> A:=Matrix([[1,-10,-3,-7],[2,-4,3,4],[3,-2,6,5],[1,8,9,3]]);
b:=<2,-3,-1,5>;
LUdecomposition(<A|b>,output='R');
```

$$A := \begin{bmatrix} 1 & -10 & -3 & -7 \\ 2 & -4 & 3 & 4 \\ 3 & -2 & 6 & 5 \\ 1 & 8 & 9 & 3 \end{bmatrix}$$

$$b := \begin{bmatrix} 2 \\ -3 \\ -1 \\ 5 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 1 & 0 & \frac{1}{3} \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$

(3.3.2.4)

$$A, b := \begin{bmatrix} 1 & 1 & 0 & 3 \\ 1 & 2 & -1 & 0 \\ 1 & 3 & -2 & -3 \\ 3 & 5 & -2 & 3 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 1 & 6 & 0 \\ 0 & 1 & -1 & -3 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

(3.3.3.1)

iv)

```
> restart;
with(LinearAlgebra):
A:= Matrix([[1,1,1],[a,b,c],[b*c,c*a,a*b]]);
bb:=<a+b+c,a*b+b*c+c*a,3*a*b*c>;
RR:=LUdecomposition(<A|bb>,output='R');
```

$$A := \begin{bmatrix} 1 & 1 & 1 \\ a & b & c \\ bc & ca & ab \end{bmatrix}$$

$$bb := \begin{bmatrix} a + b + c \\ ab + bc + ca \\ 3abc \end{bmatrix}$$

$$RR := \begin{bmatrix} 1 & 0 & 0 & -\frac{a(b^2 - 2bc + c^2)}{(a-b)(a-c)} \\ 0 & 1 & 0 & \frac{(a^2 - 2ca + c^2)b}{(b-c)(a-b)} \\ 0 & 0 & 1 & -\frac{c(-2ab + b^2 + a^2)}{ab - bc + c^2 - ca} \end{bmatrix}$$

(3.3.2.5)

```
> factor(Column(RR,4)[1]);
```

とすればさらに見やすく、変形される

$$-\frac{a(b-c)^2}{(a-b)(a-c)}$$

(3.3.2.6)

3.

```
> eqs:={x1+2*x2-x3=0,x1+x2+3*x4=0,3*x1+5*x2-2*x3+3*x4=0,
x1+3*x2-2*x3-3*x4=0};
solve(eqs,{x1,x2,x3,x4});
A1,b:=GenerateMatrix(eqs,[x1,x2,x3,x4]);
LUdecomposition(<A1|b>,output='R');
```

```
eqs := {x1 + x2 + 3.x4 = 0, x1 + 2.x2 - x3 = 0, x1 + 3.x2 - 2.x3 - 3.x4 = 0,
```

```
3.x1 + 5.x2 - 2.x3 + 3.x4 = 0}
```

```
{x1 = -6.x4 - x3, x2 = 3.x4 + x3, x3 = x3, x4 = x4}
```

逆行列(MatrixInverse)

解説

線形代数の計算にはあらかじめ関数パッケージ(LinearAlgebra)を呼び出しておく.

```
> with(LinearAlgebra):
```

行列式(Determinant)

```
> A0 := Matrix([[x,y],[z,u]]);  
Determinant(A0);
```

$$A0 := \begin{bmatrix} x & y \\ z & u \end{bmatrix}$$
$$xu - yz$$

(4.1.1.1)

逆行列(MatrixInverse)

```
> A2 := MatrixInverse(A0);  
simplify(A0.A2);
```

$$A2 := \begin{bmatrix} \frac{u}{xu-yz} & -\frac{y}{xu-yz} \\ -\frac{z}{xu-yz} & \frac{x}{xu-yz} \end{bmatrix}$$
$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

(4.1.2.1)

その他の演算

随伴(Adjoint), 対角和(Trace), ジョルダン標準形(JordanForm)などもコマンドだけで求まる. 詳しくはヘルプ参照.

課題

1. 前節の課題2の連立方程式の係数行列の行列式を求めよ.
2. 前節の課題2の連立方程式の係数行列の逆行列を求めよ. またベクトル**b**に作用して解を求めよ.

解答例

1.

```
> with(LinearAlgebra):  
eqs := {x+y-z=2, 2*x-3*y+z=4, 4*x-y+3*z=1};  
A,b := GenerateMatrix(eqs, {x,y,z});  
Determinant(A);  
eqs := {x+y-z=2, 2*x-3*y+z=4, 4*x-y+3*z=1}
```

$$A, b := \begin{bmatrix} 1 & 1 & -1 \\ 2 & -3 & 1 \\ 4 & -1 & 3 \end{bmatrix}, \begin{bmatrix} 2 \\ 4 \\ 1 \end{bmatrix}$$
$$-20$$

(4.3.1.1)

2.

```
> MatrixInverse(A);  
simplify(MatrixInverse(A).b);
```

$$\begin{bmatrix} \frac{2}{5} & \frac{1}{10} & \frac{1}{10} \\ \frac{1}{10} & -\frac{7}{20} & \frac{3}{20} \\ -\frac{1}{2} & -\frac{1}{4} & \frac{1}{4} \end{bmatrix}$$
$$\begin{bmatrix} \frac{13}{10} \\ -\frac{21}{20} \\ -\frac{7}{4} \end{bmatrix}$$

(4.3.2.1)

固有値(EigenVectors)

解説

線形代数の計算にはあらかじめ関数パッケージ(LinearAlgebra)を呼び出しておく。

```
> with(LinearAlgebra):
```

固有値(EigenVectors)

固有値(Eigenvalues)と固有ベクトルを共に求めるにはEigenvectorsを使う。下の例では、固有値と固有ベクトルを変数 λ, v に代入している。

```
> A0 := Matrix(2, 2, [[1,2], [2,1]]);  
λ, v:=Eigenvectors(A0);
```

$$A0 := \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}$$
$$\lambda, v := \begin{bmatrix} -1 \\ 3 \end{bmatrix}, \begin{bmatrix} -1 & 1 \\ 1 & 1 \end{bmatrix} \quad (5.1.1.1)$$

固有ベクトルの取り出し(Column)

行列の行を要素とするベクトル生成Columnを使って、一番目の固有値に対応する固有ベクトルを取り出す。

```
> Column(v, 1);
```

$$\begin{bmatrix} -1 \\ 1 \end{bmatrix} \quad (5.1.2.1)$$

これを使って、固有値(λ)と固有ベクトル(v)の関係

$$A0.v = \lambda.v$$

を確認できる。

```
> A0.Column(v, 1);  
1[1]*Column(v, 1);
```

$$\begin{bmatrix} 1 \\ -1 \end{bmatrix}$$
$$\begin{bmatrix} 1 \\ -1 \end{bmatrix} \quad (5.1.2.2)$$

固有値ベクトルの規格化(Normalize)

固有値ベクトルを規格化するときには、以下のコマンドを使う。

```
> Normalize(Column(v, 1), Euclidean);
```

$$\begin{bmatrix} -\frac{1}{2}\sqrt{2} \\ \frac{1}{2}\sqrt{2} \end{bmatrix} \quad (5.1.3.1)$$

対角化

固有値ベクトルで作るベクトルを用いて、次のとおり行列の対角化ができる。

```
> Transpose(v).A0.v;
```

$$\begin{bmatrix} -2 & 0 \\ 0 & 6 \end{bmatrix} \quad (5.1.4.1)$$

その他の演算

随伴(Adjoint), 対角和(Trace), ジョルダン標準形(JordanForm)などもコマンドだけで求まる。詳しくはヘルプ参照。

課題

1. 行列 $A = \begin{bmatrix} 1 & -2 & 1 \\ -1 & 2 & 1 \\ 1 & 2 & 1 \end{bmatrix}$ の固有値を固有方程式 $|A - \lambda I| = 0$ を解いて求

めよ。EigenVectorsを用いて固有値と固有ベクトルを求めよ。固有値, 固有ベクトルの関係

を確認せよ。さらに、固有ベクトルをノルム1に規格化せよ。

2. 行列 $A = \begin{bmatrix} 2 & 0 & 1 \\ 0 & 3 & 0 \\ 1 & 0 & 2 \end{bmatrix}$ を対角化する変換行列 P を求め、対角化せよ。

解答例

1.

```
> A:=Matrix([[1,-2,1],[-1,2,1],[1,2,1]]);
E:=Matrix(3,3,shape=identity);
eq:=Determinant(A-x*E);
solve(eq,x);
```

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

$$eq := 4x^2 - x^3 - 8$$

$$2, 1 + \sqrt{5}, 1 - \sqrt{5}$$

(5.3.1.1)

```
> l,v:=Eigenvectors(A);
v1:=Column(v,3);
evalf(A.v1);
evalf(l[3].v1);
Normalize(v1,Euclidean);
evalf(Normalize(v1,Euclidean));
```

$$l, v := \begin{bmatrix} 1 + \sqrt{5} \\ 1 - \sqrt{5} \\ 2 \end{bmatrix}, \begin{bmatrix} \frac{(-3 + \sqrt{5})\sqrt{5}}{-5 + 3\sqrt{5}} - \frac{(-3 - \sqrt{5})\sqrt{5}}{-5 - 3\sqrt{5}} & 1 \\ -\frac{-5 + \sqrt{5}}{-5 + 3\sqrt{5}} & -\frac{-5 - \sqrt{5}}{-5 - 3\sqrt{5}} & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

$$v := \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 2. \\ 0. \\ 2. \end{bmatrix}$$

$$\begin{bmatrix} 2. \\ 0. \\ 2. \end{bmatrix}$$

$$\begin{bmatrix} \frac{1}{2}\sqrt{2} \\ 0 \\ \frac{1}{2}\sqrt{2} \end{bmatrix}$$

$$\begin{bmatrix} 0.7071067810 \\ 0. \\ 0.7071067810 \end{bmatrix}$$

(5.3.1.2)

2.

```
> A:=Matrix([[2,0,1],[0,3,0],[1,0,2]]);
l,v:=Eigenvectors(A);
MatrixInverse(v).A.v;
```

$$A := \begin{bmatrix} 2 & 0 & 1 \\ 0 & 3 & 0 \\ 1 & 0 & 2 \end{bmatrix}$$

$$l, v := \begin{bmatrix} 3 \\ 3 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 3 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

(5.3.2.1)