# 第1章　線形代数

## 1.1　行列・ベクトル生成 (MatrixVector)

[[[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):
```

### 1.1.1　ベクトルの生成 (Vector)

ベクトルの生成は，

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

$$v1 := \begin{bmatrix} x \\ y \end{bmatrix}$$

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]);
```

$$v2 := \begin{bmatrix} x & y & z \end{bmatrix}$$

新聞の囲み記事 (列）が column，劇場の座席（行）は row．

### 1.1.2　行列の生成 (Matrix)

標準的な行列 (Matrix) の生成は，

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

リストリストからの変換は，

```
> LL1 := [[1, 2], [3, 4]]:
> A1 := Matrix(LL1);
```

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

単位行列の生成は，

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

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

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);
```

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

### 1.1.3 縦横ベクトル・行列の簡易作成法 (MVShortcut)

かぎかっこ ($< \cdots >$) を使って，ベクトルあるいは行列を直感的に作ることが可能．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; #res:参照
```

$$v1 := \begin{bmatrix} x \\ y \end{bmatrix}$$

$$v2 := \begin{bmatrix} x & y & z \end{bmatrix}$$

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

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

### 1.1.4 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];
```

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

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];
```

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

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);
```

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

$\left[\left[\left[\text{課題}\right]\right]\right]$

1. 次の行列，ベクトルを作れ．

(a) $\begin{bmatrix} 3 & 3 & 3 \\ 3 & 3 & 3 \end{bmatrix}$, (b) $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$, (c) $\begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$, (d) $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix}$, (e) $\begin{bmatrix} x & x^2 \\ x^2 & x^3 \end{bmatrix}$, (f) $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$

## [[[解答例]]]

1. (a)
```
> Matrix([3,3,3],[3,3,3]]);
> Matrix(<3,3|3,3|3,3>);
```

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

そのまま直打ちしてもいいが，少し賢い生成法も記しておく．詳しくはヘルプ参照.

```
> Matrix(2,3,3);
```

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

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

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

(c)
```
> Vector[row]([1,2,3]);
> Vector(<1|2|3>):
```

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

(d)
```
> with(LinearAlgebra):
> V:=Vector[row]([1,2,3]);
> DiagonalMatrix(V);
```

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

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

(e)
```
> f:= (i,j) -> x^(i+j-1):
> Matrix(2,f);
```

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

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

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

```
> n:=3:
> f:=(i,j)->(i-1)*n+j;
> Matrix(3,3,f);
```

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

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

## 1.2 内積外積 (DotProduct,CrossProduct)

[[[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):
```

### 1.2.1 スカラーとのかけ算

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

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

### 1.2.2 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;
```

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

### 1.2.3 内積 (DotProduct, '.')

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

$$0$$

### 1.2.4 外積 (CrossProduct, '&x')

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

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

### 1.2.5 スカラー 3 重積

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

$$v3 := \begin{bmatrix} -1 \\ 2 \\ 1 \end{bmatrix}$$

$$16$$

### 1.2.6 転置 (Transpose, '&T')

は，行列 A の ij 成分 a[i,j] を a[j,i] にする．

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

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

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);
```

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

**[[[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) $A.B$, v) $B.A$, vi) $A.v$, vii) $v.A$, viii) $v$ の転置 (Transpose) を $A$ に左側から掛けよ, ix) $A^3$

2. 2 次元平面上で原点の周りの角度 t の回転行列は

   ```
   > Ar:=t->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);
   ```

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

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

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

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

i)–vi)
   ```

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

$$\begin{bmatrix} 7 & 11 \\ 15 & 19 \end{bmatrix}$$
$$\begin{bmatrix} -1 & -1 \\ -1 & -1 \end{bmatrix}$$
$$\begin{bmatrix} 2 & 2 \\ 3 & 5 \end{bmatrix}$$
$$\begin{bmatrix} 10 & 13 \\ 22 & 29 \end{bmatrix}$$
$$\begin{bmatrix} 11 & 16 \\ 19 & 28 \end{bmatrix}$$
$$\begin{bmatrix} 5 \\ 11 \end{bmatrix}$$

vii)

```
> v.A;
```

```
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．次元を合わすためには，v に転置 (Transpose) をかけて横ベクトルにしておく必要がある．

viii)

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

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

ix)

```
> A^3;
```

$$\begin{bmatrix} 37 & 54 \\ 81 & 118 \end{bmatrix}$$

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;
```

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

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

$$Ar := t \mapsto \begin{bmatrix} \cos(t) & -\sin(t) \\ \sin(t) & \cos(t) \end{bmatrix}$$

$$\begin{bmatrix} 1/2\sqrt{3} \\ 1/2 \end{bmatrix}$$

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

ii.) 2 つの関数を別々に計算.

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

$$\begin{bmatrix} 1/4\sqrt{2}\sqrt{3} - 1/4\sqrt{2} & -1/4\sqrt{2} - 1/4\sqrt{2}\sqrt{3} \\ 1/4\sqrt{2}\sqrt{3} + 1/4\sqrt{2} & 1/4\sqrt{2}\sqrt{3} - 1/4\sqrt{2} \end{bmatrix}$$

$$\begin{bmatrix} \cos\left(\frac{5}{12}\pi\right) & -\sin\left(\frac{5}{12}\pi\right) \\ \sin\left(\frac{5}{12}\pi\right) & \cos\left(\frac{5}{12}\pi\right) \end{bmatrix}$$

2 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));
```

$$\begin{bmatrix} -0.0000000002000000000 & 0.0 \\ 0.0 & -0.0000000002000000000 \end{bmatrix}$$

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

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

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

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

## 1.3 行列の基本操作，掃き出し (LUDecomposition)

[[[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):
```

### 1.3.1 行列の基本操作

行列の掃き出しに必要となる行列の基本操作は RowOperation, ColumnOperation を参照．

### 1.3.2 掃き出し法，LU 分解 (LUDecomposition)

掃き出し法の計算は，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>;
```

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

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

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

これに LU 分解をかける．それぞれ P(permutation，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>);
```

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

さらに被約階段行列 (row reduced echelonmatrix; 後退代入までおこなって，解まで求めた状態) を求めるには，output='R' を指定する．

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

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

### 1.3.3 階数 (Rank)

行列の性質の中でも特に重要な階数 (Rank) は次のコマンドで求められる．

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

$$2$$

[[[課題]]]

1. 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，次の行基本操作をおこない，階数を求め，コマンド LUDecomposition, 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=false でやってみて，うまくいけば true にかえる．

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)
$$\begin{cases} x + y - z &= 2 \\ 2x - 3y + z &= 4 \\ 4x - y + 3z &= 1 \end{cases}$$

   (ii)
$$\begin{cases} 2x + 4y - 3z &= 1 \\ 3x - 8y + 6z &= 58 \\ x - 2y - 9z &= 23 \end{cases}$$

   (iii)
$$\begin{cases} 1x - 10y - 3z - 7u &= 2 \\ 2x - 4y + 3z + 4u &= -3 \\ x - 2y + 6z + 5u &= -1 \\ x + 8y + 9z + 3u &= 5 \end{cases}$$

   (iv)
$$\begin{cases} x + y + z &= a + b + c \\ ax + by + cz &= ab + bc + ca \\ bc\,x + ca\,y + ab\,z &= 3\,abc \end{cases}$$

3. 次の連立方程式
$$\begin{cases} x_1 + 2x_2 - x_3 &= 0 \\ x_1 + x_2 + 3x_4 &= 0 \\ x_1 + 5x_2 - 2x_3 + 3x_4 &= 0 \\ x_1 + 3x_2 - 2x_3 - 3x_4 &= 0 \end{cases}$$
   を solve を使って，$x_1, x_2, x_3, x_4$ について解け．次に GenerateMatrix を使って，拡大係数行列にした後，LUDecomposition を用いて掃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;
```
$$A := \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

   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);
```

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

結果を最初の引数 (A) に上書きする option(inplace=true) 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,[3,1],-7,inplace=true);
```

$$\begin{bmatrix} 1 & 2 & 3 \\ 0 & -3 & -6 \\ 0 & -6 & -12 \end{bmatrix}$$

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

$$\begin{bmatrix} 1 & 2 & 3 \\ 0 & 1 & 2 \\ 0 & -6 & -12 \end{bmatrix}$$

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

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

LUDecomposition による結果と見比べる．

```
> A0:=Matrix([[1,2,3],[4,5,6],[7,8,9]]):
> LUDecomposition(A0);
```

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 4 & 1 & 0 \\ 7 & 2 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 2 & 3 \\ 0 & -3 & -6 \\ 0 & 0 & 0 \end{bmatrix}$$

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);
```

$$2$$

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});
```

$$eqs := \{x+y-z=2, 2\,x-3\,y+z=4, 4\,x-y+3\,z=1\}$$

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

i)

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

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

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

$$\begin{bmatrix} 1 & 0 & 0 & \frac{13}{10} \\ 0 & 1 & 0 & -\frac{21}{20} \\ 0 & 0 & 1 & -7/4 \end{bmatrix}$$

ii)

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

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

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

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

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');
```

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

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

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

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');
```

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

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

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

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

など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.

```
> 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});
```

$$eqs := \{x1 + x2 + 3\,x4 = 0, x1 + 2\,x2 - x3 = 0,$$
$$x1 + 3\,x2 - 2\,x3 - 3\,x4 = 0, 3\,x1 + 5\,x2 - 2\,x3 + 3\,x4 = 0\}$$
$$\{x1 = -6\,x4 - x3, x2 = 3\,x4 + x3, x3 = x3, x4 = x4\}$$

```
> A1,b:=GenerateMatrix(eqs,[x1,x2,x3,x4]);
> LUDecomposition(<A1|b>,output='R');
```

$$A1, b := \begin{bmatrix} 1 & 1 & 0 & 3 \\ 1 & 2 & -1 & 0 \\ 1 & 3 & -2 & -3 \\ 3 & 5 & -2 & 3 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

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

## 1.4 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):
```

### 1.4.1 行列式 (Determinant)

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

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

$$xu - yz$$

### 1.4.2 逆行列 (MatrixInverse)

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

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

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

### 1.4.3 その他の演算

随伴 (Adjoint) 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. 次の連立方程式の係数行列の行列式を求めよ．

(i)

$$\begin{cases} x + y - z & = 2 \\ 2x - 3y + z & = 4 \\ 4x - y + 3z & = 1 \end{cases}$$

(ii)

$$\begin{cases} 2x + 4y - 3z & = 1 \\ 3x - 8y + 6z & = 58 \\ x - 2y - 9z & = 23 \end{cases}$$

(iii)

$$\begin{cases} 1x - 10y - 3z - 7u & = 2 \\ 2x - 4y + 3z + 4u & = -3 \\ x - 2y + 6z + 5u & = -1 \\ x + 8y + 9z + 3u & = 5 \end{cases}$$

(iv)

$$\begin{cases} x+y+z & = a+b+c \\ ax+by+cz & = ab+bc+ca \\ bc\,x+ca\,y+ab\,z & = 3\,abc \end{cases}$$

2. 上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);
```

$$eqs := \{x+y-z=2, 2\,x-3\,y+z=4, 4\,x-y+3\,z=1\}$$

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

$$-20$$

2. 
```
> MatrixInverse(A); simplify(MatrixInverse(A).b);
```

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

$$\begin{bmatrix} \frac{13}{10} \\ -\frac{21}{20} \\ -7/4 \end{bmatrix}$$

## 1.5 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):
```

### 1.5.1 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 l,v に代入している．

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

$$A0 := \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}$$

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

### 1.5.2 固有ベクトルの取り出し (Column)

行列の列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);
```

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

これを使って，固有値 (l) と固有ベクトル (v) の関係

$$A_0.v = \lambda.v$$

が確認できる．

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

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

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

### 1.5.3 固有値ベクトルの規格化 (Normalize)

用意されているコマンドが確かめられる．

```
> ?Normalize;
```

一般的な内積を使って長さを規格化するには，以下のコマンドを使う．

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

$$\begin{bmatrix} -1/2\sqrt{2} \\ 1/2\sqrt{2} \end{bmatrix}$$

### 1.5.4　対角化

規格化された固有値ベクトルを用いて，次のとおり行列は対角化される．

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

$$\begin{bmatrix} -2 & 0 \\ 0 & 6 \end{bmatrix}$$

### 1.5.5　その他の演算

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
$$|A - \lambda E| = 0$$
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
$$A.v = \lambda v$$
を確認せよ．さらに，固有ベクトルを長さ 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);
```

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

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

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

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

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

$$v1\ :=\ \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 2.0 \\ 0.0 \\ 2.0 \end{bmatrix}$$

$$\begin{bmatrix} 2.0 \\ 0.0 \\ 2.0 \end{bmatrix}$$

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

$$\begin{bmatrix} 0.7071067810 \\ 0.0 \\ 0.7071067810 \end{bmatrix} \tag{1.1}$$

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

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

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

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