2024 年度 4 ターム「モデリングとシミュレーション」

レポートその 2-1

ベクトルの内積と距離の性質について

教育学部 第二類 技術・情報系コース 3 年
B220052 長田 麗生

提出期限：2025 年 1 月 8 日
提出日：2025 年 1 月 10 日

担当教員：田中秀幸 先生

1. はじめに

筆者はこのレポートを自らの学習状況を示すことを目的として書く。定義、証明、定式化、等の厳密さについては、本講義で求められていると筆者が考える水準に照らして、問題ないと考える範囲において放棄する。また線型代数はごく一般的な分野であるため、広く知られているであろうことについては参考文献を記載しない。

2. ベクトルに関する諸定義および定理

問題の解答にあたって必要に応じて参照できるように、定義や定理、式と説明を記載しておく。

2.1. ベクトルの定義

ベクトルとはいくつかの実数の組である。組をなす実数の数を次元と呼び、各実数のことを成分と呼ぶ。x_1, x_2 を成分に持つ 2 次元ベクトル $\boldsymbol{x}$ は次のように書き表す。

$$\boldsymbol{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

n 次元ベクトル全体の集合のことを $\mathbb{R}^n$ と表す。

※以後この章の中では、ことわりがなければ $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^2, a \in \mathbb{R}$ とする。

(2 次元)ベクトルは次の性質を満たす。

定義 2.1.1 (ベクトルの基本演算)**:**

$$\boldsymbol{x}+\boldsymbol{y}=\begin{pmatrix}x_1+y_1\\x_2+y_2\end{pmatrix} \qquad \text{(ベクトルの和)}$$

$$a\boldsymbol{x}=\boldsymbol{x}a=\begin{pmatrix}ax_1\\ax_2\end{pmatrix} \qquad \text{(ベクトルとスカラーの積)}$$

※特に $-1\boldsymbol{x}$ のことを $-\boldsymbol{x}$ と書き、$\boldsymbol{x}+(-\boldsymbol{y})$ を $\boldsymbol{x}-\boldsymbol{y}$ と表記する。

2.2. 行ベクトルと列ベクトル、転置

ベクトルは行列の特殊な場合として見ることができる。逆に言えば、ベクトルを拡張することで行列となる。ベクトルを行列に自然に拡張するために、ベクトルには次の2種類があると考える。すなわち、成分を縦方向に書き並べる**列ベクトル**と横方向に書き並べる**行ベクトル**である。次の例では $\boldsymbol{x}_{\text{行}}$ は行ベクトル、$\boldsymbol{x}_{\text{列}}$ は列ベクトルである。

$$\boldsymbol{x}_{\text{行}}=(x_1\ \ x_2)$$

$$\boldsymbol{x}_{\text{列}}=\begin{pmatrix}x_1\\x_2\end{pmatrix}$$

行ベクトルと列ベクトルは明確に区別しないでも議論ができると判断される場合には、明示的に宣言されないことがある。本レポートでもその方針を取る。(書き並べる方向を気にしないようなベクトルのことを**数ベクトル**と呼ぶ。)

行ベクトルと列ベクトルを区別するとき、それらには**転置**という演算を考えることができる。転置によって、行ベクトルを列ベクトルに、列ベクトルを行ベクトルに変換することができる。具体的には次のように定義される。

定義 2.2.1 (ベクトルの転置)**:** 行ベクトル、列ベクトルに対して転置を次のように定める。

$$(x_1\ \cdots\ x_n)^{\top}:=\begin{pmatrix}x_1\\\vdots\\x_n\end{pmatrix}$$

$$\begin{pmatrix}x_1\\\vdots\\x_n\end{pmatrix}^{\top}:=(x_1\ \cdots\ x_n)$$

このように、転置は転置記号「$\top$」ををベクトルの右上に表記することで表現されることが多い。

行ベクトルと列ベクトルを区別するとき、ベクトルの次元(成分の数)が等しくても、行ベクトルと列ベクトルの和は計算できないことに注意する必要がある。

転置に関するいくつかの性質を示しておく。

定理 2.2.1 (転置の性質): $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^2, a \in \mathbb{R}$ とする。このとき次が成り立つ。

(1) $\left(\boldsymbol{x}^\top\right)^\top = \boldsymbol{x}$

(2) $\boldsymbol{x} + \boldsymbol{y}^\top = \boldsymbol{x}^\top + \boldsymbol{y}^\top$

(3) $a\boldsymbol{x}^\top = a\boldsymbol{x}^\top$

証明: 概略のみ示す。ベクトルを成分表示すると、転置の定義およびベクトルの満たす性質から上記の性質を導くことができる。 ■

2.3. 内積

ベクトルに対して、内積と呼ばれる演算が次のように定義される。

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle \coloneqq x_1 y_1 + x_2 y_2$$

上の式は 2 次元の場合について述べたものだが、一般に、n 次元の場合は次のようになる。

定義 2.3.1 (ベクトルの内積): n 次元ベクトル$\boldsymbol{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}, \boldsymbol{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$ に対して、内積と呼ばれる演算が次のように定義される。

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle \coloneqq \sum_{i=1}^{n} x_i y_i$$

行ベクトルと列ベクトルを区別する場合、内積を(まだ説明していないものではあるが)行列積の表現と同様に、次のように文字を並べて表現することもできる。

定義 2.3.2 (ベクトルの内積の別の表現：行ベクトルと列ベクトルの積)**:** 行ベクトルと列ベクトルの積を次のように定義する。すなわち、列ベクトル $\boldsymbol{x}, \boldsymbol{y}$ に対して

$$\boldsymbol{x}^\top \boldsymbol{y} \coloneqq \langle \boldsymbol{x}, \boldsymbol{y} \rangle = \sum_{i=1}^{n} x_i y_i$$

$\boldsymbol{x}^\top$ が行ベクトル、$\boldsymbol{y}$ が列ベクトルであること、そして行ベクトルが左側、列ベクトルが右側にあることに注意。

ただし、左側が行ベクトル、右側が列ベクトルとなっている必要がある。そのため、列ベクトル $\boldsymbol{x}, \boldsymbol{y}$ の内積は $\boldsymbol{x}^\top \boldsymbol{y}$ のように書くことになる。

定理 2.3.1 (内積の性質)**:** ベクトルの内積に関して、次が成り立つ。

(1) $\boldsymbol{x}^\top \boldsymbol{y} = \boldsymbol{y}^\top \boldsymbol{x}$ (交換法則)

(2) $\boldsymbol{x}^\top \boldsymbol{x} = 0 \Leftrightarrow \boldsymbol{x} = \boldsymbol{0}$

(3) $\boldsymbol{x}^\top \boldsymbol{x} \geq 0$

(4) $(\boldsymbol{x} + \boldsymbol{y})^\top \boldsymbol{z} = \boldsymbol{x}^\top \boldsymbol{z} + \boldsymbol{y}^\top \boldsymbol{z}$ (分配法則)

(5) $(a\boldsymbol{x})^\top \boldsymbol{z} = a(\boldsymbol{x}^\top \boldsymbol{z})$ (結合法則)

証明**:**

(1) 定義より明らか。

(3) 内積の定義より、

$$\boldsymbol{x}^\top \boldsymbol{x} = \sum_{i=1}^{n} x_i^2$$

全ての成分 x_i について $x_i^2 \geq 0$ であるから

$$\boldsymbol{x}^\top \boldsymbol{x} = \sum_{i=1}^{n} x_i^2 \geq 0$$

(2) (3)において成分のうち一つでも 0 でないものがあれば総和は 0 より大きくなることから導かれる。

(4) 諸定義から具体的に計算すればよい。

(5) 諸定義から具体的に計算すればよい。 ■

2.4. 距離、大きさ

ベクトルに対してその**大きさ**、すなわちベクトルを矢印として見たときの始点から終点までの**距離**を考えることができる。図形的に解釈すると具体的に計算ができ、三平方の定理より次のように定義するのが良いことになる。

定義 2.4.1: ベクトル $\boldsymbol{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$ の**大きさ(ユークリッドノルム)**を次のように定義する。

$$\|\boldsymbol{x}\| := \sqrt{\sum_{i=1}^{n} x_i^2}$$

これは単にベクトルの始点と終点の距離を表すものだと考えて良い。上のような式で定義される距離を**ユークリッド距離**という。**距離**とはそれが満たすべき性質を満たすもののことであり、ユークリッド距離の他にいくつかの距離の定義が考えられる。つまりユークリッド距離は広い意味での距離のうちの一つである。そして、ユークリッド距離は日常生活における距離を表しており、最も直感的で馴染み深いものである。

内積の定義を思い出せば、ベクトルの大きさは内積の表現を用いて表せることがわかる。

$$\|\boldsymbol{x}\| = \langle \boldsymbol{x}, \boldsymbol{x} \rangle = \boldsymbol{x}^\top \boldsymbol{x}$$

定理 2.4.1 (ユークリッド距離の性質): 上のように定義したベクトルの大きさに関して次の性質が成り立つ。$\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n, a \in \mathbb{R}$ とするとき、

(1) $\|\boldsymbol{x}\| = 0 \Leftrightarrow \boldsymbol{x} = \boldsymbol{0}$ (独立性)

(2) $\|a\boldsymbol{x}\| = |a|\|\boldsymbol{x}\|$ (斉次性)

(3) $\|\boldsymbol{x} + \boldsymbol{y}\| \leq \|\boldsymbol{x}\| + \|\boldsymbol{y}\|$ (三角不等式)

証明:

(1) 定理 2.3.1 (2) と $\boldsymbol{x}^\top \boldsymbol{x} = \|\boldsymbol{x}\|^2$ から導ける。

(2) 定義通りに計算すればよい。

(3) 後の 問題 3.2 の解答として示す。 ■

2.5. コーシー・シュワルツの不等式

定理 2.5.1 (コーシー・シュワルツの不等式): ベクトル$\boldsymbol{x}, \boldsymbol{y}$に対して次の関係が成り立つ。

$$\|\boldsymbol{x}\|\|\boldsymbol{y}\| \geq |\langle \boldsymbol{x}, \boldsymbol{y} \rangle|$$

証明:

$$\|\boldsymbol{x}-\boldsymbol{y}\|^2 = \|\boldsymbol{x}\|^2 + \|\boldsymbol{y}\|^2 - 2\langle \boldsymbol{x}, \boldsymbol{y} \rangle \qquad \text{(距離の定義、内積の定義より)}$$
$$\|\boldsymbol{x}-\boldsymbol{y}\|^2 = \|\boldsymbol{x}\|^2 + \|\boldsymbol{y}\|^2 - 2\|\boldsymbol{x}\|\|\boldsymbol{y}\| \cos\theta \text{ (余弦定理)}$$

上の式から下の式を辺々引くと、以下の式が成立する。

$$\|\boldsymbol{x}\|\|\boldsymbol{y}\| \cos\theta = \langle \boldsymbol{x}, \boldsymbol{y} \rangle$$

$|\cos\theta| \leq 1$ より次が成立。

$$\|\boldsymbol{x}\|\|\boldsymbol{y}\| \geq |\langle \boldsymbol{x}, \boldsymbol{y} \rangle|$$

■

2.6. 余弦定理

定理 2.6.1 (余弦定理): 2 つのベクトル $\boldsymbol{x}, \boldsymbol{y}$ とそれらのなす角 θ の間には次の関係が成り立つ。

$$\|\boldsymbol{x}-\boldsymbol{y}\|^2 = \|\boldsymbol{x}\|^2 + \|\boldsymbol{y}\|^2 - 2\|\boldsymbol{x}\|\|\boldsymbol{y}\| \cos\theta$$

証明: (線型代数の本題から遠いため省略。幾何的に証明できる。) ■

上の式はベクトルを用いて表現されているが、すべてその大きさとしてしか登場していないため、$\|\boldsymbol{x}\|, \|\boldsymbol{y}\|, \|\boldsymbol{x}-\boldsymbol{y}\|$ をそれぞれ三角形の 3 辺の長さ a, b, c と見ればベクトル特有の演算は含まれなくなる。そのため余弦定理は初等幾何的に証明できる。本レポートが線型代数を扱うという理由のもと、ここであえてベクトルを用いた表現をしているに過ぎない。

3. 問題と解答

本レポートで解答すべき問題は、第 5 回資料中の問題(1), (2)である。講義資料の問題を次のように改めて整理しておく。

- 講義資料中の「問題(1)」 の 5 つの式を 問題 3.1 の(1)〜(5)とする。
- 講義資料中の「問題(2)」 を 問題 3.2, 問題 3.3 とする。

以下、問題と解答である。

問題 3.1: $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in \mathbb{R}^2, a \in \mathbb{R}$ とする。次の(1)〜(5)を証明せよ。

(1) $(\boldsymbol{x}+\boldsymbol{y})^\top = \boldsymbol{x}^\top + \boldsymbol{y}^\top$

(2) $(a\boldsymbol{x})^\top \boldsymbol{y} = a(\boldsymbol{x}^\top \boldsymbol{y})$

(3) $(a\boldsymbol{x}+\boldsymbol{y})^\top \boldsymbol{z} = a(\boldsymbol{x}^\top \boldsymbol{z}) + \boldsymbol{y}^\top \boldsymbol{z}$

(4) $\boldsymbol{x}^\top (a\boldsymbol{y}+\boldsymbol{z}) = a(\boldsymbol{x}^\top \boldsymbol{y}) + \boldsymbol{x}^\top \boldsymbol{z}$

(5) $\|\boldsymbol{x}-\boldsymbol{y}\|^2 = \|\boldsymbol{x}\|^2 + \|\boldsymbol{y}\|^2 - 2\boldsymbol{x}^\top \boldsymbol{y}$

証明: (1)〜(5)のどれも、ベクトルの基本的な性質に関するものである。ベクトルが満たすべき性質をもとに、成分を具体的に計算することで示すことができる。

(1)

$$
\begin{aligned}
(\boldsymbol{x}+\boldsymbol{y})^\top &= \left(\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \right)^\top \\
&= \begin{pmatrix} x_1 + y_1 \\ x_2 + y_2 \end{pmatrix}^\top \\
&= (x_1 + y_1 \quad x_2 + y_2) \\
&= (x_1 \quad x_2)^\top + (y_1 \quad y_2)^\top \\
&= \boldsymbol{x}^\top + \boldsymbol{y}^\top
\end{aligned}
$$

(2)

$$
\begin{aligned}
(a\boldsymbol{x})^\top \boldsymbol{y} &= a\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}^\top \boldsymbol{y} \\
&= \begin{pmatrix} x_1 + y_1 \\ x_2 + y_2 \end{pmatrix}^\top \\
&= (x_1 + y_1 \;\; x_2 + y_2) \\
&= (x_1 \;\; x_2)^\top + (y_1 \;\; y_2)^\top \\
&= \boldsymbol{x}^\top + \boldsymbol{y}^\top
\end{aligned}
$$

(3) (上と同様に示せる)

(4) (上と同様に示せる)

(5)

$$
\begin{aligned}
\|\boldsymbol{x} - \boldsymbol{y}\|^2 &= (\boldsymbol{x} - \boldsymbol{y})^\top (\boldsymbol{x} - \boldsymbol{y}) && \text{(距離の定義)} \\
&= (\boldsymbol{x} - \boldsymbol{y})^\top \boldsymbol{x} - (\boldsymbol{x} - \boldsymbol{y})^\top \boldsymbol{y} && \text{(内積の分配法則)} \\
&= \{\boldsymbol{x}^\top \boldsymbol{x} - \boldsymbol{y}^\top \boldsymbol{x}\} - \{\boldsymbol{x}^\top \boldsymbol{y} - \boldsymbol{y}^\top \boldsymbol{y}\} && \text{(内積の分配法則)} \\
&= \{\|\boldsymbol{x}\|^2 - \boldsymbol{x}^\top \boldsymbol{y}\} - \{\boldsymbol{x}^\top \boldsymbol{y} - \|\boldsymbol{y}\|^2\} && \text{(距離の定義, 内積の交換法則)} \\
&= \|\boldsymbol{x}\|^2 + \|\boldsymbol{y}\|^2 - 2\boldsymbol{x}^\top \boldsymbol{y}
\end{aligned}
$$

■

問題 3.2: コーシー・シュワルツの不等式を用いて、以下の三角不等式を証明せよ。また、三角不等式について、三角形を書き図形的説明を与えよ。

$$\|\boldsymbol{x} + \boldsymbol{y}\| \leq \|\boldsymbol{x}\| + \|\boldsymbol{y}\|$$

証明:

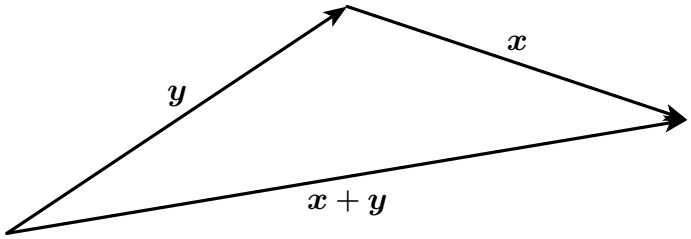
$$
\begin{aligned}
\|\boldsymbol{x} + \boldsymbol{y}\|^2 &= (\boldsymbol{x} + \boldsymbol{y})^\top (\boldsymbol{x} + \boldsymbol{y}) \\
&= \|\boldsymbol{x}\|^2 + \|\boldsymbol{y}\|^2 + 2\boldsymbol{x}^\top \boldsymbol{y}
\end{aligned}
$$

ここで、コーシー・シュワルツの不等式 $\|\boldsymbol{x}\|\|\boldsymbol{y}\| \geq |\boldsymbol{x}^\top \boldsymbol{y}|$ を用いて

$$
\begin{aligned}
\|\boldsymbol{x} + \boldsymbol{y}\|^2 &\leq \|\boldsymbol{x}\|^2 + \|\boldsymbol{y}\|^2 + 2\|\boldsymbol{x}\|\|\boldsymbol{y}\| \\
&= (\|\boldsymbol{x}\| + \|\boldsymbol{y}\|)^2
\end{aligned}
$$

よって $\|\boldsymbol{x} + \boldsymbol{y}\| \leq \|\boldsymbol{x}\| + \|\boldsymbol{y}\|$

図形的には次のように説明できる。下のように三角形を書くと、この不等式は「2辺の長さの和 $\|x\| + \|y\|$ は、残りの1辺の長さ $\|\boldsymbol{x} + \boldsymbol{y}\|$ より大きい」ことを意味していると言える。



■

問題 3.3: 以下について示せ。

$$|\|\boldsymbol{x}\| - \|\boldsymbol{y}\|| \leq \|\boldsymbol{x} + \boldsymbol{y}\|$$

証明: 与えられた不等式は次のように同値変形できる。

$$\begin{aligned} & |\|\boldsymbol{x}\| - \|\boldsymbol{y}\|| \leq \|\boldsymbol{x} + \boldsymbol{y}\| \\ \Leftrightarrow\ & \|\boldsymbol{x}\| - \|\boldsymbol{y}\| \leq \|\boldsymbol{x} + \boldsymbol{y}\| \quad \wedge \quad \|\boldsymbol{y}\| - \|\boldsymbol{x}\| \leq \|\boldsymbol{x} + \boldsymbol{y}\| \\ \Leftrightarrow\ & \|\boldsymbol{x}\| \leq \|\boldsymbol{x} + \boldsymbol{y}\| + \|\boldsymbol{y}\| \quad \wedge \quad \|\boldsymbol{y}\| \leq \|\boldsymbol{x} + \boldsymbol{y}\| + \|\boldsymbol{x}\| \end{aligned}$$

最後の式のうち左側は「$\boldsymbol{x}$ の長さは、他の2辺 $\boldsymbol{x} + \boldsymbol{y}, \boldsymbol{y}$ の長さの和より小さい」ことを表している。このことに注目し、三角不等式 $\|\boldsymbol{x} + \boldsymbol{y}\| \leq \|\boldsymbol{x}\| + \|\boldsymbol{y}\|$ を次のように利用する。

$$\begin{aligned} \|\boldsymbol{x}\| &= \|(\boldsymbol{x} + \boldsymbol{y}) + (-\boldsymbol{y})\| \\ &\leq \|\boldsymbol{x} + \boldsymbol{y}\| + \|-\boldsymbol{y}\| \qquad \text{(三角不等式より)} \\ &= \|\boldsymbol{x} + \boldsymbol{y}\| + \|\boldsymbol{y}\| \end{aligned}$$

これで最後の式のうち左側を示せた。右側についても同様に示せるので、与えられた不等式を証明できた。 ■

4. 実習の記録

4.1. 実習(1), (2) Scilab でベクトルの表示

```
1  mode(0);
2
3  // ベクトル
4  x = [sqrt(3);
5      1];
6
7  // 要素
8  x1 = x(1);
9  x2 = x(2);
10
11 // プロットするために，以下を定義.
12 ox1 = [0,x1];
13 ox2 = [0,x2];
14
15 // プロット
16 scf(1);
17 plot(ox1,ox2,'b-');
18 //plot(x1,x2,'b-x');
19
20 isoview;
```

コード 1: ベクトルのプロット

`'b-'` は「青色(blue)の実線」を意味する。コード 1 の実行結果が 図 1 である。また、18 行目のコメントアウトを外して実行した結果が 図 2 である。コメントアウトを外すと座標 `(x_1, x_2)` にバツ印が描画される。

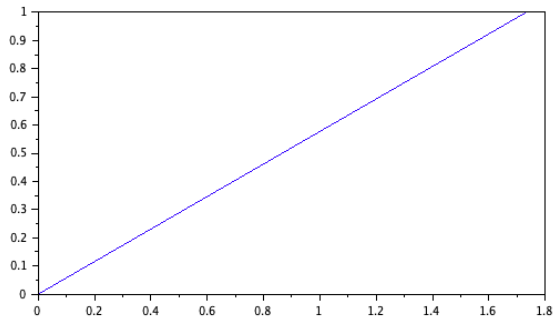


図 1: ベクトルのプロット(バツ印なし)

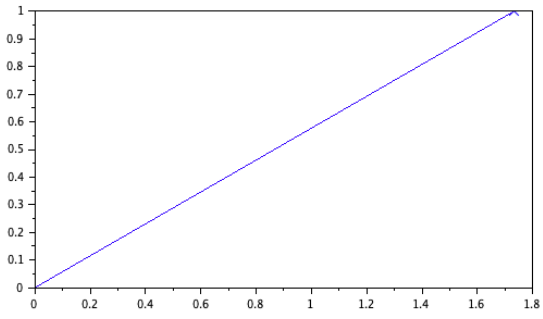


図 2: ベクトルのプロット(バツ印あり)

実習２では複数のベクトルをプロットした。コード２がプログラムでその実行結果が図 3 である。ベクトルのプロットを簡潔に記述するために、ベクトル `p` と色 `c` を指定できる関数 `plotvec(p, c)` を定義している。

```
mode(0);

function [] = plotvec(p, c)
    plot([0, p(1)], [0, p(2)], c+'-');
    plot(p(1), p(2), c+'x');
endfunction

// 一つ目のベクトル
x = [sqrt(3);
    1];

// 二つ目のベクトル
y = [1;
    sqrt(3)];

// プロット
scf(1);

plotvec(x, 'b');
plotvec(y, 'g');

a = gca();
a.isoview = 'on';
```

コード 2: 2 つのベクトルのプロット(30°, 60°)

さらに授業内で「大きさが 3, 角度が 45°, 135° のベクトルをプロットせよ」という課題が与えられた。コード 3 がプログラム、図 4 が実行結果である。

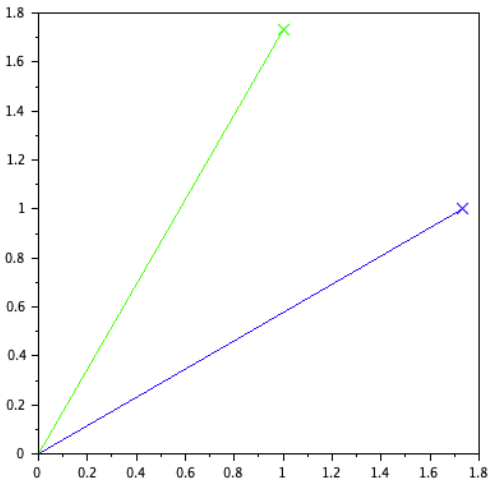


図 3: 2 つのベクトルのプロット(30°, 60°)

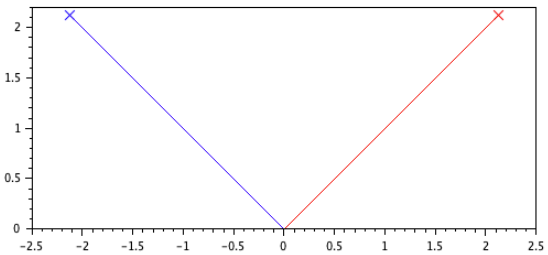


図 4: 2 つのベクトルのプロット(45°, 135°)

```
mode(0);
// プロット
scf(1);

function [] = plotvec(p, c)
    plot([0, p(1)], [0, p(2)], c+'-');
    plot(p(1), p(2), c+'x');
endfunction

// 大きさが 3, 角度が 45 度、135 度のベクトル
u = 3*[cos(1/4 * %pi);
       sin(1/4 * %pi)];
v = 3*[cos(3/4 * %pi);
       sin(3/4 * %pi)];
//plotvec(u, 'r');
//plotvec(v, 'b');

a = gca();
a.isoview = 'on';
```

コード 3: 大きさが 3, 角度が 45°, 135° のベクトルプロット

4.2. 実習(3) Scilab でベクトルの内積を計算する

内積の定義から、for ループにより内積が計算できる。また、Scilab ではベクトルの内積を行列の積として計算することができる。つまり必要に応じて転置を利用すれば次のように完結に記述することができる。 列ベクトル `x`, `y` に対してその内積は `x' * y` で書ける。

```
mode(0);

n = 2;
x = rand(n,1,'normal');
y = rand(n,1,'normal');

// for 文で計算
xy = 0;
for k = 1:n,
  xy = xy + x(k)*y(k);
end
xy

// Scilab の積で計算
xy_ = x'*y

//--------------------------------------
// 距離の計算
//--------------------------------------
// for 文で計算
xx = 0;
for k = 1:n,
  xx = xx + x(k)*x(k);
end
xx

// Scilab の積で計算
xx_ = x'*x

// y でも行ってみよう.
// for 文で計算
yy = 0;
for k = 1:n,
  yy = yy + y(k)*y(k);
end
yy

// Scilab の積で計算
yy_ = y'*y
```

コード 4: ベクトルの内積、距離の計算

```
xy =
  1.4891584
xy_ =
  1.4891584
xx =
  0.9327533
xx_ =
  0.9327533
yy =
  4.2363762
yy_ =
  4.2363762
```

コード 5: コード 4 の実行結果

4.3. 実習(4) 角度について

余弦定理(定理 2.6.1)

$$\|\boldsymbol{x}-\boldsymbol{y}\|^2 = \|\boldsymbol{x}\|^2 + \|\boldsymbol{y}\|^2 - 2\|\boldsymbol{x}\|\|\boldsymbol{y}\|\cos\theta$$

を用いることで、2 つのベクトルがなす角を計算することができる。具体的には、左辺は

$$\|\boldsymbol{x}\|^2 + \|\boldsymbol{y}\|^2 - 2\boldsymbol{x}^\top\boldsymbol{y}$$

と内積で表現できるから、余弦定理を次のように書き直すことができる。

$$\boldsymbol{x}^\top\boldsymbol{y} = \|\boldsymbol{x}\|\|\boldsymbol{y}\|\cos\theta$$

よって、

$$\cos\theta = \frac{\boldsymbol{x}^\top\boldsymbol{y}}{\|\boldsymbol{x}\|\|\boldsymbol{y}\|}$$

$$\sin\theta = \sqrt{1-\left(\frac{\boldsymbol{x}^\top\boldsymbol{y}}{\|\boldsymbol{x}\|\|\boldsymbol{y}\|}\right)^2}$$

$$\tan\theta = \frac{\sin\theta}{\cos\theta}$$

$$\theta = \arctan\frac{\boldsymbol{x}^\top\boldsymbol{y}}{\|\boldsymbol{x}\|\|\boldsymbol{y}\|}$$

これをもとに$\cos\theta, \sin\theta, \tan\theta, \theta$ を求めるプログラムが コード 6 である。

```
mode(0);

x = [sqrt(3);
    1];

y = [1;
    sqrt(3)];

//--------------------------------------
// 内積の計算
//--------------------------------------
xy = x'*y
xx = x'*x
yy = y'*y

//--------------------------------------
// cos と sin を計算(プラスだけ取る)
//--------------------------------------
cosTheta = xy/(sqrt(xx)*sqrt(yy))
sinTheta = sqrt(1 - cosTheta^2)

// tan から，角度を計算(プラスだけ取る)
tanTheta = sinTheta/cosTheta
theata_rad = atan(tanTheta)
theata_deg = (180/%pi)*theata_rad
```

コード 6: 角度を求めるプログラム

```
xy =
  3.4641016
xx =
  4.0000000
yy =
  4.0000000
cosTheta =
  0.8660254
sinTheta =
  0.5000000
tanTheta =
  0.5773503
theata_rad =
  0.5235988
theata_deg =
  30.000000
```

コード 7: コード 6 の実行結果

4.4. 実習(5) 定理の確認

2 次元の正規分布の乱数を使って、以下を確かめなさい。

(1) コーシー・シュワルツの不等式
(2) 三角不等式

それぞれ、乱数で生成したベクトルに対して複数回計算してその結果を表示し、おかしければ報告するプログラムを動作させた。

```
//---------------------------------------
// コーシーシュワルツの不等式を確かめる.
//---------------------------------------
mode(0);

for k = 1:10,
  // 乱数でベクトルを発生
  x = rand(2,1,'normal');
  y = rand(2,1,'normal');

  // ||x||||y||の計算
  xx = x'*x;
  x_ = sqrt(xx);

  yy = y'*y;
  y_ = sqrt(yy);

  xy_ = x_*y_;

  // <x,y> の計算
  xy = abs(x'*y);

  xy_xy = xy_ - xy
  if (xy_xy < 0),
      disp('Cauthy Schwarz うそつき! ');
  end

end

disp('finished!');
```

コード 8: コーシー・シュワルツの不等式が成り立っていることの確認

```
xy_xy =
  0.0340170
xy_xy =
  0.1819252
xy_xy =
  0.1448247
xy_xy =
  1.4660353
xy_xy =
  1.4258971
xy_xy =
  0.3140829
xy_xy =
  1.6179960
xy_xy =
  0.4472560
xy_xy =
  0.1325598
xy_xy =
  1.0374417
  "finished!"
```

コード 9: コード 6 の実行結果

```
//--------------------------------------
// 三角不等式を確かめる.
//--------------------------------------
mode(0);

for k = 1:10,
  // 乱数でベクトルを発生
  x = rand(2,1,'normal');
  y = rand(2,1,'normal');

  // a = ||x|| + ||y|| - ||x+y|| >=0 を確かめる
  a = sqrt(x'*x) + sqrt(y'*y) - sqrt((x+y)'*(x+y))
  if (a < 0),
      disp('Cauthy Schwarz うそつき! ');
  end

end

disp('finished!');
```

コード 10: 三角不等式が成り立っていることの確認

```
a =
  0.0725710
a =
  1.4519680
a =
  0.4522284
a =
  1.7176327
a =
  2.3055133
a =
  0.1755186
a =
  0.0207109
a =
  0.3444386
a =
  0.0010523
a =
  1.0584938
  "finished!"
```

コード 11: コード 6 の実行結果

5. まとめ

- ベクトルの内積や距離の性質について学習した
- 距離を測るのに線型代数が役に立つ
- ベクトルの内積を計算することで、距離がわかる
- 具体的な計算は計算機にさせればよい

6. 参考文献

1. 数学の景色『列ベクトルと行ベクトルの定義と違い』(2025-01-08 閲覧) https://mathlandscape.com/column-row-vector/
2. 数学の景色『数ベクトルの定義と数ベクトルにおけるノルム・内積』(2025-01-08 閲覧) https://mathlandscape.com/numerical-vector/

2024 年度 4 ターム「モデリングとシミュレーション」

レポートその 2-2

行列およびベクトルの積について

教育学部 第二類 技術・情報系コース 3 年
B220052 長田 麗生

提出期限：2025 年 1 月 8 日
提出日：2025 年 1 月 10 日

担当教員：田中秀幸 先生

1. はじめに

筆者はこのレポートを自らの学習状況を示すことを目的として書く。定義、証明、定式化、等の厳密さについては、本講義で求められていると筆者が考える水準に照らして、問題ないと考える範囲において放棄する。また線型代数はごく一般的な分野であるため、広く知られているであろうことについては参考文献を記載しない。

2. 行列に関する諸定義および定理

問題の解答にあたって必要に応じて参照できるように、定義や定理、式と説明を記載しておく。

2.1. 行列の定義

ベクトルが一方向に実数を並べたものであったのに対して、行列は実数を行と列に並べたものである。例えば次は 2 行 2 列の行列である。

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

本レポートでは、2 次元の場合について議論する。

定義 2.1.1 (行列の基本演算): $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}, B = \begin{pmatrix} e & f \\ g & h \end{pmatrix}$ のとき、

$$A + B = \begin{pmatrix} a+e & b+f \\ c+g & d+h \end{pmatrix} \qquad \text{(行列の和)}$$

$$kA = Ak = \begin{pmatrix} ka & kb \\ kc & kd \end{pmatrix} \qquad \text{(行列とスカラーの積)}$$

定義 2.1.2 (行列の成分):

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

とすると、

$$A_{11} = a, A_{12} = b, A_{21} = c, A_{22} = d$$

である。

2.2. 行列とベクトルの積

定義 2.2.1 (行列とベクトルの積): $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}, x = \boldsymbol{x}$ のとき、それらの積は次のように計算される。

$$A\boldsymbol{x} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} := \begin{pmatrix} ax_1 + bx_2 \\ cx_1 + dx_2 \end{pmatrix}$$

2.3. 連立方程式の例

問題 2.3.1 (連立方程式の例): 鶏と豚が全部で 10 匹います。全部の足を合計すると 28 本です。鶏と豚はそれぞれ何匹？

この問題は次の連立方程式で表現できる。

$$x + y = 10$$
$$2x + 4y = 28$$

この連立方程式は行列とベクトルを用いて次のように表現できる。

$$\begin{pmatrix} 1 & 1 \\ 2 & 4 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 10 \\ 28 \end{pmatrix}$$

行列で定義される基本的な演算を用いて、この連立方程式を解くことができる。手順通りに計算できて表現も単純であるため、行列で表現することは計算機に解かせる上で都合が良い。解き方の具体的な内容については今後の講義で学習する。

2.4. 行列積

定義 2.4.1 (行列積): 2 行 2 列の行列の積は次のように計算される。

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} e & f \\ g & h \end{pmatrix} := \begin{pmatrix} ae + bg & af + bh \\ ce + dg & cf + dh \end{pmatrix}$$

和の場合と異なり、積は単に対応する成分を掛け合わせるだけではない。少し複雑な計算だがこの計算が基本的なものとして定義されることによって、線型代数はその表現力を高めることができ、様々な分野に応用されている。

積の各成分に注目すると、それぞれベクトルの内積の形になっていることがわかる。すなわち、次のようにベクトルの内積で表現することができる。

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} e & f \\ g & h \end{pmatrix} = \begin{pmatrix} \begin{pmatrix} a & b \end{pmatrix} \begin{pmatrix} e \\ g \end{pmatrix} & \begin{pmatrix} a & b \end{pmatrix} \begin{pmatrix} f \\ h \end{pmatrix} \\ \begin{pmatrix} c & d \end{pmatrix} \begin{pmatrix} e \\ g \end{pmatrix} & \begin{pmatrix} c & d \end{pmatrix} \begin{pmatrix} f \\ h \end{pmatrix} \end{pmatrix}$$

行列積は非可換である。すなわち、$AB = BA$ が成り立つとは限らない。例えば次の例を計算してみよう。

問題 2.4.1 (行列積の例): $A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}, B = \begin{pmatrix} 1 & 10 \\ 10 & 1 \end{pmatrix}$ とする。AB と BA を計算し、値が異なることを確認せよ。

証明:

$$
\begin{aligned}
AB &= \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} 1 & 10 \\ 10 & 1 \end{pmatrix} \\
&= \begin{pmatrix} 1\cdot 1 + 2\cdot 10 & 1\cdot 10 + 2\cdot 1 \\ 3\cdot 1 + 4\cdot 10 & 3\cdot 10 + 4\cdot 1 \end{pmatrix} \\
&= \begin{pmatrix} 21 & 12 \\ 43 & 31 \end{pmatrix} \\
BA &= \begin{pmatrix} 1 & 10 \\ 10 & 1 \end{pmatrix} \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \\
&= \begin{pmatrix} 1\cdot 1 + 10\cdot 3 & 1\cdot 2 + 10\cdot 4 \\ 10\cdot 1 + 1\cdot 3 & 10\cdot 2 + 1\cdot 4 \end{pmatrix} \\
&= \begin{pmatrix} 31 & 42 \\ 13 & 24 \end{pmatrix}
\end{aligned}
$$

よって、$AB = \begin{pmatrix} 21 & 12 \\ 43 & 31 \end{pmatrix}$ であり、$BA = \begin{pmatrix} 31 & 42 \\ 13 & 24 \end{pmatrix}$ である。これらは異なるため、行列積は非可換である。 ■

この例は Scilab による計算も行っており、詳しくは 第 3.1.節 で述べている。

3. 実習の記録

3.1. 実習(1) Scilab における行列積の計算

Scilab による行列の積の計算の練習として 問題 2.4.1 を Scilab で計算した。Scilab では行列の積は * で表現する。また、行列の和は + で表現する。これはベクトルの回で学んだ演算と同じである。(なぜならば、行ベクトルや列ベクトルは単に行列とみなされているからである。)

実行したコードを コード 1 に、その実行結果を コード 2 に示す。

```
mode(0);

A = [1, 2;
     3, 4]
B = [1, 10;
     10, 1]

// 行列の和
A_B = A+B

// 行列のスカラー倍
A2 = A*2

// 行列の積
AB = A*B
BA = B*A
```

コード 1: Scilab による行列計算

```
A = [2x2 double]
  1.   2.
  3.   4.
B = [2x2 double]
  1.    10.
  10.   1.
A_B = [2x2 double]
  2.    12.
  13.   5.
A2 = [2x2 double]
  2.   4.
  6.   8.
AB = [2x2 double]
  21.   12.
  43.   34.
BA = [2x2 double]
  31.   42.
  13.   24.
```

コード 2: コード 1 の実行結果

3.2. 実習(2) Scilab で行(／列)だけ得る方法

Scilab では `A(1, 2)` とすれば行列 `A` の 1 行 2 列目の要素を取得できる。

カッコの中には実数だけでなくベクトルを指定することもできる。ベクトルを指定することで、複数の成分を取得することができる。例えば `A(1, 1:2)` とすれば行列 `A` の 1 行 1 列目から 2 列目までの要素からなる行列を取得できる。ここで `1:2` は`[1, 2]` の省略形である。

「i 行から j 行まで」のように具体的に範囲ではなく、全ての行を指定したいときには、: を使う。例えば `A(1, :)` とすれば行列 `A` の 1 行目の全ての要素からなる行列を取得できる。

実行したコードを コード 3 に、その実行結果を コード 4 に示す。

```
mode(0);

X = rand(3,3,'normal')

// 1 行目だけ
X(1,:)

// 2 行目だけ
X(2,:)

// 1 行目と 2 行目
X(1:2,:)

// 再度表示
X

// 1 列目だけ
X(:,1)

// 2 列目だけ
X(:,2)

// 1 列目と 2 列目
X(:, 1:2)

// 再度表示
X

// 1,2 行と 1,2 列
X(1:2, 1:2)
```

コード 3: Scilab による行列計算

```
X = [3x3 double]
   -0.2551124  -0.1158131  -0.6888704
   -0.1468231  -0.6145392   1.0452423
   -0.3398349  -0.5835608  -0.3663346
ans = [1x3 double]
   -0.2551124  -0.1158131  -0.6888704
ans = [1x3 double]
   -0.1468231  -0.6145392   1.0452423
ans = [2x3 double]
   -0.2551124  -0.1158131  -0.6888704
   -0.1468231  -0.6145392   1.0452423
X = [3x3 double]
   -0.2551124  -0.1158131  -0.6888704
   -0.1468231  -0.6145392   1.0452423
   -0.3398349  -0.5835608  -0.3663346
ans = [3x1 double]
   -0.2551124
   -0.1468231
   -0.3398349
ans = [3x1 double]
   -0.1158131
   -0.6145392
   -0.5835608
ans = [3x2 double]
   -0.2551124  -0.1158131
   -0.1468231  -0.6145392
   -0.3398349  -0.5835608
X = [3x3 double]
   -0.2551124  -0.1158131  -0.6888704
   -0.1468231  -0.6145392   1.0452423
   -0.3398349  -0.5835608  -0.3663346
ans = [2x2 double]
   -0.2551124  -0.1158131
   -0.1468231  -0.6145392
```

コード 4: コード 3 の実行結果

3.3. 実習(3) 3 種類の方法で行列積を計算する

行列の積の計算方法について理解を深めるため、3 種類の方法で行列積を計算する。1 つ目は Scilab の * を使用する方法、2 つ目は各行、各列の内積(inner product)を計算する方法、3 つ目は Scilab の for で計算する方法である。実行したコードを コード 5 に、その実行結果を コード 6 に示す。

```
mode(0);

n = 2;
X = rand(n,n,'normal')
Y = rand(n,n,'normal')

// Scilab の "*" を使用
Z = X*Y

// Scilab の内積(inner product)を使用
Zi = zeros(n,n); // 初期化
for i = 1:n,
  for j = 1:n,
    Zi(i,j) = X(i,:)*Y(:,j);
  end
end
Zi

// Scilab の for で計算
Zf = zeros(n,n); // 初期化
for i = 1:n,
  for j = 1:n,
    for k = 1:n,
      Zf(i,j) = Zf(i,j) + X(i,k)*Y(k,j);
    end
  end
end
Zf
```

コード 5: Scilab による行列計算

```
X = [2x2 double]
  -1.8546736  -0.2316
  1.1956471   0.3763408
Y = [2x2 double]
  -0.4901191   0.0932712
  -1.01017    -1.1833318
Z = [2x2 double]
  1.1429664   0.1010721
  -0.9661777  -0.3338166
Zi = [2x2 double]
  1.1429664   0.1010721
  -0.9661777  -0.3338166
Zf = [2x2 double]
  1.1429664   0.1010721
  -0.9661777  -0.3338166
```

コード 6: コード 5 の実行結果

4. まとめ

- 行列とその基本演算について学習した
- 行列で連立方程式を表現する方法を学んだ
- Scilab で行列積を計算する方法、行(／列)だけ得る方法を学んだ

5. 参考文献