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。

$$
\begin{aligned}
x + y &= 10 \\
2x + 4y &= 28
\end{aligned}
$$

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