

# プログラム演習 (2) 配列を使う

## 1 はじめに

科学技術計算には行列演算は必須であり、行列やベクトルを記述するためには配列が必要となる。ベクトル、行列の例を以下に示す。

$$\mathbf{a} = \begin{pmatrix} 1 \\ -2 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} 3 & 5 & 0 \\ 2 & -1 & 3 \\ 0 & 2 & 4 \end{pmatrix} \quad (1)$$

行列の縦の並びは列 (column) と呼び、横の並びは行 (row) と呼ぶ。上の例では、 $\mathbf{a}$  は列ベクトルである。また、行列に含まれる数値は要素 (entry、element) と呼ばれる。

行列の和算はおのこの要素に対して行えばよいが、積算にはいくつか種類がある。ここでは内積について考えることとする。列ベクトル  $\mathbf{a}$  と  $\mathbf{b}$  の内積は以下のように定義される。

$$\mathbf{a}^T \mathbf{b} = \begin{pmatrix} 2 & 5 \end{pmatrix} \begin{pmatrix} -3 \\ 4 \end{pmatrix} = 2 \cdot -3 + 5 \cdot 4 = 14 \quad (2)$$

ここで、 $\mathbf{a}$  の肩添字  $T$  は転置であることを意味する。

次に、行列とベクトルとの積算の例を示す。

$$\mathbf{A} \mathbf{b} = \begin{pmatrix} 2 & 5 \\ -1 & 3 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 4 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \cdot 4 + 5 \cdot 1 \\ -1 \cdot 4 + 3 \cdot 1 \\ 0 \cdot 4 + 0 \cdot 1 \end{pmatrix} = \begin{pmatrix} 13 \\ -1 \\ 0 \end{pmatrix} \quad (3)$$

この例では、計算結果はベクトルとなる。この例から明らかなように行列とベクトルの積算は、ベクトルの大きさ (サイズ) と行列の列の大きさが一致していなければならない。

最後に行列同士の積算の例を示す。

$$\begin{aligned} \mathbf{A} \mathbf{B} &= \begin{pmatrix} 2 & 5 & 0 \\ -1 & 3 & 4 \end{pmatrix} \begin{pmatrix} 6 & 7 \\ -2 & -3 \\ 4 & 1 \end{pmatrix} \\ &= \begin{pmatrix} 2 \cdot 6 + 5 \cdot -2 + 0 \cdot 4 & 2 \cdot 7 + 5 \cdot -3 + 0 \cdot 1 \\ -1 \cdot 6 + 3 \cdot -2 + 4 \cdot 4 & -1 \cdot 7 + 3 \cdot -3 + 4 \cdot 1 \end{pmatrix} = \begin{pmatrix} 2 & -1 \\ 4 & -12 \end{pmatrix} \end{aligned} \quad (4)$$

それでは配列を用いた C++ プログラムを以下に示していく。C++ での配列の表現の方法はいくつかあるが、ここでは標準ライブラリの `vector` を用いた例を示す。

まずは式 (1) に現れたベクトル  $\mathbf{a}$  を定義するプログラムを以下に示す。

Listing 1: prog1.cxx

```
1 #include<cmath>
2 #include<iostream>
3 #include<vector>
4
5 using namespace std;
6
7 int main(){
8     vector<float> a(2);
9
10    a[0]=1.;
11    a[1]=-2.;
12
13    return 0;
14 };
```

`vector` 型の変数 (正確には、`vector` クラスのインスタンス) を生成させているのが 9 行目であり、`float` 型のデータが格納される、大きさが 2 の配列として `a` が定義されている。`vector` クラスは標準ライブラリとして与えられており、3

行目にあるように予めインクルードさせる必要がある。なお、C++の配列では、基本的に「0」から値が格納される。すなわち、大きさが2の vector クラスのインスタンス a のデータは、a[0] と a[1] となる (a[1] と a[2] とはならない)。

次にベクトル同士の和算を行う例を以下に示す。この例では、ベクトル a と b との和の計算を行い、解を c に代入している。

Listing 2: prog2.cxx

```
1 #include<cmath>
2 #include<iostream>
3 #include<vector>
4
5 using namespace std;
6
7 int main(){
8
9     vector<float> a(2);
10    vector<float> b(2);
11    vector<float> c(2);
12
13    a[0]=1.;
14    a[1]=-2.;
15
16    b[0]=-1.;
17    b[1]=5.;
18
19    for(int i=0;i<2;i++){
20        c[i]=a[i]+b[i];
21    };
22
23    for(int i=0;i<2;i++){
24        cout<<"_c["<i<<" ]_="<<c[i]<<"\n";
25    };
26
27    return 0;
28};
```

最後に、行列を定義するプログラムを以下に示す。行列を配列で表現する場合には、この例のように二次元配列となる。

Listing 3: prog3.cxx

```
1 #include<cmath>
2 #include<iostream>
3 #include<vector>
4
5 using namespace std;
6
7 int main(){
8
9     vector< vector<float> > a(2);
10    for(int i=0;i<2;i++){
11        a[i].resize(2);
12    };
13
14    a[0][0]=1.;
15    a[0][1]=2.;
16    a[1][0]=-1.;
17    a[1][1]=5.;
18
19    return 0;
20};
```

二次元の配列は9行目のように定義する<sup>1</sup>。9行目で指定している2が、vector クラスのインスタンス a の一つ目の並びの個数を示し、10行目から12行目のループで、a[0]、a[1] それぞれの二つめの並びの個数をresizeにより指定している。なお、この例では、列、行のサイズがそれぞれ2である以下の行列を定義している。

$$A = \begin{pmatrix} 1 & 2 \\ -1 & 5 \end{pmatrix} \tag{5}$$

なお、配列は vector クラスを用いなくても定義することが出来る。例えば、Listing 2 で示したプログラムは以下のように書くことも出来る。ここでは、変数 a と b について、vector クラスではない方法によって配列として定義している。

Listing 4: prog4.cxx

```
1 #include<cmath>
2 #include<iostream>
3 #include<vector>
4
5 using namespace std;
```

<sup>1</sup>これはひとつの例である。

```

6
7 int main(){
8
9     float a[]={1., -2.};
10    float b[]={-1., 5.};
11
12    vector<float> c(2);
13
14    for(int i=0; i<2; i++){
15        c[i]=a[i]+b[i];
16    };
17
18    for(int i=0; i<2; i++){
19        cout<<" c["<<i<<" ] = " <<c[i]<<"\n";
20    };
21
22    return 0;
23 };

```

この例では配列 c は vector で定義しているが、これを a、b と同様の扱いとするならば、以下のように書く。

Listing 5: prog5.cxx

```

1 #include<cmath>
2 #include<iostream>
3 #include<vector>
4
5 using namespace std;
6
7 int main(){
8
9     float a[]={1., -2.};
10    float b[]={-1., 5.};
11
12    float *c=new float [2];
13
14    for(int i=0; i<2; i++){
15        c[i]=a[i]+b[i];
16    };
17
18    for(int i=0; i<2; i++){
19        cout<<" c["<<i<<" ] = " <<c[i]<<"\n";
20    };
21
22    delete [] c;
23
24    return 0;
25 };

```

サイズが 2 である配列 c が 12 行目で定義されている。なお、このように（動的に）定義した配列は 22 行目にあるように delete でメモリから消去させる必要がある。

問題 1：次のふたつのベクトルの和算、引き算、積算を行うプログラムを書きなさい（出力部分も）。なお、積算は行列の個々の要素の積を計算することではない（内積計算である）ことに留意せよ。

$$a = \begin{pmatrix} 3 \\ 5 \end{pmatrix}, b = \begin{pmatrix} 6 \\ -1 \end{pmatrix} \quad (6)$$

問題 2：次のふたつの行列の和算、引き算、積算を行うプログラムを書きなさい（出力部分も）。なお、積算は行列の個々の要素の積を計算することではないことに留意せよ。

$$a = \begin{pmatrix} 3 & -1 \\ 5 & 6 \end{pmatrix}, b = \begin{pmatrix} 6 & 4 \\ -1 & 3 \end{pmatrix} \quad (7)$$

問題 3：問題 2 で作成したプログラムについて、任意のサイズの行列を計算できるように修正せよ。また、作成したプログラムで問題 2 を解かせて、正しい結果が得られることを確認せよ。なお、ここで「任意」と言っているのは、プログラムの最初に int n=2; のように行列のサイズを指定した以降は、全て行列のサイズを n で記述する、ということ意図している。このようにプログラムを記述することで、解きたい問題の行列のサイズが変更になった場合でも、変数 n の値を変更するだけで対応が可能となる。