

数値解析レポート No.1

43 番 若月 耕紀

1 行列の任意の 2 行を交換するプログラムを作成し，課題 4 の A 行列の 1 行目と 3 行目を交換せよ．

作成したプログラムをリスト 1 に示す．

リスト 1 swap_row.c

```
1 void swap_row(double **matrix, int row, int col, int row1, int row2){
2     if(row1 < 1 || row1 > row || row2 < 1 || row2 > row){
3         return;
4     }
5     row1--;
6     row2--;
7     double *temp = matrix[row1];
8     matrix[row1] = matrix[row2];
9     matrix[row2] = temp;
10 }
```

引数に NAbasic の csvRead から得た行列，行数，列数，交換する行番号をそれぞれ与える．

このプログラムでは 1 未満または行列の行数より大きな行番号を入力された時のために対策をしている．また，引数を与える際には，そのままの行番号を入力するため，配列の添え字に合わせられえよう row1, row2 はデクリメントをする．

交換の動作としては，各行番号のポインタを入れ替えることで正しい動作をするようにした．

このプログラムを用いて課題 4 の A 行列である $A = \begin{pmatrix} 2 & -1 & 5 \\ -4 & 2 & 1 \\ 8 & 2 & -1 \end{pmatrix}$ の 1 行目と 3 行目を交換した結果を以下に示す．

交換前

```
2.000000 -1.000000 5.000000
-4.000000 2.000000 1.000000
8.000000 2.000000 -1.000000
```

交換後

```
8.000000 2.000000 -1.000000
-4.000000 2.000000 1.000000
2.000000 -1.000000 5.000000
```

上記より，正しく動作していることがわかる．

2 行列積を利用して、内積を求める方法を考え実装し、課題 2(予備)(2) の問題で動作確認せよ。

作成したプログラムをリスト 2 に示す。

リスト 2 matrix_product.c

```
1 double **matrix_product(double **A, int Arow, int Acol,
2                           double **B, int Brow, int Bcol) {
3
4     double **result = allocMatrix(Arow, Bcol);
5     int i, j, k;
6
7     if (Acol != Brow){
8         puts("演算ができません.");
9         return result;
10    }
11
12    for (i = 0; i < Arow; i++) {
13        for (j = 0; j < Bcol; j++) {
14            for (k = 0; k < Acol; k++) {
15                result[i][j] += A[i][k] * B[k][j];
16            }
17        }
18    }
19    return result;
20 }
```

まず、引数に NAbasic の csvRead から得た 2 つの行列とそれぞれの行数、列数を与える。

今回のプログラムは行列積を利用して内積を求めるプログラムなので、ベクトルを入力する際に、行ベクトル、列ベクトルの順で入力することで、内積を求めることができる。

このプログラムを用いて課題 2(予備)(2) の問題で動作確認した結果を以下に示す。

Vector1	Vector2
1.000000	3.000000
5.000000	7.000000
8.000000	-2.000000

result
22.000000

上記より、正しく動作していることがわかる。

3 課題 2(予備) の 2 本のベクトルがなす角を求めよ.

作成したプログラムをリスト 3 に示す.

リスト 3 formed_angle.c

```
1 double vector_norm(double *vector, int len){
2     int i;
3     double s = 0.0;
4
5     for (i = 0; i < len; i++) {
6         s += vector[i] * vector[i];
7     }
8     return sqrt(s);
9 }
10
11 double formed_angle(double **A, int Arow, int Acol,
12                     double **B, int Brow, int Bcol) {
13
14     if (Arow != 1 || Bcol != 1 || Acol != Brow) {
15         puts("演算できません");
16         return -1;
17     }
18     int len = Arow * Acol;
19     double **in_pro = matrix_product(A, Arow, Acol, B, Brow, Bcol);
20
21     return acos(in_pro[0][0]
22                / vector_norm(matrix2colVector(A, Arow, Acol), len)
23                / vector_norm(matrix2colVector(B, Brow, Bcol), len));
24 }
```

2 つのベクトル \mathbf{a}, \mathbf{b} の内積は $\mathbf{a} \cdot \mathbf{b} = |\mathbf{a}| |\mathbf{b}| \cos \theta$ と定義されるのでこの式を変形するとなす角 θ は $\theta = \arccos \frac{\mathbf{a} \cdot \mathbf{b}}{|\mathbf{a}| |\mathbf{b}|}$ で求めることができる. `vector_norm` で各ベクトルのノルムを求め, その結果を基に計算する.

このプログラムを用いて課題 2(予備) の 2 本のベクトルがなす角を求めた結果を以下に示す.

Vector1 1.000000 5.000000 8.000000

Vector2

3.000000

7.000000

-2.000000

result 1.271850[rad]

上記より正しく動作していることがわかる。

4 感想

今回のレポートは授業課題の 5,6 週目に比べると比較的簡単だと感じた。授業課題も含めここまでの課題でプログラム上で行列やベクトルを扱うことになり慣れてきたと感じた。しかし、今回のレポートは取り組み始めるのが少し遅かったため、今後難しくなると予想されるレポートを期限内に終わらせるために、次回のレポートからは計画的に取り組みたいと思う。