科学計算研究室 Pythonゼミ

～ 4. Seidel法～

2021-03-03 福田　浩

1　原理

1.1　あらまし

大学 4 年のときの実験時間中に当時の助手 (今の助教) の先生から偶然教わった方法で，個人的にはかなりショッキングな解法という印象.「連立方程式の素性が良ければ，適当な初期値を与えて，繰り返し計算すれば，まともな解に辿り着く」という，何ともいい加減な方法.

$$\begin{cases} 3x & +y & = 7 \\ -x & +4y & = -11 \end{cases} \tag{1}$$

第 1 式を変形して $x =$ の式にする．第 2 式を変形して $y =$ の式にする.

$$\begin{cases} x & = \frac{7-y}{3} \\ y & = \frac{-11+x}{4} \end{cases} \tag{2}$$

初めは $x = 0, y = 0$ とする．あとはひたすら解き進める.
1 巡目は

$$\begin{cases} x & = \frac{7-0}{3} & = 2.33 \\ y & = \frac{-11+2.33}{4} & = -2.17 \end{cases} \tag{3}$$

2 巡目は

$$\begin{cases} x & = \frac{7+2.17}{3} & = 3.06 \\ y & = \frac{-11+3.06}{4} & = -1.99 \end{cases} \tag{4}$$

3 巡目は

$$\begin{cases} x & = \frac{7+1.99}{3} & = 3.00 \\ y & = \frac{-11+3.00}{4} & = -2.00 \end{cases} \tag{5}$$

となり，$x = 3, y = -2$ が解として得られる.

1.2　前提条件

連立方程式を行列表記したときの対角要素が，他の要素の和よりも大きいこと．但しこれは十分条件であり，必要条件ではないため，この条件を満たしていなくても解ける場合がある.

1.3 アルゴリズム

1. 第 1 式から $x[1]$ を算出する
   - `x[1] -= a[1][j]*x[j]` 但し `1 != j`
   - `x[1] /= a[1][i]`

$$\vdots$$

2. 第 i 式から $x[i]$ を算出する
   - `x[i] -= a[i][j]*x[j]` 但し `i != j`
   - `x[i] /= a[i][i]`

$$\vdots$$

3. 第 n 式から $x[n]$ を算出する
   - `x[n] -= a[n][j]*x[j]` 但し `n != j`
   - `x[n] /= a[n][i]`

2 課題

Seidel 法により連立方程式を解くプログラムを Python で実装せよ

- 連立方程式の元数と行列表示にしたときの係数を，標準入力から入力する機能を備えること
- 連立方程式の解を，標準出力に出力する機能を備えること
- 以下の連立方程式を解け

$$\begin{cases} 4x & +y & +2z & +w & = 153 \\ x & +5y & +2z & +2w & = 204 \\ 2x & +y & +4z & +2w & = 197 \\ x & +y & +2z & +5w & = 398 \end{cases} \tag{6}$$

これはステンレス鋼 (SUS301) を X 線分析したときの解析例を模擬したものである.

- 以下の連立方程式を解け (Jordan 法の時の課題と同じ)

$$\begin{cases} 3x & +y & +z & = 10 \\ x & +5y & +2z & = 21 \\ x & +2y & +5z & = 30 \end{cases} \tag{7}$$

3 参考：C++ソースコード

```
#include <iostream>
#define N_MAX 10
#define C_MAX 30
using namespace std;
int main(void){

    int i, j, k, n;
    double a[10][11], x[10];

    cin >> n;

    for(i=0;i<n;i++){
        for(j=0;j<n+1;j++) cin >> a[i][j];
    }

    for(i=0; i<N_MAX; i++) x[i] = 0.0;
    for(k=0; k<C_MAX; k++){
        for (i=0; i<n; i++){
            x[i] = a[i][n];
            for (j=0; j<n; j++){
                if(i!=j) x[i] -= a[i][j]*x[j];
            }
            x[i] /= a[i][i];
        }
        cout << k << " ";
        for (i=0; i<n; i++) cout <<  x[i] <<" ";
        cout << endl;
    }
    return 0;
}
```

4 更なる検討

以下の連立方程式 (Jordan 法の時の課題と同じ) は解が発散して解けない．何故そうなるか，考察せよ．

$$\left\{\begin{array}{llllll} 0.51x_1 & +0.95x_2 & +0.80x_3 & +0.28x_4 & +0.41x_5 & = 16.7 \\ 0.39x_1 & +0.25x_2 & +0.43x_3 & +0.28x_4 & +0.88x_5 & = 9.8 \\ 0.55x_1 & +0.91x_2 & +0.12x_3 & +0.23x_4 & +0.31x_5 & = 10.4 \\ 0.26x_1 & +0.66x_2 & +0.95x_3 & +0.52x_4 & +0.57x_5 & = 17.7 \\ 0.83x_1 & +0.73x_2 & +0.62x_3 & +0.16x_4 & +0.77x_5 & = 14.1 \end{array}\right. \tag{8}$$