数値解析レポート No.2

40 番 矢野 敦大

1　LU 分解ができないときの連立方程式

$A = \begin{pmatrix} 2 & 1 & 1 \\ 1 & -1 & 5 \\ 1 & 2 & -4 \end{pmatrix}, b = \begin{pmatrix} 2 \\ -2 \\ 4 \end{pmatrix}$ としたときの $Ax = b$ の解 $x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$ を求めることを考える。この連立方程式をこれまでの LU 分解のプログラムで解こうとすると、実行途中でプログラムが停止してしまう。原因としては計算の途中で 0 除算が行われてしまうことが考えられる。これに対する対策として、0 除算が行われるまえにそれを検出し、検出した場合は「不定解」と出力するようにプログラムを書き換えることにした。

また、この連立方程式を解くと以下のようになる。

$$\left(\begin{array}{ccc|c} 2 & 1 & 1 & 2 \\ 1 & -1 & 5 & -2 \\ 1 & 2 & -4 & 4 \end{array}\right) \to \left(\begin{array}{ccc|c} 1 & -1 & 5 & -2 \\ 1 & 2 & -4 & 4 \\ 2 & 1 & 1 & 2 \end{array}\right) \to \left(\begin{array}{ccc|c} 1 & -1 & 5 & -2 \\ 0 & 3 & -9 & 6 \\ 0 & 3 & -9 & 6 \end{array}\right) \to \left(\begin{array}{ccc|c} 1 & -1 & 5 & -2 \\ 0 & 1 & -3 & 2 \\ 0 & 0 & 0 & 0 \end{array}\right)$$

この行基本変形から解は $x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 2 \\ 0 \end{pmatrix} + t \begin{pmatrix} 2 \\ -3 \\ -1 \end{pmatrix}$(t は任意定数) である。

2　LU 分解を用いて逆行列を求める

$C = \begin{pmatrix} 2 & -1 & 5 \\ -4 & 2 & 1 \\ 8 & 2 & -1 \end{pmatrix}$ の逆行列を求めることを考える。$C = LU$ と LU 分解されていれば、$C^{-1} = U^{-1}L^{-1}P$ となるので、上三角行列 U と下三角行列 L の逆行列を求めることができれば、C^{-1} を求めることができる。L と U の逆行列を求めることは比較的容易に行うことができる。

$U = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22} & a_{23} \\ 0 & 0 & a_{33} \end{pmatrix}, U^{-1} = \begin{pmatrix} a'_{11} & a'_{12} & a'_{13} \\ 0 & a'_{22} & a'_{23} \\ 0 & 0 & a'_{33} \end{pmatrix}$ と U, U^{-1} を定めると、

$$UU^{-1} = \begin{pmatrix} a_{11}a'_{11} & a_{11}a'_{12} + a_{12}a'_{22} & a_{11}a'_{13} + a_{12}a'_{23} + a_{13}a'_{33} \\ 0 & a_{22}a'_{22} & a_{22}a'_{23} + a_{23}a'_{33} \\ 0 & 0 & a_{33}a'_{33} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

これらを解くと U^{-1} は以下のようになる。

$$U^{-1} = \begin{pmatrix} a'_{11} & a'_{12} & a'_{13} \\ 0 & a'_{22} & a'_{23} \\ 0 & 0 & a'_{33} \end{pmatrix} = \begin{pmatrix} \dfrac{1}{a_{11}} & -\dfrac{a_{12}}{a_{11}a_{22}} & \dfrac{a_{12}a_{23}}{a_{11}a_{22}a_{33}} - \dfrac{a_{13}}{a_{11}a_{33}} \\ 0 & \dfrac{1}{a_{22}} & -\dfrac{a_{23}}{a_{22}a_{33}} \\ 0 & 0 & \dfrac{1}{a_{33}} \end{pmatrix}$$

L^{-1} についても同じように導出できる。$L = \begin{pmatrix} b_{11} & 0 & 0 \\ b_{21} & b_{22} & 0 \\ b_{31} & b_{32} & b_{33} \end{pmatrix}$ とすると、

$$L^{-1} = \begin{pmatrix} \frac{1}{b_{11}} & 0 & 0 \\ -\frac{b_{21}}{b_{11}b_{22}} & \frac{1}{b_{22}} & 0 \\ \frac{b_{32}b_{21}}{b_{11}b_{22}b_{33}} - \frac{b_{31}}{b_{11}b_{33}} & -\frac{b_{32}}{b_{22}b_{33}} & \frac{1}{b_{33}} \end{pmatrix}$$

ここで数値解析第 5 回課題のプログラムを利用し、C を LU 分解すると以下のようになる。

$$L = \begin{pmatrix} 1 & 0 & 0 \\ -0.5 & 1 & 0 \\ 0.25 & -0.5 & 1 \end{pmatrix}, U = \begin{pmatrix} 8 & 2 & -1 \\ 0 & 3 & 0.5 \\ 0 & 0 & 5.5 \end{pmatrix}, P = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 2 & 0 \\ 3 & 0 & 0 \end{pmatrix}$$

となる。ここから逆行列は

$$L^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0.5 & 1 & 0 \\ 0 & 0.5 & 1 \end{pmatrix}, U^{-1} = \begin{pmatrix} 0.125 & -0.083333 & 0.03030 \\ 0 & 0.333333 & -0.03030 \\ 0 & 0 & 0.18181 \end{pmatrix}$$

ここで $U^{-1}L^{-1}P$ を計算すると以下のようになる。

$$U^{-1}L^{-1}P = \begin{pmatrix} 0.030303 & -0.068188 & 0.833333 \\ -0.030303 & 0.318182 & 0.166667 \\ 0.181818 & 0.090909 & 0 \end{pmatrix}$$

手元で計算を行うと C^{-1} と一致することがわかる。

3 はさみ打ち法について

2 分法とよく似た方法にはさみうち法というものがある。2 分法では、2 点の中点を新しい端点としたのに対して、はさみうち法では、2 点が結ぶ直線が x 軸と交わる交点を新しい端点とする。[1]

はさみうち法を行う関数 regula_falsi() をソースコード 1 に示す。

リスト 1 作成した関数 regula_falsi()

```
double regula_falsi(double a, double b) {
        printf("はさみうち法の実行結果\n");

        double c;
        int i = 0;
        do {
                c = (a * f(b) - b * f(a)) / (f(b) - f(a));
                if (f(c) == 0) { break; }
                if (f(a) * f(c) < 0) { b = c; }
                if (f(a) * f(c) > 0) { a = c; }
                printf("試行回数:␣%2d,␣x=%.9lf\n", ++i, c);
```

```
                if (i == 10)break;
        } while (fabs(f(c)) > 1e-10);
        return c;
}
```

今回は数値解析第 3 回課題を 2 分法とはさみうち法でときこれらについて比較してみる。k3-input.csv について実行した結果を以下に示す。

はさみ打ち法の実行結果

試行回数： 1，x=1.181818182
試行回数： 2，x=1.330162049
試行回数： 3，x=1.447303924
試行回数： 4，x=1.536568968
試行回数： 5，x=1.602390729
試行回数： 6，x=1.649612800
試行回数： 7，x=1.682775076
試行回数： 8，x=1.705697489
試行回数： 9，x=1.721362780
試行回数：10，x=1.731983564

二分法の実行結果

試行回数： 1，x=2.000000000
試行回数： 2，x=1.500000000
試行回数： 3，x=1.750000000
試行回数： 4，x=1.875000000
試行回数： 5，x=1.812500000
試行回数： 6，x=1.781250000
試行回数： 7，x=1.765625000
試行回数： 8，x=1.757812500
試行回数： 9，x=1.753906250
試行回数：10，x=1.751953125

これを見るとはさみ打ち法のほうが収束が遅いことがわかる。しかし、k3-input_test.csv に対してこのプログラムを実行すると以下のようになる。

はさみうち法の実行結果

試行回数： 1，x=-2.333333333
試行回数： 2，x=-2.464639788
試行回数： 3，x=-2.508055719
試行回数： 4，x=-2.521499786
試行回数： 5，x=-2.525575287
試行回数： 6，x=-2.526802702
試行回数： 7，x=-2.527171632

```
試行回数： 8，x=-2.527282456
試行回数： 9，x=-2.527315741
試行回数：10，x=-2.527325738
二分法の実行結果
試行回数： 1，x=-2.500000000
試行回数： 2，x=-2.750000000
試行回数： 3，x=-2.625000000
試行回数： 4，x=-2.562500000
試行回数： 5，x=-2.531250000
試行回数： 6，x=-2.515625000
試行回数： 7，x=-2.523437500
試行回数： 8，x=-2.527343750
試行回数： 9，x=-2.525390625
試行回数：10，x=-2.526367188
```

こちらのデータでははさみ打ち法のほうが収束が早いことがわかる。これらの結果から与えられるデータによって、収束の早さが変わることがわかる。

4　ヤコビ法やガウス・ザイデル法が収束しない条件

ヤコビ法とガウス・ザイデル法が収束する条件として、「係数行列の各行について、対角項の絶対値がその行の対角項以外の成分の和の絶対値より大きい」というものが知られている。[3] つまりこれを満たさないような連立方程式は、ヤコビ法やガウス・ザイデル法が収束しないと言える。

例として次のような連立方程式を考える。

$$\begin{pmatrix} 3 & 3 & 3 \\ 2 & 4 & 6 \\ 9 & 2 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 18 \\ 28 \\ 16 \end{pmatrix}$$

この方程式の解は、$\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$ であるが、初期値を $\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$ としてヤコビ法で解を求めようとすると、

$$\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \to \begin{pmatrix} 4 \\ 5 \\ 5 \end{pmatrix} \to \begin{pmatrix} -4 \\ -2.5 \\ -30 \end{pmatrix} \to \begin{pmatrix} 38.5 \\ 54 \\ 57 \end{pmatrix} \to \begin{pmatrix} -105 \\ -97.75 \\ -438.5 \end{pmatrix}$$

というふうに解が発散してしまい、収束しない。この連立方程式はガウス・ザイデル法を用いても収束しない。

5　感想

今回のレポート課題は、初めて聞くようなアルゴリズムもあり、前回のレポート課題よりは少し難しかった。しかし調べてみるといろんな大学で行われている数値解析の授業で扱われているような内容だったので、しっかり覚えておくべき内容だと思った。次回のレポート課題も頑張りたい。

参考文献

[1] はさみうち法 (非線形方程式の数値解法), Qiita, @omu58n, ”https://qiita.com/omu58n/items/98886614ff92c00e5f05”

[2] 線形方程式の解法：反復法, 東京大学, ”http://nkl.cc.u-tokyo.ac.jp/13n/SolverIterative.pdf”

[3] パソコンで数値計算, ヤコビ法, ”http://nkl.cc.u-tokyo.ac.jp/13n/SolverIterative.pdf”

[4] 理数アラカルト, 上三角行列/下三角行列の性質, ”http://nkl.cc.u-tokyo.ac.jp/13n/SolverIterative.pdf” https://slpr.sakura.ne.jp/qp/zeroinsearch/