数値解析レポート No.2

42 番 鷲尾 優作

2023 年 1 月 6 日

1 問題 1.1 (連立方程式)

$$
\begin{aligned}
Ax &= b \\
\begin{pmatrix} 2 & 1 & 1 \\ 1 & -1 & 5 \\ 1 & 2 & -4 \end{pmatrix} x &= \begin{pmatrix} 2 \\ -2 \\ 4 \end{pmatrix}
\end{aligned}
$$

通常連立方程式を計算する場合，ガウス消去法が考えられる．この場合，解が求まらないのは，解が存在しない場合と，解が無限に存在する場合である．拡大係数行列から，掃き出しを行ってどちらか確認する．

$$
\begin{aligned}
\left[\begin{array}{ccc|c} 2 & 1 & 1 & 2 \\ 1 & -1 & 5 & -2 \\ 1 & 2 & -4 & 4 \end{array}\right] &\rightarrow \left[\begin{array}{ccc|c} 2 & 1 & 1 & 2 \\ 3 & 0 & 6 & 0 \\ 1 & 2 & -4 & 4 \end{array}\right] \\
&\rightarrow \left[\begin{array}{ccc|c} 2 & 1 & 1 & 2 \\ 3 & 0 & 6 & 0 \\ -3 & 0 & -6 & 0 \end{array}\right] \\
&\rightarrow \left[\begin{array}{ccc|c} 2 & 1 & 1 & 2 \\ 3 & 0 & 6 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right]
\end{aligned}
$$

3 行目は z にどんな値を与えても成立するため，不定解となる．

対策として，プログラム上で任意の行が全て 0 になった場合，エラー出力を行うか，解が存在しないことを示す値を返すようにする．

2 問題 1.2 (LU 分解)

LU 分解を $A = LU$ と定義する．

通常連立方程式を計算する場合，$Ax = b$ より，$x = A^{-1}b$ を利用する．このとき，A^{-1} は A の逆行列である．LU 分解の定義より，$Ax = b$ より，$LUx = b$ となる．ここで，右辺を単位行列 E として，$LUx = E$ とすると，$x = U^{-1}L^{-1}E$．この方程式で $x = A^{-1}$ であるから，$A^{-1} = U^{-1}L^{-1}E$ となる．

したがって，上三角行列 U、下三角行列 L の逆行列を組み合わせることで元の逆行列を復元できる．

2.0.1 L^{-1}，U^{-1} の導出

L，U が正方行列であるとして，L^{-1}，U^{-1} を求める．

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

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

LU 分解時の下三角行列，上三角行列の特性は逆行列を求めても維持される．したがって，逆行列が存在するとき，L，U の各要素に対して，計算に必要な L^{-1}，U^{-1} の各要素を求めることができる．

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

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

2.0.2 実装

L^{-1}, U^{-1} を求める処理をリスト 1 に示す。

リスト 1: 逆行列の生成

```
for (i = 1; i < data[0].row; i++) {
      for (j = 0; j < data[0].row - i; j++) {
          for (k = 0; k < i; k++) {
              Ui[j][j + i] -= data[1].matrix[j][j + k + 1] * Ui[j][j] * Ui[j + k + 1][j + i];
              Li[j + i][j] -= data[0].matrix[j + k + 1][j] * Li[j][j] * Li[j + i][j + k + 1];
          }
      }
}
```

また，行列積はリスト 2 のように求められる．

リスト 2: 行列積の生成

```
  for (*rowOut = 0; *rowOut < data[0].row; (*rowOut)++) {
    for (*colOut = 0; *colOut < data[1].col; (*colOut)++) {
      element = 0;
      for (i = 0; i < data[1].row; i++) {
        element += data[0].matrix[*rowOut][i] * data[1].matrix[i][*colOut];
      }
      matrixOut[*rowOut][*colOut] = element;
    }
  }
```

2.0.3 実行結果

小数点以下 3 桁として，出力結果を示す.

$$A^{-1} = \begin{pmatrix} 0.030 & -0.068 & 0.083 \\ -0.030 & 0.318 & 0.166 \\ 0.181 & 0.090 & 0 \end{pmatrix}$$

3 課題 2 2 分法とよく似た方法にはさみうち法がある．はさみうち法について調査し，2 分法との違いをまとめよ．また，同じ条件で 10 回繰り返した時の値について比較せよ.

はさみうち法は，二分法で二点の中点を新しい端点としたのに対して、二点を結ぶ直線が xx 軸と交わる点を新しい端点とする検索方法である.

作成したはさみうち法プログラムをリスト 3 に示す。

リスト 3: hasamiuch

```
double anatanokokorowohasamiuch(double left, double right, dynamicMemory *data, int attempts) {
  double point;
  point =  (a1 * a2 - a2 * a1) / (a2 - a1);
  if ((equationSolver(point, data) * equationSolver(left, data)) < 0) {
    right = point;
  } else if ((equationSolver(mid, data) * equationSolver(right, data)) < 0) {
    left = point;
  }
  if (attempts > 1) {
    return dichotomy(left, right, data, attempts - 1);
  } else {
    printf("att:␣%d,␣left:␣%lf,␣right:␣%lf,␣result:␣%lf\n", attempts, left,
           right, point);
    return point;
  }
}
```

一般的にははさみうち法のほうが二分法よりも収束が早いといわれている.

課題 3 ニュートン法の課題で求めた解と比較すると，$x \approx 1.753664$ として，はさみうち法のほうがわずかに収束が早いことがわかる.

二分法 $x = -2.526855469$

はさみうち法 $x = -2.527319846$

ニュートン法 $x = -2.527330028$

4 課題 3 ヤコビ法やガウス・ザイデル法が収束しない条件を，例をあげながら解説せよ

ヤコビ法やガウス・ザイデル法は，対角優位性を満たすときに収束する.

対角優位性とは，対角成分の絶対値が，その他の成分の絶対値の和よりも大きいことである．対角優位性を満たさないときには，収束しない．式で表すと

$$\sum_{j=1}^{n} |a_{ij}| < |a_{ii}|$$

となる.

例を挙げると，以下のような行列は対角優位性を満たしている.

$$A^{-1} = \begin{pmatrix} 3 & 1 & 1 \\ 2 & 4 & -1 \\ -2 & -1 & -8 \end{pmatrix}$$

この場合,

$$
\begin{aligned}
3 &> 1+1 \quad (1)\\
4 &> 2-1 \quad (2)\\
|8| &> -2+-1 \quad (3)
\end{aligned}
$$

である．これを満たさない行列を作成すれば，対角優位性を満たさず，収束しない．

参考文献

[1] LU 分解を用いた連立方程式の解法と逆行列の導出の仕組みと Python でのプログラミング,
lib-arts,
"https://www.hello-statisticians.com/explain-terms-cat"

[2] ヤコビ法，ガウス・ザイデル法の収束条件と対角優位性,
"http://nkl.cc.u-tokyo.ac.jp/15n/SolverIterative-25-29.pdf"