

電気電子計算工学及演習

三軒家 佑将 (さんげんや ゆうすけ)

3 回生

1026-26-5817

a0146089

1 前進代入

1.1 採用したアルゴリズム

$$y_i = b_i - \sum_{k=0}^{i-1} l_{ik} y_k$$

として、 $i = 0, 1, 2$ の順に y_i を求めた。

1.2 プログラムに関する情報

ファイル名

- 1.go
- forward.go
- print.go

コンパイルコマンド

```
go run 1.go print.go forward.go
```

作成した主な関数

PrintVector

ベクトル (1 次元配列) を表示する関数。引数としてベクトルを渡す。

PrintMatrix

行列 (2 次元配列) を表示する関数。引数としてベクトルを渡す。

Forward

前進代入法によって方程式の解を求める関数。第一引数として行列 (下三角行列) を、第二引数としてベクトル ($b = Lx$ のときの b) を渡すと、方程式の解をベクトルとして返す。

1.3 結果

手計算の結果は、

$$y = \begin{pmatrix} 9 \\ 8 \\ -4 \end{pmatrix}$$

であった。また、プログラムによる数値解は、

$$y = \begin{pmatrix} 9.000 \\ 8.000 \\ -4.000 \end{pmatrix}$$

であった。

1.4 考察

手計算による解とプログラムによる数値解は一致していた。

2 後退代入

2.1 採用したアルゴリズム

$$x_i = \frac{1}{u_{ii}} \left(c_i - \sum_{k=i+1}^{n-1} u_{ik} x_k \right)$$

として、 $i = 2, 1, 0$ の順に x_i を求めた。

2.2 プログラムに関する情報

ファイル名

- 2.go
- backward.go

コンパイルコマンド

```
go run 2.go print.go backward.go
```

作成した主な関数

Backward

後退代入法によって方程式の解を求める関数。第一引数として行列（上三角行列）を、第二引数としてベクトル（ $b = Ux$ のときの b ）を渡すと、方程式の解をベクトルとして返す。

2.3 結果

手計算の結果は、

$$y = \begin{pmatrix} 2 \\ -3 \\ -2 \end{pmatrix}$$

であった。また、プログラムによる数値解は、

$$y = \begin{pmatrix} 2.000 \\ -3.000 \\ -2.000 \end{pmatrix}$$

であった。

2.4 考察

手計算による解とプログラムによる数値解は一致していた。

3 行列の計算

3.1 結果

手計算により、

$$\begin{pmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{pmatrix} = \begin{pmatrix} 3 & 1 & -3 \\ -3 & -3 & 2 \\ -6 & -8 & 5 \end{pmatrix} \quad (1)$$

と求められた。

4 LU 分解

4.1 採用したアルゴリズム

$$u_{ij} = a_{ij} - \sum_{k=0}^{i-1} l_{jk} u_{kj}$$
$$l_{ij} = \frac{1}{u_{jj}} \left(a_{ij} - \sum_{k=0}^{j-1} l_{jk} u_{kj} \right)$$

として、 u_{ij}, l_{ij} を求めた。

このとき、問に与えられた順序に従って計算するのは面倒なので、添字から u_{ij}, l_{ij} を計算する関数をそれぞれ用意し、内部的に再帰呼び出しするように実装した。

4.2 プログラムに関する情報

ファイル名

- 4.go
- lu.go

コンパイルコマンド

```
go run 4.go print.go lu.go
```

作成した主な関数

Decomp

matrix.Decomp のように呼び出すと、上三角行列と下三角行列を返す関数。内部的に lower 関数と upper 関数を呼び出し、2次元配列に入れて、それを返すだけの関数。

upper

matrix.upper(i,j) のように呼び出すと、添字 (i と j) に対応する上三角行列の要素を返す関数。内部的に lower 関数を呼び出している。

lower

matrix.lower(i,j) のように呼び出すと、添字 (i と j) に対応する下三角行列の要素を返す関数。内部的に upper 関数を呼び出している。

4.3 結果

式 (??) を Decomp 関数により LU 分解した結果、

$$L = \begin{pmatrix} 1.000 & 0.000 & 0.000 \\ -1.000 & 1.000 & 0.000 \\ -2.000 & 3.000 & 1.000 \end{pmatrix}$$
$$U = \begin{pmatrix} 3.000 & 1.000 & -3.000 \\ 0.000 & -2.000 & -1.000 \\ 0.000 & 0.000 & 2.000 \end{pmatrix}$$

4.4 考察

$N = 11$ 程度までは大丈夫だった。それ以上になると、memoize が必要。