数値計算　後期期末レポート

人間情報システム工学科 4 年 36 号
松山　京介

2022 年 1 月 24 日

課題

課題 1　偏微分方程式 1(ラプラス方程式)

図 1 のような三角形の境界を S とし，三角形で囲まれた領域を R とする問題領域 $G = R \cup S$ がある．領域 R 内はラプラス方程式

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = 0$$

を満たしており，境界条件は以下である．

$$\begin{cases} \phi(x,y) = 20x, & (x,y) \in S_1 \\ \phi(x,y) = 20y, & (x,y) \in S_2 \\ \phi(x,y) = 160, & (x,y) \in S_3 \end{cases}$$

刻み幅を 2 として領域 R 内で定義されたラプラス方程式の解 $\phi_{2,2}$，$\phi_{2,4}$，$\phi_{4,2}$ を求めよ．(解：$\phi_{2,2} = 80$，$\phi_{2,4} = 120$，$\phi_{4,2} = 120$)

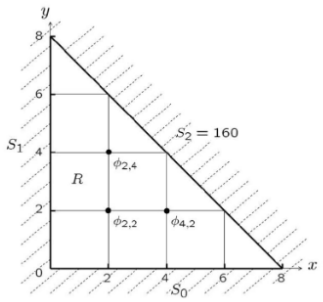

図 1: 偏微分方程式 1

アルゴリズム

ここでは，定常温度場や電磁場の解析によく見られる偏微分方程式 1 であるポアソン方程式について，閉領域 R で定義され，R の境界 S 上において与えられた境界条件のもとに解く問題：

$$\frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2} = \rho(x,y) \tag{1}$$
$$\text{ただし}, (x,y) \in R(\text{閉領域}).$$

境界条件：$f(x,y) = g(x,y), (x,y) \in S(\text{境界})$ を満たす解 $f(x,y)$ を求める問題である．
($\rho(x,y) = 0$ のときはラプラス方程式という)

閉領域 R 上で定義されたポアソン方程式 (1) を，境界 S の上において与えられた境界条件のもとに解く問題を考える．解法は，偏微分方程式 (1) を差分方程式に変換し，連立 1 次方程式を解く問題に帰着させる．ま

ず関数 $f(x,y)$ の点 (x_i, y_i) での x についてのテーラー展開は,

$$f_{i+1,j} = f_{i,j} + hf_x + \frac{h^2}{2!}f_{x,x} + \cdots \tag{2}$$

ただし,

$$h = x_{i+1} - x_i,\ f_{i,j} = f(x_i, y_j),\ f_{i+1,j} = f(x_{i+1},\ y_j),$$
$$f_x = \frac{\partial f(x,y)}{\partial x}\Big|_{x=x_i,\ y=y_j},\ f_{x,x} = \frac{\partial^2 f(x,y)}{\partial x^2}\Big|_{x=x_i,\ y=y_j}$$

また，h を $-h$ として,

$$f_{i-1,j} = f_{i,j} - hf_x + \frac{h^2}{2!}f_{x,x} + \cdots \tag{3}$$

h^3 以降の項を十分小さいものとして無視すると，eq. (2)+eq. (3) より

$$f_{x,x} \approx \frac{1}{h^2}(f_{i+1,j} - 2f_{i,j} + f_{i-1,j}) \tag{4}$$

よって点 (x_i, y_j) での $f_{x,x}$ は 3 点 $f_{i+1,j}, f_{i,j}, f_{i-1,j}$ で表せる (2 参照)．同様にして,

$$f_{y,y} \approx \frac{1}{k^2}(f_{i,j+1} - 2f_{i,j} + f_{i,j-1}) \tag{5}$$
$$\text{ただし}, k = y_{i+1} - y_i$$

この eq. (4)，eq. (5) の近似式をポアソン方程式 (1) に代入すると点 (x_i, y_j) に関して,

$$\frac{1}{h^2}(f_{i+1,j} - 2f_{i,j} + f_{i-1,j}) + \frac{1}{k^2}(f_{i,j+1} - 2f_{i,j} + f_{i,j-1}) = \rho_{i,j} \tag{6}$$

となり，ポアソン方程式が差分方程式に変換された．さらに，6 において $\rho_{i,j} = 0$ となるラプラス方程式では $h = k$ のとき次の簡単な差分方程式となる (3).

$$f_{i,j} = \frac{1}{4}(f_{i-1,j} + f_{i+1,j} + f_{i,j-1} + f_{i,j+1}) \tag{7}$$

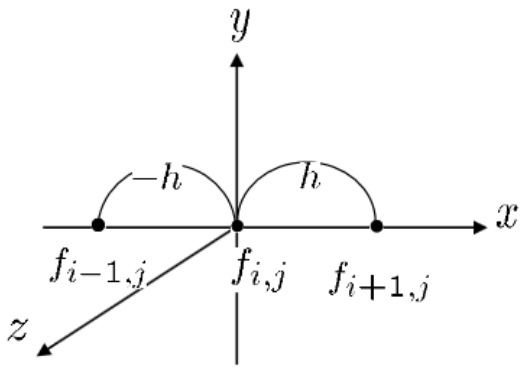

図 2: 離散点の関係

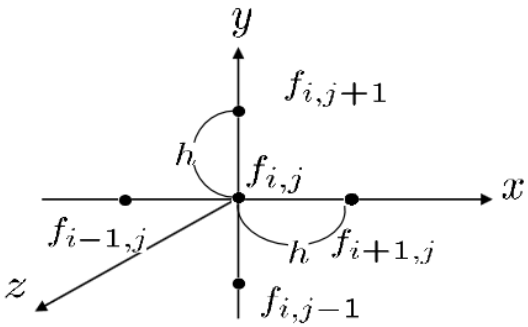

図 3: ラプラス方程式 (Laplace Equation) の差分方程式化

プログラムリスト

ソースコード 1: ラプラス方程式のプログラム

```
import numpy as np

mat=[[-4,1,1,-80],
   [1,-4,0,-400],
   [0,1,-4,-400]]

np.array(mat)
for k in range(0,3):
    for i in range(0,3):
        if(k==i):
            continue
        X=mat[i][k]/mat[k][k]
        for j in range(k,4):
            mat[i][j]=mat[i][j]-mat[k][j]*X

for l in range(0,3):
    print(mat[l][3]/mat[l][l])
```

実行結果

実行結果を以下に示す.

```
82.71186440677965
120.67796610169492
130.16949152542372
```

考察

差分方程式を計算すると、((-4,1,1)，(1,-4,0)，(1,0,-4))（ϕ 2.2, ϕ 2.4, ϕ 4.2）＝（-80,-400,-400）となる．これを解くと

ϕ 2.2 = 82.71186440677965

ϕ 2.4 = 120.67796610169492

ϕ 4.2 = 130.16949152542372

となり解析解とほぼ等しい値となる事がわかる.

課題 2　偏微分方程式 2(ラプラス方程式)

図 4 のような領域 R 内で定義されたラプラス方程式

$$\frac{\partial^2\phi}{\partial x^2}+\frac{\partial^2\phi}{\partial y^2}=0$$

の解を求める．ただし，境界条件は以下である．

$$\begin{cases}\phi(x,y)=10x, & (x,y)\in S_1\\ \phi(x,y)=10y, & (x,y)\in S_2\\ \phi(x,y)=30, & (x,y)\in S_3\end{cases}$$

刻み幅を 1 として領域 R 内で定義されたラプラス方程式の解 $\phi_{1,1}$，$\phi_{2,1}$，$\phi_{1,2}$ を求めよ．(解：$\phi_{1,1}=17.14$，$\phi_{2,1}=24.29$，$\phi_{1,2}=24.29$)

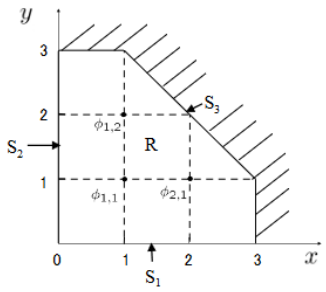

図 4: 偏微分方程式 2

アルゴリズム

ラプラス方程式のアルゴリズムは課題 1 に同じ.

プログラムリスト

```
1 import numpy as np
2
3 mat=[[-4,1,1,-20],
4     [1,-4,0,-80],
5     [1,0,-4,-80]]
6
7 np.array(mat)
8 for k in range(0,3):
```

```
    for i in range(0,3):
        if(k==i):
            continue
        X=mat[i][k]/mat[k][k]
        for j in range(k,4):
            mat[i][j]=mat[i][j]-mat[k][j]*X

for l in range(0,3):
    print(mat[l][3]/mat[l][l])
```

実行結果

実行結果を以下に示す.

```
17.142857142857146
24.285714285714285
24.28571428571429
```

考察

差分方程式を計算すると、((-4,1,1),(1,-4,0),(1,0,-4))(ϕ 1.1, ϕ 2.1, ϕ 1.2) = (-20,-80,-80) となる。これを解くと、

ϕ 1.1 = 17.142857142857146

ϕ 2.1 = 24.285714285714285

ϕ 1.2 = 24.28571428571429

となる。よって解析解とほぼ等しい値となる.

課題 3　(固有値問題 1)

行列 $A = \begin{pmatrix} 4 & 1 \\ 1 & 0 \end{pmatrix}$ の絶対値最大の固有値とその固有ベクトルをべき乗法により求めよ.

解：固有値 $\lambda_1 = 4.236$, 固有ベクトル $u_1 = \begin{pmatrix} 0.973 \\ 0.230 \end{pmatrix}$

(ただし初期値により固有ベクトルの向きが反対になる場合もある．)

アルゴリズム

固有値問題は,

$$A\boldsymbol{x} = \lambda\boldsymbol{x} \tag{8}$$

において，初期ベクトル $\boldsymbol{x}^{(0)}$ を適当に選び,

$$\boldsymbol{x}^{(k)} = A\boldsymbol{x}^{(k-1)} \tag{9}$$

を反復して $k \to \infty$ のとき $\boldsymbol{x}^{(k)}$ は行列 A の絶対値最大の固有値に対応する固有ベクトルに収束する．このことを利用して絶対値最大の固有値とその固有ベクトルを求める方法がべき乗法である．次にこのべき乗法について説明する．

(1) $n \times n$ 行列 A の n 個の固有値は全て実数で，$\lambda_1, \lambda_2, \ldots, \lambda_n$ とする．ただし，

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_n| \tag{10}$$

対応する固有ベクトルを $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_n$ $(A\boldsymbol{u}_i = \lambda_i \boldsymbol{u}_i,\ (i = 1, \ldots, n))$ とし互いに一次独立とする．任意の初期ベクトルを $\boldsymbol{x}$ とすると，$\boldsymbol{x}$ は A の固有ベクトルの線形結合：

$$\boldsymbol{x} = c_1\boldsymbol{u}_1 + c_2\boldsymbol{u}_2 + \cdots + c_n\boldsymbol{u}_n = \sum_{i=1}^{n} c_i\boldsymbol{u}_i \quad (c_i (i = 1, \ldots, n) \text{ は定数}) \tag{11}$$

で表せる．両辺に行列 A をかけていくと，

(2) $k \to \infty$ のとき右辺第 2 項は 0 に近づき，行列 A の絶対値最大の固有値に対応する固有ベクトルに収束する．

$$A^k\boldsymbol{x} \to c_1\lambda_1^k\boldsymbol{u}_1 \tag{12}$$

(3) 次に固有ベクトルに対する固有値を求める．eq. (12) で求まった固有ベクトル $c_1\lambda_1^k\boldsymbol{u}_1$ は $\boldsymbol{u}_1$ とおけるので，この $\boldsymbol{u}_1$ を用いて，

$$\frac{(\boldsymbol{u}_1, A\boldsymbol{u}_1)}{(\boldsymbol{u}_1, \boldsymbol{u}_1)} = \frac{\boldsymbol{u}_1^{\mathsf{T}} \cdot A\boldsymbol{u}_1}{\boldsymbol{u}_1^{\mathsf{T}}\boldsymbol{u}_1} = \frac{\boldsymbol{u}_1^{\mathsf{T}} \cdot \lambda_1\boldsymbol{u}_1}{\boldsymbol{u}_1^{\mathsf{T}}\boldsymbol{u}_1} = \lambda_1 \frac{u_1^2 + u_2^2 + \cdots + u_n^2}{u_1^2 + u_2^2 + \cdots + u_n^2} = \lambda_1 \tag{13}$$

により絶対値最大の固有値 λ_1 が求まる．

以下にべき乗法のアルゴリズムを示す．このアルゴリズムにはオーバーフローを防ぐために反復計算ごとに固有ベクトルの正規化：

を行っている．

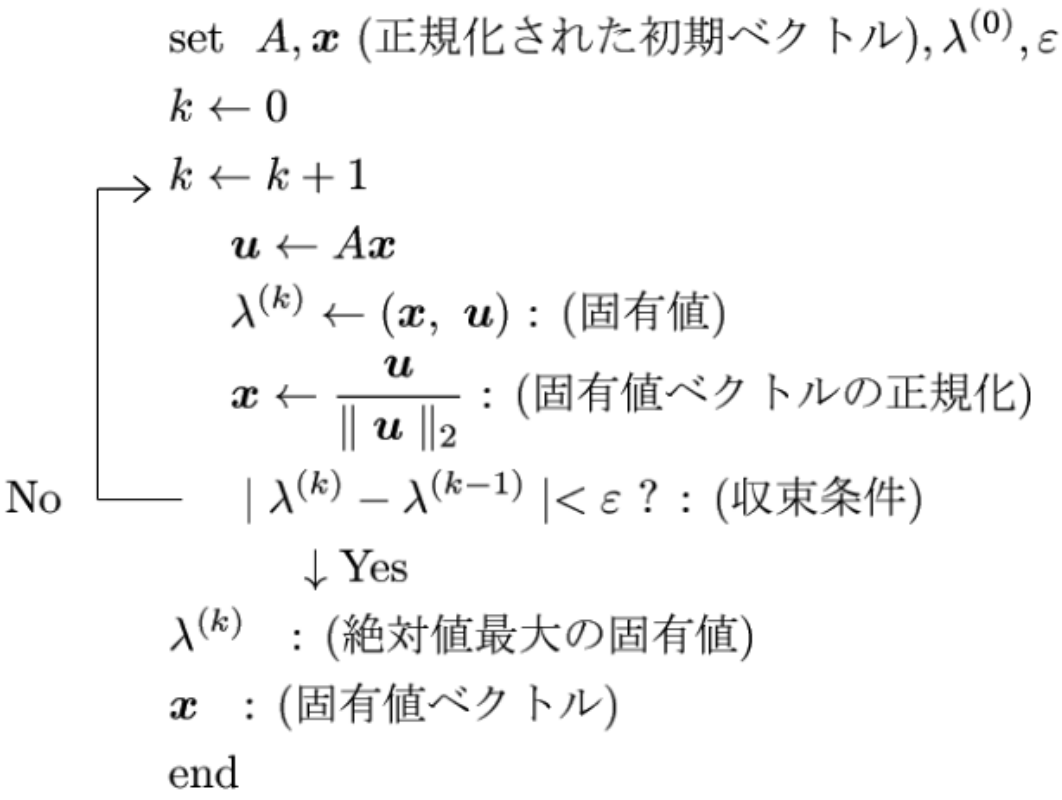

図 5: べき乗法 (Power Method) のアルゴリズム

プログラムリスト

ソースコード 2: 固有値問題のプログラム

```
import matplotlib.pyplot as plt
import numpy as np
import math

A=np.array([[4,1],[1,0]])
x=np.array([[1],[0]])

ramuda=[]
ramuda.append(1)
eps=0.0000001
k=0

while(1):
    k+=1
    u=np.dot(A,x)
    ramuda.append(np.dot(x.T,u))
```

```
    x=u/math.sqrt(np.dot(u.T,u))

    if(abs(ramuda[k]-ramuda[k-1])<eps):
        break
print("固有値:"+str(ramuda[k]))
print("固有ベクトル:"+str(x))
```

実行結果

実行結果を以下に示す.

```
固有値:[[4.23606798]]
固有ベクトル:[[0.97324896],[0.22975304]]
```

考察

問題で与えられた値は, 固有値＝ 4.236, 固有ベクトル＝（0.973，0.230）
プログラムの実行結果の値は, 固有値＝ 4.23606798、固有ベクトル＝（0.97324896，0.22975304）であることから解析解と等しい解が求められていることがわかる.

課題 4　(固有値問題 2)

行列 $A = \begin{pmatrix} 1 & 3 & 2 \\ 3 & 5 & -1 \\ 2 & -1 & 3 \end{pmatrix}$ の絶対値最大の固有値とその固有ベクトルをべき乗法により求めよ.

解：固有値 $\lambda_1 = 6.607$, 固有ベクトル $u_1 = \begin{pmatrix} -0.4776 \\ -0.8783 \\ -0.02128 \end{pmatrix}$

アルゴリズム

アルゴリズムは課題 3 に同じ.

プログラムリスト

ソースコード 3: 固有値問題のプログラム 2

```
import matplotlib.pyplot as plt
import numpy as np
import math

A=np.array([[1,3,2],[3,5,-1],[2,-1,3]])
x=np.array([[-1],[-1],[-1]])

```

```
ramuda=[]
ramuda.append(1)
eps=0.0000001
k=0

while(1):
    k+=1
    u=np.dot(A,x)
    ramuda.append(np.dot(x.T,u))
    x=u/math.sqrt(np.dot(u.T,u))

    if(abs(ramuda[k]-ramuda[k-1])<eps):
       break
print("固有値:"+str(ramuda[k]))
print("固有ベクトル:"+str(x))
```

実行結果

実行結果を以下に示す.

```
固有値:[[6.60687207]]
固有ベクトル:[[-0.47758451],[-0.87832631],[-0.02135261]]
```

考察

問題で与えられた値は, 固有値＝ 6.607, 固有ベクトル＝（-0.4776, -0.8783,-0.02128）である.
プログラムの実行結果の値は, 固有値=6.60687207, 固有ベクトル=(-0.47758451,-0.87832631,-0.02135261) である.
以上より解析解と等しい値が求められていることがわかる.

課題 5　(固有値問題 3)

行列 $A = \begin{pmatrix} 3 & 0 & 0 & 1 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 3 \end{pmatrix}$ の絶対値最大の固有値とその固有ベクトルをべき乗法により求めよ.

解：固有値 $\lambda_1 = 4.000$, 固有ベクトル $u_1 = \begin{pmatrix} 0.707 \\ 0 \\ 0 \\ 0.707 \end{pmatrix}$

アルゴリズム

べき乗法のアルゴリズムは課題 3 に同じ.

プログラムリスト

ソースコード 4: 固有値問題のプログラム 3

```
import matplotlib.pyplot as plt
import numpy as np
import math

A=np.array([[3,0,0,1],[0,3,0,0],[0,0,1,0],[1,0,0,3]])
x=np.array([[-1],[-1],[-1],[-1]])

ramuda=[]
ramuda.append(1)
eps=0.0000001
k=0

while(1):
    k+=1
    u=np.dot(A,x)
    ramuda.append(np.dot(x.T,u))
    x=u/math.sqrt(np.dot(u.T,u))

    if(abs(ramuda[k]-ramuda[k-1])<eps):
       break
print("固有値:"+str(ramuda[k]))
print("固有ベクトル:"+str(x))
```

実行結果

実行結果を以下に示す.

```
固有値:[[3.99999991]]
固有ベクトル:[[-7.07106763e-01],[-2.24491740e-04],[-9.81307762e-18],[-7.07106763e
    -01]]
```

考察

問題で与えられた値は、固有値＝ 4.000、固有ベクトル＝（0.707, 0,0,0.707）である.
プログラムの実行結果の値は固有値＝ 3.99999991 固有ベクトル＝ (-7.07106763e-01, -2.24491740e-04, -9.81307762e-18, -7.07106763e-01）である.
以上より解析解と等しい値が求められていることがわかる.

感想

一年間を通じて様々な計算をプログラムを用いて求めた。プログラムを書き、実行し、正しい数値が得られるまで大変なこともあったが最後までやり遂げることが出来た。また、この授業から初めて LaTeX を使用したが、他の文書作成ソフトなどよりも美しく見やすいレポートを作成することが出来るため、今では他の科目のレポートも TeX を用いて書くようになった。これからもレポートを書く時は積極的に TeX を用いていきたい。