

Poisson 方程式に対する差分解

桂田 祐史

2003年11月19日, 2010年2月11日

目次

第1章	差分解の厳密解への収束証明	3
1.1	差分方程式	3
1.2	差分方程式に関する最大値原理	4
1.3	誤差評価	6
1.4	近似境界データ $\tilde{\varphi}$ の定め方	7
第2章	正方形領域における数値実験	9
2.1	ターゲット問題	9
2.2	差分法による離散化	9
2.3	連立1次方程式の行列、ベクトル表現	10
2.3.1	ポアソン方程式に対する差分法 revisited	14
2.4	定常反復法による連立1次方程式の解法 (計算手順の復習)	15
2.4.1	Jacobi 法	15
2.4.2	Gauss-Seidel 法	16
2.4.3	SOR 法	16
2.5	連立方程式 (2.3), (2.4) を SOR 法により解く	17
2.6	SOR 法によるサンプル・プログラム	18
2.7	課題	20
2.8	消去法によるサンプル・プログラム	20
第3章	円盤、円環における差分法	24
3.1	展望	24
3.2	差分近似について	24
3.2.1	2次元の場合	24
3.2.2	3次元の場合	25
3.3	2次元円盤領域における Poisson 方程式の Dirichlet 境界値問題	25
付録A	1次元 Poisson 方程式	29
A.1	同次 Dirichlet 境界条件	29

A.2 同次 Neumann 境界条件	32
A.3 周期境界条件	35

第1章 差分解の厳密解への収束証明

この章の記述は John [2] による (少し古めかしいかもしれない)。田端 [7] も参考になる。熱方程式と同様に、最大値原理を使うのが鍵となる。

1.1 差分方程式

Ω は \mathbf{R}^2 の有界領域、 Γ をその境界、 $f: \Omega \rightarrow \mathbf{R}$, $\varphi: \Gamma \rightarrow \mathbf{R}$ が与えられているとき、Poisson 方程式の境界値問題

$$(1.1) \quad -\Delta u = f \quad (\text{in } \Omega),$$

$$(1.2) \quad u = \varphi \quad (\text{on } \Gamma)$$

を考える。

正数 h を固定し、 \mathbf{R}^2 に幅 h の格子を導入して、

$$(1.3) \quad G \stackrel{\text{def.}}{=} \bar{\Omega} \text{ に属する格子点の全体,}$$

$$(1.4) \quad G_i \stackrel{\text{def.}}{=} \{P \in G; P \text{ とその上下左右の格子点} \in \bar{\Omega}\},$$

$$(1.5) \quad G_b \stackrel{\text{def.}}{=} G \setminus G_i$$

とおく。 G_i の点を内部格子点、 G_b の点を境界格子点と呼ぶ。

もちろん $G_i \subset \Omega$. 正方形領域等を格子で切る場合は $G_b \subset \Gamma$ とできるが、一般には駄目。

U が G で定義されているとき、 G_i 上の関数 LU を

$$LU(x, y) = \frac{U(x+h, y) - 2U(x, y) + U(x-h, y)}{h^2} + \frac{U(x, y+h) - 2U(x, y) + U(x, y-h)}{h^2}$$

で定める。

$\varphi: \Gamma \rightarrow \mathbf{R}$ の近似となっている $\tilde{\varphi}: G_b \rightarrow \mathbf{R}$ を適当に選ぶ (具体的な取り方は後述)。

以下、方程式

$$(1.6) \quad -LU(P) = f(P) \quad (P \in G_i),$$

$$(1.7) \quad U(P) = \tilde{\varphi}(P) \quad (P \in G_b)$$

を考える。この方程式を差分方程式と呼び、その解 $\{U(P)\}_{P \in G}$ が存在するとき、それを差分解と呼ぶ。

(1.6) を $U(P)$ について解くと

$$U(P) = \frac{1}{4} \sum_{Q \sim P} U(Q) + \frac{h^2}{4} f(P)$$

となる。ただし「 Q は P の上下左右の格子点である」ということを $Q \sim P$ と表した。

1.2 差分方程式に関する最大値原理

補題 1.2.1 (離散最大値原理) U が $-LU = f$ (on G_i) の解で、 $f(P) < 0$ ($\forall P \in G_i$) であれば、 U は G_i で最大値 ($= \max_G U$) を取り得ない。

証明 U が $P_0 \in G_i$ で最大値 M を取ったと仮定しよう。

$$M = \max_{P \in G} U(P) = U(P_0) = \frac{1}{4} \sum_{Q \sim P_0} U(Q) + \frac{h^2}{4} f(P_0) \leq \frac{1}{4} \sum_{Q \sim P_0} M + \frac{h^2}{4} f(P_0) < M.$$

これは矛盾である。ゆえに U は G_i で最大値を取り得ない。 ■

命題 1.2.2 U が差分方程式 $-LU = f$ (on G_i) の解ならば、

$$\max_{P \in G} |U(P)| \leq \max_{P \in G_b} |U(P)| + \frac{d^2}{4} \max_{P \in G_i} |f(P)|.$$

ただし $d \stackrel{\text{def.}}{=} G$ の直径 $= \sup_{x, y \in G} |x - y|$.

証明 まず

$$(1.8) \quad U(P) \leq \max_{P \in G_b} |U(P)| + \frac{d^2}{4} \max_{P \in G_i} |f(P)| \quad (P \in G)$$

を示そう。 U が最大値を G_b で取る場合は何も証明する必要はない。 U が $P_0 \in G_i$ で最大値を取ったと仮定する:

$$\max_{P \in G} U(P) = U(P_0).$$

P_0 では U と同じ値で、差分作用素 $-L$ を施すと負の値を取るような補助関数 \bar{U} を作ろう。 ε を定数として、

$$\bar{U}(P) \stackrel{\text{def.}}{=} U(P) + \varepsilon \overline{P_0 P^2}$$

とおく。

$$\frac{(x+h)^2 - 2x^2 + (x-h)^2}{h^2} + \frac{(y+h)^2 - 2y^2 + (y-h)^2}{h^2} = 4$$

に注意すると、 $(\overline{P_0 P} = (x, y)$ として使って)

$$-L\bar{U}(P) = -(LU(P) + 4\varepsilon) = f(P) - 4\varepsilon$$

が成り立つことがわかる。

そこで、 $f(P) - 4\varepsilon < 0$ ($P \in G$) となるように、つまり

$$(1.9) \quad \varepsilon > \max_{P \in G_i} \frac{1}{4} f(P)$$

であるように ε を取ると、 \bar{U} は G_i で最大値を取り得ないことが分かる (実際、 $\tilde{f} = f - 4\varepsilon$ とおくと、 G_i の上で $-L\bar{U} = \tilde{f}$, $\tilde{f} < 0$ が成り立つので、上の補題が適用できる)。

ゆえに、 $\bar{U}(P_0)$ は $\max_G \bar{U} = \max_{G_b} \bar{U}$ よりも真に小さいので、

$$\begin{aligned} \max_{P \in G} U(P) &= U(P_0) = \bar{U}(P_0) \\ &< \max_{P \in G_b} \bar{U}(P) = \max_{P \in G_b} \left(U(P) + \varepsilon \overline{P_0 P^2} \right) \leq \max_{P \in G_b} U(P) + \varepsilon d^2. \end{aligned}$$

この式は (1.9) を満たす任意の ε について成立するので、実は

$$\varepsilon = \max_{P \in G_i} \frac{1}{4} f(P)$$

についても成立する。代入すると

$$\max_{P \in G} U(P) \leq \max_{P \in G_b} U(P) + \frac{d^2}{4} \max_{P \in G_i} f(P) \leq \max_{P \in G_b} U(P) + \frac{d^2}{4} \max_{P \in G_i} |f(P)|.$$

すなわち (1.8) が証明できた。\$U\$ の代わりに \$-U\$ について議論すれば、

$$\max_{P \in G} (-U(P)) \leq \max_{P \in G_b} (-U(P)) + \frac{d^2}{4} \max_{P \in G_i} |f(P)| \quad (P \in G)$$

が得られるので、二つをまとめると所望の不等式を得る。■

系 1.2.3 差分方程式 (1.6), (1.7) の解は一意である。

証明 \$U\$ が

$$-LU(P) = 0 \quad (P \in G_i), \quad U(P) = 0 \quad (P \in G_b)$$

を満たすとする。上の記号で \$f = 0, \tilde{\varphi} = 0\$ ということだから、

$$\max_{P \in G} |U(P)| \leq 0 + 0 = 0.$$

これから \$U \equiv 0\$。ゆえに差分方程式の解は一意である。■

差分方程式は、適当な \$A \in M(n; \mathbf{R}), b \in \mathbf{R}^n\$ によって \$Ax = b\$ の形に書けることに注意すると、この系から差分方程式が常に可解であることも分かる。

1.3 誤差評価

\$u\$ は 4 階までの微分係数が \$\Omega\$ で存在して有界とする。

$$E(P) \stackrel{\text{def.}}{=} U(P) - u(P)$$

とおく。明らかに

$$LE(P) = LU(P) - Lu(P) = -f(P) - Lu(P).$$

Taylor の定理から、

$$-Lu(P) = -\Delta u(P) - \frac{h^2}{12} \bar{u}_{xxxx} - \frac{h^2}{12} \bar{u}_{yyyy} = f(P) - \frac{h^2}{12} (\bar{u}_{xxxx} + \bar{u}_{yyyy})$$

ただし

$$\begin{aligned}\bar{u}_{xxxx} &= P \text{ の近くの二点における } u_{xxxx} \text{ の平均値,} \\ \bar{u}_{yyyy} &= P \text{ の近くの二点における } u_{yyyy} \text{ の平均値.}\end{aligned}$$

ゆえに

$$|-Lu(P) - f(P)| \leq \frac{u_4 h^2}{6}, \quad u_4 \stackrel{\text{def.}}{=} \sup_{|\alpha|=4, x \in \Omega} |D^\alpha u(x)|.$$

ゆえに

$$\tilde{f}(P) \stackrel{\text{def.}}{=} -LE(P)$$

とおくとき、

$$|\tilde{f}(P)| \leq \frac{u_4 h^2}{6} \quad (P \in G).$$

さて、 E は

$$LE(P) = \tilde{f}(P) \quad (P \in G_i), \quad E(P) = u(P) - \tilde{\varphi}(P) \quad (P \in G_b)$$

の解であるから、

$$\max_{P \in G} |E(P)| \leq \max_{P \in G_b} |E(P)| + \frac{d^2}{4} \max_{P \in G_i} |\tilde{f}(P)| \leq \max_{P \in G_b} |\varphi(P) - \tilde{\varphi}(P)| + \frac{u_4 d^2 h^2}{24}.$$

例えば $G_b \subset \Gamma$ のように選べるときは、 $P \in G_b$ に対して、 $\tilde{\varphi}(P) = \varphi(P)$ とできるので、

$$\max_{P \in G} |U(P) - u(P)| \leq \frac{u_4 d^2 h^2}{24} = O(h^2) \quad (h \rightarrow 0).$$

1.4 近似境界データ $\tilde{\varphi}$ の定め方

$P_0 \in G_b$ に対し、 $\tilde{\varphi}(P_0)$ をどう定めるか。

方法 I (P_0 の近くの点における φ の値を採用) 実は

$$(1.10) \quad \exists Q \in \Gamma \quad \text{s.t.} \quad \overline{QP_0} \leq h.$$

が成り立つので、このような Q を一つ適当に選んで、 $\tilde{\varphi}(P) = \varphi(Q)$ とおけばよい。

以下 (1.10) を証明する。もしも $P_0 \in \Gamma$ ならば $Q = P_0$ でよい。 $P_0 \notin \Gamma$ の場合を考えよう。 $P_0 \in \bar{\Omega}$ であったから $P_0 \in \Omega$ である。 $P_0 \in G_b$ で、 G_b の定義から、 $Q \sim P_0$ となる Q のうち少なくとも一つ P_1 は $\bar{\Omega}$ に属さない。線分 $P_0 P_1$ におい

て、 $P_0 \in \Omega$, $P_1 \notin \bar{\Omega}$ であるから、その線分上の点 Q で $Q \in \Gamma$ であるものがある。
 このとき $\overline{P_0Q} < h$.

この場合、

$$\max_{P \in G_b} |u(P) - \tilde{\varphi}(P)| = O(h).$$

方法 II P_0 の上下左右の格子点のうち、 P_0 をはさんで向かい合う 2 点 P_1, P_2 が

$$P_1 \in G_i, \quad P_2 \notin G$$

を満たすとする。このような P_1, P_2 が常に存在するわけではないが (これは図を描いて考えれば明らか)、 Γ が滑らかで、格子が十分細かければ、これが成り立つことが期待できる (証明できるかな...).

すると

$$\exists Q \in \text{線分 } P_0P_3 \cap \Gamma.$$

こういう Q を適当に選んでおいて、

$$\varphi(P_0) = \frac{\theta U(P_1) + \varphi(Q)}{1 + \theta}, \quad \theta = \frac{\overline{P_0Q}}{h}$$

とおく。

この場合、

$$\max_{P \in G_b} |u(P) - \tilde{\varphi}(P)| = O(h^2).$$

第2章 正方形領域における数値実験

この節の内容は、菊地・山本 [1]、森・杉原・室田 [6] などが参考になる。

2.1 ターゲット問題

2次元正方形領域 $\Omega = (0, 1) \times (0, 1)$ における Poisson 方程式の Dirichlet 境界値問題

$$(2.1) \quad -\Delta u(x, y) \equiv -\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right)u(x, y) = f(x, y) \quad ((x, y) \in \Omega),$$

$$(2.2) \quad u(x, y) = g(x, y) \quad ((x, y) \in \Gamma)$$

を考える。ただし Γ は Ω の境界で、 $f: \Omega \rightarrow \mathbf{R}$, $g: \Gamma \rightarrow \mathbf{R}$ は与えられているとする。

2.2 差分法による離散化

正方形領域の各辺を N 等分して格子に区切ることにして、次のようにおく。

$$\begin{aligned} h &= \frac{1}{N}, \\ x_i &= ih \quad (0 \leq i \leq N), \\ y_j &= jh \quad (0 \leq j \leq N), \\ u_{i,j} &= u(x_i, y_j) \quad (0 \leq i \leq N, 0 \leq j \leq N), \\ f_{i,j} &= f(x_i, y_j) \quad (1 \leq i \leq N-1, 1 \leq j \leq N-1) \end{aligned}$$

$u_{i,j}$ の近似値 $U_{i,j}$ を次の方程式の解として定義する。

$$\begin{aligned} -\left[\frac{U_{i+1,j} - 2U_{i,j} + U_{i-1,j}}{h^2} + \frac{U_{i,j+1} - 2U_{i,j} + U_{i,j-1}}{h^2}\right] &= f_{i,j} \quad ((x_i, y_j) \in \Omega \text{ となる } (i, j)), \\ U_{i,j} &= g(x_i, y_j) \quad ((x_i, y_j) \in \Gamma \text{ となる } (i, j)) \end{aligned}$$

で定義すると、

$$C = -4I_{N-1} + J_{N-1}$$

と表される。さらに Kronecker 積 (テンソル積) \otimes を用いると¹、

$$(2.8) \quad A = -4I_{N-1} \otimes I_{N-1} + J_{N-1} \otimes I_{N-1} + I_{N-1} \otimes J_{N-1}$$

と表現できる (MATLAB などでは Kronecker 積が用意されていて、簡単に A が用意できる)。

余談 2.3.1 (こう書くともっと素敵さ) 初めて (2.8) を見たときはとても感心したのであるが、今はもう一息と思っている。次のように書くべきである。

折角だから、一般化して書く。 $\Omega := (a, b) \times (c, d)$ を x 方向に N_x 等分、 y 方向に N_y 等分して解く場合、

$$\begin{aligned} h_x &:= \frac{b-a}{N_x}, & h_y &:= \frac{d-c}{N_y}, \\ x_i &:= a + ih_x \quad (i = 0, 1, \dots, N_x), & y_j &:= c + jh_y \quad (j = 0, 1, \dots, N_y), \\ m &:= N_x - 1, & n &:= N_y - 1, \\ \omega &:= \{1, \dots, m\} \times \{1, \dots, n\}, \\ f_{ij} &:= f(x_i, y_j) \quad ((i, j) \in \omega) \end{aligned}$$

とおくと差分方程式は

$$-\left(\frac{U_{i+1,j} - 2U_{i,j} + U_{i-1,j}}{h_x^2} + \frac{U_{i,j+1} - 2U_{i,j} + U_{i,j-1}}{h_y^2} \right) = f_{ij} \quad ((i, j) \in \omega).$$

格子点 (x_i, y_j) ($(i, j) \in \omega$) を

$$(x_1, y_1), (x_2, y_1), \dots, (x_m, y_1), (x_1, y_2), (x_2, y_2), \dots, (x_m, y_2), \dots, (x_1, y_n), (x_2, y_n), \dots, (x_m, y_n).$$

のように並べること、つまり (x_i, y_j) が³

$$\varphi(i, j) := (j-1)(N_x-1) + i$$

¹

$$A \otimes B := \begin{pmatrix} a_{11}B & a_{12}B & \cdots & a_{1n}B \\ a_{21}B & a_{22}B & \cdots & a_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}B & a_{m2}B & \cdots & a_{mn}B \end{pmatrix} \quad (A = (a_{ij}) \in \mathbf{R}^{m \times n}, \quad B \in \mathbf{R}^{k \times \ell}).$$

番目であると番号づけすることを、「I 方向に」番号づけするという。

$l = \varphi(i, j)$ のとき、

$$i = (\ell \bmod m) + 1, \quad j = (\ell \operatorname{div} m) + 1$$

である。

$l = \varphi(i, j)$ とするとき、 $U_\ell := U_{i,j}$, $f_\ell := U_{i,j}$ と書くことにすると、差分方程式は

$$A \begin{pmatrix} U_1 \\ U_2 \\ \vdots \\ U_{mn} \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_{mn} \end{pmatrix}$$

と表される。ただし

$$(2.9) \quad A := I_n \otimes \left[\frac{1}{h_x^2} (2I_m - J_m) \right] + \left[\frac{1}{h_y^2} (2I_n - J_n) \right] \otimes I_m.$$

ここで $\frac{1}{h_x^2} (2I_m - J_m)$ と $\frac{1}{h_y^2} (2I_n - J_n)$ は、それぞれ (a, b) 上の微分作用素 $-\frac{d^2}{dx^2}$ と (c, d) 上の微分作用素 $-\frac{d^2}{dy^2}$ の差分近似を表す係数行列である。■

問 $\vec{b} \in \mathbf{R}^n$ の成分を求めよ (場合分けして書くか、計算するプログラムを作れ)。

MATLAB 互換の Octave による計算例

```
oyabun% octave
GNU Octave, version 2.0.16 (sparc-sun-solaris2.6).
Copyright (C) 1996, 1997, 1998, 1999, 2000 John W. Eaton.
This is free software with ABSOLUTELY NO WARRANTY.
For details, type 'warranty'.

octave:1> n=4
n = 4
octave:2> I=eye(n)
I =

    1  0  0  0
    0  1  0  0
    0  0  1  0
    0  0  0  1

octave:3> B=diag(ones(n-1,1),1)+diag(ones(n-1,1),-1)
B =

    0  1  0  0
    1  0  1  0
    0  1  0  1
    0  0  1  0

octave:4> A = - 4 * kron(I,I) + kron(B,I) + kron(I,B)
A =

   -4    1    0    0    1    0    0    0    0    0    0    0    0    0    0
    1   -4    1    0    0    1    0    0    0    0    0    0    0    0    0
    0    1   -4    1    0    0    1    0    0    0    0    0    0    0    0
    0    0    1   -4    0    0    0    1    0    0    0    0    0    0    0
    1    0    0    0   -4    1    0    0    1    0    0    0    0    0    0
    0    1    0    0    1   -4    1    0    0    1    0    0    0    0    0
    0    0    1    0    0    1   -4    1    0    0    1    0    0    0    0
    0    0    0    1    0    0    1   -4    1    0    0    1    0    0    0
    0    0    0    0    1    0    0    0   -4    1    0    0    1    0    0
    0    0    0    0    0    1    0    0    1   -4    1    0    0    1    0
    0    0    0    0    0    0    1    0    0    1   -4    0    0    0    1
    0    0    0    0    0    0    0    1    0    0    1   -4    1    0    0
    0    0    0    0    0    0    0    0    1    0    0    1   -4    1    0
    0    0    0    0    0    0    0    0    0    1    0    0    1   -4    1
    0    0    0    0    0    0    0    0    0    0    1    0    0    1   -4
```

```
octave:5> quit
```

2.3.1 ポアソン方程式に対する差分法 revisited

(2016年某月某日、久しぶりにポアソン方程式に対する差分法を扱う学生が出て来て、この文書を見返して気づいたこと。)

$i = N_x - 1, i = 1, j = N_y - 1, j = 1$ の場合の (x_i, y_j) に対応する方程式はどの行にあるのか、非同次方程式の場合は連立1次方程式はどうなるか、説明を補足した方がいいかも。前者は「熱方程式に対する差分法 I」(桂田 [4]) を読めば分かるはず (と思いたい)。

以下、係数行列

$$A = I_{N_y-1} \otimes \frac{1}{h_x^2}(2I_{N_x-1} - J_{N_x-1}) + \frac{1}{h_y^2}(2I_{N_y-1} - J_{N_y-1}) \otimes I_{N_x-1}$$

の性質について少し述べる。

A は既約かつ優対角 ($|a_{ii}| \geq \sum_{\substack{1 \leq j \leq n \\ j \neq i}} |a_{ij}|$) かつ

$$|a_{11}| > \sum_{\substack{1 \leq j \leq n \\ j \neq 1}} |a_{1j}|$$

を満たすから、既約優対角行列である (既約性の証明は、定義を知っていればすぐ分かると思われるが、桂田 [5] には一応書いてある)。ゆえに A は正則である。

一方 A が対称であることは明らかで、 A の固有値が実数であることが分かる。さらに Gerschgorin の定理から、固有値は非負であることが分かるが、上で示したように A は正則であるから、0 は A の固有値ではない。ゆえに A の固有値はみな正であり、 A は正値対称行列である。

なお、 A の固有値は次のように具体的に求めることが出来る。

まず準備として、 $K_N = 2I_N - J_N$ の固有ベクトルは

$$\mathbf{u}_1 := \begin{pmatrix} \sin x_1 \\ \sin x_2 \\ \vdots \\ \sin x_N \end{pmatrix}, \mathbf{u}_2 := \begin{pmatrix} \sin 2x_1 \\ \sin 2x_2 \\ \vdots \\ \sin 2x_N \end{pmatrix}, \dots, \mathbf{u}_k := \begin{pmatrix} \sin kx_1 \\ \sin kx_2 \\ \vdots \\ \sin kx_N \end{pmatrix}, \dots, \mathbf{u}_N := \begin{pmatrix} \sin Nx_1 \\ \sin Nx_2 \\ \vdots \\ \sin Nx_N \end{pmatrix}.$$

対応する固有値は...

$K_N := 2I_N - J_N$ の固有値を $\lambda_i^{(N)}$ ($i = 1, \dots, N$) とおくと、 A の固有値は

$$\frac{1}{h_x^2} \lambda_i^{(N_x-1)} + \frac{1}{h_y^2} \lambda_j^{(N_y-1)} \quad (1 \leq i \leq N_x - 1, 1 \leq j \leq N_y - 1)$$

である。

2.4 定常反復法による連立1次方程式の解法 (計算手順の復習)

$A\vec{x} = \vec{b}$ を同値な方程式

$$\vec{x} = M\vec{x} + \vec{c}$$

の形の方程式に変形し、これを逐次近似法で解く。

これは次の原理に基づく。適当な初期ベクトル $\vec{x}^{(0)}$ を選んで、漸化式

$$\vec{x}^{(k+1)} = M\vec{x}^{(k)} + \vec{c}$$

によってベクトル列 $\{\vec{x}^{(k)}\}_{k \in \mathbf{N}}$ を定義するとき、

$$\rho(M) \stackrel{\text{def.}}{=} \max\{|\lambda|; \lambda \text{ は } M \text{ の固有値}\} < 1$$

であれば、 $\lim_{k \rightarrow \infty} \vec{x}^{(k)}$ が存在し、それは方程式の解になる。そこで十分大きな k に対して $\vec{x}^{(k)}$ を近似解として採用することができる。

A, b を与えられたとき、 M, c をどう定めるかは色々な方法がある (以下に代表的なものを3つあげ、計算手順のみ解説する)。注意すべきは、それらの方法は一般に通用するものではなく、 A がある種の良い性質を持っている場合にのみ適用できるということである。

次小節の準備として、 $A\vec{x} = \vec{b}$ の第 i 行

$$\sum_{j=1}^n a_{i,j}x_j = b_i$$

について、 $a_{ii} \neq 0$ と仮定して、 x_i について解いた式を書いておく:

$$x_i = \frac{1}{a_{ii}} \left(b_i - \sum_{j \neq i} a_{ij}x_j \right).$$

2.4.1 Jacobi 法

ベクトル列 $x^k = (x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)})^T$ ($k = 0, 1, 2, \dots$) を

for ($i = 1; i \leq n; i++$)

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j \neq i} a_{ij}x_j^{(k)} \right)$$

という漸化式で生成する反復法が Jacobi 法である。

2.4.2 Gauss-Seidel 法

ベクトル列 $x^k = (x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)})^T$ ($k = 0, 1, 2, \dots$) を

$$\text{for } (i = 1; i \leq n; i++) \\ x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij} x_j^{(k)} \right)$$

という漸化式で生成する反復法が Gauss-Seidel 法である。

2.4.3 SOR 法

ベクトル列 $x^k = (x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)})^T$ ($k = 0, 1, 2, \dots$) を

$$\text{for } (i = 1; i \leq n; i++) \{ \\ y_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij} x_j^{(k)} \right); \\ x_i^{(k+1)} = x_i^{(k)} + \omega \left(y_i^{(k+1)} - x_i^{(k)} \right); \\ \}$$

という漸化式で生成する反復法が SOR 法である。ここで ω は $0 < \omega < 2$ を満たす定数であり、緩和パラメータと呼ばれる。 $\omega = 1$ のとき、SOR 法は Gauss-Seidel 法に一致する。

変数の内容を上書きすることが可能な C のようなプログラミング言語では次のように簡単なコードで実現できる。

$$\text{for } (i = 1; i \leq n; i++) \{ \\ y = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x[j] - \sum_{j=i+1}^n a_{ij} x[j] \right); \\ x[i] = x[i] + \omega (y - x[i]); \\ \}$$

(2.6), (2.7) で定義される A の場合、理論的に完全な解析が可能で、結果は以下のようなになる。「最適の」 ω は

$$(2.10) \quad \omega_0 = \frac{2}{1 + \sin(\pi/N)}$$

であり、そのとき「収束率」は

$$\frac{1 - \sin(\pi/N)}{1 + \sin(\pi/N)}$$

に等しい (森・杉原・室田 [6])。

このように最適値が分かることは滅多になく、実際には実験的に最適値の見当をつけるしかない。しかし N が大きくなったときに、最適値がどのように変化するのか、その場合、どの程度の収束の速さが期待できるのか、大いに参考になるだろう。

2.5 連立方程式 (2.3), (2.4) を SOR 法により解く

(2.3) を U_{ij} について解くと

$$U_{ij} = \frac{1}{4} (U_{i+1,j} + U_{i-1,j} + U_{i,j+1} + U_{i,j-1} + h^2 f_{i,j}) \quad (1 \leq i \leq N-1, 1 \leq j \leq N-1).$$

右辺には $U_{i,j}$ ($i = 0$ or $i = N$ or $j = 0$ or $j = N$) が現れるが、これは未知数ではなく、境界条件から $g(x_i, y_j)$ に等しい。

余談 2.5.1 $f \equiv 0$ のとき (つまり u が調和関数のとき)、 U_{ij} が上下左右の格子点での値 $U_{i+1,j}$, $U_{i-1,j}$, $U_{i,j+1}$, $U_{i,j-1}$ の平均に等しい、という式になる。これは調和関数の性質をよく表わしていると言える²。■

従って SOR 法の反復計算は次のようになる。

²Laplace 方程式を定常版熱伝導方程式と考えると、「各点の温度が周囲の平均温度に等しい」と解釈できる。あるいは「調和関数の球面平均の定理」を見よ。

```

for (i = 1; i < N; i++)
  for (j = 1; j < N; j++) {
    y =  $\frac{1}{4} (h^2 f_{ij} + U[i+1][j] + U[i-1][j] + U[i][j+1] + U[i][j-1])$ ;
    U[i][j] = U[i][j] +  $\omega (y - U[i][j])$ ;
  }

```

実はこの方法は (U の適当な初期値が分かっている場合を除いて) あまり効率的とは言えないが、プログラムが大変書きやすいため、それなりに利用されることが多い。

2.6 SOR 法によるサンプル・プログラム

前小節で説明した方法によるプログラムを示す。

```

1 /*
2  * poisson-by-sor.c
3  *   正方形領域  $\Omega=(0,1) \times (0,1)$  における Poisson 方程式
4  *    $-\Delta u=f$  in  $\Omega$ ,  $u=g$  on  $\Gamma=\partial\Omega$ 
5  *   を SOR 法で解く。
6  *
7  * gcc -I/usr/local/include -o poisson-by-sor poisson-by-sor.c -L/usr/local/lib -lmatrix -lm
8  */
9
10 #include <stdio.h>
11 #include <math.h>
12 #include <matrix.h>
13
14 double f(double, double);
15 double g(double, double);
16 double Square(double);
17
18 int main()
19 {
20     int N, n, i, j, k, MaxIter;
21     matrix U, F;
22     double h, h2, omega, x, y, U_GS, Delta_U;
23
24     printf("N: "); scanf("%d", &N);
25     h = 1.0 / N; h2 = h * h;
26     n = (N - 1) * (N - 1);

```

```

27  U = new_matrix(N + 1, N + 1);
28  F = new_matrix(N + 1, N + 1);
29
30  /* f の値を事前に計算して記憶しておく */
31  for (i = 1; i < N; i++)
32      for (j = 1; j < N; j++)
33          F[i][j] = f(i * h, j * h);
34
35  /* 特に情報がないので、初期値は 0 にしておく */
36  for (i = 1; i < N; i++)
37      for (j = 1; j < N; j++)
38          U[i][j] = 0.0;
39
40  /* 境界条件から境界上にある格子点での値は分かる */
41  for (i = 0; i <= N; i++) {
42      x = i * h;
43      U[i][0] = g(x, 0.0); U[i][N] = g(x, 1.0);
44  }
45  for (j = 0; j <= N; j++) {
46      y = j * h;
47      U[0][j] = g(0.0, y); U[N][j] = g(1.0, y);
48  }
49
50  /* SOR 法の緩和パラメーターの選択 (とりあえず) */
51  omega = 1.6;
52
53  /* SOR 法の反復計算 */
54  MaxIter = 10000;
55  for (k = 1; k <= MaxIter; k++) {
56      Delta_U = 0.0;
57      /* SOR 反復 */
58      for (i = 1; i < N; i++)
59          for (j = 1; j < N; j++) {
60              U_GS = (h2 * F[i][j]
61                  + U[i+1][j] + U[i-1][j] + U[i][j+1] + U[i][j-1]) / 4;
62              U[i][j] += omega * (U_GS - U[i][j]);
63              /* 修正量を計算する */
64              Delta_U += Square(U_GS - U[i][j]);
65          }
66      Delta_U = omega * sqrt(Delta_U);
67      printf("反復回数=%d, 修正量=%g\n", k, Delta_U);
68      if (Delta_U < 1.0e-10)
69          break;
70  }
71  return 0;
72 }
73
74 double Square(double x)
75 {

```

```

76     return x * x;
77 }
78
79 double f(double x, double y)
80 {
81     return 0.0;
82 }
83
84 double g(double x, double y)
85 {
86     return x * x - y * y;
87 }

```

なお、 $f \equiv 0$, $g(x, y) = x^2 - y^2$ としてあるので、厳密解は $u(x, y) = x^2 - y^2$ である。

2.7 課題

- 「収束」したときの差分解 U_{ij} が厳密解 u_{ij} に近いことを観察せよ。誤差の N に関する挙動を調べよ。予想外のことが起こった場合は、 f や g を変えて実験せよ。
- 修正量 $\varepsilon_k \stackrel{\text{def.}}{=} \|U^{(k+1)} - U^{(k)}\|$ が「等比数列的」であることを観察せよ。
- N を変化したとき、最適な ω がどう変わるか実験により調べて、理論値 (2.10) と比較せよ。

余裕があれば次もやってみよ。

- (2.5) の \vec{b} を求めることによって、連立 1 次方程式を定常反復法以外の直接法 (例えば消去法) あるいは非定常反復法 (CG 法の系統) で解くプログラムを作成し、計算に要する CPU time を比較せよ。
 … 色々と細かい工夫が可能である。消去法を用いる場合は、半バンド幅 N の帯行列と考えて A を記憶することでメモリーを節約できる。CG 法を用いる場合は、 A を記憶するために 5 本ないし (対称性を利用して) 3 本のベクトルを用意すれば十分である。また前処理を施すことも考えられる。

2.8 消去法によるサンプル・プログラム

以下に掲げるプログラムは連立 1 次方程式を解く部分が非効率的であるが、参考までに。

```

/*
 * poisson.c
 */

/*
C   **   FINITE DIFFERENCE METHOD   **
C   **   FOR BOUNDARY VALUE PROBLEM   **
C   **   OF 2-D POISSON EQUATION   **
C   N : NO. OF DIVISIONS IN ONE DIRECTION
C   H : MESH SIZE
C   A : COEFFICIENT MATRIX
C   F : FREE TERM & SOLUTION VECTOR
*/

#include <stdio.h>
#include <matrix.h>

void gauss(matrix, vector, int);

int main()
{
    int i,j,n,nn,ij,ij1,ij2;
    double h,h2;
    vector f;
    matrix a;

    /* INPUT */
    printf("INPUT : N ="); scanf("%d", &n);
    if (n <= 1) return 0;
    nn = (n-1) * (n-1);

    /* */
    if ((a = new_matrix(nn, nn)) == NULL) {
        fprintf(stderr, "cannot allocate a matrix\n");
        exit(1);
    }
    if ((f = new_vector(nn)) == NULL) {
        fprintf(stderr, "cannot allocate a vector\n");
        exit(1);
    }

    /* INITIAL CLEAR */
    for (i = 0; i < nn; i++)
        for (j = 0; j < nn; j++)
            a[i][j] = 0.0;

    /* COEFFICIENT MATRIX */
    h = 1.0 / n;
    h2 = h * h;

```

```

for (i = 1; i < n; i++) {
    for (j = 1; j < n; j++) {
        ij = (n-1) * (j-1) + i - 1;
        ij1 = ij - (n-1);
        ij2 = ij + (n-1);
        a[ij][ij] = 4.0 / h2;
        if (i != 1) a[ij][ij-1] = - 1.0 / h2;
        if (j != 1) a[ij][ij1] = - 1.0 / h2;
        if (i != n-1) a[ij][ij+1] = - 1.0 / h2;
        if (j != n-1) a[ij][ij2] = - 1.0 / h2;
    }
}
/* FREE TERM */
for (i = 0; i < nn; i++)
    f[i] = 1.0;
/* CALCULATION */
gauss(a,f,nn);
/* OUTPUT */
printf(" MESH SIZE=%12.4e\n", h);
printf(" ** NODAL VALUES OF SOLUTION **\n");
for (i = 0; i < 5; i++)
    printf(" I      U(I)      ");
printf("\n");
for (i = 0; i < nn; i++) {
    printf("%3d%12.4e ", i, f[i]);
    if (i % 5 == 4) printf("\n");
}
printf("\n");
return 0;
}

```

```

void gauss(matrix a, vector f, int n)
{
    int i,j,k,nm1;
    double aa;
    nm1 = n - 1;
    /* FORWARD ELIMINATION */
    for (i = 0; i < nm1; i++) {
        for (j = i + 1; j < n; j++) {
            aa = a[j][i] / a[i][i];
            for (k = i + 1; k < n; k++)
                a[j][k] = a[j][k] - aa * a[i][k];
            f[j] = f[j] - aa * f[i];
        }
    }
    /* BACKWARD SUBSTITUTION */
    f[nm1] = f[nm1] / a[nm1][nm1];
    for (i = n - 2; i >= 0; i--) {
        for (j = i + 1; j < n; j++)

```

```
        f[i] = f[i] - a[i][j] * f[j];  
    f[i] = f[i] / a[i][i];  
    }  
}
```


第3章 円盤、円環における差分法

3.1 展望

以前は極座標変換をすれば良い、つまり2次元であれば

$$x = r \cos \theta, \quad y = r \sin \theta$$

によって、円盤領域 $\{(x, y) \in \mathbf{R}^2; x^2 + y^2 < R^2\}$ を長方形領域 $\{(r, \theta) \in \mathbf{R}^2; r \in [0, R), \theta \in [0, 2\pi]\}$ に写像して、そちらで差分方程式を立てて解くことでうまく行くと思っていた。しかし実際に熱方程式や波動方程式を解くプログラムを作成して実験してみて、安定性の問題が生じることが分かった。それからは Shortley-Weller 近似や、有限要素法にもアンテナを張っている。

3.2 差分近似について

3.2.1 2次元の場合

$$\Delta u = \frac{\partial^2 u}{\partial r^2} + \frac{1}{r} \frac{\partial u}{\partial r} + \frac{1}{r^2} \frac{\partial^2 u}{\partial \theta^2}.$$

$r \neq 0$ において、 Δu を差分近似することは簡単であるが、 $r = 0$ における取り扱いには注意が必要である。

$$\Delta u(0) = 4 \frac{u_{\text{mean}} - u(0)}{(\Delta r)^2} + O(\Delta r^2)$$

が成り立つ。ただし u_{mean} は $r = \Delta r$ における u の平均値を表わす。(詳しくは桂田 [3] の9章を見よ。)

$r = 0$ がない場合は、 $R = \log r$ と変数変換すると

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial r^2} + \frac{1}{r} \frac{\partial u}{\partial r}$$

は

$$e^{2R} \frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial R^2}.$$

3.2.2 3次元の場合

$$\Delta u = \frac{\partial^2 u}{\partial r^2} + \frac{2}{r} \frac{\partial u}{\partial r} + \frac{\cot \theta}{r} \frac{\partial u}{\partial \theta} + \frac{1}{r^2} \frac{\partial^2 u}{\partial \theta^2} + \frac{1}{r^2 \sin^2 \theta} \frac{\partial^2 u}{\partial \phi^2}.$$

$$\Delta u(0) = 6 \frac{u_{\text{mean}} - u(0)}{(\Delta r)^2} + O(\Delta r^2)$$

ただし u_{mean} は $r = \Delta r$ における u の平均値を表わす。

$r = 0$ がない場合は、 $u = w/r$ と従属変数の変数変換を行なうと

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial r^2} + \frac{2}{r} \frac{\partial u}{\partial r}$$

は

$$\frac{\partial w}{\partial t} = \frac{\partial^2 w}{\partial r^2}$$

に変換される。

3.3 2次元円盤領域における Poisson 方程式の Dirichlet 境界値問題

$\Omega = \{(x, y) \in \mathbf{R}^2; x^2 + y^2 < R^2\}$, $f: \Omega \rightarrow \mathbf{R}$ が与えられたとき、

$$(3.1) \quad -\Delta u = f \quad \text{in } \Omega,$$

$$(3.2) \quad u = 0 \quad \text{on } \Gamma := \partial\Omega$$

を解くことを考える。

$N_r, N_\theta \in \mathbf{N}$ に対して、

$$h_r := \frac{R}{N_r}, \quad h_\theta := \frac{2\pi}{N_\theta}, \quad \omega := \{1, 2, \dots, N_r - 1\} \times \{0, 1, \dots, N_\theta - 1\}$$

とおき、

$$r_i := ih_r \quad (i = 0, 1, \dots, N_r), \quad \theta_j := jh_\theta \quad (j = 0, 1, \dots, N_\theta - 1)$$

とおく。なお、

$$\theta_{-1} = \theta_{N_\theta - 1}, \quad \theta_{N_\theta} = \theta_0 = 0$$

と約束する。

(3.1) は極座標変換すると

$$-\left(u_{rr} + \frac{1}{r}u_r + \frac{1}{r^2}u_{\theta\theta}\right) = f$$

になるので、対応する差分方程式として

$$-\left[\frac{U_{i+1,j} - 2U_{ij} + U_{i-1,j}}{h_r^2} + \frac{1}{r_i} \frac{U_{i+1,j} - U_{i-1,j}}{2h_r} + \frac{1}{r_i^2} \frac{U_{i,j+1} - 2U_{ij} + U_{i,j-1}}{h_\theta^2}\right] = f_{ij} \quad ((i,j) \in \omega)$$

が得られる。これから

$$\left(\frac{2r_i^2}{h_r^2} + \frac{2}{h_r^2}\right)U_{ij} - \left(\frac{r_i^2}{h_r^2} + \frac{r_i}{2h_r}\right)U_{i+1,j} - \left(\frac{r_i^2}{h_r^2} - \frac{r_i}{2h_r}\right)U_{i-1,j} - \frac{1}{h_\theta^2}U_{i,j+1} - \frac{1}{h_\theta^2}U_{i,j-1} = r_i^2 f_{ij}.$$

すなわち

$$(a_i + 2d)U_{ij} - b_i U_{i+1,j} - c_i U_{i-1,j} - dU_{i,j+1} - dU_{i,j-1} = r_i^2 f_{ij} \quad ((i,j) \in \omega),$$

ただし

$$b_i := \frac{r_i^2}{h_r^2} + \frac{r_i}{2h_r}, \quad c_i := \frac{r_i^2}{h_r^2} - \frac{r_i}{2h_r}, \quad a_i := b_i + c_i, \quad d := \frac{1}{h_\theta^2}$$

とおいた。一方原点 $(0,0)$ での値 U_0 は、

$$\frac{1}{h_r^2} \left(U_0 - \frac{1}{N_\theta} \sum_{j=0}^{N_\theta-1} U_{1,j} \right) = f_0.$$

$$e := \frac{1}{N_\theta h_r^2}$$

とおくと、

$$N_\theta e U_0 - e \sum_{j=0}^{N_\theta-1} U_{1,j} = f_0.$$

これは行列とベクトルによる方程式

$$(\Lambda + \Theta)U = f$$

で表される。

$$\Theta := \left(\begin{array}{cccc|c|c|c} 0 & 0 & \cdots & 0 & & & \\ 0 & 2d & -d & -d & & & \\ 0 & -d & 2d & \ddots & & & \\ \vdots & & \ddots & \ddots & -d & & \\ 0 & -d & & -d & 2d & & \\ \hline & & & & & \ddots & \\ \hline & & & & & 2d & -d & -d \\ & & & & & -d & 2d & -d \\ & & & & & & \ddots & \ddots & \ddots \\ & & & & & & & -d & 2d & -d \\ & & & & & & & -d & & 2d \end{array} \right) = \left(\begin{array}{c|c} 0 & \mathbf{0}^T \\ \hline \mathbf{0} & D \\ & & \ddots & \\ & & & D \end{array} \right),$$

$$\Lambda := \left(\begin{array}{cccc|c|c|c|c} N_\theta e & -e & \cdots & -e & & & & \\ -c_1 & a_1 & & & -b_1 & & & \\ \vdots & & \ddots & & & \ddots & & \\ -c_1 & & & a_1 & & & -b_1 & \\ \hline & -c_2 & & & a_2 & & -b_2 & \\ & & \ddots & & & \ddots & & \\ & & & -c_2 & & a_2 & & -b_2 \\ \hline & & & & & & \ddots & \\ & & & & & & & \ddots & \\ & & & & & & & & \ddots \\ \hline & & & & & -c_{m-1} & & a_{m-1} & & -b_{m-1} \\ & & & & & & \ddots & & \ddots & \\ & & & & & & & -c_{m-1} & & a_{m-1} \\ \hline & & & & & & & -c_m & & a_m \\ & & & & & & & & \ddots & \\ & & & & & & & & -c_m & \end{array} \right)$$

$$m := N_r - 1, \quad N := (N_r - 1)N_\theta + 1,$$

$$\Theta(2 : N, 2 : N) = I_m \otimes D,$$

$$\Lambda(2 : N, 2 : N) = \begin{pmatrix} a_1 & -b_1 & & & \\ -c_2 & a_2 & -b_2 & & \\ & \ddots & \ddots & \ddots & \\ & & -c_{m-1} & a_{m-1} & -b_{m-1} \\ & & & -c_m & a_m \end{pmatrix} \otimes I_{N_\theta}$$

付録 A 1次元 Poisson 方程式

A.1 同次 Dirichlet 境界条件

$$-u''(x) = f(x) \quad (x \in (0, 1)), \quad u(0) = u(1) = 0.$$

Green 関数による解は

$$u(x) = \int_0^1 G(x, y) f(y) dy, \quad G(x, y) := \begin{cases} y(1-x) & (0 \leq y \leq x \leq 1) \\ x(1-y) & (0 \leq x \leq y \leq 1). \end{cases}$$

Fourier 級数による解は

$$u(x) = \sum_{n=1}^{\infty} \frac{b_n}{n^2 \pi^2} \sin n\pi x, \quad b_n := 2 \int_0^1 f(x) \sin n\pi x dx \quad (n \in \mathbf{N}).$$

固有値、固有関数は

$$\lambda_n = n^2 \pi^2, \quad \varphi_n(x) = \sin n\pi x \quad (n \in \mathbf{N}).$$

$N \in \mathbf{N}$ に対して、 $h := 1/N$ とおくと、素直に作った差分方程式は

$$(A.1) \quad \frac{1}{h^2} \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix} = \begin{pmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_{N-2}) \\ f(x_{N-1}) \end{pmatrix}.$$

$$\begin{aligned} & \sin n\pi x_{j+1} - 2 \sin n\pi x_j + \sin n\pi x_{j-1} \\ &= \sin n\pi(x_j + h) - 2 \sin n\pi x_j + \sin n\pi(x_j - h) \\ &= \sin n\pi x_j \cos n\pi h + \cos n\pi x_j \sin n\pi h - 2 \sin n\pi x_j + \sin n\pi x_j \cos n\pi h - \cos n\pi x_j \sin n\pi h \\ &= 2(\cos n\pi h - 1) \sin n\pi x_j \\ &= -4 \sin^2(n\pi h/2) \sin n\pi x_j \end{aligned}$$

であるから

$$-\frac{1}{h^2} (\sin n\pi x_{j+1} - 2 \sin n\pi x_j + \sin n\pi x_{j-1}) = \frac{4 \sin^2(n\pi h/2)}{h^2} \sin n\pi x_j.$$

$$\mu_n = \frac{4 \sin^2(n\pi h/2)}{h^2}, \quad \mathbf{v}_n := \begin{pmatrix} \sin n\pi x_1 \\ \sin n\pi x_2 \\ \vdots \\ \sin n\pi x_{N-2} \\ \sin n\pi x_{N-1} \end{pmatrix} = \begin{pmatrix} \sin \frac{n\pi}{N} \\ \sin \frac{2n\pi}{N} \\ \vdots \\ \sin \frac{(N-1)n\pi}{N} \end{pmatrix} \quad (n = 1, 2, \dots, N-1)$$

とおき、(A.1) の左辺の行列を A とすると、

$$A\mathbf{v}_n = \mu_n \mathbf{v}_n, \quad \mathbf{v}_n \neq \mathbf{0}, \quad 0 < \mu_1 < \mu_2 < \dots < \mu_{N-1}$$

が成り立つ¹。 \mathbf{v}_n は、実対称行列 A の固有値 μ_n (相異なる) に属する固有ベクトルであり、互いに直交するので、直交基底をなす。(固有関数は、離散化する前と同じ $\sin n\pi x$ で、そのうち独立なものは $n = 1, 2, \dots, N-1$ の $N-1$ 個だけ、というのが理解しやすいかもしれない)。『熱方程式に対する差分法』² を見よ。

¹ $Nh = 1$ ゆえ $0 < \frac{1 \cdot \pi h}{2} < \frac{2 \cdot \pi h}{2} < \dots < \frac{(N-1)\pi h}{2} < \frac{\pi}{2}$ に注意せよ。

²<http://nalab.mind.meiji.ac.jp/~mk/labo/text/heat-fdm-1.pdf>

```
% 1次元 Laplacian (Dirichlet 境界条件) の箱型行列の固有値と固有ベクトルを計算する関数
function [a,p]=LD_eigen(N)
```

```
    e=ones(N-1,1);
    a=spdiags([-e,2*e,-e],[-1:1,N-1,N-1]);
    h=1/N;
    x=(h:h:(N-1)*h)';
    p=zeros(N-1,N-1);
    for n=1:N-1
        p(:,n)=sin(n*pi*x);
    end
    d=p'*a*p;
    for i=1:N-1
        for j=1:N-1
            if abs(d(i,j)) < 1e-12
                d(i,j)=0;
            endif
        end
    end
end
d
```

```
> LD_eigen(8);
d =
```

0.60896	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
0.00000	2.34315	0.00000	0.00000	0.00000	0.00000	0.00000
0.00000	0.00000	4.93853	0.00000	0.00000	0.00000	0.00000
0.00000	0.00000	0.00000	8.00000	0.00000	0.00000	0.00000
0.00000	0.00000	0.00000	0.00000	11.06147	0.00000	0.00000
0.00000	0.00000	0.00000	0.00000	0.00000	13.65685	0.00000
0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	15.39104

```
octave-3.2.2.exe:6:C:\cygwin\home\mk
>
```


A.2 同次Neumann境界条件

$$-u''(x) = f(x) \quad (x \in (0, 1)), \quad u'(0) = u'(1) = 0.$$

これは解の一意性がない (定数だけの不定さがある)。とりあえず一つ解を求めるということで考える。Fourier 級数を用いると

$$u(x) = \sum_{n=1}^{\infty} \frac{a_n}{n^2\pi^2} \cos n\pi x, \quad a_n = 2 \int_0^1 f(x) \cos n\pi x dx \quad (n \in \mathbf{N}).$$

(この解は $\int_0^1 u(x) dx = 0$ という性質を持つ。その条件の下で一意性がある。) 固有値・固有関数は

$$\lambda_n = n^2\pi^2, \quad \varphi_n(x) = \cos n\pi x \quad (n = 0, 1, 2, \dots).$$

仮想格子点の技法を用いると、差分方程式は

$$\frac{1}{h^2} \begin{pmatrix} 2 & -2 & & & 0 \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ 0 & & & -2 & 2 \end{pmatrix} = \begin{pmatrix} f(x_0) \\ f(x_1) \\ \vdots \\ f(x_{N-1}) \\ f(x_N) \end{pmatrix}.$$

あるいは (最初と最後の成分に 1/2 をかけて)

$$(A.2) \quad \frac{1}{h^2} \begin{pmatrix} 1 & -1 & & & 0 \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ 0 & & & -1 & 1 \end{pmatrix} = \begin{pmatrix} f(x_0)/2 \\ f(x_1) \\ \vdots \\ f(x_{N-1}) \\ f(x_N)/2 \end{pmatrix}.$$

$$\begin{aligned} & \cos n\pi x_{j+1} - 2 \cos n\pi x_j + \cos n\pi x_{j-1} \\ &= \cos n\pi x_j \cos n\pi h - \sin n\pi x_j \sin n\pi h - 2 \cos n\pi x_j + \cos n\pi x_j \cos n\pi h + \sin n\pi x_j \sin n\pi h \\ &= 2(\cos n\pi h - 1) \cos n\pi x_j \\ &= -4 \sin^2(n\pi h/2) \cos n\pi x_j \end{aligned}$$

であるから

$$-\frac{1}{h^2} (\cos n\pi x_{j+1} - 2 \cos n\pi x_j + \cos n\pi x_{j-1}) = \frac{4 \sin^2(n\pi h/2)}{h^2} \cos n\pi x_j.$$

ゆえに

$$\mu_n = \frac{4 \sin^2(n\pi h/2)}{h^2}, \quad \mathbf{u}_n := \begin{pmatrix} \cos n\pi x_0 \\ \cos n\pi x_1 \\ \vdots \\ \cos n\pi x_{N-1} \\ \cos n\pi x_N \end{pmatrix} = \begin{pmatrix} 1 \\ \cos \frac{n\pi}{N} \\ \cos \frac{2n\pi}{N} \\ \vdots \\ \cos \frac{(N-1)n\pi}{N} \\ \cos \frac{Nn\pi}{N} \end{pmatrix} \quad (n = 0, 1, 2, \dots, N)$$

とおき、(A.2) の左辺の行列を A とおくと、

$$A\mathbf{u}_n = \mu_n \mathbf{u}_n, \quad \mathbf{u}_n \neq \mathbf{0}, \quad 0 = \mu_0 < \mu_1 < \mu_2 < \dots < \mu_{N-1} < \mu_N = \frac{4}{h^2}$$

が成り立つ³。 \mathbf{u}_n は、実対称行列 A の固有値 μ_n (相異なる) に属する固有ベクトルであり、互いに直交するので、直交基底をなす。(固有関数は、離散化する前と同じ $\cos n\pi x$ で、そのうち独立なものは $n = 0, 1, 2, \dots, N$ の $N + 1$ 個だけ、というのが理解しやすいかもしれない)。「熱方程式に対する差分法」⁴を見よ。

³ $Nh = 1$ ゆえ $0 = \frac{0 \cdot \pi h}{2} < \frac{1 \cdot \pi h}{2} < \frac{2 \cdot \pi h}{2} < \dots < \frac{(N-1)\pi h}{2} < \frac{N\pi h}{2} = \frac{\pi}{2}$ に注意せよ。

⁴<http://nalab.mind.meiji.ac.jp/~mk/lab/text/heat-fdm-1.pdf>

```

% LN_eigen.m --- 1次元 Laplacian (Neumann境界匠) う箱cdadad
cda4 Yc6a5好
function [a,p]=LN_eigen(N)
    e=ones(N+1,1);
    d=2*e; d(1)=1; d(N+1)=1;
    a=spdiags([-e,d,-e],-1:1,N+1,N+1);
    h=1/N;
    x=(0:h:1)';
    p=zeros(N+1,N+1);
    for n=0:N
        p(:,n+1)=cos(n*pi*x);
    end
    d=p'*a*p;
    for i=1:N+1
        for j=1:N+1
            if abs(d(i,j)) < 1e-12
                d(i,j)=0;
            endif
        end
    end
end
d

```

```

> LN_eigen(8);
d =
    0.00000    0.00000    0.00000    0.00000    0.00000    0.00000    0.00000    0.00000    0.00000
    0.00000    0.60896    0.00000    0.00000    0.00000    0.00000    0.00000    0.00000    0.00000
    0.00000    0.00000    2.34315    0.00000    0.00000    0.00000    0.00000    0.00000    0.00000
    0.00000    0.00000    0.00000    4.93853    0.00000    0.00000    0.00000    0.00000    0.00000
    0.00000    0.00000    0.00000    0.00000    8.00000    0.00000    0.00000    0.00000    0.00000
    0.00000    0.00000    0.00000    0.00000    0.00000    11.06147    0.00000    0.00000    0.00000
    0.00000    0.00000    0.00000    0.00000    0.00000    0.00000    13.65685    0.00000    0.00000
    0.00000    0.00000    0.00000    0.00000    0.00000    0.00000    0.00000    15.39104    0.00000
    0.00000    0.00000    0.00000    0.00000    0.00000    0.00000    0.00000    0.00000    32.00000

octave-3.2.2.exe:5:C:\cygwin\home\mk

```

A.3 周期境界条件

$$-u''(x) = f(x) \quad (x \in (-\pi, \pi)), \quad u(-\pi) = u(\pi), \quad u'(-\pi) = u'(\pi).$$

この問題についても解の一意性は成り立たない。

$$a_n := \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \cos nx \, dx, \quad b_n := \frac{1}{\pi} \int_{-\pi}^{\pi} f(x) \sin nx \, dx$$

とおく。まず

$$\int_{-\pi}^{\pi} f(x) \, dx = - \int_{-\pi}^{\pi} u''(x) \, dx = - [u'(x)]_{-\pi}^{\pi} = -(u'(\pi) - u'(-\pi)) = 0$$

であるから、 $a_0 = 0$ 。このことに注意すると、

$$u(x) := \sum_{n=1}^{\infty} \frac{1}{n^2} (a_n \cos nx + b_n \sin nx) \quad (x \in [-\pi, \pi])$$

とおくとき、

$$-u''(x) = \sum_{n=1}^{\infty} (a_n \cos nx + b_n \sin nx) = f(x) \quad (x \in [-\pi, \pi])$$

が成り立つことが分かる。

$N \in \mathbb{N}$ に対して、 $h := 2\pi/N$ とおく。自然に差分方程式を作って

$$(A.3) \quad \frac{1}{h^2} \begin{pmatrix} 2 & -1 & & -1 \\ -1 & 2 & -1 & \\ & \ddots & \ddots & \ddots \\ & & -1 & 2 & -1 \\ -1 & & & -1 & 2 \end{pmatrix} = \begin{pmatrix} f(x_0) \\ f(x_1) \\ \vdots \\ f(x_{N-2}) \\ f(x_{N-1}) \end{pmatrix}.$$

これまでと同様に

$$-\frac{1}{h^2} (\cos nx_{j+1} - 2 \cos nx_j + \cos nx_{j-1}) = \frac{4 \sin^2(nh/2)}{h^2} \cos nx_j,$$

$$-\frac{1}{h^2} (\sin nx_{j+1} - 2 \sin nx_j + \sin nx_{j-1}) = \frac{4 \sin^2(nh/2)}{h^2} \sin nx_j.$$

ゆえに

$$\mu_n := \frac{4 \sin^2(nh/2)}{h^2}, \quad \mathbf{u}_n := \begin{pmatrix} \cos nx_0 \\ \cos nx_1 \\ \vdots \\ \cos nx_{N-1} \end{pmatrix}, \quad \mathbf{v}_n := \begin{pmatrix} \sin nx_0 \\ \sin nx_1 \\ \vdots \\ \sin nx_{N-1} \end{pmatrix}$$

とおき、(A.3) の左辺の行列を A とおくと、

$$A\mathbf{u}_n = \mu_n \mathbf{u}_n, \quad A\mathbf{v}_n = \mu_n \mathbf{v}_n,$$
$$0 = \mu_0 < \mu_2 < \cdots < \mu_{[(N+1)/2]} = \frac{1}{h^2}.$$

が成り立つ⁵。

$0 \leq n \ll N$ のとき

$$\mu_n \doteq n^2$$

であることを注意しておく。

$$\mu_0 = 0, \quad \mathbf{v}_0 = \mathbf{0}$$

であって、 μ_0 の固有ベクトルは \mathbf{u}_0 の定数倍のみである。 N が偶数ならば μ_j ($1 \leq j \leq N/2$) は二重の固有値である。 N が奇数ならば μ_j ($1 \leq j \leq N-1/2$) は二重の固有値で、 $\mu_{(N+1)/2}$ は....

⁵ $Nh = 2\pi$ ゆえ、 $0 = \frac{0 \cdot h}{2} < \frac{1 \cdot h}{2} < \cdots < \frac{N/2 \cdot h}{2} = \frac{\pi}{2}$

関連図書

- [1] 菊地文雄, 山本昌宏. 微分方程式と計算機演習. 山海堂, 1991.
- [2] F. John. *Lectures on Advanced Numerical Analysis*. Gordon and Breach, Science Publishers, Inc., 1967. 邦訳: F. ジョン著, 藤田宏・名取亮訳, 数値解析講義, 産業図書 (1975).
- [3] 桂田祐史. 熱方程式に対する差分法 II — 円盤における熱方程式 —. <http://nalab.mind.meiji.ac.jp/~mk/labo/text/heat-fdm-2.pdf>, 1998～.
- [4] 桂田祐史. 熱方程式に対する差分法 I — 区間における熱方程式 —. <http://nalab.mind.meiji.ac.jp/~mk/labo/text/heat-fdm-1.pdf>, 1998 年～.
- [5] 桂田祐史. 連立 1 次方程式 iii. <http://nalab.mind.meiji.ac.jp/~mk/labo/text/linear-eq-3.pdf>, 2002～.
- [6] 森正武, 杉原正顯^{まさあき}, 室田一雄^{むろた かずお}. 線形計算. 岩波講座 応用数学. 岩波書店, 1994.
- [7] 田端正久. 偏微分方程式の数値解法. 岩波書店, 2010. もともとは岩波講座応用数学の「微分方程式の数値解法 II」(1994)であった。