目　次

付録 A

有限要素法の実行

附録 A では，MATLAB を用いた有限要素法の実際の計算方法を説明する．第 XX 章と第 XX 章 XX 節を既習であることを前提としている．あくまで概要を説明するのみであるから，より詳しいことは，[1]，[2]，[3]，[4] などを参照されたい．以下で紹介するプログラムは，著者が作成したものであるが，メッシュデータの構造に関しては，[5] を参考にした．

A.1　例題の設定

$L > 0$ を定数とする．正方形領域 $\Omega = (0, L) \times (0, L)$ の 4 つの辺を $\Gamma_N = \{(x, y) \mid 0 < x < L,\ y = 0\}$，$\Gamma_D = \partial\Omega \backslash \overline{\Gamma_N}$ のように分割する（図 A.1 を見よ）．

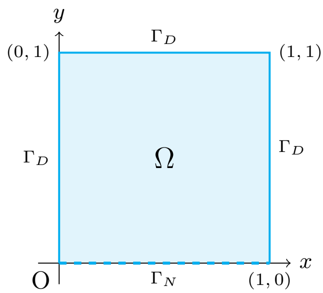


図 A.1　$L = 1$ のときの正方形領域 Ω とその境界 Γ_D，Γ_N．

fem_matlab:<2021/11/24>(9:48)

そして，Ω 上で，次の楕円型方程式を満たす関数 $u=u(x,y)$ の近似解を有限要素法で求めよう．

$$-c\Delta u+qu=f \qquad (x,y)\in\Omega, \tag{A.1a}$$

$$-c\frac{\partial u}{\partial n}=\kappa(u-g)-r \qquad (x,y)\in\Gamma_N, \tag{A.1b}$$

$$u=g \qquad (x,y)\in\Gamma_D. \tag{A.1c}$$

ただし，$c>0, q\ge 0, \kappa\ge 0$ は与えられた定数，$f=f(x,y)$，$g=g(x,y)$，$r=r(x,y)$ は与えれた関数である．g と r は，それぞれ，Γ_D と Γ_N の上だけで定義されていればよいが，ここでは，簡単のため，$\overline{\Omega}=[0,L]\times[0,L]$ 上で定義された連続関数とする．同様に，f も $\overline{\Omega}$ 上の連続関数としよう．

例 A.1. $c=q=\kappa=1$ として，(A.1) の解が，

$$u(x,y)=e^{-5(x-1/2)^2-5(y-1/2)^2}$$

となるように，f，g と r を求めておき，具体的な計算を行う際のデータとする．

A.2　メッシュデータの作成

有限要素法を適用するために，三角形分割を導入する．まず，一般的な方針を述べる．領域を三角形に分割した後に，すべての節点に通し番号 $1,2,\ldots,\texttt{np}$ を振り，P_s $(s=1,2,\ldots,\texttt{np})$ と名前を付ける．`np` は全節点数である．各節点の座標を $\mathrm{P}_s=(x_s,y_s)$ とする．同様に，すべての要素に通し番号 $1,2,\ldots,\texttt{nt}$ を振り，T_l $(l=1,2,\ldots,\texttt{nt})$ と名前を付ける．`nt` は全要素数である．このようにしてできた三角形分割を $\mathcal{T}_h=\{T_l\}_{l=1}^{\texttt{nt}}$ と表す．

いま，図 A.2 の左図のように，T_l を一つ固定し，その 3 節点を $\mathrm{Q}_1^l,\mathrm{Q}_2^l,\mathrm{Q}_3^l$ とする．これらを，T_l の局所的な節点と言う．ただし，$\mathrm{Q}_1^l\to\mathrm{Q}_2^l\to\mathrm{Q}_3^l$ の向きが，反時計回りになるようにしておく．これに対して，P_s は大域的な節点と言われる．局所的な節点には，大域的な節点が対応しているはずである．それらを，$\mathrm{Q}_1^l=\mathrm{P}_s$，$\mathrm{Q}_2^l=\mathrm{P}_t$，$\mathrm{Q}_3^l=\mathrm{P}_v$ とする．局所的な節点と対応する大域的な節点の通し番号の対応を，$\#\mathrm{Q}_1^l=s$ のように書く．

kbdbook6<2017/01/25>: pLaTeX2e<2020-10-01>+1 (based on LaTeX2e<2020-10-01>+2):

fem_matlab:<2021/11/24>(9:48)

次に，Γ_N を構成する辺の数を ne とし，すべての辺に $1,2,\ldots,\mathtt{ne}$ と通し番号を振り，e_k $(k = 1,2,\ldots,\mathtt{ne})$ と名前を付ける．すなわち，$\overline{\Gamma_N} = e_1 \cup e_2 \cup \cdots \cup e_{\mathtt{ne}}$ である．図 A.2 の右図のように，各 e_k の両端の 2 節点を $\mathrm{E}_1^k, \mathrm{E}_2^k$ とする．ここで，e_k を辺に持つ要素 T_l を考えた際に，$\mathrm{E}_1^k \to \mathrm{E}_2^k$ の向きが，反時計回りになるようにしておく．E_1^k，E_2^k が，それぞれ，大域的な節点に $\mathrm{E}_1^k = \mathrm{P}_s$ と $\mathrm{E}_2^k = \mathrm{P}_t$ のように対応しているとする．大域的な節点の通し番号を $\#\mathrm{E}_1^k = s$ と参照する．Γ_D についても同様に，これを構成する要素の辺の数を nd として，すべての辺を d_k $(k = 1,2,\ldots,\mathtt{nd})$ とする．各 d_k の両端の 2 節点を $\mathrm{D}_1^k = \mathrm{P}_s, \mathrm{D}_2^k = \mathrm{P}_t$ として，$\#\mathrm{D}_1^k = s$ と参照する．

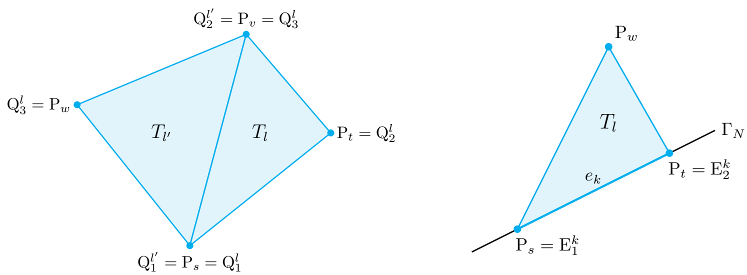


図 A.2　三角形分割における節点と要素

以上を確認した上で，問題の正方形領域 Ω の三角形分割を行おう．図 A.3 のように，Ω の各辺を m 等分し，$h = 1/m$ とおき，合同な $(m+1)^2$ 個の直角三角形に分割する（図 A.3 では $m = 5$ としている）．節点数は $\mathtt{np} = (m+1)^2$，要素数は $\mathtt{ne} = 2m^2$ となる．各節点の座標は，

$$\mathrm{P}_i = (sh, th), \quad i = (t-1)m + s, \quad (s, t = 1, \ldots, m+1) \tag{A.2}$$

である．要素の番号付けも図 A.3 のように行い，これで，T_l $(l = 1,\ldots,\mathtt{nt})$ を作る．Γ_N を構成する辺は，$e_s = \overline{\mathrm{P}_s\mathrm{P}_{s+1}}$ $(s = 1,2,\ldots,5)$ であり，残りの境界上に位置する辺はすべて，Γ_D を構成する辺となる．特に，$\mathtt{ne} = 5$，$\mathtt{nd} = 15$ である．

これらの三角形分割に関する情報を元に，表 A.2 に示すような形式のファイル (mesh.dat) を作る．そこに現れる $\mathtt{idn}_l$ などは，データの整理の目的で付加しているものであり，以後の計算には使わない．

kbdbook6<2017/01/25>: pLaTeX2e<2020-10-01>+1 (based on LaTeX2e<2020-10-01>+2):

fem_matlab:<2021/11/24>(9:48)

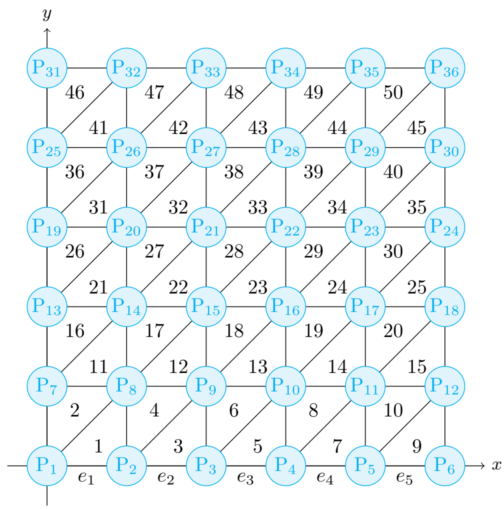


図 A.3　Ω の三角形分割

表 A.1　メッシュファイル (mesh.dat) の形式

(行番号)	mesh.dat				
1	np	nt	ne	nd	← パラメータ
2	x_1	y_1	$\mathtt{idn}_1$	0	← 節点座標
⋮	⋮	⋮	⋮	⋮	($\mathtt{idn}_l = 0$ 内部節点)
$\mathtt{np}+1$	$x_{\mathtt{np}}$	$y_{\mathtt{np}}$	$\mathtt{idn}_{\mathtt{np}}$	0	($\mathtt{idn}_l > 0$ 境界節点)
$\mathtt{np}+2$	$\#\mathrm{Q}_1^1$	$\#\mathrm{Q}_2^1$	$\#\mathrm{Q}_3^1$	0	← 要素の節点番号
⋮	⋮	⋮	⋮	⋮	
$\mathtt{np}+\mathtt{nt}+1$	$\#\mathrm{Q}_1^{\mathtt{nt}}$	$\#\mathrm{Q}_2^{\mathtt{nt}}$	$\#\mathrm{Q}_3^{\mathtt{nt}}$	0	
$\mathtt{np}+\mathtt{nt}+2$	$\#\mathrm{R}_1^1$	$\#\mathrm{R}_2^1$	$\mathtt{idr}_1$	0	← Robin を課す辺の節点番号
⋮	⋮	⋮	⋮	⋮	($\mathtt{idr}_k$: 偶数)
$\mathtt{np}+\mathtt{nt}+\mathtt{ne}+1$	$\#\mathrm{R}_1^{\mathtt{ne}}$	$\#\mathrm{R}_2^{\mathtt{ne}}$	$\mathtt{idr}_{\mathtt{ne}}$	0	
$\mathtt{np}+\mathtt{nt}+\mathtt{ne}+2$	$\#\mathrm{D}_1^1$	$\#\mathrm{D}_2^1$	$\mathtt{idd}_1$	0	← Dirichlet を課す辺の節点番号
⋮	⋮	⋮	⋮	⋮	($\mathtt{idd}_k$: 奇数)
$\mathtt{np}+\mathtt{nt}+\mathtt{ne}+\mathtt{nd}+1$	$\#\mathrm{D}_1^{\mathtt{nd}}$	$\#\mathrm{D}_2^{\mathtt{nd}}$	$\mathtt{idd}_{\mathtt{nd}}$	0	

kbdbook6<2017/01/25>: pLaTeX2e<2020-10-01>+1 (based on LaTeX2e<2020-10-01>+2):

リスト A.1 に示した mk_square1.m は，これを実行する MATLAB プログラムである．L と m を与えると，square1.dat に結果が保存される．これを用いて，$L=1$，$m=5$ としたメッシュデータファイル square1_5.dat と，$L=1$，$m=20$ のときの square1_20.dat を用意する（図 A.4 を見よ）．

```
function mk_square1(L, m)
h=L/m; mm=m+1; np=mm^2; ne=2*m^2; nr=m; nd=3*m;
x=linspace(0,L,mm); dat=[]; t=[]; p=[]; er=[]; ed=[];
% total numbers
dat=[dat;np,ne,nr,nd];
% nodes
for j=1:mm
    for i=1:mm
        k=(j-1)*mm + i; id=0; tmp=(i-1)*(j-1)*(i-mm)*(j-mm);
        if tmp==0 id=1; end
        p=[p;x(i),x(j),id,0];
    end
end
% elements
enum=0;
for j=1:m
    for i=1:m
        k=(j-1)*mm + i; kk=k+mm;
        t = [t;k, k+1, kk+1, 0]; t=[t;k, kk+1, kk, 0];
    end
end
% Robin boundaries
for i=1:m er=[er;i,i+1,2,0]; end
% Dirichlet boundaries
for j=1:m q=j*mm; ed=[ed;q,q+mm,1,0]; end
for i=1:m q=mm^2-i+1; ed=[ed;q,q-1,1,0]; end
for j=1:m q=mm^2-m-(j-1)*mm;ed=[ed;q,q-mm,1,0]; end
% output
dat=[dat;p;t;er;ed]'; F1 = fopen("square1.dat","w");
fprintf(F1,"%f %f %f %f\n",dat); fclose(F1);
end
```

リスト A.1　mk_square1.m

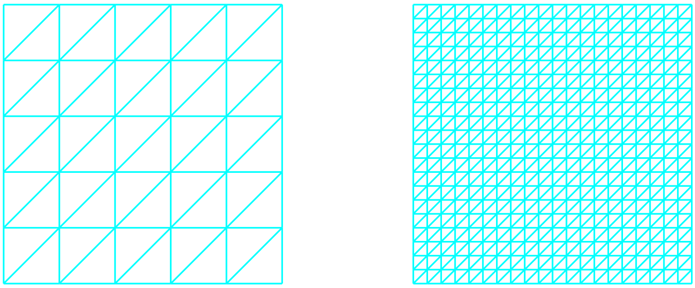


図 A.4　square1_5.dat と square1_20.dat

A.3　弱形式とその行列・ベクトル表現

関数空間

$$V = \{v \in H^1(\Omega) \mid v(x,y) = 0,\ (x,y) \in \Gamma_D\},$$

$$V(g) = \{v \in H^1(\Omega) \mid v(x,y) = g(x,y),\ (x,y) \in \Gamma_D\}$$

を導入すると，(A.1) の弱形式は，次のようになる：$u \in V(g)$，かつ，

$$\nu \iint_\Omega \nabla u \cdot \nabla v\ dxdy + q \iint_\Omega uv\ dxdy + \kappa \int_{\Gamma_N} uv\ dS = \iint_\Omega fv\ dxdy + \int_{\Gamma_N} (\kappa g + r) v\ dS \quad (\forall v \in V). \quad \text{(A.3)}$$

$\mathcal{T}_h$ 上で，P1 有限要素空間 X_h を考える（**??**節で導入した X_h と同じ空間である）．Π_h を $\mathcal{T}_h$ に対応する Lagrange 補間作用素として，$f_h = \Pi_h f$，$g_h = \Pi_h g$，$r_h = \Pi_h r$ とする．V と $V(g)$ に対応した有限要素空間

$$V_h = \{v_h \in X_h \mid v_h(x,y) = 0,\ (x,y) \in \Gamma_D\},$$

$$V_h(g_h) = \{v_h \in X_h \mid v_h(x,y) = g_h(x,y),\ (x,y) \in \Gamma_D\}$$

を導入すると，(A.3) に対する有限要素近似は次のようになる：$u_h \in V(g_h)$，かつ，

$$\underbrace{\nu \iint_\Omega \nabla u_h \cdot \nabla v_h\ dxdy}_{=I_1} + \underbrace{q \iint_\Omega u_h v_h\ dxdy}_{=I_2} + \underbrace{\kappa \int_{\Gamma_N} u_h v_h\ dS}_{=I_3}$$

$$= \underbrace{\iint_\Omega f_h v_h \ dxdy}_{=I_4} + \underbrace{\int_{\Gamma_N} (\kappa g_h + r_h) v_h \ dS}_{=I_5} \quad (\forall v_h \in V_h). \tag{A.4}$$

??節で説明したように，(A.5) は連立一次方程式に帰着される．それを具体的に述べるために，記号を設定しよう．$Z = \{1, 2, \ldots, N\}$ とおく．節点 P_i $(i \in \overline{\Lambda})$ に対応する X_h の P1 基底関数を ϕ_i とする．そして，一般に，$u_h, v_h \in X_h$ を，

$$u_h = \sum_{j=1}^{N} u_j \phi_j, \quad v_h = \sum_{i=1}^{N} v_i \phi_i \tag{A.5}$$

と表す．ただし，$u_j = u_h(\mathrm{P}_j)$，$v_i = u_h(\mathrm{P}_i)$ と書いている．さらに，

$$\boldsymbol{u} = \begin{pmatrix} u_1 \\ \vdots \\ u_N \end{pmatrix}, \quad \boldsymbol{v} = \begin{pmatrix} v_1 \\ \vdots \\ v_N \end{pmatrix} \tag{A.6}$$

と書くと，

$$I_1 = \sum_{i=1}^{N} \sum_{j=1}^{N} v_i u_j \iint_\Omega \nabla\phi_j \cdot \nabla\phi_i \ dxdy = \boldsymbol{v} \cdot \mathbf{K}\boldsymbol{u}, \tag{A.7a}$$

$$I_4 = \sum_{i=1}^{N} v_i \iint_\Omega f_h \phi_i \ dxdy = \boldsymbol{v} \cdot \boldsymbol{\beta} \tag{A.7b}$$

と書ける．ただし，$\mathbf{K} = (K_{i,j})_{1 \le i,j \le N}$ と $\boldsymbol{\beta} = (\beta_i)_{1 \le i \le N}$ を

$$K_{i,j} = \iint_\Omega \nabla\phi_j \cdot \nabla\phi_i \ dxdy, \quad \beta_i = \iint_\Omega f_h \phi_i \ dxdy \tag{A.8}$$

で定めている．

ここで，$\overline{\Gamma_D}$ 上に位置する節点番号の集合を Δ として，$\Lambda = Z \backslash \Delta$ とする．すなわち，$\Lambda = \{i \in Z \mid \mathrm{P}_i \in \overline{\Omega} \backslash \overline{\Gamma_D}\}$，$\Delta = \{i \in Z \mid \mathrm{P}_i \in \overline{\Gamma_D}\}$ である．図 A.3 の三角形分割では $\Delta = \{1, 6, 7, 12, 13, 18, 19, 24, 25, 30, 31, 32, 33, 34, 35, 36\}$ となっている．ただし，以下では，説明の便宜上，$\Lambda = \{1, 2, \ldots, n\}$，

$\Delta = \{n+1, \cdots, N\}$ の形を仮定する（必要ならば，適当に番号を付け直せば良い）．その上で,

$$\boldsymbol{u} = \begin{pmatrix} \boldsymbol{u}_{\mathrm{I}} \\ \boldsymbol{u}_{\Delta} \end{pmatrix}, \quad \boldsymbol{u}_{\mathrm{I}} = \begin{pmatrix} u_1 \\ \vdots \\ u_n \end{pmatrix}, \quad \boldsymbol{u}_{\Delta} = \begin{pmatrix} u_{n+1} \\ \vdots \\ u_N \end{pmatrix}, \tag{A.9a}$$

$$\mathbf{K} = \begin{pmatrix} \mathbf{K}_{\mathrm{I}} & \mathbf{K}_{12} \\ \mathbf{K}_{21} & \mathbf{K}_{\Delta} \end{pmatrix}, \quad \begin{array}{l} \mathbf{K}_{\mathrm{I}} = (K_{i,j})_{1\le i,j\le n}, \\ \mathbf{K}_{12} = (K_{i,j})_{1\le i\le n, n+1\le j\le N} \end{array} \quad \text{など} \tag{A.9b}$$

と書くことにしよう．$\boldsymbol{v}$ や $\boldsymbol{\beta}$ についても, 同様に, $\boldsymbol{v} = \begin{pmatrix} \boldsymbol{v}_{\mathrm{I}} \\ \boldsymbol{v}_{\Delta} \end{pmatrix}$ や $\boldsymbol{\beta} = \begin{pmatrix} \boldsymbol{\beta}_{\mathrm{I}} \\ \boldsymbol{\beta}_{\Delta} \end{pmatrix}$ という表記を用いるが，$v_h \in V_h$ のときは，$\boldsymbol{v}_{\Delta} = \mathbf{0}$ となる．この表記を用いれば，$v_h \in V_h$ に対して，(A.7) は,

$$I_1 = \boldsymbol{v}_{\mathrm{I}} \cdot \mathbf{K}_{\mathrm{I}} \boldsymbol{u}_{\mathrm{I}} + \boldsymbol{v}_{\mathrm{I}} \cdot \mathbf{K}_{12} \boldsymbol{u}_{\Delta}, \quad I_4 = \boldsymbol{v}_{\mathrm{I}} \cdot \boldsymbol{\beta}_{\mathrm{I}} \tag{A.10}$$

となる．

他の項についても同様に計算すれば，結局，(A.5) は，任意の $v_h \in V_h$，すなわち，任意の $\boldsymbol{v}_{\mathrm{I}} \in \mathbb{R}^n$ に対して,

$$\begin{aligned} &\boldsymbol{v}_{\mathrm{I}} \cdot [\underbrace{(\nu \mathbf{K}_{\mathrm{I}} + q\mathbf{M}_{\mathrm{I}} + \kappa \mathbf{R}_{\mathrm{I}})}_{=\mathbf{A}_{\mathrm{I}}} \boldsymbol{u}_{\mathrm{I}} + \underbrace{(\nu \mathbf{K}_{12} + q\mathbf{M}_{12} + \kappa \mathbf{R}_{12})}_{=\mathbf{A}_{12}} \boldsymbol{u}_{\Delta}] \\ &\qquad = \boldsymbol{v}_{\mathrm{I}} \cdot (\boldsymbol{\beta}_{\mathrm{I}} + \boldsymbol{\rho}_{\mathrm{I}}) \end{aligned} \tag{A.11}$$

が成り立つことと同値である．ここで，$\mathbf{M}_{\mathrm{I}} = (K_{i,j})_{1\le i,j\le n}$，$\mathbf{R}_{12} = (R_{i,j})_{1\le i\le n, n+1\le j\le N}$ や $\boldsymbol{\rho}_{\mathrm{I}} = (\rho_i)_{1\le i\le n}$, などは,

$$M_{i,j} = \iint_{\Omega} \phi_j \phi_i \, dxdy, \quad R_{i,j} = \int_{\Gamma_N} \phi_j \phi_i \, dS, \quad \rho_i = \int_{\Gamma_N} (\kappa g_h + r_h) \phi_i \, dS \tag{A.12}$$

を用いて，$\mathbf{M}_{\mathrm{I}}$，$\mathbf{M}_{12}$ や $\boldsymbol{\beta}_{\mathrm{I}}$ に準じて定められる．

そして，$\boldsymbol{u}_{\Delta}$ は，境界条件 $u_h \in V_h(g_h)$ を用いて,

$$\boldsymbol{u}_\Delta = \boldsymbol{g}_\Delta = \begin{pmatrix} g_h(\mathrm{P}_{n+1}) \\ \vdots \\ g_h(\mathrm{P}_N) \end{pmatrix} \tag{A.13}$$

とすれば良い．

まとめると，(A.1) に対する有限要素近似 (A.3) は，行列・ベクトル形式では，

$$\boldsymbol{v}_\mathrm{I} \cdot (\mathbf{A}_\mathrm{I}\boldsymbol{u}_\mathrm{I} + \mathbf{A}_{12}\boldsymbol{g}_\Delta) = \boldsymbol{v}_\mathrm{I} \cdot (\boldsymbol{\beta}_\mathrm{I} + \boldsymbol{\rho}_\mathrm{I}) \quad (\forall \boldsymbol{v}_\mathrm{I} \in \mathbb{R}^n)$$

となる．$\boldsymbol{v}_\mathrm{I}$ は任意なのだから，これは，次と同値である：

$$\mathbf{A}_\mathrm{I}\boldsymbol{u}_\mathrm{I} = \boldsymbol{\beta}_\mathrm{I} + \boldsymbol{\rho}_\mathrm{I} - \mathbf{A}_{12}\boldsymbol{g}_\Delta \tag{A.14}$$

求めるべき未知ベクトルは，もちろん，$\boldsymbol{u}_\mathrm{I} \in \mathbb{R}^n$ である．

A.4　メイン関数

A.2 節で説明したメッシュデータファイル square1_20.m などを用意しておく．また，例 A.1 で用意した f，g，r の関数形を記述した fem_func1.m を用意しておく（リスト A.2）．

```
function z=fem_func1(id, x, y)
a=0.5; b=0.5; lam=5; r=(x-a).^2+(y-b).^2; w=exp(-5*r);
switch id
    case 0 % right-hand side funtion
    z = w .* (1+4*lam-4*lam^2.*r);
    %z=ones(size(x));
    case 1 % Dirichlet g
    z = w;
    case 2 % Neumann r
    z = 2*lam*(y-b) .* w;
    case 3 % exact solution u
    z = w;
    case 4 % exact solution u_x
    z = -2*lam*(x-a) .* w;
    case 5 % exact solution u_y
    z = -2*lam*(y-b) .* w;
end
```

```
end
```

リスト A.2　fem_func1.m

(A.1) を有限要素法で解く際に，実行するのは，リスト A.3 に示した MATLAB プログラム p1fem1.m である（このようなプログラムをメイン関数という）.

```
function p1fem1(datafile,givenfunc)
% read mesh data
[deg,p,idp,t,idt,e,ide,d,idd] = getmesh(datafile);
% coefficients
%coef_c = 1.0; coef_q = 1.0; coef_kappa = 1.0;
coef_c = 1.0; coef_q = 0.0; coef_kappa = 0.0;
% stiffnes/mass matrices and force term
[K, M, b] = matrix1(p,t,givenfunc);
% Robin BC
[R, rho] = boundary1(p,e,coef_kappa,givenfunc);
% global coefficient matrix
Aglobal = coef_c*K + coef_q*M + coef_kappa*R;
% global right-hand side vector
bglobal = b + rho;
% Dirichlet BC and solving Au=b
[A,b,usol] = dirichlet1(p,d,Aglobal,bglobal,givenfunc);
% 3D plot
figure(5); plot_p1fem(p,t,usol); saveas(5,"p1fem1.pdf");
% output resuts
F1=fopen("p1fem1.res","w"); Z=[p;usol'];
fprintf(F1,"%f %f %f\n",Z); fclose(F1);
end
```

リスト A.3　p1fem1.m

getmesh.m（リスト A.4 を見よ）で，square1_20.dat などから，次のデータを読み取る.

$$\mathtt{p}=\begin{pmatrix} x_1, & \cdots, & x_{\mathrm{np}} \\ y_1, & \cdots, & y_{\mathrm{np}} \end{pmatrix},\quad \mathtt{t}=\begin{pmatrix} \#\mathrm{Q}_1^1, & \cdots, & \#\mathrm{Q}_1^{\mathrm{nt}} \\ \#\mathrm{Q}_2^1, & \cdots, & \#\mathrm{Q}_2^{\mathrm{nt}} \\ \#\mathrm{Q}_3^1, & \cdots, & \#\mathrm{Q}_3^{\mathrm{nt}} \end{pmatrix},$$

$$\mathtt{e}=\begin{pmatrix} \#\mathrm{R}_1^1, & \cdots, & \#\mathrm{R}_1^{\mathrm{ne}} \\ \#\mathrm{R}_2^1, & \cdots, & \#\mathrm{R}_2^{\mathrm{ne}} \end{pmatrix},\quad \mathtt{d}=\begin{pmatrix} \#\mathrm{D}_1^1, & \cdots, & \#\mathrm{D}_1^{\mathrm{nd}} \\ \#\mathrm{D}_2^1, & \cdots, & \#\mathrm{D}_2^{\mathrm{nd}} \end{pmatrix}.$$

idn$_l$ なども配列に入れるが，結局使わない．

p1fem1.m の他の部分については，以下の節で順に説明する．

```
function [deg,p,idp,t,idt,e,ide,d,idd] = getmesh(filename)
FILE1 = fopen(filename,'r'); formatSpec = '%f %f %f %f';
sizeD = [4 Inf]; D = fscanf(FILE1,formatSpec,sizeD); fclose(FILE1);
D=D';np=D(1,1);nt=D(1,2);ne=D(1,3);nd=D(1,4);
deg=[np nt ne nd]; p=D(2:np+1,1:2)'; idp=D(2:np+1,3)';
t=D(np+2:np+nt+1,1:3)'; idt=D(np+2:np+nt+1,4)';
e=D(np+nt+2:np+nt+ne+1,1:2)'; ide=D(np+nt+2:np+nt+ne+1,3)';
d=D(np+nt+ne+2:np+nt+ne+nd+1,1:2)';
idd=D(np+nt+ne+2:np+nt+ne+nd+1,3)';
end
```

リスト A.4　getmesh.m

A.5　係数行列の構成

$\mathbf{K}$, $\mathbf{M}$, $\mathbf{R}$, $\boldsymbol{\beta}$ と $\boldsymbol{\rho}$ を具体的に計算しよう．

まず，$\mathbf{K}$ の各成分 $K_{i,j}$ の構成について説明する．$i,j\in Z$ に対して，$K_{i,j}$ は (A.8) で定められているのだった．もし，P_i と P_j が辺を共有していない（すなわち，同一の要素上の節点でない）ときには，$K_{i,j}=0$ である．P_i と P_j が辺を共有している場合を考える．さらに，P_i と P_j のうち少なくとも一つが Ω の内部に位置するとしよう．このとき，これらを両端点に辺は 2 つの要素 T_e と $T_{e'}$ の共通部分になっており，さらに，

$$K_{i,j}=\underbrace{\iint_{T_e}\nabla\phi_i\nabla\phi_j\ dxdy}_{=K^e_{i,j}}+\underbrace{\iint_{T_{e'}}\nabla\phi_i\nabla\phi_j\ dxdy}_{=K^{e'}_{i,j}} \tag{A.15}$$

と計算できる．したがって，i と j を Z の中で動かして，T_e や $T_{e'}$ の重心座標を用いて，$K_{i,j}$ を計算することができる．

しかし，プログラミングの立場からは，このような“節点志向”でなく，以下で説明する“要素志向”で $\mathbf{K}$ の各成分を構成する方が考えやすい．そのために，(A.7a) を次のように書いてみる：

$$I_1 = \sum_{l=1}^{\mathtt{nt}} \underbrace{\iint_{T_l} \nabla u_h \cdot \nabla v_h \; dxdy}_{=J_l} . \tag{A.16}$$

しばらくの間，T_l を固定して，J_l を計算しよう．T_e の 3 頂点を $\mathrm{P}_i, \mathrm{P}_j, \mathrm{P}_k$ とする．ただし，$\mathrm{P}_i \to \mathrm{P}_j \to \mathrm{P}_k$ が反時計回りの向きになると仮定して，$\mathrm{Q}_1 = \mathrm{P}_i$，$\mathrm{Q}_2 = \mathrm{P}_j$，$\mathrm{Q}_3 = \mathrm{P}_k$ とする．その上で，T_e に対応する重心座標を $\{\lambda_\nu = \lambda_{l,\nu} = \lambda_{T_l,\nu}\}_{\nu=1}^3$ とする．$\hat{u}$ と $\hat{v}$ と，それぞれ，u_h と v_h の T_e 上への制限として，

$$\hat{u} = \sum_{\mu=1}^{3} \hat{u}_\mu \lambda_\mu, \quad \hat{v} = \sum_{\nu=1}^{3} \hat{v}_\nu \lambda_\nu$$

と書こう．ここで，$\hat{u}_1 = \hat{u}(Q_1) = u_h(\mathrm{P}_i)$ などどおいている．

そうすると，J_e は，

$$J_l = \sum_{\nu=1}^{3} \sum_{\mu=1}^{3} \hat{v}_\nu \hat{u}_\mu \underbrace{\iint_{T_l} \nabla\lambda_\mu \cdot \nabla\lambda_\nu \; dxdy}_{=\hat{K}_{\nu,\mu}} \tag{A.17}$$

となる．各重心座標を $\lambda_\nu = a_\nu + b_\nu x + c_\nu y$ と書いておく．??節の??を用いれば，$\hat{\mathbf{K}}^l = (\hat{K}_{\nu,\mu})_{1\le\nu,\mu\le3}$ は，

$$\hat{\mathbf{K}}^l = \frac{1}{4|T_l|} \left[\begin{pmatrix} b_2 - b_3 \\ b_3 - b_1 \\ b_1 - b_2 \end{pmatrix} \begin{pmatrix} b_2 - b_3 \\ b_3 - b_1 \\ b_1 - b_2 \end{pmatrix}^{\mathrm{T}} + \begin{pmatrix} a_3 - a_2 \\ a_1 - a_3 \\ a_2 - a_1 \end{pmatrix} \begin{pmatrix} a_3 - a_2 \\ a_1 - a_3 \\ a_2 - a_1 \end{pmatrix}^{\mathrm{T}} \right] \tag{A.18}$$

と計算できる．

ここで，(A.7a)，(A.15)，(A.16) と (A.17) を比べてみよう．$\hat{u}_1 = u_i, \hat{u}_2 = u_j, \hat{v}_1 = v_i, \hat{v}_2 = v_j$ であるから，$K^e_{i,j} = \hat{K}_{i,j}$ であることがわかる．$K_{i,j}$ の値を求めるには、$K^{e'}_{i,j}$ の値も必要であるが，これは，$T_{e'}$ に対する $J_{e'}$ を計算する際に算出される．

- $\mathbf{K} = \boldsymbol{O}$ としておく．

- $e = 1, 2, \ldots, E$ に対して，$\hat{\mathbf{K}}^e$ の各成分を計算して，$\mathbf{K} = (K_{i,j})_{1 \le i,j \le N}$ の各成分の値を次のように更新する：

$$\begin{array}{lll} K_{i,i} \leftarrow K_{i,i} + \hat{K}_{1,1} & K_{i,j} \leftarrow K_{i,j} + \hat{K}_{1,2} & K_{i,k} \leftarrow K_{i,k} + \hat{K}_{1,3} \\ K_{j,i} \leftarrow K_{j,i} + \hat{K}_{2,1} & K_{j,j} \leftarrow K_{j,j} + \hat{K}_{2,2} & K_{j,k} \leftarrow K_{j,k} + \hat{K}_{2,3} \\ K_{k,i} \leftarrow K_{k,i} + \hat{K}_{3,1} & K_{k,j} \leftarrow K_{k,j} + \hat{K}_{3,2} & K_{k,k} \leftarrow K_{k,k} + \hat{K}_{3,3}. \end{array}$$

$\mathbf{M}$ の導出も同様に考えれば良い．(A.5) で定めた I_2 を，

$$I_2 = \sum_{l=1}^{E} \sum_{\nu=1}^{2} \sum_{\mu=1}^{3} \hat{v}_\nu \hat{u}_\mu \underbrace{\iint_{T_l} \lambda_\mu \lambda_\nu \, dxdy}_{=\hat{M}_{\nu,\mu}}$$

と書いておくと，(**??**) により，$\hat{\mathbf{M}}^l = (\hat{M}_{\nu,\mu})_{1 \le \nu,\mu \le 3}$ は，

$$\hat{\mathbf{M}}^l = \frac{|T_l|}{12} \begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{pmatrix}$$

と計算できる．$\mathbf{K}$ のときと同様に，$\hat{M}_{\nu,\mu}$ を対応する $M_{i,j}$ に加えていけば良い．

リスト A.5 に示した `matrix1.m` において，$\mathbf{K}$ と $\mathbf{M}$ を計算する．`matrix1.m` では，各 $l = 1, 2, \ldots,$ `nt` に対して，`tlocal` $= (\#\mathrm{Q}_1^l, \#\mathrm{Q}_2^l, \#\mathrm{Q}_3^l)^{\mathrm{T}}$，`x` $=$ `p(1,tlocal)` $= (x_s, x_t, x_v)$，`y` $=$ `p(2,tlocal)` $= (y_s, y_t, y_v)$ とする．ただし，$\#\mathrm{Q}_1^l = s$，$\#\mathrm{Q}_2^l = t$，$\#\mathrm{Q}_3^l = v$ としている．そして，リスト A.6 の `P1grad.m` で (A.18) の計算で使う b_ν や c_ν を計算し，$\hat{\mathbf{K}}^l$，$\hat{\mathbf{M}}^l$ の各成分を計算する．それらを用いて，対応する $\mathbf{K}$，$\mathbf{M}$ の成分を計算している．

```
function [K, M, force] = matrix1(p,t,givenfunc)
np = size(p,2); nt = size(t,2);
K = sparse(np,np); M = sparse(np,np); force = zeros(np,1);
for l = 1:nt
  tlocal=t(1:3,l); x=p(1,tlocal); y=p(2,tlocal); [area,b,c]=P1grad(x,y);
  % stiffness matrix
```

```
  Klocal=(b*b'+c*c')*area; K(tlocal,tlocal)=K(tlocal,tlocal)+ Klocal;
  % mass matrix
  Mlocal = [2 1 1; 1 2 1; 1 1 2]/12*area;
  M(tlocal,tlocal)=M(tlocal,tlocal)+Mlocal;
  %force term
  flocal=Mlocal*givenfunc(0,x,y)'; force(tlocal)=force(tlocal)+flocal;
end
end
```

リスト A.5　matrix1.m

```
function [area,b,c] = P1grad(x,y)
area=polyarea(x,y);
b=[y(2)-y(3); y(3)-y(1); y(1)-y(2)]/2/area;
c=[x(3)-x(2); x(1)-x(3); x(2)-x(1)]/2/area;
end
```

リスト A.6　P1grad.m

A.6　右辺ベクトルの構成と Robin 境界条件の取り込み

(A.7b) の I_4 を計算して，$\boldsymbol{\beta}$ を求めよう．いままでの表記を用いると，T_l 上では，$f_h = \sum_{\nu=1}^{3} \hat{f}_\nu \lambda_\nu$，$v_h = \sum_{\nu=1}^{3} \hat{v}_\nu \lambda_\nu$ と書けているのだから，

$$I_4 = \sum_{l=1}^{\mathtt{nt}} \iint_{T_l} f_h v_h \, dxdy = \sum_{l=1}^{\mathtt{nt}} \sum_{\nu=1}^{3} \sum_{\mu=1}^{3} \hat{v}_\nu \hat{f}_\mu \hat{M}^l_{\nu,\mu}$$

となる．ここで，$\hat{f}_1 = f(Q_1) = f(\mathrm{P}_i)$ などを意味する（$f_h(Q_1) = f(Q_1)$ などに注意せよ）．したがって，

$$\hat{\boldsymbol{\beta}}^l = \hat{\mathbf{M}}^l \begin{pmatrix} f(Q_1) \\ f(Q_2) \\ f(Q_3) \end{pmatrix}$$

を計算して，対応する $\boldsymbol{\beta}$ の成分に加えていけば良い．これを行うのが force1.m である．リスト A.7 を見よ．この中で，$f(x,y)$ の値は，givenfunc(0,x,y) で参照される．

```
function [force] = force1(p,t,givenfunc)
np = size(p,2); nt = size(t,2); force = zeros(np,1);
for l = 1:nt
  tlocal=t(1:3,l); x=p(1,tlocal); y=p(2,tlocal); [area,b,c]=P1grad(x,y);
  Mlocal = [2 1 1; 1 2 1; 1 1 2]/12*area;
  %force term
  %flocal=givenfunc(0,x,y)'/3*area;
  flocal=Mlocal*givenfunc(0,x,y)'; force(tlocal)=force(tlocal)+flocal;
end
end
```

リスト A.7　force1.m

次に，Robin 境界条件 (A.1b) からの寄与となる

$$I_3 = \sum_{k=1}^{\mathrm{ne}} \int_{e_k} u_h v_h \ dS, \quad I_5 = \sum_{k=1}^{\mathrm{ne}} \int_{e_k} \underbrace{(\kappa g_h + r_h)}_{=z_h} v_h \ dS$$

について考える（e_k はすべて線分である）．しばらく k は固定して，e_k の両端の節点を $\mathrm{P}_i, \mathrm{P}_j$，長さを l_k する．e_k は境界上に位置しているので，これを一辺として含む要素 T_l は唯一である．e_k を $(x,y)^{\mathrm{T}} = (1-\xi)\overrightarrow{\mathrm{OP}_i} + \xi\overrightarrow{\mathrm{OP}_j}$ $(0 \le \xi \le l)$ とパラメータ ξ で表示すると，

$$\int_{e_k} u_h v_h \ dS = \int_0^l \hat{u}(\xi)\hat{v}(\xi) \ d\xi$$

と書ける（こうなるように P_i と P_j に向きついている）．ここで，例えば，$\hat{u}$ は，$[0,l]$ 上の一次関数で $\hat{u}(0) = u_h(\mathrm{P}_i)$，$\hat{u}(l) = u_h(\mathrm{P}_j)$ となるものである．この積分は，“e_k 上の重心座標”

$$\eta_1(\xi) = \frac{l-\xi}{l}, \quad \eta_2(\xi) = \frac{\xi}{l}$$

を導入することにより，

$$I_3 = \sum_{k=1}^{\mathrm{ne}} \sum_{\nu=1}^{2} \sum_{\mu=1}^{2} \hat{v}_\nu \hat{u}_\mu \underbrace{\int_0^l \eta_j \eta_i \ d\xi}_{=\hat{R}_{\nu,\mu}}, \quad I_5 = \sum_{k=1}^{\mathrm{ne}} \sum_{\nu=1}^{2} \sum_{\mu=1}^{2} \hat{v}_\nu \hat{z}_\mu \hat{R}_{\nu,\mu}$$

となり，さらに，$\hat{\mathbf{R}}^k=(\hat{R}_{\nu,\mu})_{1\le\nu,\mu\le2}$ と $\hat{\boldsymbol{\rho}}^k=(\hat{\rho}_\nu)_{1\le\nu\le2}$ は，

$$\hat{\mathbf{R}}^k=\frac{|e_k|}{6}\begin{pmatrix}2&1\\1&2\end{pmatrix},\quad \hat{\boldsymbol{\rho}}^k=\hat{\mathbf{R}}^k\begin{pmatrix}z(\mathrm{R}_1)\\z(\mathrm{R}_2)\end{pmatrix}$$

と計算できる．これらを，それぞれ，対応する $\mathbf{R}$ と $\boldsymbol{\beta}$ の成分に加えていけば良いのは今までと同じである．リスト A.8 の force1.m でこれを行う．この中で，$g(x,y)$ の値は givenfunc(1,x,y) で，$r(x,y)$ の値は givenfunc(2,x,y) で参照される．

```
function [R,rho] = boundary1(p,e,coef_kappa,givenfunc)
np = size(p,2); ne = size(e,2);
R = sparse(np,np); rho = zeros(np,1);
for k = 1:ne
  elocal = e(1:2,k); x = p(1,elocal); y = p(2,elocal);
  length = sqrt((x(1)-x(2))^2+(y(1)-y(2))^2);
  % for mass matrix
  Rlocal=[2 1; 1 2]*length/6; R(elocal,elocal)=R(elocal,elocal)+Rlocal;
  % for right-hand side vector
  % xc = mean(x); yc = mean(y); rlocal = tmp*[1; 1]*length/2;
  tmp = coef_kappa*givenfunc(1,x,y)+givenfunc(2,x,y);
  rlocal = Rlocal*tmp'; rho(elocal) = rho(elocal) + rlocal;
end
```

リスト A.8　boundary1.m

A.7　Dirichlet 境界条件の取り込みと連立一次方程式の解法

ここまでの計算で，係数行列 $c\mathbf{K}+q\mathbf{M}+\kappa\mathbf{R}$ と右辺ベクトル $\boldsymbol{\beta}+\boldsymbol{\rho}$ が得られている．Dirichlet 境界条件 (A.1c) を取り込むために，$1,2,\ldots,$ np を，Dirichlet 境界条件を課す節点番号の集合 Δ とそれ以外の集合 Λ に分けて，(A.11) のように A_{I} などを作る．$\boldsymbol{u}_\Delta=\boldsymbol{g}_\Delta$ は，Dirichlet 境界条件から直接に計算できる．あとは，連立一方程式 (A.14) を解いて $\boldsymbol{u}_{\mathrm{I}}$ を求めれば，最終的な答えである $\boldsymbol{u}$ が得られるのである．

これを行うのがリスト A.9 の dirichlet1.m である．fixed が Δ，free が Λ を表している．図 A.7 のような形式で，結果が p1fem1.res というファイル

表 A.2　結果ファイル (p1fem1.res) の形式

x_1	y_1	$u_h(\mathrm{P}_1)$
x_2	y_2	$u_h(\mathrm{P}_2)$
$\vdots$	$\vdots$	$\vdots$
x_{np}	y_{np}	$u_h(\mathrm{P}_{\mathrm{np}})$

に書き込まれる.

```
function [A,b,u] = dirichlet1(p,d,A,b,givenfunc)
np = size(p,2); u = zeros(np,1);
% boundary and interior nodes
fixed=unique(d); free=setdiff([1:np],fixed);
% set boundary value
g=zeros(np,1); x=p(1,fixed); y=p(2,fixed); g(fixed)=givenfunc(1,x,y);
% re-allocate A and b
b = b(free)-A(free,fixed)*g(fixed); A = A(free,free);
% solving Au = b
u(fixed) = g(fixed); u(free) = A\b;
end
```

リスト A.9　dirichlet1.m

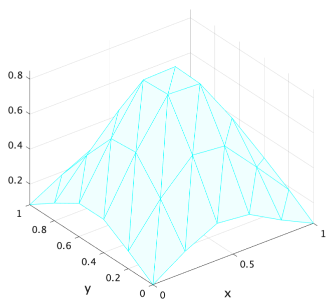
A.8　解の可視化

得られた解を plot_p1fem.m で可視化する．結果は図 A.5 の通りである．

```
function plot_p1fem(p,t,u)
nt=size(t,2); X=zeros(3*nt,1); Y=zeros(3*nt,1); uu=zeros(3*nt,1);
%
i=t(1,:); j=t(2,:); k=t(3,:);
X(1:3:end)=p(1,i); X(2:3:end)=p(1,j); X(3:3:end)=p(1,k);
Y(1:3:end)=p(2,i); Y(2:3:end)=p(2,j); Y(3:3:end)=p(2,k);
uu(1:3:end)=u(i); uu(2:3:end)=u(j); uu(3:3:end)=u(k);
% using only cyan
%trisurf(reshape([1:3*nt],3,nt)',X,Y,uu,'FaceColor',[0.95 1 1],'EdgeColor
     ',[0 1 1]);
% full color
trisurf(reshape([1:3*nt],3,nt)',X,Y,uu);
%colormap gray; view(49,47);
```

```
% axes settings
xlabel('x','FontSize',14); ylabel('y','FontSize',14);
%axis vis3d equal; grid on;
end
```

リスト A.10　plot_p1fem.m



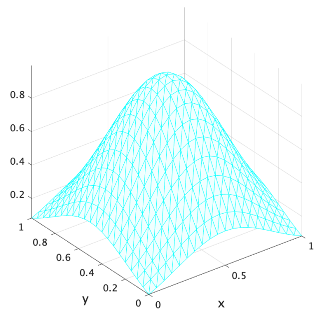


図 A.5　square1_5.dat と square2_20.dat を使って，例 A.1 のデータに対して，p1fem1.m で (A.5) を解いた結果.

参考文献

[1] 川原睦人, 有限要素法流体解析, 日科技連, 1985 年.
[2] 菊地文雄, 有限要素法概説 [新訂版], サイエンス社, 1999 年.
[3] 森正武, FORTRAN77 数値計算プログラミング, 岩波書店, 1987 年.
[4] 日本計算工学会・編集, 有限要素法による流れのシミュレーション [第 3 版]—OpenMP に基づく Fortran ソースコード付, 丸善出版, 2017 年.
[5] M. G. Larson and F. Bengzon, *The Finite Element Method: Theory, Implementation, and Applications*, Springer, 2013.