

# 応用数値解析特論 第5回

～1次元 Poisson 方程式に対する有限要素法, 2次元 Poisson 方程式に  
対する有限要素法～

かつらだ まさし  
桂田 祐史

<http://m-katsurada.sakura.ne.jp/ana2024/>

2024年5月14日

# 目次

- 1 本日の内容
- 2 1次元の有限要素法
  - モデル問題とその弱定式化
  - 有限要素解の定義
    - 有限要素への分割
    - 区分的 1 次多項式の空間の基底関数
    - 有限要素空間, 有限要素解
    - 蛇足の話
  - 有限要素解を求めるアルゴリズム
    - 長さ座標
    - 弱形式の分割
    - 要素係数行列, 要素自由項ベクトル
    - 直接剛性法 (近似方程式の組み立て)
    - 具体的にすることのまとめ
  - 連立 1 次方程式の具体形
  - サンプル・プログラム fem1d.c
    - 問題
    - プログラムの解説
    - 実験
    - 参考: 昔の練習問題

- 2 2次元の有限要素法

# 本日の内容

- いきなり 2 次元の有限要素法と格闘する前に、1 次元の場合 (菊地 [?] の 3 章) を解説する。1 次元であっても、多くの重要な考え方に触れることができる。
- 2 次元 Poisson 方程式に対する有限要素法 (菊地 [?] の 4 章):  
大筋は前回の 1 次元 Poisson 方程式と同様である。
  - 三角形要素上の 1 次関数を重み座標  $L_i$  を導入する。
  - 要素係数行列, 要素自由項ベクトルの公式が求まる。
  - 2 次元でも直接剛性法が出来る。やや技巧的な部分もあるが、学ぶ価値がある。
- コーディングの詳細は省略する。

## 4 1次元の有限要素法

有限要素法が実際に利用されるのは、空間2次元、3次元の問題がほとんどであるが、ここでは計算手順の概要 (特に[直接剛性法](#)) を理解するために、1次元の Poisson 方程式の境界値問題に対する有限要素法の説明を行う。

このすぐ後に説明する2次元の場合を分かりやすくするためという趣旨である (いきなり全部やると大変)。

以上は、菊地 [?] を踏襲したものだが、私自身の経験から「分かりやすい」と思っている。

## 4.1 モデル問題とその弱定式化

問題 (P) の 1 次元版である、常微分方程式の境界値問題

$$(1) \quad \begin{cases} -u''(x) = f(x) & (x \in (0, 1)) \\ u(0) = \alpha, \quad u'(1) = \beta \end{cases}$$

を考える。ここで  $f$  は  $(0, 1)$  上定義された既知の関数、 $\alpha$  と  $\beta$  は既知の実定数である。(要するに  $n = 1$ ,  $\Omega = (0, 1)$ ,  $\Gamma = \{0, 1\}$ ,  $\Gamma_1 = \{0\}$ ,  $\Gamma_2 = \{1\}$ ,  $g_1 = \alpha$ ,  $g_2 = \beta$  である。)

$$X_{g_1} := \{w \in H^1(I) \mid w(0) = \alpha\}, \quad X := \{v \in H^1(I) \mid v(0) = 0\},$$

$$\langle u, v \rangle := \int_0^1 u'(x)v'(x) dx, \quad (f, v) := \int_0^1 f(x)v(x) dx$$

とおくと、(1) の弱解とは、弱形式

$$(2) \quad \langle u, v \rangle = (f, v) + \beta v(1) \quad (v \in X)$$

を満たす  $u \in X_{g_1}$  のことである。

## 4.2 有限要素解の定義 要点

要点はすでに予告してある。

有限要素法は区分的多項式を試行関数、試験関数に用いる Ritz-Galerkin 法である。

一般に、 $X_{g_1}$ ,  $X$  の有限次元近似  $\hat{X}_{g_1}$ ,  $\hat{X}$  を定めて、(1つの) Ritz-Galerkin 解が定義される。

区分的多項式というものを定義して、それを用いて適切に  $\hat{X}_{g_1}$ ,  $\hat{X}$  を定めることで有限要素解が定義できる。

## 4.2.1 有限要素, 区分1次多項式

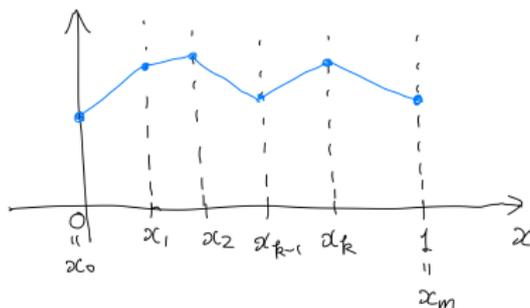
区間  $[0, 1]$  を  $m$  個の小区間に分割する:

$$0 = x_0 < x_1 < x_2 < \cdots < x_m = 1.$$

$x_i$  ( $0 \leq i \leq m$ ) を**節点 (node)** と呼ぶ。

区間  $e_k := [x_{k-1}, x_k]$  ( $k = 1, \dots, m$ ) を**有限要素 (finite element)** と呼ぶ。

区間  $[0, 1]$  全体で連続で、各要素  $e_k$  上で1次関数に等しい関数を**区分的1次多項式**と呼び、区分的1次多項式の全体を  $\tilde{X}$  と表す。  $\dim \tilde{X} = m + 1$  である。



試行関数 (近似解)  $\hat{u}$ , 試験関数  $\hat{v}$  として、区分的1次多項式を採用しよう。言い換えると、試行関数の空間  $\hat{X}_{g_1}$ , 試験関数の空間  $\hat{X}$  は、 $\hat{X}_{g_1} \subset \tilde{X}$ ,  $\hat{X} \subset \tilde{X}$  を満たすよう定める。

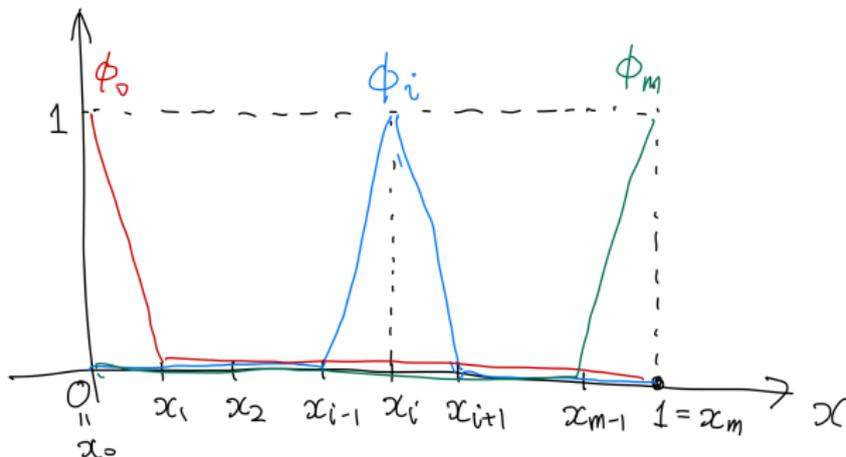
## 4.2.2 区分的1次多項式の空間の基底関数

$\tilde{X}$  の基底関数として、以下に定義する  $\{\phi_i\}_{i=0}^m$  を採用できる。

—  $\phi_i$  の定義 —

$\phi_i$  は区分的1次多項式で、 $x_i$  では1, 他の節点  $x_j$  ( $j \neq i$ ) では0という値を取る:

- ❶  $\phi_i \in C[0, 1]$
- ❷  $(\forall k \in \{1, \dots, m\}) (\exists p, q \in \mathbb{R}) (\forall x \in e_k) \phi_i(x) = px + q$
- ❸  $\phi_i(x_j) = \delta_{ij}$  ( $j = 0, 1, \dots, m$ ).



## 4.2.2 区分的1次多項式の空間の基底関数

### 補題 5.1 (基底関数 $\phi_i$ の性質)

$w_i \in \mathbb{R}$  ( $0 \leq i \leq m$ ) に対して

$$\hat{w}(x) := \sum_{i=0}^m w_i \phi_i(x)$$

とおくと

$$\hat{w}(x_j) = w_j \quad (0 \leq j \leq m).$$

すなわち  $\hat{w}$  の ( $\phi_j$  に関する) 係数  $w_j$  は、節点  $x_j$  における関数値である。

### 証明

任意の  $j \in \{0, 1, \dots, m\}$  に対して

$$\hat{w}(x_j) = \sum_{i=0}^m w_i \phi_i(x_j) = \sum_{i=0}^m w_i \delta_{ij} = w_j \delta_{jj} = w_j.$$

### 定理 5.2

区分的1次多項式の全体  $\tilde{X}$  は、 $m+1$ 次元の線形空間で、 $\{\phi_j\}_{0 \leq j \leq m}$  は基底をなす。

## 4.2.3 有限要素空間, 有限要素解

試行関数の空間  $\hat{X}_{g_1}$  と試験関数の空間  $\hat{X}$  として、次のものを採用する。

$$\hat{X}_{g_1} := \left\{ \hat{w} \in \tilde{X} \mid \hat{w}(0) = \alpha \right\}, \quad \hat{X} := \left\{ \hat{v} \in \tilde{X} \mid \hat{v}(0) = 0 \right\}.$$

基底関数を用いて表すと

$$(3) \quad \hat{X}_{g_1} = \left\{ \alpha\phi_0 + \sum_{i=1}^m a_i\phi_i \mid a_i \in \mathbb{R} \quad (i = 1, 2, \dots, m) \right\},$$

$$(4) \quad \hat{X} = \left\{ \sum_{i=1}^m a_i\phi_i \mid a_i \in \mathbb{R} \quad (i = 1, 2, \dots, m) \right\}.$$

このとき定まる Ritz-Galerkin 解を  $\hat{u}$  とする。すなわち  $\hat{u}$  は

$$(5a) \quad \hat{u} \in \hat{X}_{g_1},$$

$$(5b) \quad \langle \hat{u}, \hat{v} \rangle = (f, \hat{v}) + \beta\hat{v}(1) \quad (\hat{v} \in \hat{X}).$$

を満たす。この  $\hat{u}$  を区分的 1 次要素 (P1 要素) を用いた **有限要素解** と呼ぶ。

## 4.2.4 蛇足の話

(実は必要がないのだけれど) 式で書くと、 $1 \leq i \leq m-1$  に対しては

$$\phi_i(x) = \begin{cases} \frac{x - x_{i-1}}{x_i - x_{i-1}} & (x \in [x_{i-1}, x_i]) \\ \frac{x_{i+1} - x}{x_{i+1} - x_i} & (x \in [x_i, x_{i+1}]) \\ 0 & (\text{その他}), \end{cases}$$

$i = 0$  に対しては

$$\phi_0(x) = \begin{cases} \frac{x_1 - x}{x_1 - x_0} & (x \in [x_0, x_1]) \\ 0 & (\text{その他}), \end{cases}$$

$i = m$  に対しては

$$\phi_m(x) = \begin{cases} \frac{x - x_{m-1}}{x_m - x_{m-1}} & (x \in [x_{m-1}, x_m]) \\ 0 & (\text{その他}). \end{cases}$$

このように式で書けるけれど、そうしてもほとんど使いみちがない。

$\phi_i(x_j) = \delta_{ij}$  を満たす連続な区分的 1 次関数ということと、グラフのイメージを覚えた方がよい。

## 4.3 有限要素解を求めるアルゴリズム

前項までに有限要素解  $\hat{u}$  は定義された。 $\hat{u}$  は

$$\hat{u} = \alpha\phi_0 + \sum_{i=1}^m u_i\phi_i$$

と表すことができるが、 $u_i$  を並べた

$$\mathbf{u}^* = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{pmatrix} = (u_1, u_2, \dots, u_m)^\top$$

について、連立1次方程式  $\mathbf{A}\mathbf{u}^* = \mathbf{f}^*$  が得られることは、既に(原理的)に分かっている。

しかし、 $A = (\langle\phi_j, \phi_i\rangle)$  や  $\mathbf{f}^*$  を実際に計算するのは、やり方を知らないと案外難しい。有限要素法ではこのあたりが良く整備されていて、明快なアルゴリズムが確立されている。

有限要素  $e_k$  ごとに、要素係数行列、要素自由項ベクトルというものを求め、それから  $A$  と  $\mathbf{f}$  を“組み立てる”。後半の操作を構造力学の用語にちなみ直接剛性法と呼ぶ(直接合成法ではない)。…… 少し長い込み入った話になる。

## 4.3.1 長さ座標

各要素  $e_k = [x_{k-1}, x_k]$  において

$$L_0(x_{k-1}) = 1, \quad L_0(x_k) = 0,$$

$$L_1(x_{k-1}) = 0, \quad L_1(x_k) = 1$$

で定まる 1 次関数  $L_0, L_1$  を  $e_k$  の長さ座標と呼ぶ (グラフを自分で描こう)。

$$(6) \quad L_0 + L_1 \equiv 1$$

が成り立つ。

また節点の座標  $x_{k-1}, x_k$  を用いて

$$(7) \quad L_0(x) = \frac{x_k - x}{x_k - x_{k-1}}, \quad L_1(x) = \frac{x - x_{k-1}}{x_k - x_{k-1}}$$

と具体的に表わせる (こちらは  $\phi_i$  と違って、後でちょっと用いる)。

$\hat{w} \in \tilde{X}$  に対して  $w_i := \hat{w}(x_i)$  とおくと、次式が成り立つ:

$$(8) \quad \hat{w}(x) = w_{k-1}L_0(x) + w_kL_1(x) = \sum_{j=0}^1 w_{k+j-1}L_j \quad (x \in e_k).$$

(たった 2 項なのに  $\sum$  を使うのは大げさなようだけれど…)

## 4.3.2 弱形式の分割

各要素  $e_k$  について

$$(9) \quad \langle u, v \rangle_{e_k} := \int_{x_{k-1}}^{x_k} u'(x)v'(x)dx, \quad (f, v)_{e_k} := \int_{x_{k-1}}^{x_k} f(x)v(x)dx$$

とおくと、Galerkin 法の弱形式

$$(再掲 2) \quad \langle \hat{u}, \hat{v} \rangle = (f, \hat{v}) + \beta \hat{v}(1) \quad (\hat{v} \in \hat{X})$$

は

$$(10) \quad \sum_{k=1}^m \langle \hat{u}, \hat{v} \rangle_{e_k} = \sum_{k=1}^m (f, \hat{v})_{e_k} + \beta \hat{v}(1) \quad (\hat{v} \in \hat{X})$$

と書き直せる ( $\because \int_0^1 = \sum_{k=1}^m \int_{e_k}$ )。

### 4.3.3 要素係数行列, 要素自由項ベクトル

このスライドの目標:  $\langle \hat{u}, \hat{v} \rangle_{e_k}$ ,  $(f, \hat{v})_{e_k}$ ,  $\hat{v}(1)$  を成分で表す。

$$\begin{aligned}\langle \hat{u}, \hat{v} \rangle_{e_k} &= \left\langle \sum_{j=0}^1 u_{k+j-1} L_j, \sum_{i=0}^1 v_{k+i-1} L_i \right\rangle_{e_k} = \sum_{j=0}^1 \sum_{i=0}^1 u_{k+j-1} v_{k+i-1} \langle L_j, L_i \rangle_{e_k} \\ &= \sum_{i=0}^1 \sum_{j=0}^1 v_{k+i-1} A_{ij}^{(k)} u_{k+j-1},\end{aligned}$$

ただし

$$A_{ij}^{(k)} := \langle L_j, L_i \rangle_{e_k}.$$

一方、

$$(f, \hat{v})_{e_k} = \left( f, \sum_{j=0}^1 v_{k+j-1} L_j \right)_{e_k} = \sum_{j=0}^1 v_{k+j-1} (f, L_j)_{e_k} = \sum_{j=0}^1 v_{k+j-1} f_j^{(k)},$$

ただし

$$f_j^{(k)} := (f, L_j)_{e_k}.$$

また  $\hat{v}(1) = v_m$  より

$$\beta \hat{v}(1) = \beta v_m.$$

## 2次形式と行列を用いた表記

$x_1, \dots, x_m$  に対する “純粹の2次式”

$$(11) \quad \sum_{i,j=1}^m a_{ij}x_i x_j$$

を2次形式とよぶ

$A := (a_{ij})$  とおくと、 $A$  は  $m$  次正方行列であるが

$$\sum_{i,j=1}^m a_{ij}x_i x_j = \sum_{i=1}^m \left( x_i \sum_{j=1}^m a_{ij}x_j \right) = Ax \text{ と } x \text{ の内積} = x^T Ax.$$

ここで  $^T$  は転置 (transpose) を表す。 $\mathbf{b}^T \mathbf{a} = \mathbf{a} \cdot \mathbf{b}$  である。

$A$  を2次形式 (11) の係数行列とよぶ。普通は対称行列を選ぶ。

((11) という書き方には冗長性があるので、 $a_{ij} = a_{ji}$  という条件を課すことができる。例えば  $3x_1x_2 + x_2x_1 = 2x_1x_2 + 2x_2x_1$  と書き直せる。)

### 4.3.3 要素係数行列, 要素自由項ベクトル

このスライドの目標:  $\langle \hat{u}, \hat{v} \rangle_{e_k}$ ,  $(f, \hat{v})_{e_k}$ ,  $\hat{v}(1)$  をベクトル、行列で表す。

そこで

$$(12a) \quad \mathbf{u}_k := \begin{pmatrix} u_{k-1} \\ u_k \end{pmatrix}, \quad \mathbf{v}_k := \begin{pmatrix} v_{k-1} \\ v_k \end{pmatrix}, \quad \mathbf{f}_k := \begin{pmatrix} f_0^{(k)} \\ f_1^{(k)} \end{pmatrix},$$

$$(12b) \quad A_k := \begin{pmatrix} A_{00}^{(k)} & A_{01}^{(k)} \\ A_{10}^{(k)} & A_{11}^{(k)} \end{pmatrix},$$

$$(12c) \quad \mathbf{g}_m := \begin{pmatrix} 0 \\ \beta \end{pmatrix}$$

とおくと、次式が得られる。

$$(13) \quad \langle \hat{u}, \hat{v} \rangle_{e_k} = \mathbf{v}_k^\top A_k \mathbf{u}_k, \quad (f, \hat{v})_{e_k} = \mathbf{v}_k^\top \mathbf{f}_k \quad (k = 1, \dots, m), \quad \beta \hat{v}(1) = \mathbf{v}_m^\top \mathbf{g}_m.$$

$\mathbf{u}_k$ ,  $\mathbf{v}_k$  は要素節点パラメーター・ベクトル、 $\mathbf{f}_k$  は要素自由項ベクトル、 $A_k$  は要素係数行列と呼ばれる。

### 4.3.3 要素係数行列, 要素自由項ベクトル

プログラムを読み書きするときのために、実際に  $A_k$ ,  $f_k$  を求めよう。

$$\langle L_i, L_j \rangle_{e_k} = \int_{x_{k-1}}^{x_k} L_i'(x) L_j'(x) dx = \int_{x_{k-1}}^{x_k} \frac{\varepsilon}{(x_k - x_{k-1})^2} dx = \frac{\varepsilon}{x_k - x_{k-1}},$$
$$\varepsilon := \begin{cases} 1 & (i = j) \\ -1 & (i \neq j) \end{cases}$$

であるから、

$$(14) \quad A_k = \frac{1}{x_k - x_{k-1}} \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}.$$

一方

$$f_j^{(k)} = (f, L_j)_{e_k} = \int_{x_{k-1}}^{x_k} f(x) L_j(x) dx \quad (j = 0, 1).$$

この右辺の積分は、 $f$  に応じて何らかの手段 (例えば数値積分) で計算しておく。

### 4.3.3 要素係数行列, 要素自由項ベクトル

(このスライドはスキップしても良い。)

$f$  が複雑な関数の場合は、 $f_j^{(k)}$  は厳密に計算できないかもしれないが、節点での値さえ分かれば近似計算 (数値積分) は難しくない。例えば

$$f \doteq f(x_{k-1})L_0 + f(x_k)L_1 \quad (\text{on } e_k)$$

と 1 次補間近似を利用して

$$f_j^{(k)} = (f, L_j)_{e_k} \doteq (f(x_{k-1})L_0 + f(x_k)L_1, L_j)_{e_k} = f(x_{k-1})(L_0, L_j)_{e_k} + f(x_k)(L_1, L_j)_{e_k}.$$

$x = x_{k-1} + (x_k - x_{k-1})t$  ( $0 \leq t \leq 1$ ) と変数変換して

$$(L_0, L_0)_{e_k} = (L_1, L_1)_{e_k} = (x_k - x_{k-1}) \int_0^1 t^2 dt = \frac{x_k - x_{k-1}}{3},$$

$$(L_0, L_1)_{e_k} = (L_1, L_0)_{e_k} = (x_k - x_{k-1}) \int_0^1 t(1-t) dt = \frac{x_k - x_{k-1}}{6}.$$

ゆえに

$$(15) \quad \mathbf{f}_k \doteq \frac{(x_k - x_{k-1})}{6} \begin{pmatrix} 2f(x_{k-1}) + f(x_k) \\ f(x_{k-1}) + 2f(x_k) \end{pmatrix}.$$

### 4.3.3 要素係数行列, 要素自由項ベクトル

以下の話で必要になる式を再掲しておく。

弱形式は次のように書き直される。

$$(16) \quad \sum_{k=1}^m \langle \hat{u}, \hat{v} \rangle_{e_k} = \sum_{k=1}^m (f, \hat{v})_{e_k} + \beta \hat{v}(1) \quad (\hat{v} \in \hat{X})$$

$\mathbf{u}_k = \begin{pmatrix} u_{k-1} \\ u_k \end{pmatrix}$ ,  $\mathbf{v}_k = \begin{pmatrix} v_{k-1} \\ v_k \end{pmatrix}$ , さらに  $\mathbf{f}_k$ ,  $A_k$ ,  $\mathbf{g}_m$  を適当に定義すると

$$(17) \quad \langle \hat{u}, \hat{v} \rangle_{e_k} = \mathbf{v}_k^\top A_k \mathbf{u}_k, \quad (f, \hat{v})_{e_k} = \mathbf{v}_k^\top \mathbf{f}_k \quad (k = 1, \dots, m),$$

$$(18) \quad \beta \hat{v}(1) = \mathbf{v}_m^\top \mathbf{g}_m.$$

(16) に代入して

$$(19) \quad \sum_{k=1}^m \mathbf{v}_k^\top A_k \mathbf{u}_k = \sum_{k=1}^m \mathbf{v}_k^\top \mathbf{f}_k + \mathbf{v}_m^\top \mathbf{g}_m.$$

### 4.3.4 直接剛性法 (近似方程式の組み立て)

(12a), (12b), (12c) で与えたベクトル、行列を  $m+1$  次元に拡大する。まず

$$\mathbf{u} := \begin{pmatrix} u_0 \\ \vdots \\ u_m \end{pmatrix}, \quad \mathbf{v} := \begin{pmatrix} v_0 \\ \vdots \\ v_m \end{pmatrix}$$

とおく。繰り返しになるが、 $u_i = \hat{u}(x_i)$ ,  $v_i = \hat{v}(x_i)$ .

$\mathbf{f}_k$ ,  $A_k$ ,  $\mathbf{g}_m^*$  については、0 を補って、 $\mathbb{R}^{m+1}$  や  $M(m+1; \mathbb{R})$  の元に拡大する:

$$\mathbf{f}_k^* := \begin{pmatrix} 0 \\ \vdots \\ 0 \\ f_0^{(k)} \\ f_1^{(k)} \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad A_k^* := \left( \begin{array}{c|cc|c} 0 & & & 0 \\ 0 & A_{00}^{(k)} & A_{01}^{(k)} & 0 \\ \hline 0 & A_{10}^{(k)} & A_{11}^{(k)} & 0 \\ \hline 0 & & 0 & 0 \end{array} \right) \quad (k = 1, \dots, m), \quad \mathbf{g}_m^* := \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \beta \end{pmatrix}.$$

これらを用いると  $(\langle \hat{u}, \hat{v} \rangle)_{e_k} = \mathbf{v}_k^\top A_k \mathbf{u}_k$ ,  $(f, \hat{v})_{e_k} = \mathbf{v}_k^\top f_k$ ,  $\beta \hat{v}(1) = \mathbf{v}_m^\top \mathbf{g}_m$  であるから)

$$(20) \quad \langle \hat{u}, \hat{v} \rangle_{e_k} = \mathbf{v}^\top A_k^* \mathbf{u}, \quad (f, \hat{v})_{e_k} = \mathbf{v}^\top f_k^* \quad (k = 1, 2, \dots, m), \quad \beta \hat{v}(1) = \mathbf{v}^\top \mathbf{g}_m^*.$$

## 4.3.4 直接剛性法 (近似方程式の組み立て)

(20) を用いると、弱形式を書き直した (19) はさらに次のように書き直される。

$$(21) \quad \sum_{k=1}^m \mathbf{v}^\top \mathbf{A}_k^* \mathbf{u} = \sum_{k=1}^m \mathbf{v}^\top \mathbf{f}_k^* + \mathbf{v}^\top \mathbf{g}_m^* \quad (\mathbf{v} \in Y).$$

ここで  $Y$  は、 $\hat{v} = \sum_{i=0}^m v_i \phi_i$  が  $\hat{X}$  に属するような  $\mathbf{v} = (v_0, \dots, v_m)^\top$  の全体、すなわち

$$Y := \left\{ (v_0, v_1, \dots, v_m)^\top \in \mathbb{R}^{m+1} \mid v_0 = 0 \right\}.$$

(21) は

$$\mathbf{v}^\top \left( \sum_{k=1}^m \mathbf{A}_k^* \right) \mathbf{u} = \mathbf{v}^\top \left( \sum_{k=1}^m \mathbf{f}_k^* + \mathbf{g}_m^* \right) \quad (\mathbf{v} \in Y)$$

と書き直せる。ゆえに

$$(22) \quad \mathbf{A}^* := \sum_{k=1}^m \mathbf{A}_k^*, \quad \mathbf{f}^* := \sum_{k=1}^m \mathbf{f}_k^* + \mathbf{g}_m^*$$

とおけば

$$(23) \quad \mathbf{v}^\top (\mathbf{A}^* \mathbf{u} - \mathbf{f}^*) = 0 \quad (\mathbf{v} \in Y).$$

## 4.3.4 直接剛性法 (近似方程式の組み立て)

(再掲 23) 
$$\mathbf{v}^\top (\mathbf{A}^* \mathbf{u} - \mathbf{f}^*) = 0 \quad (\mathbf{v} \in Y).$$

これは次と同値である。

$$\mathbf{A}^* \mathbf{u} - \mathbf{f} \in Y^\perp = \left\{ (\lambda, 0, \dots, 0)^\top \in \mathbb{R}^{m+1} \mid \lambda \in \mathbb{R} \right\}.$$

つまりは

$$(\mathbf{A}^* \mathbf{u} - \mathbf{f}^*) \text{ の最初の成分以外} = 0.$$

すなわち

(24) 
$$\mathbf{A}^{**} \mathbf{u} = \mathbf{f}^{**}.$$

ここで

$$\begin{aligned} \mathbf{A}^{**} &:= \mathbf{A}^* \text{ の第 } 0 \text{ 行を除いた } m \times (m+1) \text{ 行列,} \\ \mathbf{f}^{**} &:= \mathbf{f}^* \text{ の第 } 0 \text{ 成分を除いた } m \text{ 次元縦ベクトル.} \end{aligned}$$

部分配列を表すための MATLAB 風の記法を使うと、 $\mathbf{A}^{**} = \mathbf{A}^*(\mathbf{1}:m, \mathbf{0}:m)$ ,  $\mathbf{f}^{**} = \mathbf{f}^*(\mathbf{1}:m)$  と書ける。この記法は便利なので以下でも使うことにする。

## 4.3.4 直接剛性法 (近似方程式の組み立て)

$A^{**}$  は正方行列ではない。しかし  $\mathbf{u}$  の成分のうち  $u_0$  は未知ではない:  $u_0 = \hat{u}(0) = \alpha$ .  
その部分を右辺に移項しよう。

$$\mathbf{u}^* := \mathbf{u} \text{ の第 } 0 \text{ 成分を除いた } m \text{ 次元縦ベクトル} = (u_1, \dots, u_m)^\top$$

$$\text{i.e. } \mathbf{u} = \begin{pmatrix} \alpha \\ \mathbf{u}^* \end{pmatrix},$$

$A := A^{**}$  の第 0 列を除いた  $m$  次正方行列

$$\text{i.e. } A^{**} = \left( \begin{array}{c|c} A_{10}^* & \\ \vdots & \\ A_{m0}^* & A \end{array} \right), \quad A := \begin{pmatrix} A_{11}^* & \cdots & A_{1m}^* \\ \vdots & & \vdots \\ A_{m1}^* & \cdots & A_{mm}^* \end{pmatrix}$$

とおくと

$$A^{**} \mathbf{u} = \left( \begin{array}{c|c} A_{10}^* & \\ \vdots & \\ A_{m0}^* & A \end{array} \right) \begin{pmatrix} \alpha \\ \mathbf{u}^* \end{pmatrix} = \begin{pmatrix} A_{10}^* \\ \vdots \\ A_{m0}^* \end{pmatrix} \alpha + A \mathbf{u}^* = \alpha \begin{pmatrix} A_{10}^* \\ \vdots \\ A_{m0}^* \end{pmatrix} + A \mathbf{u}^*.$$

$\mathbf{f} := \mathbf{f}^{**} - \alpha \begin{pmatrix} A_{10}^* \\ \vdots \\ A_{m0}^* \end{pmatrix}$  とおけば、(24)  $A^{**} \mathbf{u} = \mathbf{f}^{**}$  は、 $A \mathbf{u}^* = \mathbf{f}$  に書き換えられる。

## 4.3.4 直接剛性法 (近似方程式の組み立て)

以上のように、局所的な (要素の) 情報から方程式を組み立てる操作を**直接剛性法** (direct stiffness method) という。

(参考情報: 「直接剛性法」は、有限要素法の直接のルーツである**構造力学**に由来する用語である。構造力学の問題において、 $A$  は剛性行列という名前が付いている。)

次のことを覚えておくとよい。

- 係数行列は Dirichlet 境界条件を課す節点の節点番号の行と列を除いたもの
- Dirichlet 境界条件の情報は右辺のベクトルに組み込む
- 未知数は節点パラメーターであり、基底関数は節点に対応して作る

## 4.3.5 具体的にすることのまとめ (1枚で十分)

**第1段** 各要素  $e_k$  ( $k = 1, 2, \dots, m$ ) について、 $A_k, \mathbf{f}_k$  を求める:

$$A_k = \begin{pmatrix} \langle L_0, L_0 \rangle_{e_k} & \langle L_1, L_0 \rangle_{e_k} \\ \langle L_0, L_1 \rangle_{e_k} & \langle L_1, L_1 \rangle_{e_k} \end{pmatrix}, \quad \mathbf{f}_k = \begin{pmatrix} f(x_{k-1})(L_0, L_0)_{e_k} + f(x_k)(L_1, L_0)_{e_k} \\ f(x_{k-1})(L_0, L_1)_{e_k} + f(x_k)(L_1, L_1)_{e_k} \end{pmatrix},$$

$$\langle L_j, L_i \rangle_{e_k} = \begin{cases} \frac{1}{x_k - x_{k-1}} & (i = j) \\ -\frac{1}{x_k - x_{k-1}} & (i \neq j), \end{cases} \quad (L_j, L_i)_{e_k} = \begin{cases} \frac{x_k - x_{k-1}}{3} & (i = j) \\ \frac{x_k - x_{k-1}}{6} & (i \neq j) \end{cases}$$

を計算して、それを  $m+1$  次正方行列  $A_k^*$ ,  $m+1$  次元ベクトル  $\mathbf{f}_k^*$  に拡大して、

$$A^* := \sum_{k=1}^m A_k^*, \quad \mathbf{f}^* := \sum_{k=1}^m \mathbf{f}_k^* + \mathbf{g}_m^*,$$

それから  $A := A^*(1:m, 1:m)$ ,  $\mathbf{f}^{**} := \mathbf{f}^*(1:m)$ ,  $\mathbf{f} := \mathbf{f}^{**} - \alpha \begin{pmatrix} A_{10} \\ \vdots \\ A_{m0} \end{pmatrix}$ .

**第2段** 連立1次方程式  $A \begin{pmatrix} u_1 \\ \vdots \\ u_m \end{pmatrix} = \mathbf{f}$  を解いて  $u_1, \dots, u_m$  を求めて

$$\hat{u} = \alpha \phi_0 + \sum_{i=1}^m u_i \psi_i.$$

## 4.4 連立 1 次方程式の具体形

$\bar{\Omega} = \overline{(0,1)} = [0,1]$  を 4 等分して、各小区間を有限要素と考える。つまり  $m = 4$  で

$$x_i := ih \quad (i = 0, 1, 2, 3, 4),$$

ただし  $h = 1/4$ . そして

$$e_k := [x_{k-1}, x_k] \quad (k = 1, 2, 3, 4).$$

すると

$$A_k = \frac{1}{x_k - x_{k-1}} \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} = \frac{1}{h} \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix},$$

$$\mathbf{f}_k = \begin{pmatrix} f_0^{(k)} \\ f_1^{(k)} \end{pmatrix}, \quad f_j^{(k)} = \int_{e_k} f(x) L_j(x) dx \quad (L_j \text{ は } k \text{ によるので記号が変}),$$

$$\mathbf{g}_4 = \begin{pmatrix} 0 \\ \beta \end{pmatrix}.$$

特に (簡単のため)  $f(x) \equiv \bar{f}$  (定数関数) とすると、

$$f_j^{(k)} = \frac{\bar{f}h}{2} \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

## 4.4 連立 1 次方程式の具体形

$$\begin{aligned}A^* &= A_1^* + A_2^* + A_3^* + A_4^* \\&= \frac{1}{h} \begin{pmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} + \frac{1}{h} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \\&\quad + \frac{1}{h} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} + \frac{1}{h} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{pmatrix} \\&= \frac{1}{h} \begin{pmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{pmatrix},\end{aligned}$$

$$\begin{aligned}\mathbf{f}^* &= \mathbf{f}_1^* + \mathbf{f}_2^* + \mathbf{f}_3^* + \mathbf{f}_4^* + \mathbf{g}_4^* \\&= \frac{\bar{f}h}{2} \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \frac{\bar{f}h}{2} \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} + \frac{\bar{f}h}{2} \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} + \frac{\bar{f}h}{2} \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \beta \end{pmatrix} \\&= \bar{f}h \begin{pmatrix} 1/2 \\ 1 \\ 1 \\ 1 \\ 1/2 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \beta \end{pmatrix}.\end{aligned}$$

## 4.4 連立 1 次方程式の具体形

最後に  $u_0 = u(0) = \alpha$  を代入して  $u_0$  を消去すると

$$\frac{1}{h} \begin{pmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{pmatrix} = \bar{f}h \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1/2 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ \beta \end{pmatrix} + \begin{pmatrix} \alpha/h \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

この最後の方程式は、(仮想格子点を導入して、Neumann 境界条件を中心差分近似した) **差分法で得られる連立 1 次方程式と同じ**である。つまり

- 規則的な有限要素分割をしたとき、有限要素法は差分法に近い。
- 差分法で自明でない工夫 (仮想格子点の導入) をして得られた Neumann 境界条件の近似に相当することが、有限要素法ではごく自然に得られる。有限要素法は Neumann 境界条件の近似に強い。

## 4.4 連立 1 次方程式の具体形

(おまけ) 最後に、境界条件を ( $u(0) = \alpha$ ,  $u'(1) = \beta$  から)

$$u(0) = \alpha, \quad u(1) = \beta$$

に替えた、Dirichlet 境界値問題を調べておこう。この場合は、次の連立 1 次方程式が得られる。

$$\frac{1}{h} \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \bar{f}h \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + \begin{pmatrix} \alpha/h \\ 0 \\ \beta/h \end{pmatrix}.$$

## 4.5 サンプル・プログラム fem1d.c 4.5.1 問題

以下に紹介する C プログラム fem1d.c は

`https://m-katsurada.sakura.ne.jp/program/fem/fem1d.c`

に置いてある。現象数理学科 Mac ならば、ターミナルから

```
curl -O https://m-katsurada.sakura.ne.jp/program/fem/fem1d.c
```

で入手できる。コンパイル、実行の仕方はプログラムの先頭部分に注釈として書いてある。

このプログラムが対象としている問題は、 $f \equiv 1$  で、境界条件は同次、すなわち  $\alpha = \beta = 0$  の場合である。具体的に書き下すと

$$(25) \quad -u'' = 1, \quad u(0) = u'(1) = 0.$$

この問題の厳密解は  $u(x) = x(2-x)/2$  である。

## 4.5.2 プログラムの解説

- `main()` を読むと分かるように、最初に
  - `nnode` 総節点数 (the number of nodes)
  - `nelmt` 総要素数 (the number of elements)
  - `nbc` ディリクレ境界にある接点の個数 (1 または 2)
  - `x[]` 節点の座標
  - `ibc` ディリクレ境界にある接点の節点番号を決めている。
- 連立 1 次方程式を構成するのは、関数 `assem()` で行っている (assemblage)。作業内容は 3 つに分かれる。
  - ① `am`, `fm` を 0 クリアする。
  - ② すべての有限要素について、要素係数行列 `ae`, 要素自由ベクトル `fe` を関数 `ecm()` で計算して (element coefficient matrix)、それぞれ全体係数行列 `am`、全体自由項ベクトル `fm` に算入する。
  - ③ ディリクレ境界上にある節点に対応する部分を修正する。

## 4.5.2 プログラムの解説

- 関数  $\text{ecm}()$  で必要となる事項の復習。  $e_k = [x_{k-1}, x_k]$  とすると、

$$A_k = \frac{1}{x_k - x_{k-1}} \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}, \quad \mathbf{f}_k = \begin{pmatrix} (f, L_0)_{e_k} \\ (f, L_1)_{e_k} \end{pmatrix}$$

であったが、 $f$  を

$$f(x) \doteq f(x_{k-1})L_0(x) + f(x_k)L_1(x) \quad (x \in e_k)$$

と 1 次近似することにすれば、

$$\mathbf{f}_k \doteq \frac{x_k - x_{k-1}}{6} \begin{pmatrix} 2f(x_{k-1}) + f(x_k) \\ f(x_{k-1}) + 2f(x_k) \end{pmatrix}.$$

## 4.5.3 実験

fem1d.c のコンパイル&実行& gnuplot によるグラフ描画

```
$ cc -o fem1d fem1d.c
$ ./fem1d
nodal values of u (節点での u の値)
  i      u      i      u      i      u
  0  0.000e+00  1  9.500e-02  2  1.800e-01
  3  2.550e-01  4  3.200e-01  5  3.750e-01
  6  4.200e-01  7  4.550e-01  8  4.800e-01
  9  4.950e-01 10  5.000e-01

$ cat fem1d.out
0.000000 0.000000
0.100000 0.095000
0.200000 0.180000
0.300000 0.255000
0.400000 0.320000
0.500000 0.375000
0.600000 0.420000
0.700000 0.455000
0.800000 0.480000
0.900000 0.495000
1.000000 0.500000

$ gnuplot
gnuplot> plot "fem1d.out" with lp, x*(2-x)/2
```

## 4.5.3 実験

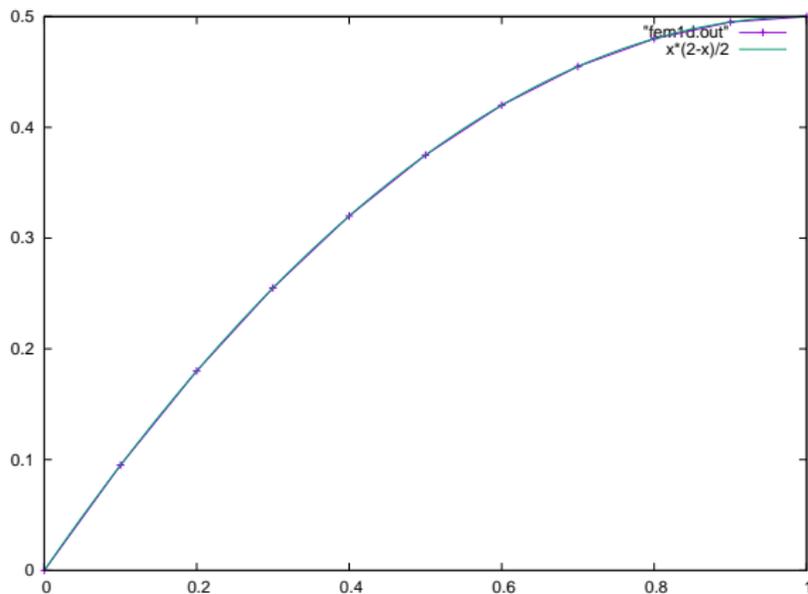


図 1: fem1d.c の計算結果 ( $m=10$ ) と厳密解  $\frac{x(2-x)}{2}$  のグラフを重ね書き

## 4.5.4 参考: 昔の練習問題

FreeFem++ がまだなかった頃、有限要素法のプログラムを、C 言語や Fortran のようなプログラミング言語で書いていました。

そのときは (アルゴリズムの理解する助けになると考えて) 以下のような練習問題を出していました。参考まで。

- ① 両側ディリクレ条件  $u(0) = u(1) = 0$  の問題を解く。
- ② 非同次ディリクレ条件  $u(0) = \alpha$  の問題を解く。
- ③ 非同次 Neumann 条件  $u'(1) = \beta$  の問題を解く。
- ④  $-(pu')' = f$  という一般の楕円型方程式の問題を解く。  
( $p$  は  $\min_x p(x) > 0$  を満たす既知の関数)

## 5 2次元の有限要素法

第2回の授業で扱った Poisson 方程式の境界値問題を題材に、2次元領域における有限要素法を説明する (菊地 [?] 第5章)。

(思い出す)  $\Omega$  は  $\mathbb{R}^2$  の有界領域で、その境界  $\Gamma$  は区分的に十分滑らかであるとする。また  $\Gamma_1, \Gamma_2$  は条件

$$\Gamma = \bar{\Gamma}_1 \cup \bar{\Gamma}_2, \quad \Gamma_1 \cap \Gamma_2 = \emptyset, \quad \Gamma_1 \neq \emptyset$$

を満たすとする。  $f: \Omega \rightarrow \mathbb{R}$ ,  $g_1: \Gamma_1 \rightarrow \mathbb{R}$ ,  $g_2: \Gamma_2 \rightarrow \mathbb{R}$  が与えられたとき、次の Poisson 方程式の境界値問題を考える。

問題 (P)

次式を満たす  $u$  を求めよ:

$$(26) \quad -\Delta u = f \quad \text{in } \Omega,$$

$$(27) \quad u = g_1 \quad \text{on } \Gamma_1,$$

$$(28) \quad \frac{\partial u}{\partial \mathbf{n}} = g_2 \quad \text{on } \Gamma_2,$$

ここで  $\mathbf{n}$  は  $\Gamma$  の外向き単位法線ベクトルを表す。

大筋は1次元の場合と同様だが、(i) 各要素内の計算に面積座標を使うところと、(ii) 直接剛性法が2次元でも実現可能であることを理解するところが要点となる。

## 5.1 三角形要素への分割と区分的1次多項式

領域  $\Omega \subset \mathbb{R}^2$  に対し、 $\bar{\Omega}$  を三角形  $e_k$  ( $1 \leq k \leq N_e$ ) に分割する:

$$\bar{\Omega} \doteq \hat{\Omega} := \bigcup_{k=1}^{N_e} e_k.$$

ただし、重なりや、すき間、頂点が他の三角形の辺上にあることは避けることにする。

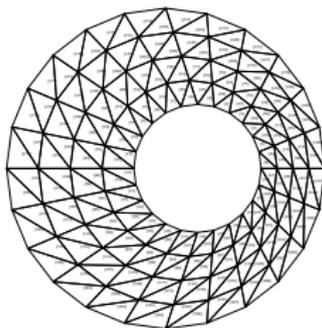


図 2: 二つの円で囲まれた閉領域  $\bar{\Omega}$  を三角形の合併で近似する

$\Omega$  が多角形でない限り、境界は「曲がっている」。  $\bar{\Omega} = \bigcup_{k=1}^{N_e} e_k$  は期待できない。

各三角形を **三角形 (有限) 要素** とよぶ。

## 5.1 三角形要素への分割と区分的1次多項式

$N_e$  は要素の総数 (the number of elements) で、プログラムでは NELMT のような名前の変数で記憶されることが多い。

有限要素の頂点を節点 (node) と呼び、 $\{P_i\}_{i=1}^m$  のように番号をつけておく。

$m$  は節点の総数 (the number of nodes) で、プログラムでは NNODE のような名前の変数で記憶されることが多い。

**注意** 1次元の場合、節点の個数 = 要素の個数 + 1 という簡単な関係が成立していた。(区間を  $m$  等分したとき、 $m$  を用いて、要素を  $e_k$  ( $1 \leq k \leq m$ ), 節点を  $x_k$  ( $0 \leq k \leq m$ ) と番号づけることが出来た。) 2次元の場合は、そのような関係はない。

## 5.1 三角形要素への分割と区分的1次多項式

$\hat{\Omega}$  上連続で、各有限要素  $e_k$  上で  $x$  と  $y$  の1次多項式関数に等しいものを、**区分的1次多項式**と呼び、その全体を  $\tilde{X}$  で表わす ( $\tilde{X}$  はここだけの記号)。

$\{e_k\}_{k=1}^{N_e}$  と  $\tilde{X}$  の対を**区分的1次有限要素** (piecewise linear finite element) と呼ぶ。

2変数の1次関数  $z = a + bx + cy$  のグラフは平面であるから、 $\tilde{X}$  のグラフは、空間内の三角形を連続につなげた“折れ面”である。

試行関数の空間  $\hat{X}_{g_1}$ , 試験関数の空間  $\hat{X}$  は次のように選ぶ。

$$(29) \quad \hat{X}_{g_1} = \left\{ \hat{w} \in \tilde{X} \mid \hat{w} = \hat{g}_1 \quad \text{on } \Gamma_1 \right\}, \quad \hat{X} = \left\{ \hat{v} \in \tilde{X} \mid \hat{v} = 0 \quad \text{on } \Gamma_1 \right\}.$$

( $\hat{g}_1$  は  $g_1$  に近い計算しやすい関数)

$\tilde{X}$  の基底関数  $\{\phi_i\}_{i=1}^m$  は

$$(30a) \quad \phi_i \in \tilde{X} \quad (i = 1, 2, \dots, m),$$

$$(30b) \quad \phi_i(P_j) = \delta_{ij} \quad (i, j = 1, 2, \dots, m)$$

満たすものを採用する (この条件で一意に確定し、線形独立であることに注意)。

## 5.2 三角形 $e$ 上の 1 次関数 $L_i$ と $(L_j, L_i)_e, \langle L_j, L_i \rangle_e$

### 5.2.1 三角形の面積

平面上の三角形要素  $e$  を考える (本来は、以下の  $N_i, L_i$  も含めて、要素になるので、 $e_k, N_i^k, L_i^k$  のように書いた方が良くもしいないが、うっとうしいので  $k$  は略する)。

$e$  に属する節点を、反時計まわりに  $N_i = (x_i, y_i)$  ( $i = 0, 1, 2$ ) とする。

後でしばしば必要になるので、 $e$  の面積  $|e|$  を計算しておこう。

$$\begin{aligned} |e| &= \frac{1}{2} \det \begin{pmatrix} x_1 - x_0 & x_2 - x_0 \\ y_1 - y_0 & y_2 - y_0 \end{pmatrix} \\ &= \frac{1}{2} [(x_1 - x_0)(y_2 - y_0) - (y_1 - y_0)(x_2 - x_0)]. \end{aligned}$$

1 次多項式全体を  $P^1$  と表す:

$$P^1 := \{ \hat{u} \mid (\exists \alpha_0, \alpha_1, \alpha_2 \in \mathbb{R}) \hat{u}(x, y) = \alpha_0 + \alpha_1 x + \alpha_2 y \}.$$

任意の  $\hat{u} \in P^1$  は、3 節点  $N_i$  における値  $u^i := \hat{u}(N_i)$  ( $i = 0, 1, 2$ ) を指定すれば定まる ( $u_i = \hat{u}(P_i)$  と混同しないように上に添字をつけた)。これは、直観的に明らかであるが (平面は、その上にある 3 点 (ただし同一直線上にはないとする) を指定すれば定まる)、すぐ後で証明する。

## 5.2.2 三角形要素の面積座標 $L_i$

節点  $N_i$  で 1, 他の節点で 0 となる 1 次関数を  $L_i$  とする ( $i = 0, 1, 2$ )。つまり

$$(31a) \quad L_i \in P^1,$$

$$(31b) \quad L_i(N_j) = L_i(x_j, y_j) = \delta_{ij} \quad (j = 0, 1, 2).$$

$(L_0, L_1, L_2)$  は  $P^1$  の基底になる。実際、1 次独立性はすぐ分かり、 $\dim P^1 = 3$  であるから、任意の  $\hat{u} \in P^1$  は、

$$(32) \quad \hat{u}(x, y) = \sum_{i=0}^2 u^i L_i(x, y) \quad ((x, y) \in e)$$

と表される。

任意の  $P \in e$  に対して、3 実数  $(L_0(P), L_1(P), L_2(P))$  を  $P$  の面積座標 (area coordinate) あるいは重心座標 (barycentric coordinate) と呼ぶ。(色々裏があるけれど今回は駆け足で進む。)

任意の  $P \in e$  に対して次式が成り立つ。

$$L_0(P) + L_1(P) + L_2(P) = 1.$$

( $P^1$  の基底としては 3 つ必要だが、座標としては 2 つで十分ということになる。)

以下、 $L_i$  のグラフの鳥瞰図と等高線を描こう。

# Mathematica のコード例と実行結果

```
xs = {0, 3, 1}; ys = {0, 1, 2};  
{a, b, c} = Inverse[Transpose[{{1, 1, 1}, xs, ys}]];  
L[i_, x_, y_] := a[[i]] + b[[i]]*x + c[[i]]*y  
xmin = Min[xs]; xmax = Max[xs]; ymin = Min[ys]; ymax = Max[ys];  
gbase = RegionPlot[L[1, x, y] > 0 && L[2, x, y] > 0 && L[3, x, y] > 0,  
  {x, xmin, xmax}, {y, ymin, ymax}]  
gb=Table[Plot3D[L[i, x, y], {x, xmin, xmax}, {y, ymin, ymax},  
  RegionFunction -> Function[{x, y, z},  
  L[1, x, y] > 0 && L[2, x, y] > 0 && L[3, x, y] > 0]], {i, 3}]  
gc=Table[ContourPlot[L[i, x, y], {x, xmin, xmax}, {y, ymin, ymax},  
  RegionFunction -> Function[{x, y, z},  
  L[1, x, y] > 0 && L[2, x, y] > 0 && L[3, x, y] > 0]], {i, 3}]
```

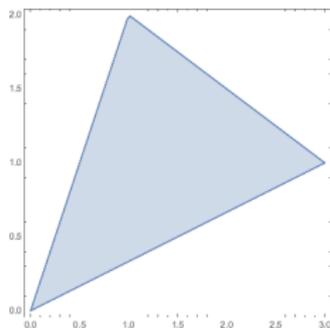


図 3: 三角形要素

—— ターミナルで入手して開く ——

```
curl -O https://m-katsurada.sakura.ne.jp/ana/20240514.nb  
open 20240514.nb
```

# Mathematica のコード例と実行結果

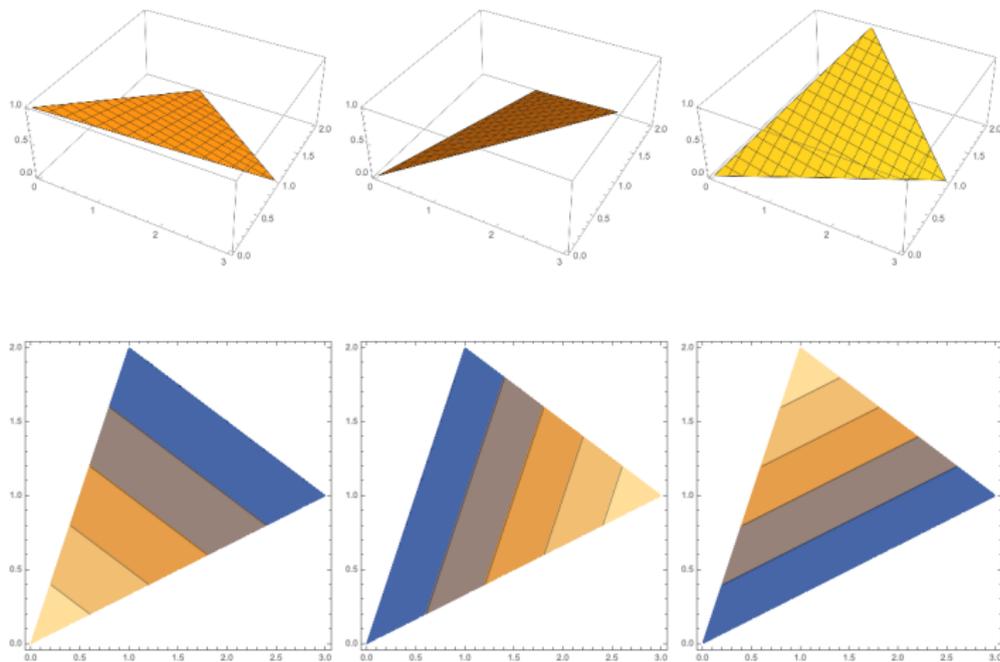


図 4: 左から  $L_0$ ,  $L_1$ ,  $L_2$  の鳥瞰図と等高線

## 5.2.3 面積座標の積の積分と $(L_j, L_i)_e$

面積座標の積の積分については、便利な公式がある。

0 以上の任意の整数  $i, j, k$  に対して

$$(33) \quad \iint_e L_0(x, y)^i L_1(x, y)^j L_2(x, y)^k dx dy = 2|e| \frac{i!j!k!}{(i+j+k+2)!}.$$

## 5.2.3 面積座標の積の積分と $(L_j, L_i)_e$

**証明**  $P_0 = (0, 0)$ ,  $P_1 = (1, 0)$ ,  $P_2 = (0, 1)$  で囲まれる三角形を  $\Delta$  とし、1 次関数  $\varphi: \mathbb{R}^2 \rightarrow \mathbb{R}^2$  を

$$\varphi(P_0) = N_0, \quad \varphi(P_1) = N_1, \quad \varphi(P_2) = N_2$$

で定める。このとき

$$\varphi(\Delta) = e,$$

$$\varphi(u, v) = \begin{pmatrix} x_1 - x_0 & x_2 - x_0 \\ y_1 - y_0 & y_2 - y_0 \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} + \begin{pmatrix} x_0 \\ y_0 \end{pmatrix},$$

$$\det \varphi'(u, v) = \det \begin{pmatrix} x_1 - x_0 & x_2 - x_0 \\ y_1 - y_0 & y_2 - y_0 \end{pmatrix} = 2|e|.$$

ゆえに変数変換  $(x, y) = \varphi(u, v)$  を行なうと

$$\begin{aligned} \iint_e L_0^i L_1^j L_2^k dx dy &= \iint_{\Delta} L_0(\varphi(u, v))^i L_1(\varphi(u, v))^j L_2(\varphi(u, v))^k |\det \varphi'(u, v)| du dv. \\ &= 2|e| \iint_{\Delta} L_0(\varphi(u, v))^i L_1(\varphi(u, v))^j L_2(\varphi(u, v))^k du dv. \end{aligned}$$

## 5.2.3 面積座標の積の積分と $(L_j, L_i)_e$

$L_i(\varphi(u, v))$  は (1 次関数と 1 次関数の合成であるから)、 $u, v$  についての 1 次関数で、

$$L_i(N_j) = L_i(\varphi(P_j)) = \delta_{ij}$$

を満たすことから

$$L_0(\varphi(u, v)) = 1 - u - v, \quad L_1(\varphi(u, v)) = u, \quad L_2(\varphi(u, v)) = v$$

(各等式の両辺は 1 次関数で、 $N_j$  での値が一致するから、全体で一致する)。

ゆえに

$$\begin{aligned} \iint_e L_0^i L_1^j L_2^k dx dy &= 2|e| \iint_{\Delta} (1-u-v)^i u^j v^k du dv \\ &= 2|e| \int_0^1 u^j \left( \int_0^{1-u} (1-u-v)^i v^k dv \right) du. \end{aligned}$$

右辺の内側の積分で、 $v = (1-u)t$  ( $0 \leq t \leq 1$ ) と変数変換すると

$$dv = (1-u)dt, \quad (1-u-v)^i = ((1-u) - (1-u)t)^i = (1-u)^i (1-t)^i$$

であるから

$$\begin{aligned} \int_0^{1-u} (1-u-v)^i v^k dv &= \int_0^1 (1-u)^i (1-t)^i (1-u)^k t^k \cdot (1-u) dt \\ &= (1-u)^{i+k+1} \int_0^1 (1-t)^i t^k dt. \end{aligned}$$

## 5.2.3 面積座標の積の積分と $(L_j, L_i)_e$

ゆえに

$$\begin{aligned}\iint_e L_0^i L_1^j L_2^k dx dy &= 2|e| \int_0^1 (1-u)^{i+k+1} u^j du \int_0^1 (1-t)^i t^k dt \\ &= 2|e| B(i+k+2, j+1) B(i+1, k+1).\end{aligned}$$

ただし  $B$  は次式で定義されるベータ関数である:

$$B(p, q) = \int_0^1 (1-t)^{p-1} t^{q-1} dt \quad (p, q > 0).$$

このベータ関数と、ガンマ関数

$$\Gamma(x) := \int_0^\infty t^{x-1} e^{-t} dt \quad (x > 0)$$

について、次の有名な公式が成り立つ (証明は例えば桂田 [?] § E.2 の 命題 E.2.4)。

$$B(p, q) = \frac{\Gamma(p)\Gamma(q)}{\Gamma(p+q)}, \quad \Gamma(n+1) = n! \quad (n \in \mathbb{N}).$$

ゆえに

$$\begin{aligned}\iint_e L_0^i L_1^j L_2^k dx dy &= 2|e| \frac{\Gamma(i+k+2)\Gamma(j+1)}{\Gamma(i+j+k+3)} \cdot \frac{\Gamma(i+1)\Gamma(k+1)}{\Gamma(i+k+2)} \\ &= 2|e| \frac{\Gamma(i+1)\Gamma(j+1)\Gamma(k+1)}{\Gamma(i+j+k+3)} = 2|e| \frac{i!j!k!}{(i+j+k+2)!}. \quad \square\end{aligned}$$

## 5.2.3 面積座標の積の積分と $(L_j, L_i)_e$

$i = j$  の場合、それ以外の添字 ( $\in \{0, 1, 2\}$ ) を  $k, \ell$  とすると

$$(L_j, L_i)_e = \iint_e L_i^2 L_k^0 L_\ell^0 dx dy = 2 |e| \frac{2!0!0!}{(2+0+0+2)!} = \frac{1}{6} |e|.$$

$i \neq j$  の場合、それ以外の添字 ( $\in \{0, 1, 2\}$ ) を  $k$  とすると

$$(L_j, L_i)_e = \iint_e L_i^1 L_j^1 L_k^0 dx dy = 2 |e| \frac{1!1!0!}{(1+1+0+2)!} = \frac{1}{12} |e|.$$

## 5.2.4 $L_i(x, y) = a_i + b_i x + c_i y$ の係数決定と $\langle L_j, L_i \rangle_e$

$$L_i(x, y) = a_i + b_i x + c_i y$$

とおくと

$$\delta_{ij} = L_j(x_i, y_i) = a_j + b_j x_i + c_j y_i = (1 \ x_i \ y_i) \begin{pmatrix} a_j \\ b_j \\ c_j \end{pmatrix}$$

であるから

$$(\heartsuit) \quad \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & x_0 & y_0 \\ 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \end{pmatrix} \begin{pmatrix} a_0 & a_1 & a_2 \\ b_0 & b_1 & b_2 \\ c_0 & c_1 & c_2 \end{pmatrix}.$$

$$A := \begin{pmatrix} 1 & x_0 & y_0 \\ 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \end{pmatrix}$$

とおくと

$$\det A = \det \begin{pmatrix} 1 & x_0 & y_0 \\ 0 & x_1 - x_0 & y_1 - y_0 \\ 0 & x_2 - x_0 & y_2 - y_0 \end{pmatrix} = \begin{vmatrix} x_1 - x_0 & y_1 - y_0 \\ x_2 - x_0 & y_2 - y_0 \end{vmatrix} = 2|e| > 0.$$

ゆえに  $A$  は逆行列を持つ。

## 5.2.4 $L_i(x, y) = a_i + b_i x + c_i y$ の係数決定と $\langle L_j, L_i \rangle_e$

(♡) から

$$\begin{pmatrix} a_0 & a_1 & a_2 \\ b_0 & b_1 & b_2 \\ c_0 & c_1 & c_2 \end{pmatrix} = A^{-1}.$$

Cramer の公式によって

$$\begin{aligned} A^{-1} &= \frac{1}{\det A} \begin{pmatrix} (-1)^{1+1} \begin{vmatrix} x_1 & y_1 \\ x_2 & y_2 \end{vmatrix} & (-1)^{1+2} \begin{vmatrix} 1 & y_1 \\ 1 & y_2 \end{vmatrix} & (-1)^{1+3} \begin{vmatrix} 1 & x_1 \\ 1 & x_2 \end{vmatrix} \\ (-1)^{2+1} \begin{vmatrix} x_0 & y_0 \\ x_2 & y_2 \end{vmatrix} & (-1)^{2+2} \begin{vmatrix} 1 & y_0 \\ 1 & y_2 \end{vmatrix} & (-1)^{3+3} \begin{vmatrix} 1 & x_0 \\ 1 & x_2 \end{vmatrix} \\ (-1)^{3+1} \begin{vmatrix} x_0 & y_0 \\ x_1 & y_1 \end{vmatrix} & (-1)^{3+2} \begin{vmatrix} 1 & y_0 \\ 1 & y_1 \end{vmatrix} & (-1)^{3+3} \begin{vmatrix} 1 & x_0 \\ 1 & x_1 \end{vmatrix} \end{pmatrix}^T \\ &= \frac{1}{\det A} \begin{pmatrix} x_1 y_2 - y_1 x_2 & -(y_2 - y_1) & x_2 - x_1 \\ -(x_0 y_2 - y_0 x_2) & y_2 - y_0 & -(x_2 - x_0) \\ x_0 y_1 - y_0 x_1 & -(y_1 - y_0) & x_1 - x_0 \end{pmatrix}^T \\ &= \frac{1}{\det A} \begin{pmatrix} x_1 y_2 - y_1 x_2 & x_2 y_0 - y_2 x_0 & x_0 y_1 - y_0 x_1 \\ y_1 - y_2 & y_2 - y_0 & y_0 - y_1 \\ x_2 - x_1 & x_0 - x_2 & x_1 - x_0 \end{pmatrix}. \end{aligned}$$

## 5.2.4 $L_i(x, y) = a_i + b_i x + c_i y$ の係数決定と $\langle L_j, L_i \rangle_e$

$A^{-1}$  の下 2 行  $b_k, c_k$  ( $k = 0, 1, 2$ ) が得られれば、 $\langle L_j, L_i \rangle_e$  が計算できる:

$$(34) \quad \begin{aligned} \langle L_j, L_i \rangle_e &= \iint_e \nabla L_j \cdot \nabla L_i \, dx \, dy = \iint_e \begin{pmatrix} b_j \\ c_j \end{pmatrix} \cdot \begin{pmatrix} b_i \\ c_i \end{pmatrix} \, dx \, dy \\ &= (b_j b_i + c_j c_i) |e|. \end{aligned}$$

## 5.3 要素係数行列の計算

「積分は積分範囲を分割して計算し、後から和を取ればよい」ので、弱形式

$$\langle \hat{u}, \hat{v} \rangle = (f, \hat{v}) + [g_2, \hat{v}] \quad (\hat{v} \in \hat{X})$$

は

$$(35) \quad \sum_{k=1}^{N_e} \langle \hat{u}, \hat{v} \rangle_{e_k} = \sum_{k=1}^{N_e} (f, \hat{v})_{e_k} + [g_2, \hat{v}] \quad (\hat{v} \in \hat{X})$$

となる。ただし

$$\langle \hat{u}, \hat{v} \rangle_{e_k} := \int_{e_k} \nabla \hat{u}(x) \cdot \nabla \hat{v}(x) dx, \quad (f, \hat{v})_{e_k} := \int_{e_k} f(x) \hat{v}(x) dx, \quad [g_2, \hat{v}] := \int_{\Gamma_2} g_2 \hat{v} d\sigma.$$

そこで  $u^j := \hat{u}(N_j)$ ,  $v^j := \hat{v}(N_j)$  ( $j = 0, 1, 2$ ) を用いて、

$$\hat{u} = \sum_{j=0}^2 u^j L_j, \quad \hat{v} = \sum_{i=0}^2 v^i L_i \quad (e_k \text{ 上})$$

と表すと

$$(36) \quad \langle \hat{u}, \hat{v} \rangle_{e_k} = \sum_{j=0}^2 \sum_{i=0}^2 v^i A_{ij}^{(k)} u^j, \quad (f, \hat{v})_{e_k} = \sum_{i=0}^2 v^i f_i^{(k)},$$

ただし

$$(37) \quad A_{ij}^{(k)} := \langle L_j, L_i \rangle_{e_k}, \quad f_i^{(k)} := (f, L_i)_{e_k}.$$

## 5.3 要素係数行列の計算

そこで

$$(38a) \quad \mathbf{u}_k := \begin{pmatrix} u^0 \\ u^1 \\ u^2 \end{pmatrix}, \quad \mathbf{v}_k := \begin{pmatrix} v^0 \\ v^1 \\ v^2 \end{pmatrix}, \quad \mathbf{f}_k := \begin{pmatrix} f_0^{(k)} \\ f_1^{(k)} \\ f_2^{(k)} \end{pmatrix},$$

$$(38b) \quad A_k := \begin{pmatrix} A_{00}^{(k)} & A_{01}^{(k)} & A_{02}^{(k)} \\ A_{10}^{(k)} & A_{11}^{(k)} & A_{12}^{(k)} \\ A_{20}^{(k)} & A_{21}^{(k)} & A_{22}^{(k)} \end{pmatrix}$$

とおくと、 $A_k$  は対称行列で、

$$(39) \quad \langle \hat{u}, \hat{v} \rangle_{e_k} = \mathbf{v}_k^T A_k \mathbf{u}_k, \quad (f, \hat{v})_{e_k} = \mathbf{v}_k^T \mathbf{f}_k.$$

(線積分  $[g_2, \hat{v}]$  については、今回の授業では説明を省略する。とりあえず  $g_2 = 0$  と考えて授業を聴いて下さい。桂田 [?] には書いてある。)

$A_{ij}^{(k)} = \langle L_j, L_i \rangle_{e_k}$  の計算は (34) で済んでいる。

## 5.3 要素係数行列の計算 具体的な成分の計算

$f_i^{(k)} = (f, L_i)_{e_k}$  については、例えば  $L_i$  を 1 次関数補間して

$$(f, L_i)_{e_k} \doteq \left( \sum_{j=0}^2 f(N_j) L_j, L_i \right)_{e_k} = \sum_{j=0}^2 f(N_j) (L_j, L_i)_{e_k}.$$

のように近似すれば

$$(L_j, L_i)_{e_k} = \begin{cases} |e_k|/6 & (i = j \text{ のとき}), \\ |e_k|/12 & (i \neq j \text{ のとき}). \end{cases}$$

であるから

$$f_0^{(k)} \doteq \frac{|e_k|}{12} (2f(N_0) + f(N_1) + f(N_2)),$$

$$f_1^{(k)} \doteq \frac{|e_k|}{12} (f(N_0) + 2f(N_1) + f(N_2)),$$

$$f_2^{(k)} \doteq \frac{|e_k|}{12} (f(N_0) + f(N_1) + 2f(N_2)).$$

## 5.4 近似方程式の組み立て — 直接剛性法

$u_i := \hat{u}(P_i)$ ,  $v_i := \hat{v}(P_i)$  ( $i = 1, 2, \dots, m$ ) として、

$$(40) \quad \mathbf{u} := \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{pmatrix}, \quad \mathbf{v} := \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \end{pmatrix}$$

とおく。

有限要素  $e_k$  ( $k = 1, 2, \dots, N_e$ ) の節点  $N_0, N_1, N_2$  に対して、

$$N_0 = P_{i_{k,0}}, \quad N_1 = P_{i_{k,1}}, \quad N_2 = P_{i_{k,2}}$$

となる整数  $i_{k,0}, i_{k,1}, i_{k,2}$  を取る (これらを**全体節点番号**と呼ぶ)。

$\mathbf{f}_k^* \in \mathbb{R}^m$  を  $i_{k,0}$  成分 =  $f_0^{(k)}$ ,  $i_{k,1}$  成分 =  $f_1^{(k)}$ ,  $i_{k,2}$  成分 =  $f_2^{(k)}$  で、それ以外の成分はすべて 0 であるようなベクトルとする。例えば  $i_0^{(k)} < i_1^{(k)} < i_2^{(k)}$  ならば

$$\mathbf{f}_k^* = \begin{pmatrix} 1 & & & & & & & & & & \\ \downarrow & & & & & & & & & & \\ 0 & \dots & 0 & f_0^{(k)} & 0 & \dots & 0 & f_1^{(k)} & 0 & \dots & 0 & f_2^{(k)} & 0 & \dots & 0 \end{pmatrix}^\top.$$

このように  $\mathbf{f}_k^*$  を定義すると、次が成り立つ。

$$(41) \quad (\mathbf{f}, \hat{v})_{e_k} = \mathbf{v}_k^\top \mathbf{f}_k = \mathbf{v}^\top \mathbf{f}_k^* \quad (k = 1, 2, \dots, N_e).$$

## 5.4 近似方程式の組み立て — 直接剛性法

同様の考え方で、行列  $A_k^* = (a_{ij}^{(k)})$  を

$$\begin{aligned} a_{i_{k,0}i_{k,0}}^{(k)} &= A_{00}^{(k)}, & a_{i_{k,0}i_{k,1}}^{(k)} &= A_{01}^{(k)}, & a_{i_{k,0}i_{k,2}}^{(k)} &= A_{02}^{(k)}, \\ a_{i_{k,1}i_{k,0}}^{(k)} &= A_{10}^{(k)}, & a_{i_{k,1}i_{k,1}}^{(k)} &= A_{11}^{(k)}, & a_{i_{k,1}i_{k,2}}^{(k)} &= A_{12}^{(k)}, \\ a_{i_{k,2}i_{k,0}}^{(k)} &= A_{20}^{(k)}, & a_{i_{k,2}i_{k,1}}^{(k)} &= A_{21}^{(k)}, & a_{i_{k,2}i_{k,2}}^{(k)} &= A_{22}^{(k)}, \\ && \text{それ以外} &= 0 \end{aligned}$$

で定める。例えば  $i_{k,0} < i_{k,1} < i_{k,2}$  ならば

$$A_k^* = \begin{pmatrix} i_{k,0} & i_{k,1} & i_{k,2} & & \\ \downarrow & \downarrow & \downarrow & & \\ A_{00}^{(k)} & A_{01}^{(k)} & A_{02}^{(k)} & \leftarrow i_{k,0} & \\ A_{10}^{(k)} & A_{11}^{(k)} & A_{12}^{(k)} & \leftarrow i_{k,1} & \\ A_{20}^{(k)} & A_{21}^{(k)} & A_{22}^{(k)} & \leftarrow i_{k,2} & \end{pmatrix} \quad (\text{書いてない成分は } 0).$$

これを用いると

$$(42) \quad \langle \hat{u}, \hat{v} \rangle_{e_k} = \mathbf{v}_k^\top A_k \mathbf{u}_k = \mathbf{v}^\top A_k^* \mathbf{u} \quad (k = 1, 2, \dots, N_e).$$

## 5.4 近似方程式の組み立て — 直接剛性法

ゆえに弱形式 (35) は ( $g_2 = 0$  と考えている)

$$\sum_{k=1}^{N_e} \mathbf{v}^\top A_k^* \mathbf{u} = \sum_{k=1}^{N_e} \mathbf{v}^\top \mathbf{f}_k^*$$

すなわち

$$\mathbf{v}^\top \left( \sum_{k=1}^{N_e} A_k^* \mathbf{u} - \sum_{k=1}^{N_e} \mathbf{f}_k^* \right) = 0$$

と同値になる。

ゆえに

$$\mathbf{A}^* := \sum_{k=1}^{N_e} A_k^*, \quad \mathbf{f}^* := \sum_{k=1}^{N_e} \mathbf{f}_k^*$$

とおけば

$$(43) \quad \mathbf{v}^\top (\mathbf{A}^* \mathbf{u} - \mathbf{f}^*) = 0.$$

## 5.4 近似方程式の組み立て — 直接剛性法

ここで  $\mathbf{v}$  は

$$Y := \left\{ \begin{pmatrix} v_1 \\ \vdots \\ v_m \end{pmatrix} \in \mathbb{R}^m; v_i = 0 \quad (P_i \in \Gamma_1 \text{ なる } i) \right\}$$

の任意の元であるから、(43) は次と同値である。

$$\mathbf{A}^* \mathbf{u} - \mathbf{f} \in Y^\perp = \left\{ \begin{pmatrix} w_1 \\ \vdots \\ w_m \end{pmatrix} \in \mathbb{R}^m; w_i = 0 \quad (P_i \notin \Gamma_1 \text{ なる } i) \right\}$$

すなわち

$$(44) \quad \mathbf{A}^{**} \mathbf{u} = \mathbf{f}^{**}.$$

ここで

$\mathbf{A}^{**} := \mathbf{A}^*$  の第  $i$  行 ( $P_i \in \Gamma_1$  なる  $i$ ) を除いた行列,

$\mathbf{f}^{**} := \mathbf{f}^*$  の第  $i$  成分 ( $P_i \in \Gamma_1$  なる  $i$ ) を除いた縦ベクトル.

## 5.4 近似方程式の組み立て — 直接剛性法

$$I := \{i \in \mathbb{N} \mid 1 \leq i \leq m\}, \quad I_1 := \{i \in I \mid P_i \in \Gamma_1\}$$

とおく ( $I_1$  は  $\Gamma_1$  上にある節点の節点番号全体の集合)。

条件

$$u_i = g_1(P_i) \quad (i \in I_1)$$

があるから、これを代入して  $u_i$  ( $i \in I_1$ ) を消去できる。以下それを実行する。

$A^{**}$  を列ベクトルで

$$A^{**} = (\mathbf{a}_1 \mathbf{a}_2 \cdots \mathbf{a}_m)$$

のように表示すると、(44) は

$$(45) \quad \sum_{i=1}^m \mathbf{a}_i u_i = \mathbf{f}^{**}.$$

左辺の  $\sum_{i=1}^m$  を  $\sum_{i \in I \setminus I_1} + \sum_{i \in I_1}$  と分解して、移項すると

$$\sum_{i \in I \setminus I_1} \mathbf{a}_i u_i = \mathbf{f}^{**} - \sum_{i \in I_1} \mathbf{a}_i u_i.$$

## 5.4 近似方程式の組み立て — 直接剛性法

ゆえに

$$A\mathbf{u}^* = \mathbf{f},$$

ただし

$\mathbf{u}^* := \mathbf{u}$  の第  $i$  成分 ( $i \in I_1$ ) を除いた縦ベクトル,

$A := A^{**}$  の第  $i$  列 ( $i \in I_1$ ) を除いた正方行列,

$$\mathbf{f} := \mathbf{f}^{**} - \sum_{i \in I_1} \mathbf{a}_i u_i.$$

実際にプログラムを作成するとき、 $A$  や  $\mathbf{f}$  が容易に求められることは次回解説する。

## 5.5 連立1次方程式の具体例

簡単な問題に対する有限要素法の連立1次方程式を実際に求めてみよう。

$$\Omega = (0, 1) \times (0, 1),$$

$$\Gamma_1 = \{(x, y) \mid x = 0, 0 \leq y \leq 1\} \cup \{(x, y) \mid 0 \leq x \leq 1, y = 0\},$$

$$\Gamma_2 = \{(x, y) \mid x = 1, 0 < y \leq 1\} \cup \{(x, y) \mid 0 < x \leq 1, y = 1\},$$

$$g_1 \equiv 0, \quad g_2 \equiv 0, \quad f \equiv \text{定数関数 } \bar{f}.$$

すなわち

$$-\Delta u = \bar{f} \quad (\text{in } \Omega), \quad u = 0 \quad \text{on } \Gamma_1, \quad \frac{\partial u}{\partial \mathbf{n}} = 0 \quad \text{on } \Gamma_2.$$

## 5.5 連立1次方程式の具体例

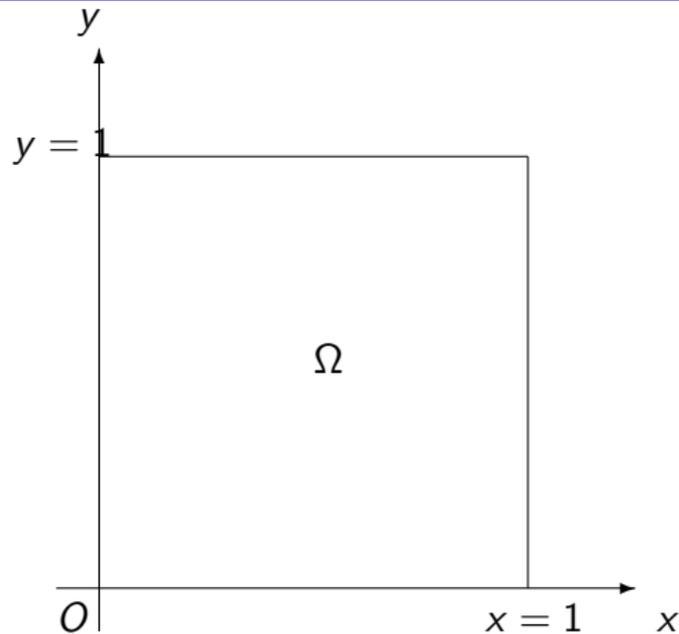


図 5: 領域  $\Omega$

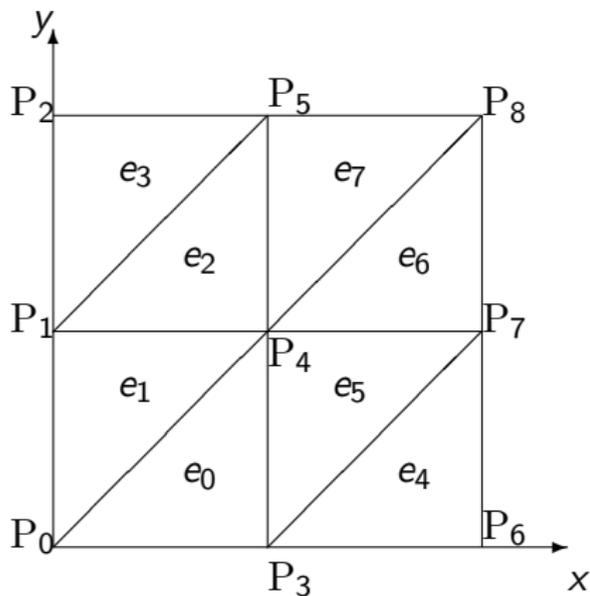


図 6: 要素分割

## 5.5 連立1次方程式の具体例

正方形領域  $\Omega$  を図2のように3三角形要素によって要素分割する。

有限要素は次の二つのタイプがある (タイプ I, II と呼ぶことにする)。各々に図3のように局所節点番号をつける (左下から反時計回り)。

タイプ I  $e_0, e_2, e_4, e_6$ .

タイプ II  $e_1, e_3, e_5, e_7$ .

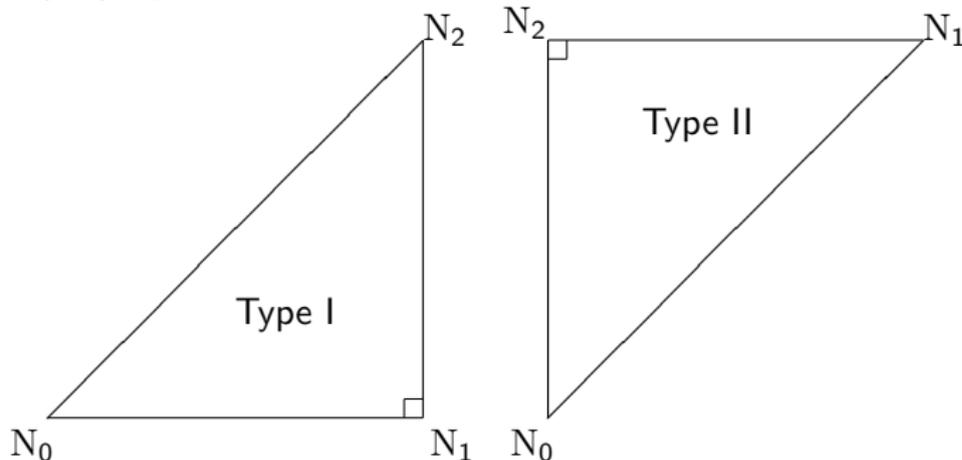


図 7: 二つのタイプの有限要素と局所節点番号

## 5.5 連立1次方程式の具体例

タイプが同じならば、要素係数行列  $A_k$ , 要素自由項ベクトル  $\mathbf{f}_k$  が等しいことはすぐ分かる。それぞれ  $A_I, A_{II}, \mathbf{f}_I, \mathbf{f}_{II}$  で表すことにする。

タイプ I については

$$D = h^2, \quad S = \frac{h^2}{2},$$

$$A_I = \frac{1}{2} \begin{pmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{pmatrix}, \quad \mathbf{f}_I = \frac{\bar{f}h^2}{6} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}.$$

タイプ II については

$$D = h^2, \quad S = \frac{h^2}{2},$$

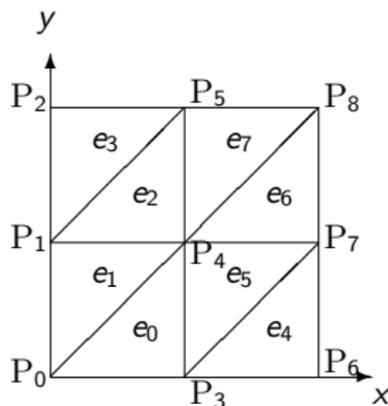
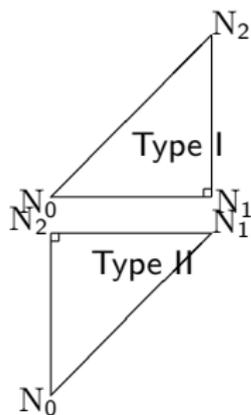
$$A_{II} = \frac{1}{2} \begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \\ -1 & -1 & 2 \end{pmatrix}, \quad \mathbf{f}_{II} = \frac{\bar{f}h^2}{6} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}.$$

## 5.5 連立1次方程式の具体例

これらから、全体的な近似方程式を作ろう。

そのために局所的な節点番号と、全体的な節点番号の対応づけが必要である。そこで以下のような対応表を用意する (スライド見る場合も手で写すことを推奨 — 要素タイプは不要)。

要素	$e_0$	$e_1$	$e_2$	$e_3$	$e_4$	$e_5$	$e_6$	$e_7$
要素タイプ	I	II	I	II	I	II	I	II
$N_0$ の全体節点番号	0	0	1	1	3	3	4	4
$N_1$ の全体節点番号	3	4	4	5	6	7	7	8
$N_2$ の全体節点番号	4	1	5	2	7	4	8	5



## 5.5 連立1次方程式の具体例

これから Galerkin 法の弱形式は

$$\mathbf{v}^\top \frac{1}{2} \begin{pmatrix} 2 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 4 & -1 & 0 & -2 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & 0 & 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 4 & -2 & 0 & -1 & 0 & 0 \\ 0 & -2 & 0 & -2 & 8 & -2 & 0 & -2 & 0 \\ 0 & 0 & -1 & 0 & -2 & 4 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 & 0 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & -2 & 0 & -1 & 4 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 2 \end{pmatrix} \begin{pmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \\ u_8 \end{pmatrix} = \mathbf{v}^\top \frac{\bar{f}h^2}{6} \begin{pmatrix} 2 \\ 3 \\ 1 \\ 3 \\ 6 \\ 3 \\ 1 \\ 3 \\ 2 \end{pmatrix}$$

for

$$\forall \mathbf{v} \in \left\{ (v_0, v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8)^\top \in \mathbb{R}^9 \mid v_0 = v_1 = v_2 = v_3 = v_6 = 0 \right\}.$$

## 5.5 連立1次方程式の具体例 Dirichlet境界条件の考慮

$P_i \in \Gamma_1$  となる  $i$  について (今の場合  $i = 0, 1, 2, 3, 6$ )、第  $i$  行は削除してよい。

$$\frac{1}{2} \begin{pmatrix} 0 & -2 & 0 & -2 & 8 & -2 & 0 & -2 & 0 \\ 0 & 0 & -1 & 0 & -2 & 4 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & -2 & 0 & -1 & 4 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 2 \end{pmatrix} \begin{pmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \\ u_8 \end{pmatrix} = \frac{\bar{f}h^2}{6} \begin{pmatrix} 6 \\ 3 \\ 3 \\ 3 \\ 2 \end{pmatrix}.$$

また  $P_i \in \Gamma_1$  となる  $i$  について、 $u_i = g_1(P_i)$  ( $= 0$ )。これを代入して移項すると

$$\frac{1}{2} \begin{pmatrix} 8 & -2 & -2 & 0 \\ -2 & 4 & 0 & -1 \\ -2 & 0 & 4 & -1 \\ 0 & -1 & -1 & 2 \end{pmatrix} \begin{pmatrix} u_4 \\ u_5 \\ u_7 \\ u_8 \end{pmatrix} = \frac{\bar{f}h^2}{6} \begin{pmatrix} 6 \\ 3 \\ 3 \\ 2 \end{pmatrix} + \begin{pmatrix} g_1(P_1) + g_1(P_3) \\ g_1(P_2)/2 \\ g_1(P_6)/2 \\ 0 \end{pmatrix}.$$

すなわち

$$\begin{pmatrix} 4 & -1 & -1 & 0 \\ -1 & 2 & 0 & -1/2 \\ -1 & 0 & 2 & -1/2 \\ 0 & -1/2 & -1/2 & 1 \end{pmatrix} \begin{pmatrix} u_4 \\ u_5 \\ u_7 \\ u_8 \end{pmatrix} = \begin{pmatrix} \bar{f}h^2 \\ \bar{f}h^2/2 \\ \bar{f}h^2/2 \\ \bar{f}h^2/3 \end{pmatrix}.$$

## 5.5 連立1次方程式の具体例 Dirichlet境界条件の考慮

上のやり方では、係数行列とベクトルが“縮小”される。いくつか留意すべき点:

- 例えば MATLAB では、 $(0, 1, 2, 3, 6)$ ,  $(4, 5, 7, 8)$  という添字ベクトルを用意すれば、全体係数行列と全体自由項ベクトルを求めるのは(コーディング上は)容易である。
- 自分で疎行列を扱うコードを書いていたりする場合はそれなりに面倒。
- データの移動にも計算コストがかかる。

Dirichlet 境界条件の処理には、他のやり方 (行列とベクトルを縮小しない方法) もある。

## 5.5 連立1次方程式の具体例 Dirichlet境界条件の考慮

### ベクトル、行列の縮小を避ける方法 (1)

$P_i \in \Gamma_1$  となる  $i$  (この例では  $i = 0, 1, 2, 3, 6$ ) に対して

- $i$  番目の方程式 ( $\mathbf{v}^\top$  のせいで  $\mathbf{0}^\top \mathbf{u} = 0$ ) を  $u_i = g_1(P_i)$  で置き換える。  
(結果的に係数行列の第  $i$  行は  $\mathbf{e}_i^\top$  で置き換える)

とすることで  $\mathbf{v}^\top$  が外せる。

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & -1 & 4 & -1 & 0 & -1 & 0 \\ 0 & 0 & -1/2 & 0 & -1 & 2 & 0 & 0 & -1/2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & -1/2 & 2 & -1/2 \\ 0 & 0 & 0 & 0 & 0 & -1/2 & 0 & -1/2 & 1 \end{pmatrix} \begin{pmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \\ u_8 \end{pmatrix} = \begin{pmatrix} g_1(P_0) \\ g_1(P_1) \\ g_1(P_2) \\ g_1(P_3) \\ \bar{f}h^2 \\ \bar{f}h^2/2 \\ g_1(P_6) \\ \bar{f}h^2/2 \\ \bar{f}h^2/3 \end{pmatrix}$$

これは正しい方程式であるが、係数行列が対称でなくなっている (数値計算で不利)。

そこで、係数行列の  $i$  列に **0でない非対角要素**があれば、それと  $g_1(P_i)$  との積を右辺に移項する。  
(その結果は次のスライド)

## 5.5 連立1次方程式の具体例 Dirichlet境界条件の考慮

$$\begin{pmatrix}
 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 4 & -1 & 0 & -1 & 0 \\
 0 & 0 & 0 & 0 & -1 & 2 & 0 & 0 & -1/2 \\
 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
 0 & 0 & 0 & 0 & -1 & 0 & 0 & 2 & -1/2 \\
 0 & 0 & 0 & 0 & 0 & -1/2 & 0 & -1/2 & 1
 \end{pmatrix}
 \begin{pmatrix}
 u_0 \\
 u_1 \\
 u_2 \\
 u_3 \\
 u_4 \\
 u_5 \\
 u_6 \\
 u_7 \\
 u_8
 \end{pmatrix}
 =
 \begin{pmatrix}
 g_1(P_0) \\
 g_1(P_1) \\
 g_1(P_2) \\
 g_1(P_3) \\
 \bar{f}h^2 \\
 \bar{f}h^2/2 \\
 g_1(P_6) \\
 \bar{f}h^2/2 \\
 \bar{f}h^2/3
 \end{pmatrix}
 +
 \begin{pmatrix}
 0 \\
 0 \\
 0 \\
 0 \\
 g_1(P_1) + g_1(P_3) \\
 g_1(P_2)/2 \\
 0 \\
 g_1(P_6)/2 \\
 0
 \end{pmatrix}
 .$$

## 5.5 連立1次方程式の具体例 Dirichlet境界条件の考慮

(説明のため、Galerkin法の弱形式を再度掲示)

$$\mathbf{v}^T \frac{1}{2} \begin{pmatrix} 2 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 4 & -1 & 0 & -2 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & 0 & 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 4 & -2 & 0 & -1 & 0 & 0 \\ 0 & -2 & 0 & -2 & 8 & -2 & 0 & -2 & 0 \\ 0 & 0 & -1 & 0 & -2 & 4 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 & 0 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & -2 & 0 & -1 & 4 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 2 \end{pmatrix} \begin{pmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \\ u_8 \end{pmatrix} = \mathbf{v}^T \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \\ f_8 \end{pmatrix}$$

for

$$\forall \mathbf{v} \in \left\{ (v_0, v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8)^T \in \mathbb{R}^9 \mid v_0 = v_1 = v_2 = v_3 = v_6 = 0 \right\}.$$

## 5.5 連立1次方程式の具体例 Dirichlet境界条件の考慮

### ベクトル、行列の縮小を避ける方法 (2) (FreeFem++で採用?)

$P_i \in \Gamma_1$  となる  $i$  に対して、行列の  $(i, i)$  成分を “terrible great value”  $\text{tgv}$  ( $= 10^{30}$ ) で置き換え、右辺のベクトルの第  $i$  成分を  $\text{tgv} \times g_1(P_i)$  で置き換える。方程式が近似方程式に置き換わってしまうが、以下の利点がある。

- 解は実質的にはほぼ変わらない (演算精度を 10 進 16 桁弱と想定してる)。
- 行列、ベクトルの縮小 (サイズ変更) は不要。
- 係数行列の対称性は保たれる。
- コーディングの負担 (手間) が少ない。

## 5.5 連立1次方程式の具体例 Dirichlet境界条件の考慮

$T$  を  $\text{tgv} (= 10^{30})$  として

$$\begin{pmatrix} T & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & T & -1 & 0 & -2 & 0 & 0 & 0 & 0 \\ 0 & -1 & T & 0 & 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & T & -2 & 0 & -1 & 0 & 0 \\ 0 & -2 & 0 & -2 & 8 & -2 & 0 & -2 & 0 \\ 0 & 0 & -1 & 0 & -2 & 4 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 & 0 & T & -1 & 0 \\ 0 & 0 & 0 & 0 & -2 & 0 & -1 & 4 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 2 \end{pmatrix} \begin{pmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \\ u_8 \end{pmatrix} = \begin{pmatrix} Tg_1(P_0) \\ Tg_1(P_1) \\ Tg_1(P_2) \\ Tg_1(P_3) \\ f_4 \\ f_5 \\ Tg_1(P_6) \\ f_7 \\ f_8 \end{pmatrix}$$

(実は、この方法が数値計算的にも妥当なものであるか、私自身は納得できていないところがある (行列の条件数が大きくなるが大丈夫?))

## 5.6 プログラム 5.6.1 方程式を立てるのに必要なもの

有限要素解を計算する (連立 1 次方程式を作る) ため、何が必要かまとめる。

上の例 ( $f \equiv \bar{f}$  (定数),  $g_1 \equiv 0$ ,  $g_2 \equiv 0$ ) では

- 節点の座標 ( $i = 1, \dots, m$  に対して  $P_i$  の座標  $(x_i, y_i)$ )
- $\Gamma_1$  上にある節点の全体節点番号
- 各要素  $e_k$  を構成する節点の全体節点番号  $i_{k,0}, i_{k,1}, i_{k,2}$

が必要になった。

一般の問題では、次のものも必要になる。

- Ⓐ  $\Omega$  に属する節点  $P_i$  での  $f$  の値  $f(P_i)$
- Ⓑ  $\Gamma_1$  上にある節点での  $g_1$  の値
- Ⓒ  $\Gamma_2$  上にある節点の全体節点番号
- Ⓓ  $\Gamma_2$  上にある節点での  $g_2$  の値

以上の情報があれば、Poisson 方程式の境界値問題を解くための一般的な方程式が作成できる。

( $\Omega, \Gamma_1, \Gamma_2, \{e_k\}, \{P_i\}, f, g_1, g_2$  などの情報は、プログラムの中に埋め込まずに入力データとして与えることが出来る。)

## 5.6.2 サンプルプログラム紹介

菊地 [?] にはサンプル・プログラム (FORTRAN, C 言語) も用意されている。

[?] の初版の FORTRAN プログラムを、移植した C 言語プログラムを紹介する。長いので別資料として紹介する。

# 6 C言語による2次元要素法サンプル・プログラムの紹介

## 6.1 進行表

- ① 百聞は一見しかず。まず実行例を見てもらう。
- ② プログラムが何をするか、入力と出力を理解する。  
有限要素解を求めるプログラム (`naive`, `band`) では、領域や三角形分割の情報を入力データとする。そのため一般性が高くなっている。
- ③ `naive` と `band` の比較をする。数学的にはやること同じ。効率の違いは？
- ④ プログラムの心臓部分 `assem()` と `ecm()` の解説 (説明したことの確認)。

## 6.2 試しに実行

参考 授業 WWW サイトの「有限要素法のサンプル C プログラム」

— 入手、展開、ファイル名確認 —

```
curl -O https://m-katsurada.sakura.ne.jp/program/fem/fem-mac-20221031.tar.gz
tar xzf fem-mac-20221031.tar.gz
cd fem-mac-20221031
ls
```

とりあえず動作チェック (実行には、cc, ccg (あるいは cglsc), make 等が必要)

— コンパイル&テスト —

make	プログラムのコンパイル
make test1	naive の動作確認 (辺を 2,4,8 分割したときの有限要素解の数値データ)
make test2	band の動作確認 (辺を 2,4,8 分割したときの有限要素解の数値データ)
make test3	band の動作確認 (辺を 2,4,8,16,32 分割したときの有限要素解の等高線表示) 等高線を描いたウィンドウをクリックすると次を表示

途中で引っかけた場合、相談して下さい。

もしかすると、今の院生の Mac には ccg がインストールされていないかも。  
その場合は `make test3` は実行できない。

ソースプログラム `naive.c`, `band.c` は、それぞれ 321 行、397 行である。

## 6.3 有限要素解を求めるプログラム naive, band の理解

2次元多角形領域  $\Omega$  における Poisson 方程式の同次 Dirichlet, Neumann 境界値問題

$$(46) \quad -\Delta u(x, y) = f(x, y) := \mathbf{1} \quad ((x, y) \in \Omega),$$

$$(47) \quad u(x, y) = g_1(x, y) := \mathbf{0} \quad ((x, y) \in \Gamma_1),$$

$$(48) \quad \frac{\partial u}{\partial \mathbf{n}}(x, y) = g_2(x, y) := \mathbf{0} \quad ((x, y) \in \Gamma_2)$$

を有限要素法で解くプログラムである。

**Q** ここで  $\Omega, \Gamma_1, \Gamma_2$  は何か？

**A** 実は  $\Omega, \Gamma_1, \Gamma_2$  についてはデータとして入力する。

naive, band とともに、**任意の領域&境界についての計算ができる。**

( $f, g_1, g_2$  については、簡単のため、特殊な値  $\mathbf{1}, \mathbf{0}, \mathbf{0}$  が仮定されている。これを一般化するのは適度の演習問題である。)

## 6.3 有限要素解を求めるプログラム naive, band の理解

入力データの例 input.dat

9	8	5			
0.0		0.0			
0.0		0.5			
0.0		1.0			
0.5		0.0			
0.5		0.5			
0.5		1.0			
1.0		0.0			
1.0		0.5			
1.0		1.0			
0	3	4	0	4	1
1	4	5	1	5	2
3	6	7	3	7	4
4	7	8	4	8	5
0	1	2	3	6	

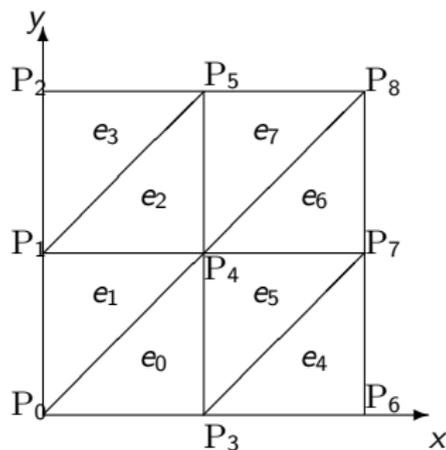


図 8: 要素分割 (各辺を 2 等分してから要素分割)

- 1 行目には、節点数 (nnode)、要素数 (nelmt)、 $\Gamma_1$  に属している節点数 (nbc)
- 2~10 行は、節点の座標  $(x_i, y_i)$  ( $i = 0, 1, \dots, \text{nnode} - 1$ )
- 11~14 行は、各要素を構成する節点の全体節点番号 (0 から nelmt-1 までの通し番号) 節点は各要素を左回りに回るように順序付けてある。
- 最後に  $\Gamma_1$  に属する節点の全体節点番号 (nbc 個の番号)

## 6.3 有限要素解を求めるプログラム naive, band の理解

この形式のデータがあれば、図が描ける (幾何的状況が分かる) ことを理解しよう。

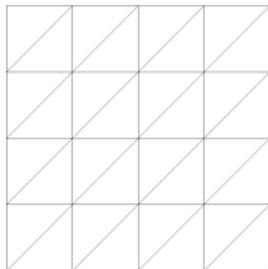
—— 三角形と (結果として) $\Omega$  を描く ——

```
./disp-glsc3d input.dat  
./disp-glsc3d input4.dat  
cat input4.dat | ./disp-glsc3d  
./make-input | ./disp-glsc3d
```

(最後のコマンドに対して、辺を何等分するか、数値 (例えば 64 とか) を入力しよう。)

コマンド 1 | コマンド 2 でコマンド 1 の出力をコマンド 2 に入力できる (パイプ機能)。

disp-glsc3d は上の形式のデータを図示するプログラム、make-input は正方形領域に対して上の形式のデータを作成するプログラムである。



## 6.3 有限要素解を求めるプログラム naive, band の理解

naive, band は上の形式の入力データから、有限要素解を計算するプログラム。

両者は同じ計算を行う。連立 1 次方程式の係数行列が**帯行列** (band matrix) であることを利用して、計算の効率化の工夫をしたのが band で、それをしないのが naive である。

—— 一辺 64 分割で解き比べ (CPU 時間計測), 解の等高線表示 ——

```
echo 64 | ./make-band-input > input64.dat
```

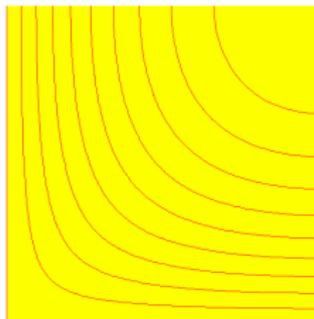
```
./disp-glsc3d input64.dat
```

```
time ./naive input64.dat
```

```
time ./band input64.dat
```

あるマシンで 17.5 秒 vs 0.02 秒 (naive は実際的ではない, ちなみに節点数 4225)

```
./contour-glsc3d band.out
```



## 6.4 プログラム naive の内部構造

主な関数には以下のようなものがある。

<code>main()</code>	
<code>input()</code>	入力データ読み込み
<code>assem()</code>	全体係数行列 $A$ , 全体自由項ベクトル $f$ の計算 (直接剛性法)
<code>ecm()</code>	要素係数行列 $A_e$ , 要素自由項ベクトル $f_e$ の計算
<code>solve()</code>	
<code>output()</code>	節点パラメーター (節点での解の値) を出力
<code>f()</code>	Poisson 方程式 $-\Delta u = f$ の右辺の既知関数 $f$

主な変数名

<code>nnode</code>	節点の総数
<code>nelmt</code>	有限要素の総数
<code>nbc</code>	$\Gamma_1$ (Dirichlet 境界条件を課す) 上の節点の総数
<code>x[nnode], y[nnode]</code>	節点の座標
<code>ielmt[nelmt].node[3]</code>	各有限要素を構成する節点の番号
<code>ibc[nbc]</code>	基本境界条件を課す節点の番号
<code>am[][]</code>	全体係数行列
<code>fm[]</code>	全体自由項ベクトル

## 6.4 プログラム naive の内部構造 `assem()`

`assem()` は連立 1 次方程式を組み立てる関数。

$A^* := \sum_{k=0}^{N_e-1} A_k^*$  と  $f^* := \sum_{k=0}^{N_e-1} f_k^*$  を次のように計算する。

```
/* assemblage of total matrix and vector; */
for (k = 0; k < nelmt; k++) {
    ecm(k, ielmt, x, y, ae, fe);
    for (i = 0; i < 3; i++) {
        ii = ielmt[k].node[i];
        fm[ii] += fe[i];
        for (j = 0; j < 3; j++) {
            jj = ielmt[k].node[j];
            am[ii][jj] += ae[i][j];
        }
    }
}
```

`ielmt[k].node[i]` は、要素  $e_k$  の、局所節点番号が  $i$  の節点  $N_i$  の全体節点番号

## 6.4 プログラム naive の内部構造 ecm()

ecm() は要素係数行列  $A_k$ 、要素自由項ベクトル  $f_k$  を求める関数。

```
/* 節点の座標を求める */
for (i = 0; i < 3; i++) {
    j = ielmt[k].node[i];
    xe[i] = x[j];
    ye[i] = y[j];
}
```

節点の座標さえ求まれば、 $A_k$ ,  $f_k$  の成分は公式に従って計算するだけである。(それを確かめたければ、naive.c あるいは band.c を見よ。)

## 6.5 参考課題

以前、FreeFem++ が使えなかった頃は、授業で次のような課題を出していた。

私は「百見は一験にしかず」と考えていて、次のような実験をすることは有益と知っているが、この科目では要求しない。

- a) このプログラムで解ける問題は、境界条件が同次境界条件

$$u = 0 \quad \text{on } \Gamma_1, \quad \frac{\partial u}{\partial \mathbf{n}} = 0 \quad \text{on } \Gamma_2$$

であるが、これを非同次境界条件

$$u = g_1 \quad \text{on } \Gamma_1, \quad \frac{\partial u}{\partial \mathbf{n}} = g_2 \quad \text{on } \Gamma_2$$

に変える。

- b) 自分で選んだ領域を三角形分割して、このプログラムに入力できるデータを生成するプログラムを書く。

# 参考文献