

# 数値解析入門

小俣正朗（金沢大学 理工研究域）

## 概要

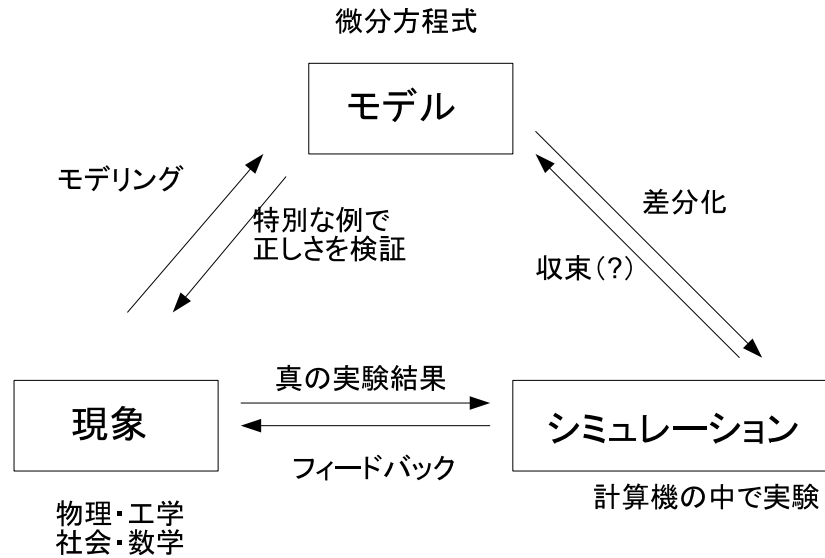
偏微分方程式の数値解析方法, 特に, 有限要素法について多角的な角度から考察を行い, 実際のプログラム作成までの学習を行う.

## 講義のあらすじ

本講義は, 次の順序で行う.

- 0 導入, 講義のあらすじ紹介
- 1 熱型方程式入門 (解の構成・差分法)
- 2 常微分方程式の数値解法
- 3 ガウスの消去法・LU 分解・三重対角行列
- 4 有限要素法とヒルベルト空間
- 5 二次元楕円型方程式
- 6 CG 法 (CG 法紹介, 速度向上, 他の方法との比較)
- 7 定常反復法
- 8 離散勾配流法
- A 付録: 熱方程式の解の評価
- B 付録: クリロフ部分空間
- C 付録: MPI プログラミング

# 1 微分方程式を中心としたシミュレーションの世界



- モデルの正しさ (微分方程式と現象は?)
- モデルの正しさ (微分方程式と差分方程式の関係は?)

⇒

- 現象の予測 (例：天気予報)
- 機器の設計 (例：飛行機, 原子炉)
- 意外な例 ⇒ 金融商品の設計

Computational Science

知りたいこと

- 現象の **メカニズム**
- 現象の **予測**

これらを明確にしたい. そのための **数理的手法の確立** が目標となる.

熱伝導方程式を例にとって Computational Science をやってみる.

(1) 現象とその理解

1次元の現象を考えることにする.

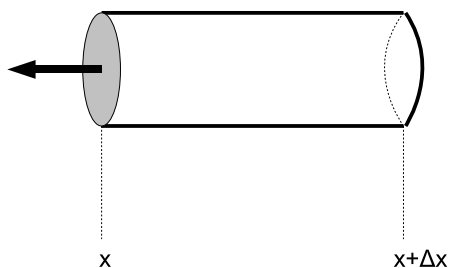


1. 「細い」金属棒.
2. 放射は無視.
3. 熱の伝わり方 …… 何を基準にするか?

(ア) 熱量 ~ エネルギーに着目.

(イ) 測定できるものは温度であるから, (ア) の量を温度を使って表したい.

⇒ 熱量は温度に比例すると仮定する.



$u(x, t)$  を温度分布とする (比熱:  $Q$ ). また,  $x$  での熱の流量は「温度勾配に比例する」とする. ここで微小領域  $(x, x + \Delta x)$  での熱の流量を考えると

$$\begin{cases} \text{熱の流量 (左向き at } x) = -c \frac{\partial u}{\partial x}(x, t) \\ \text{熱の流量 (右向き at } x + \Delta x) = c \frac{\partial u}{\partial x}(x + \Delta x, t) \end{cases}$$

となるので, 熱の変化量は

$$\Delta x \frac{\partial u}{\partial t} = c \left( \frac{\partial u}{\partial x}(x + \Delta x, t) - \frac{\partial u}{\partial x}(x, t) \right)$$

だから

$$\frac{\partial u}{\partial t} = c \frac{\left( \frac{\partial u}{\partial x}(x + \Delta x, t) - \frac{\partial u}{\partial x}(x, t) \right)}{\Delta x}$$

極限をとると

$$\frac{\partial u}{\partial t} = c \frac{\partial^2 u}{\partial x^2}$$

これが求める方程式になる.

## 1.1 熱方程式

簡単の為に  $c = 1$  とする.

$$\begin{cases} u_t = \Delta u, & (x, t) \in (0, \pi) \times (0, T) \\ u(x, 0) = f(x), & (f(0) = f(\pi) = 0 \text{ を仮定}) \\ u(0, t) = u(\pi, t) = 0 \end{cases}$$

変数分離法とフーリエ級数による解法を説明する. まず  $u = X(x)T(t)$  と置く. 方程式に代入すれば

$$XT' = TX''$$

となるので,

$$\frac{T'}{T} = \frac{X''}{X} = \lambda \text{ (定数)}$$

ここで,

$$X'' = \lambda X, \quad X(0) = X(\pi) = 0$$

であるから,  $\lambda = -n^2$  ( $n = 1, 2, 3, \dots$ ) でなければ,  $X(0) = X(\pi) = 0$  をみたすような解はない.  $\lambda = -n^2$  ならば,

$$X(x) = \sin nx$$

が求める解である.

一方,

$$T' = -n^2 T, \quad T(0) = 1$$

を解けば,

$$T = e^{-n^2 t}$$

となる. よって熱方程式とその境界条件を満たす  $u = XT$  は

$$e^{-n^2 t} \sin nx$$

と書くことができる. 「解の和は解」また「解の定数倍は解」(線形性)より

$$\sum_{n=1}^{\infty} c_n e^{-n^2 t} \sin nx \quad (c_n \text{ は定数})$$

が解であることが期待される. これが真の解になっていることは付録で証明する.

## 1.2 差分法

本小節でも方程式  $\frac{\partial u}{\partial t} = c \frac{\partial^2 u}{\partial x^2}$  を考察する. ただし  $0 \leq t \leq T$  とする.

$(0, \pi)$  を  $N$  等分して,  $\frac{\pi}{N} = \Delta x$  とおく. また,  $(0, T)$  も同様に  $M$  等分し,

$$\frac{T}{M} = \Delta t \text{ とおく.}$$

### 1.2.1 陽解法 (Explicit Method)

まずは、陽解法から紹介しよう。  $\frac{\partial u}{\partial t} = c \frac{\partial^2 u}{\partial x^2}$  を次のように差分化する。

$$\frac{u(x, t + \Delta t) - u(x, t)}{\Delta t} = c \frac{u(x + \Delta x, t) - 2u(x, t) + u(x - \Delta x, t)}{(\Delta x)^2}$$

これを整理すると、

$$\begin{aligned} u(x, t + \Delta t) &= u(x, t) + \frac{c(\Delta t)}{(\Delta x)^2} (u(x + \Delta x, t) - 2u(x, t) + u(x - \Delta x, t)) \\ &= \frac{c(\Delta t)}{(\Delta x)^2} (u(x + \Delta x, t) + u(x - \Delta x, t)) + \left(1 - \frac{2c(\Delta t)}{(\Delta x)^2}\right) u(x, t) \end{aligned}$$

となる。

ここで、 $u_{i,j} = u(x, y)$  とし、 $u_{i+k,j+h} = u(x + k\Delta x, t + h\Delta t)$  などとおくと、次の漸化式を得る。

$$u_{i,j+1} = \frac{c(\Delta t)}{(\Delta x)^2} (u_{i+1,j} + u_{i-1,j}) + \left(1 - \frac{2c(\Delta t)}{(\Delta x)^2}\right) u_{i,j}$$

ここで、陽解法の特徴を2つ挙げておこう。

(i) 次の繰り返し文で、プログラムが可能であり、容易にできる。

```
for j = 0, 1, ..., M - 1 do :
  for i = 1, 2, ..., N - 1 do :
     $u_{i,j+1} = \frac{c\Delta t}{(\Delta x)^2} (u_{i+1,j} + u_{i-1,j}) + \left(1 - \frac{2c\Delta t}{(\Delta x)^2}\right) u_{i,j}$ 
  end do
end do
```

(ii)  $1 - \frac{2c(\Delta t)}{(\Delta x)^2}$  の符号が変わりうるため、不安定である。

### 1.2.2 陽解法の安定性解析

陽解法の安定性を調べる。陽解法は格子点上で以下の漸化式に従っている：

$$u_{m,j+1} = \mu u_{m-1,j} + (1 - 2\mu) u_{m,j} + \mu u_{m+1,j}$$

ただし、 $\mu = c \frac{\Delta t}{(\Delta x)^2}$ 、 $x_m = m\Delta x$  である。

ここで、この漸化式に波数  $k$  の正弦波成分を代入して、その成長を見よう。 Fourier

の方法をふまえると波数  $k$  の成分は格子点上で以下のようにふるまうとしてよい :

$$u_{m,j} = \lambda^j \exp(ikx_m) \quad (1.1)$$

なぜならば, Fourier 級数による解を見ればわかるように, ある三角関数成分は  $n = k$  のとき,

$$\exp(-k^2 t) \sin kt$$

と表すことができるので,  $m$  番目のメッシュでは

$$(\exp(-k^2))^{j\Delta t} \sin km \cdot \Delta x = (\exp(-k^2 \Delta t))^j \sin(k \cdot \Delta x)m$$

とふるまうと思ってよい. このようにふるまう解は一般に, 格子点  $(m, j)$  上で

$$(\exp(-k^2 \Delta t))^j \exp(i(k \cdot \Delta x)m)$$

と表現できるからである.

このような関数のもつ特徴が, メッシュ上の差分計算だけで保てればよい. この場合そうはいかない場合がある. 式 (1.1) についてもしある波数  $k$  に対して  $|\lambda| > 1$  であるならば, 波数  $k$  の成分が指数的に成長してしまう (数値的に不安定: すなわち実際の Fourier 成分は減衰するのに数値的には爆発することになる).

よって式 (1.1) については (差分計算をいくらかえしても) 全ての波数  $k$  に対し  $|\lambda| \leq 1$  でなければならない. 実際そのようになるかどうかチェックしてみよう.

$$\begin{aligned} \lambda^{j+1} \exp(ikx_m) &= \mu \lambda^j \exp(ikx_{m+1}) + (1 - 2\mu) \lambda^j \exp(ikx_m) + \mu \lambda^j \exp(ikx_{m-1}) \\ \lambda &= \mu \exp(ik\Delta x) + (1 - 2\mu) + \mu \exp(-ik\Delta x) \\ &= 1 - 2\mu(1 - \cos k\Delta x) = 1 - 4\mu \sin^2 \frac{k\Delta x}{2} \end{aligned}$$

よって  $\lambda \leq 1$  は常に成立する.  $\lambda \geq -1$  を満たすためには

$$1 - 4\mu \sin^2 \frac{k\Delta x}{2} \geq -1 \quad \text{for all } k$$

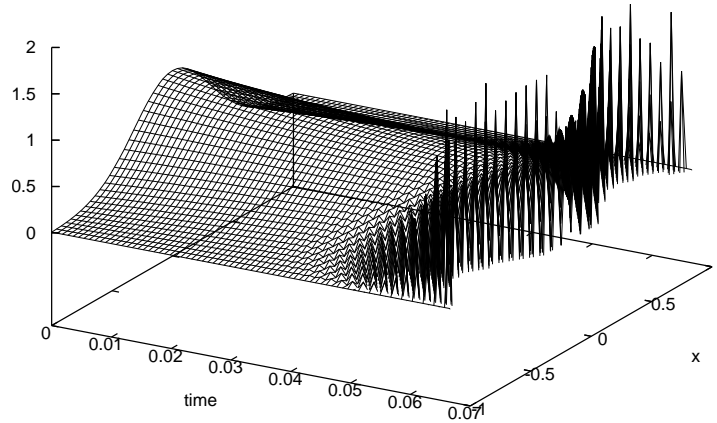
$$1 - 4\mu \geq -1$$

ゆえに

$$c \frac{\Delta t}{(\Delta x)^2} \leq \frac{1}{2}$$

となることが求められる. これが陽解法が安定であるための条件である. 陽解法が不安定である場合の数値実験例を次に図示する.

$$(\Delta x = 0.04, \Delta t = 0.0009, c = 1.0)$$



### 1.2.3 陰解法 (Implicit Method)

次に陰解法を説明しよう. 陰解法は,  $\frac{\partial u}{\partial t} = c \frac{\partial^2 u}{\partial x^2}$  を次のように差分する.

$$\frac{u(x, t) - u(x, t - \Delta t)}{\Delta t} = c \frac{u(x + \Delta x, t) - 2u(x, t) + u(x - \Delta x, t)}{(\Delta x)^2}$$

この式を, 左辺を時刻  $t$  で, 右辺を時刻  $t - \Delta t$  で整理すると,

$$u(x, t) - \frac{c(\Delta t)}{(\Delta x)^2} (u(x + \Delta x, t) - 2u(x, t) + u(x - \Delta x, t)) = u(x, t - \Delta t)$$

となり,

$$-\frac{c(\Delta t)}{(\Delta x)^2} u(x + \Delta x, t) + \left(1 + 2\frac{c(\Delta t)}{(\Delta x)^2}\right) u(x, t) - \frac{c(\Delta t)}{(\Delta x)^2} u(x - \Delta x, t) = u(x, t - \Delta t)$$

前と同じ記号を使えば, 次のような式になる.

$$\left(1 + 2\frac{c(\Delta t)}{(\Delta x)^2}\right) u_{i,j} - \frac{c(\Delta t)}{(\Delta x)^2} (u_{i+1,j} + u_{i-1,j}) = u_{i,j-1}$$

これは次の連立方程式で表現できて, 各  $u_{i,j}$  を求めることは, それを解くことにあたる.

$$\frac{c(\Delta t)}{(\Delta x)^2} = a \text{ とおくと,}$$

$$\begin{bmatrix} 1+2a & -a & 0 & \cdots & \cdots & 0 \\ -a & 1+2a & -a & 0 & \cdots & 0 \\ 0 & -a & 1+2a & 0 & \cdots & 0 \\ \vdots & & & & & \vdots \\ \vdots & & & & & \vdots \\ 0 & \cdots & 0 & -a & 1+2a & -a \\ 0 & \cdots & 0 & 0 & -a & 1+2a \end{bmatrix} \begin{bmatrix} u_{1,j} \\ u_{2,j} \\ \vdots \\ \vdots \\ \vdots \\ u_{N-2,j} \\ u_{N-1,j} \end{bmatrix} = \begin{bmatrix} u_{1,j-1} + au_{0,j} \\ u_{2,j-1} \\ \vdots \\ \vdots \\ \vdots \\ u_{N-2,j-1} \\ u_{N-1,j-1} + au_{N,j} \end{bmatrix}$$

ただし,  $u_{0,j}, u_{N,j}$  は境界条件より定まっている.  
ここで, 陰解法の特徴を二つあげておこう.

- (i) 連立方程式を解かねばならない.
- (ii) 安定な解法である. (次節で述べる)

### 1.2.4 陰解法の安定性解析

同様に陰解法の安定性についても考えてみる. 対応する漸化式は以下のようになる.

$$-\mu u_{m+1,j+1} + (1+2\mu)u_{m,j+1} - \mu u_{m-1,j+1} = u_{m,j}$$

ただし,  $\mu = c \frac{\Delta t}{(\Delta x)^2}$ ,  $x_m = m\Delta x$  である.

$$\begin{aligned} -\mu \lambda^{j+1} \exp(ikx_{m+1}) + (1+2\mu)\lambda^{j+1} \exp(ikx_m) - \mu \lambda^{j+1} \exp(ikx_{m-1}) &= \lambda^j \exp(ikx_m) \\ \lambda(-\mu \exp(ik\Delta x) + (1+2\mu) - \mu \exp(-ik\Delta x)) &= 1 \\ \lambda\{1+2\mu(1-\cos k\Delta x)\} &= 1 \\ \lambda &= \frac{1}{1+4\mu \sin^2 \frac{k\Delta x}{2}} \end{aligned}$$

ゆえに常に  $|\lambda| \leq 1$  だから, 陰解法は無条件安定である.

## 2 常微分方程式の数値解法

ここでは, 以下の常微分方程式の数値解法について考える.

$$\begin{cases} \frac{du}{dt} = f(u) \\ u(0) = u_0 \end{cases} \quad (2.1)$$



方程式 (2.1) の両辺を時間区間  $(t, t + \Delta t)$  で積分すると,

$$u(t + \Delta t) - u(t) = \int_t^{t+\Delta t} f(u(\tau))d\tau \quad (2.2)$$

となる. 式 (2.2) の右辺を近似的求めることで数値計算を行う. 近似の仕方により様々な数値解法が存在する.

## 2.1 Euler 法

積分に対して長方形公式を適用する場合, 次の漸化式を得る.

$$u(t + \Delta t) = u(t) + \Delta t f(u(t)) \quad (2.3)$$

式 (2.3) を用いて微分方程式を解いていく方法が Euler 法である. この方法では打ちきり誤差が  $O(\Delta t^2)$  である.

## 2.2 2 次の Runge-Kutta 法

2 次の Runge-Kutta 法の場合, 式 (2.2) の積分の近似には台形則を用いる.

$$\int_t^{t+\Delta t} f(u(\tau))d\tau \simeq \frac{\Delta t}{2} (f(u(t)) + f(u(t + \Delta t)))$$

ここで  $u(t + \Delta t)$  は未知の値であるので Euler 法の式 (2.3) を用いて,

$$f(u(t + \Delta t)) \simeq f(u(t) + \Delta t f(u(t)))$$

と近似的に求める. 従って, 漸化式は

$$u(t + \Delta t) = u(t) + \frac{\Delta t}{2} (f(u(t)) + f(u(t) + \Delta t f(u(t)))) \quad (2.4)$$

となる. この方法では打ちきり誤差が  $O(\Delta t^3)$  となり Euler 法と比べて精度が向上している.

## 2.3 4 次の Runge - Kutta 法

式 (2.2) の右辺をシンプソン則で積分を計算する.

$$\int_t^{t+\Delta t} f(u(\tau))d\tau = \frac{\Delta t}{6} (f(u(t)) + 4f(u(t + \Delta t/2)) + f(u(t + \Delta t)))$$

未知関数を含む  $f(u(t + \Delta t/2))$ ,  $f(u(t + \Delta t))$  は次のように扱う. まず, Euler 法を用いて,

$$u_1 = u(t) + \frac{\Delta t}{2} f(u(t))$$

と置く. さらに, 次の近似を用いる.

$$u_2 = u(t) + \frac{\Delta t}{2} f(u_1)$$

$u_1, u_2$  はいずれも  $u(t + \Delta t/2)$  を近似的に求めたものである. これらを用いて,  $f(u(t + \Delta t/2))$  を以下のように近似する.

$$f(u(t + \Delta t/2)) \simeq \frac{1}{2}(f(u_1) + f(u_2))$$

とする. また,  $f(u(t + \Delta t))$  は

$$u_3 = u(t) + \Delta t f(u_2)$$

として,

$$f(u(t + \Delta t)) \simeq f(u_3)$$

と近似する. まとめると, 4 次の Runge-kutta 法は, 次の形式に整理される.

$$u(t + \Delta t) = u(t) + \frac{1}{6}(k_0 + 2k_1 + 2k_2 + k_3) \quad (2.5)$$

$$k_0 = \Delta t f(u(t))$$

$$k_1 = \Delta t f(u(t) + k_0/2)$$

$$k_2 = \Delta t f(u(t) + k_1/2)$$

$$k_3 = \Delta t f(u(t) + k_2) .$$

ここで各  $k_i$  は  $k_0$  から順番に計算できるようになっていることに注意しよう. この方法は打ちきり誤差が  $O(\Delta t^4)$  となる. 精度の高さやアルゴリズムの簡単さにより, 常微分方程式の数値解法にはよく用いられる方法である.

## 2.4 数値計算例

具体的な例として, 以下の微分方程式を各数値解法で解いた結果を紹介する.

$$\begin{cases} \frac{d^2 x}{dt^2} = -x \\ x(0) = 1, \left. \frac{dx}{dt} \right|_{t=0} = 0 \end{cases} \quad (2.6)$$

2階微分方程式の場合,

$$\frac{dz}{dt} = g(z), \quad z = (x, v), \quad g = (v, -x)$$

と1階微分方程式として各数値解法を用いる方法の他に, 式 (2.6) を

$$\frac{x(t + \Delta t) - 2x(t) + x(t - \Delta t)}{\Delta t^2} = -x(t) \quad (2.7)$$

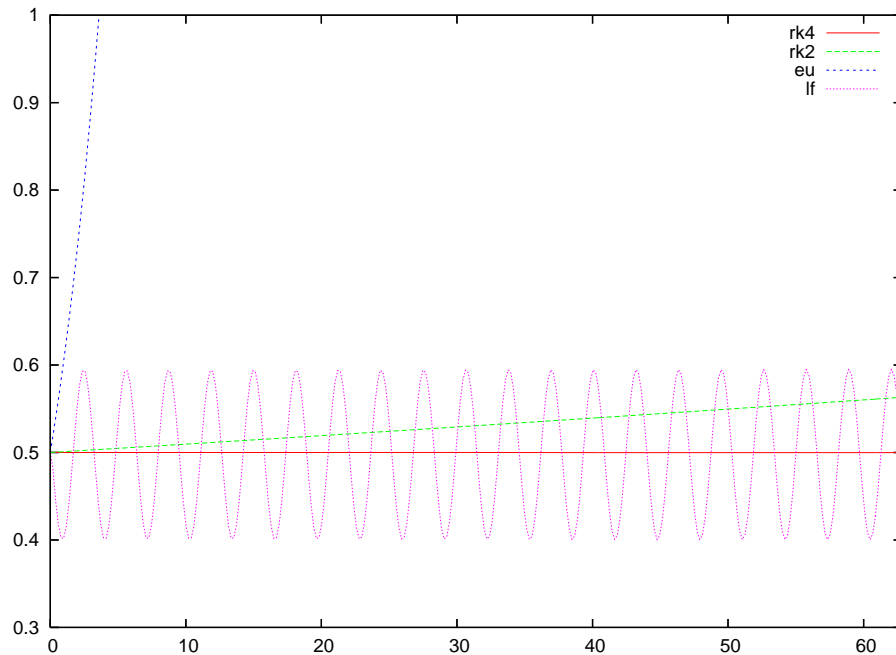
$$x(0) = 1, \quad x(-\Delta t) = (1 - \Delta t^2/2)x(0)$$

と離散化して解く方法がある. この場合, 式 (2.7) は leap-frog 法と呼ばれる.

式 (2.6) の解は  $x(t) = \cos(t)$  である. 各手法を用いて得られた解の性質を見てみよう. 方程式 (2.6) はエネルギー

$$\frac{1}{2} \left( x^2 + \frac{dx^2}{dt} \right)$$

が保存する.  $\Delta t = \pi/2$  として Euler 法, 2 次の Runge-Kutta 法, 4 次の Runge-Kutta 法, leap-frog 法で解き, エネルギーの保存性を確かめる.



上の図は横軸を時刻  $t$ , 縦軸をエネルギーで取ったものである. 青色の線が Euler 法, 緑色は 2 次の Runge-Kutta 法, 赤色は 4 次の Runge-Kutta 法, ピンク色は leap-frog 法の場合である. Euler 法, 2 次の Runge-Kutta 法, により得られた解ではエネルギーが時間と共に増大してしまっている. 対して leap-frog 法で得られた解はエネルギーは時間的に振動する. 4 次の Runge-Kutta 法は図ではエネルギーが保存している様に見えるが, 実際には時間と共に減少してしまう.

### 3 行列の取り扱い

陰解法を解くためには、行列方程式を解かねばならない。本章では、基本的な行列方程式の解き方を紹介しよう。

#### 3.1 ガウスの消去法

前節の陰解法で連立方程式が出てきた。このような方程式を解く方法のうち、比較的簡単なものをここで議論する。

$N$  元連立一次方程式

$$Ax = b \quad (3.1)$$

の解を求める方法の一つにガウスの消去法がある。ここで

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,N} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,N} \\ \vdots & & & \vdots \\ a_{N,1} & a_{N,2} & \cdots & a_{N,N} \end{pmatrix}, \quad x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix}, \quad b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{pmatrix}$$

である。

ガウスの消去法は方程式 (2.1) に対して、両辺に左から行の基本変形行列を乗じ、 $A$  を上三角行列に変形出来ることに基づく。具体的には  $k = 1, \dots, N$  に対して順に次の処理を行なうことによって変形することが出来る (前進消去)。

$k$  行目の全ての要素  $a_{k,k}, a_{k,k+1}, \dots, a_{k,N}$  及び  $b_k$  を  $a_{k,k}$  で割る。そしてこの  $k$  行目に  $a_{l,k}$ , ( $k < l \leq N$ ) を掛けて  $l$  行目から引き、 $k+1$  行目以降の第  $k$  列成分を全て 0 とする。前進消去の過程で、絶えず行列  $A$  の要素と右辺  $b$  の値が更新される。

```
for  $k = 1, 2, \dots, N$  do:
  for  $l = k, k + 1, \dots, N$  do:
     $a_{k,l} = \frac{a_{k,l}}{a_{k,k}}$ 
  end do
   $b_k = \frac{b_k}{a_{k,k}}$ 
  for  $l = k + 1, \dots, N$  do:
    for  $m = k, k + 1, \dots, N$  do:
       $a_{l,m} = a_{l,m} - a_{l,k}a_{k,m}$ 
    end do
     $b_l = b_l - a_{l,k}b_k$ 
  end do
end do
```

このとき、ある  $k$  行目で  $a_{k,k} = 0$  となると次の  $k+1$  段目へ進むことができなくなってしまう。これを避けるために、 $k$  行目の前進消去の前に  $a_{k,k}, a_{k,k+1}, \dots, a_{k,n}$  の中で絶対値が最大になるものを  $a_{k,k}$  に来るように行列を変形する。具体的には最大要素が  $a_{k,m}$  であったとき  $k$  列目と  $m$  列目を交換する (ピボット選択)。

前進消去によって係数行列  $A$  は単位上三角に変形されており、 $x_N$  の値が  $b_N$  として得られている。従って  $N-1$  行目から遡って代入操作を行っていけば次々と解  $x_{N-1}, x_{N-2}, \dots, x_1$  が求まる (後退代入)。

for  $k = N-1, N-2, \dots, 1$  do:  

$$x_k = b_k - \sum_{k < l \leq N} a_{k,l} x_l$$
  
 end do

### 3.2 LU 分解法

前節のガウスの消去法でも考えたように、係数行列が上三角、或は下三角であれば順次代入操作を行なうだけで線形方程式は求解できる。そこで係数行列  $A$  を下三角行列  $L$  と上三角行列  $U$  の積に分解する ( $A = LU$ )。

$A, L, U$  をそれぞれ

$$A = (a_{i,j}) = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,N} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,N} \\ \vdots & & & \vdots \\ a_{N,1} & a_{N,2} & \cdots & a_{N,N} \end{pmatrix}$$

$$L = (l_{i,j}) = \begin{pmatrix} l_{1,1} & 0 & \cdots & 0 \\ l_{2,1} & l_{2,2} & \cdots & 0 \\ \vdots & & & \vdots \\ l_{N,1} & l_{N,2} & \cdots & l_{N,N} \end{pmatrix}$$

$$U = (u_{i,j}) = \begin{pmatrix} u_{1,1} & u_{1,2} & \cdots & u_{1,N} \\ 0 & u_{2,2} & \cdots & u_{2,N} \\ \vdots & & & \vdots \\ 0 & 0 & \cdots & u_{N,N} \end{pmatrix}$$

と表すと、行列の積  $LU$  は具体的に次のように計算できる：

$$(LU)_{i,j} = \sum_{1 \leq k \leq N} l_{i,k} u_{k,j} = \sum_{1 \leq k \leq \min\{i,j\}} l_{i,k} u_{k,j} = \begin{cases} \sum_{1 \leq k \leq j} l_{i,k} u_{k,j} & (i \geq j) \\ \sum_{1 \leq k \leq i} l_{i,k} u_{k,j} & (j \geq i) \end{cases}$$

したがって,

$$LU = \begin{pmatrix} l_{1,1}u_{1,1}^{(1)} & l_{1,1}u_{1,2}^{(2)} & l_{1,1}u_{1,3}^{(3)} & \cdots \\ l_{2,1}u_{1,1}^{(N+1)} & l_{2,1}u_{1,2} + l_{2,2}u_{2,2}^{(N+2)} & l_{2,1}u_{1,3} + l_{2,2}u_{2,3}^{(N+3)} & \cdots \\ l_{3,1}u_{1,1}^{(2N+1)} & l_{3,1}u_{1,2} + l_{3,2}u_{2,2}^{(2N+2)} & l_{3,1}u_{1,3} + l_{3,2}u_{2,3} + l_{3,3}u_{3,3}^{(2N+3)} & \cdots \\ \dots & \dots & \dots & \dots \end{pmatrix}.$$

である.

さて,  $A = LU$  より  $a_{i,j} = (LU)_{i,j}$  ( $1 \leq i, j \leq N$ ) であるから条件は  $N^2$  である. 他方, 未知変数は  $\{l_{i,j}\}_{1 \leq i \leq j \leq N}$  の  $N(N+1)/2$  個と  $\{u_{i,j}\}_{1 \leq j \leq i \leq N}$  の  $N(N+1)/2$  個の合計  $N(N+1)$  個であるから条件過多である. そこで通常  $l_{i,i} = 1$  ( $1 \leq i \leq N$ ) の条件を課し  $L$  を単位下三角行列として求める.  $l_{i,j}, u_{i,j}$  については条件 (1), (2),  $\dots$ , (N), (N+1),  $\dots$ , (N+2),  $\dots$ , ( $N^2$ ) という順に用いていけば求めることができる.

具体的には

$$\begin{aligned} (1) \quad & u_{1,1} = a_{1,1} \\ (2) \quad & u_{1,2} = a_{1,2} \\ (3) \quad & u_{1,3} = a_{1,3} \\ & \dots \\ (N) \quad & u_{1,N} = a_{1,N} \\ & \dots \\ (N+1) \quad & l_{2,1} = a_{2,1}/u_{1,1} \\ (N+2) \quad & u_{2,2} = a_{2,2} - l_{2,1}u_{1,2} \\ (N+3) \quad & u_{2,3} = a_{2,3} - l_{2,1}u_{1,3} \\ (N+4) \quad & u_{2,4} = a_{2,4} - l_{2,1}u_{1,4} \\ & \dots \\ (2N+1) \quad & l_{3,1} = a_{3,1}/u_{1,1} \\ (2N+2) \quad & l_{3,2} = [a_{3,2} - l_{3,1}u_{1,2}]/u_{2,2} \\ & \dots \\ & \dots \end{aligned}$$

となる. 以上より LU 分解の条件式は

$$\begin{aligned} a_{i,j} &= \sum_{1 \leq k \leq j} l_{i,k}u_{k,j} \quad (i \geq j) \\ a_{i,j} &= \sum_{1 \leq k \leq i} l_{i,k}u_{k,j} \quad (j \geq i) \end{aligned} \tag{3.2}$$

ただし

$$l_{i,j} = \begin{cases} 0 & (i < j) \\ 1 & (i = j) \end{cases}, \quad u_{i,j} = 0 \quad (i > j)$$

である. (2.2) 式から

$$\begin{cases} l_{i,j} = \left[ a_{i,j} - \sum_{1 \leq k < j} l_{i,k} u_{k,j} \right] / u_{j,j} & (i > j) \\ u_{i,j} = a_{i,j} - \sum_{1 \leq k < i} l_{i,k} u_{k,j} & (j > i) \end{cases} \quad (3.3)$$

従って LU 分解のプロセスは次のようにまとめることができる.

```
for i = 1, 2, ..., N do:
  for j = i, i + 1, ..., N do:
    if i > j then
       $l_{i,j} = \left[ a_{i,j} - \sum_{1 \leq k < j} l_{i,k} u_{k,j} \right] / u_{j,j}$ 
    end if
    if i ≤ j then
       $u_{i,j} = a_{i,j} - \sum_{1 \leq k < i} l_{i,k} u_{k,j}$ 
    end if
  end do
end do
```

LU 分解を行なえば, 方程式 (2.1) は  $Ly = b$  かつ  $Ux = y$  と同値である. 前者は

$$y_i = b_i - \sum_{1 \leq j < i} l_{i,j} y_j \quad (i = 1, 2, \dots, N) \quad (3.4)$$

で求解でき, 後者は

$$x_i = \left[ y_i - \sum_{i < j \leq N} u_{i,j} x_j \right] / u_{i,i} \quad (i = N, N - 1, \dots, 1) \quad (3.5)$$

で求解できる.

### 3.3 三重対角行列の数値解法

さて,  $N$  元連立一次方程式

$$Ax = b$$

が与えられているとき, 行列  $A$  が以下のような三重対角行列である場合の数値計算法を紹介しよう.

$$\begin{pmatrix} a_1 & c_1 & 0 & 0 & \cdots & 0 & 0 \\ d_1 & a_2 & c_2 & 0 & \cdots & 0 & 0 \\ 0 & d_2 & a_3 & c_3 & \cdots & 0 & 0 \\ \vdots & & & & & & \vdots \\ 0 & 0 & 0 & 0 & \cdots & a_{N-1} & c_{N-1} \\ 0 & 0 & 0 & 0 & \cdots & d_{N-1} & a_N \end{pmatrix}$$

前節で解説した  $LU$  分解法を用いてこの行列を  $LU$  分解すると,  $L, U$  はそれぞれ以下ようになる.

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ l_{2,1} & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & l_{3,2} & 1 & 0 & \cdots & 0 & 0 \\ \vdots & & & & & & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & 0 & 0 & \cdots & l_{N,N-1} & 1 \end{pmatrix}$$

$$U = \begin{pmatrix} u_{1,1} & u_{1,2} & 0 & 0 & \cdots & 0 & 0 \\ 0 & u_{2,2} & u_{2,3} & 0 & \cdots & 0 & 0 \\ 0 & 0 & u_{3,3} & u_{3,4} & \cdots & 0 & 0 \\ \vdots & & & & & & \vdots \\ 0 & 0 & 0 & 0 & \cdots & u_{N-1,N-1} & u_{N-1,N} \\ 0 & 0 & 0 & 0 & \cdots & 0 & u_{N,N} \end{pmatrix}$$

このように,  $LU$  分解した後も 0 の要素が非常に多くなる. よって,  $LU$  分解するとき, 最終的に 0 ではなくなるインデックスの要素の計算のみ行えばいいことになる.

あらかじめ  $L, U$  を用意しておけば, 右辺定数項  $b$  が変化しても簡単な代入操作だけで連立方程式の解が求まる. 以下に  $L, U$  を用いた求解法を示す.

始めに  $Ly = b, Ux = y$  とおく. 上式は前進代入で解く.

$$y_1 = b_1$$

$$y_i = b_i - l_{i,i-1}y_{i-1} \quad (i = 2, \dots, N)$$



下式は後退代入を行う.

$$\begin{aligned} x_N &= \frac{y_N}{u_{N,N}} \\ x_i &= \frac{y_i - u_{i,i+1}x_{i+1}}{u_{i,i}} \quad (i = N-1, \dots, 1) \end{aligned}$$

空間次元が 1 の問題については, 行列が対角に近いところ以外は 0 要素のみになり今までの議論で十分である. しかし高次元の場合はそうはいかない. これは後ほど論ずることにする.

## 4 有限要素法

本章では, 有限要素法を扱う. 対象とする方程式はつぎのポアソン方程式とする.

$$\begin{cases} \Delta u = f(x) & \text{in } \Omega \\ u|_{\partial\Omega} = g(x) & \text{on } \partial\Omega \end{cases} \quad (g \in C^0(\partial\Omega)) \quad (4.1)$$

ここで,  $\Delta u = \sum_{i=1}^n \frac{\partial^2 u}{\partial x_i^2}$ ,  $f(x) \in L^2(\Omega)$  とし,  $\Omega \subset R^n$  は単連結な開領域, 境界  $\partial\Omega$  はなめらかとする. (4.1) の  $C^2$ -級の解を (もし存在するならば) 強い解と呼ぶことにする. これに対して, 次の弱い方程式とその解 (弱解) が定義される.

**Definition 4.1.** (弱いポアソン方程式)

$$-\int_{\Omega} \nabla u \nabla \zeta dx = \int_{\Omega} f(x) \zeta dx \quad \forall \zeta \in C_0^\infty(\Omega)$$

をみたく  $u \in W^{1,2}(\Omega)$  が存在するとき  $u$  を  $\Delta u = f(x)$  の弱解とよぶ.

注意 :  $W^{1,2}(\Omega) = \{u \in L^2(\Omega); u_{x_i} \in L^2(\Omega)\}$

**Proposition 4.1.** 強解であるならば弱解である.

強い方程式の両辺にテスト関数  $\zeta$  を掛けて部分積分すると,

$$-\int_{\Omega} \nabla u \nabla \zeta dx = \int_{\Omega} f(x) \zeta dx$$

となる.

## 4.1 ソボレフ空間

変分問題の数学的取り扱いのために、ソボレフ空間  $W^{1,2}$  を導入する (注 ;  $W^{1,2}$  は通常の内積によりヒルベルト空間となる). この空間は 1 階 (弱) 微分までが  $L^2$  に属するという関数空間である. 最初に弱微分の定義をしよう.

**Definition 4.2.** 弱微分 ( 超関数 (*distribution*) の意味の微分 )  
 $u \in L^1_{\text{loc}}(\Omega)$  に対して

$$-\int_{\Omega} v \varphi dx = \int_{\Omega} u \frac{\partial \varphi}{\partial x_i} dx \quad \text{for all } \varphi \in C_0^\infty(\Omega)$$

を満たす  $v \in L^1_{\text{loc}}(\Omega)$  が存在するとき,  $v$  を  $u$  の  $x_i$  方向の弱微分とよび,  $v = \frac{\partial u}{\partial x_i}$  と書く. もしも  $u \in C^1(\Omega)$  であるならば, 上式は部分積分の公式であり,  $v$  すなわち  $\frac{\partial u}{\partial x_i}$  は従来の微分と一致していることが分かる.

関数空間  $L^2(\Omega)$  は次の性質をもつ.

- (1)  $L^2(\Omega) := \{f : \text{可測関数 } \int_{\Omega} |f|^2 dx < +\infty\}$
- (2)  $L^2(\Omega)$  の内積は  $(u, v) = \int_{\Omega} uv dx$  で定義される.
- (3)  $L^2$  はノルム  $\|u\|_{L^2} := \sqrt{(u, u)}$  について完備である.

また, 関数空間  $W^{1,2}(\Omega)$  は次の性質をもつ.

- (1)  $W^{1,2}(\Omega) = \{f \in L^2(\Omega) : f_{x_i} \in L^2(\Omega) (i = 1, \dots, n)\}$   
ここで  $f_{x_i}$  は  $f$  の  $x_i$  方向の弱微分である.
- (2)  $W^{1,2}$  は次の内積とノルムをもつ.

$$\langle f, g \rangle = \int_{\Omega} f g dx + \sum_{i=1}^n \int_{\Omega} f_{x_i} g_{x_i} dx$$
$$\|f\|_{W^{1,2}} = \sqrt{\langle f, f \rangle}$$

- (3)  $W^{1,2}$  は  $\|\cdot\|_{W^{1,2}}$  について完備

最後の完備性について ( $L^2$  の完備性をみとめたうえで) 簡単に示しておこう.

**Theorem 4.1.**  $W^{1,2}$  は完備である.

[proof.]  $L^2(\Omega)$  の完備性はみとめた上でこのことを示しておく. まず,  $\|u\|_{W^{1,2}} \approx \|u\|_{L^2} + \|u_x\|_{L^2}$  である.  $W^{1,2}$  の中から Cauchy 列を選ぶ. すなわち,  $\|u_m - u_n\|_{W^{1,2}} \rightarrow 0$  ( $m, n \rightarrow \infty$ ) である. この  $\{u_n\}$  に対して  $\|u_m - u_n\|_{L^2} + \|u_{m,x} - u_{n,x}\|_{L^2} \rightarrow 0$  となる. すると,  $L^2$  の完備性より  $u_n \rightarrow \exists u \in L^2, u_{x,n} \rightarrow \exists v \in L^2$  であり,  $u_n$  は弱微分可能であるから,

$$-\int_{\Omega} \frac{\partial u_n}{\partial x_i} \varphi dx = \int_{\Omega} u_n \frac{\partial \varphi}{\partial x_i} dx$$

がすべての  $n$  に対して成り立つ. そして, 極限をとると

$$-\int_{\Omega} v_i \varphi dx = \int_{\Omega} u \frac{\partial \varphi}{\partial x_i} dx,$$

すなわち,  $v_i = \frac{\partial u}{\partial x_i}$  がいえる. (証明終わり)

$L^2$  や  $W^{1,2}$  は, それぞれの内積により Hilbert space となる. Hilbert space の大切な性質を示しておこう.

**Proposition 4.2.**  $H$  を Hilbert space とすると,  $H$  の有界列  $\{u_n\}$  から弱収束部分列を選ぶことができる. すなわち,

$$\exists \{u_{n_j}\} \subset \{u_n\}, \quad \exists u \text{ s.t. } \langle u_{n_j}, \varphi \rangle \rightarrow \langle u, \varphi \rangle \quad (j \rightarrow \infty)$$

である.

注意: 弱収束は次で定義される.

**Definition 4.3.**  $H$  の点列  $f_n$  が  $f$  に弱収束する ( $f_n \rightarrow f$  (弱) in  $H$ ) とは

$$\lim_{n \rightarrow \infty} \langle f_n, \varphi \rangle = \langle f, \varphi \rangle \quad \forall \varphi \in H$$

が成り立つことである.

上記の証明をしよう.  $X$ : Banach space とし,  $X^*$  を  $X$  上の有界線形汎関数全体とする.  $f$  が  $X$  から  $R$  への有界線形汎関数であるとは,

$$\begin{aligned} \exists M > 0, \text{ s.t. } |f(u)| &\leq M \|u\|_X \quad u \in X \text{ (有界性)} \\ f(\alpha u + \beta v) &= \alpha f(u) + \beta f(v) \quad \alpha, \beta \in R, u, v \in X \text{ (線形性)} \end{aligned}$$

をみたすことである. このもとで,

**Proposition 4.3.** 有界線形汎関数は連続

$$u_n \rightarrow u \text{ in } X \Rightarrow \lim_{n \rightarrow \infty} f(u_n) = f(u)$$

[proof . ]

$$|f(u_n) - f(u)| = |f(u_n - u)| \leq M \|u_n - u\|_X \longrightarrow 0 \quad (n \rightarrow \infty)$$

**Example .**  $X$  と  $X^*$  の例を述べよう .  $X = L^p(\Omega)$  とすると , すなわち  $u \in L^p(\Omega)$  ,  $\int_{\Omega} |u|^p dx < +\infty$  とすると ,

$$(L^p(\Omega))^* = L^q(\Omega)$$

である . ただし  $\frac{1}{p} + \frac{1}{q} = 1$  .

**Definition .**  $X^*$  をノルム空間とし ,  $f \in X^*$  のとき  $X^*$  のノルムは次で定義される .

$$\|f\|_{X^*} := \sup_{u \in X, \|u\|=1} |f(u)|$$

**Proposition 4.4.**  $X$  を Banach 空間とすると ,  $X^{**} \supset X$

[proof . ]  $\forall u \in X, \forall f \in X^*$  に対して ,

$$u(f) := f(u)$$

とすれば ,  $f$  にとって  $u$  は汎関数である .

$$\begin{aligned} u(\alpha f + \beta g) &= \alpha u(f) + \beta u(g) \\ |u(f)| &\leq \|f\|_* \|u\| \end{aligned}$$

であるから  $u$  は  $f$  にとって有界線形汎関数である . 従って  $X^{**}$  は  $X$  を含む . (証明終わり) .

特に ,  $X^{**} = X$  のとき , 回帰的 Banach 空間と呼ぶ .

次に  $H$  を Hilbert 空間とする . 簡単のため  $H$  が可分であるとする . また  $\{\varphi_j\}$  を

$$\{\varphi_j\}_{j=1}^{\infty} \subset H, \overline{\{\varphi_j\}_{j=1}^{\infty}} = H$$

となるような dense な可算個の集合とする .

**Theorem 4.2.**  $H^*$  から可算個の  $\{u_n\}$  を選び ,  $\|u_n\|_{H^*} \leq M$  であるとする ,

$$\exists \{u_{n_j}\} \subset \{u_n\} \text{ s.t. } * \text{ 弱収束する .}$$

すなわち ,  $\forall \varphi \in H$  について

$$u_{n_j}(\varphi) \rightarrow u(\varphi) .$$

(証明の概略) さて,  $a_n = \langle u_n, \varphi_1 \rangle$  とすると,  $a_n$  が有界であるから収束部分列を持つ. この部分列を  $u_n^1$  と表記する ( $\langle u_n^1, \varphi_1 \rangle$  は収束列である.) 次に  $\varphi_2$  に対して,  $\langle u_n^1, \varphi_2 \rangle$  は有界である. よって, 部分列  $\{u_n^2\} \subset \{u_n^1\}$  を選ぶと  $\langle u_n^2, \varphi_2 \rangle$  も収束列とすることができる. 以下同様にして  $\varphi_k$  に対して  $\{u_n^k\}$  を収束列になるように選ぶことができる. このように決めた列に対して, 新たに  $\{u_k^k\}$  を考える. すると, 対角線論法により,  $\forall j \in N$  について  $\langle u_k^k, \varphi_j \rangle$  は収束列となる.  $\{\varphi_j\}$  が dense であることなどから少しがんばると  $\forall \varphi \in H$  に対して  $\langle u_k^k, \varphi \rangle$  は収束列となる.

ここで,  $\forall \varphi \in H$  について  $u(\varphi) := \lim_{k \rightarrow \infty} \langle u_k^k, \varphi \rangle$  と定義すると, これは  $H$  上の汎関数となる. (証明終わり)

### Theorem 4.3. Riesz の表現定理

$H$  上の有界線形汎関数  $f$  に対して,  $f(u) = \langle v, u \rangle \forall u \in H$  となる  $v \in H$  が存在する.

[proof.]

$\ker f := \{u \text{ in } H ; f(u) = 0\}$  とすると, これは  $H$  の閉部分空間となる. よって  $H = \ker f^\perp \oplus \ker f$  一意にと分解できる.

次に  $\ker f^\perp$  は高々 1 次元であることを示す. 今, 2 次元以上であるとする.  $u_1, u_2 \in \ker f^\perp$  と, 1 次独立に選ぶことができる. すると

$$f(u_1) \neq 0, f(u_2) \neq 0$$

であるから, 0 でない実数  $\alpha, \beta$  があって  $\alpha f(u_1) = -\beta f(u_2)$  とできるが, 線形性より

$$f(\alpha u_1 + \beta u_2) = 0$$

これは矛盾. そこで kernel の直行補空間の元を  $u^\perp$ , ( $\|u^\perp\| = 1$ ) とすると,  $H$  の元  $u$  は  $u = \alpha u^\perp + u^{\ker}$ , と一意的に分解できる.  $f(u) = f(\alpha u^\perp + u^{\ker}) = \alpha f(u^\perp)$  であるから,  $\langle v, u \rangle = \alpha f(u^\perp)$  となるように  $v$  を決めればよい. すなわち

$$v = \frac{f(u^\perp)u^\perp}{\langle u^\perp, u^\perp \rangle}$$

とすればよい.

## 4.2 一次元の有限要素法

まず始めは, (4.1) の 1 次元の一番単純な場合で考えよう. 例えば,  $n = 1$ ,  $\Omega = (0, 1)$ , 境界条件:  $u(0) = a, u(1) = b$  外力項  $f(x) \equiv 0$  とおくと, 方程式は次のようになる.

Problem 4.5.

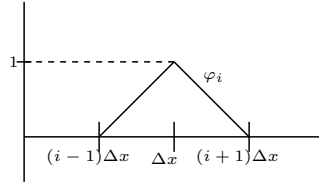
$$\begin{cases} \frac{d^2 u}{dx^2} = 0 & \text{in } \Omega \\ u(0) = a, u(1) = b \end{cases} \quad (4.2)$$

有限要素法は，後述の弱解からスタートする．これは，(3.2) の両辺に滑らかでコンパクトサポートを持つ関数を掛けて，部分積分して得られる．書き下すと，

$$-\int_0^1 u' \zeta' dx = 0 \quad (\forall \zeta \in C_0^\infty(0, 1))$$

となる．これをみたす解のことを，(3.2) の弱解と呼び，関係式を「弱い方程式」と呼ぶ．この方程式の近似解のある基底関数  $\{\varphi_i\}_i^N$  で表される関数の中で求める． $u = \sum_{i=0}^N a_i \varphi_i$  ( $a_i \in R$ ) とおくことにする．ここで， $(0, 1)$  を  $N$  等分して， $\frac{|\Omega|}{N} = \Delta x$  とおく．基底関数  $\varphi_i$  を次のように選択する．

$$\varphi_i = \begin{cases} 0 & x < (i-1)\Delta x \\ \frac{1}{\Delta x}x - (i-1) & (i-1)\Delta x \leq x \leq i\Delta x \\ -\frac{1}{\Delta x}x + (i+1) & i\Delta x \leq x \leq (i+1)\Delta x \\ 0 & (i+1)\Delta x < x \end{cases}$$



いま， $\zeta = \varphi_j$  ( $j = 1, \dots, N-1$ ) とおくと，

$$\begin{aligned} -\int_0^1 \left( \sum_{i=0}^N a_i \varphi_i \right)' \varphi_j' dx &= 0 \quad (j = 1, \dots, N-1) \\ -\sum_{i=0}^N a_i \int_0^1 \varphi_i' \varphi_j' dx &= 0 \quad (j = 1, \dots, N-1) \end{aligned}$$

となる．

ここで， $\varphi_i' \varphi_j'$  の積分は，次のようになり

$$\int_0^1 \varphi_i' \varphi_j' dx = \begin{cases} -\frac{1}{\Delta x} & (i = j+1, j-1) \\ \frac{2}{\Delta x} & (i = j) \\ 0 & (i \geq j+2, i \leq j-2) \end{cases}$$

係数,  $a_i$  は, 次の式をみます.

$$\sum_{i=1}^{N-1} a_i \int_0^1 \varphi'_i \varphi'_j dx = \frac{1}{\Delta x} (2a_j - a_{j+1} - a_{j-1}) \quad (j = 1, \dots, N-1)$$

$[c_{ij}] = \int_0^1 \varphi'_i \varphi'_j dx$  とすると  $a_i$  は  $n-1$  次の行列  $[c_{ij}]$  を用いて表される, 次の方程式をみます.

$$\frac{1}{\Delta x} \begin{bmatrix} 2 & -1 & 0 & \cdots & \cdots & \cdots & 0 \\ -1 & 2 & -1 & 0 & \cdots & \cdots & 0 \\ 0 & -1 & 2 & -1 & 0 & \cdots & 0 \\ \vdots & & & & & & \vdots \\ \vdots & & & & & & \vdots \\ 0 & \cdots & \cdots & 0 & -1 & 2 & -1 \\ 0 & \cdots & \cdots & 0 & 0 & -1 & 2 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ \vdots \\ a_{n-2} \\ a_{n-1} \end{bmatrix} = \begin{bmatrix} \frac{1}{\Delta x} a_0 \\ 0 \\ 0 \\ \vdots \\ \vdots \\ 0 \\ \frac{1}{\Delta x} a_n \end{bmatrix}$$

### 4.3 変分問題とラプラス方程式

Problem 4.6. 変分問題

$$I(v) = \int_{\Omega} |\nabla v|^2 dx$$

$I$  を最小化する関数を 許容関数集合  $\mathcal{K} := \{v \in W^{1,2}(\Omega); v|_{\partial\Omega} = g\}$  内で求めよ.

もし  $\inf_{v \in \mathcal{K}} I(v) = I(u)$  となる  $u \in \mathcal{K}$  が存在するならば,  $u$  は弱解、すなわち

$$-\int_{\Omega} \nabla u \nabla \zeta dx = 0 \quad \forall \zeta \in C_0^\infty(\Omega) \quad (4.3)$$

をみます.

このことを示そう.  $u$  が最小化関数であるとき,  $u + \varepsilon \zeta$  ( $\forall \zeta \in C_0^\infty(\Omega), \forall \varepsilon \in \mathbb{R}$ ) という関数が  $\mathcal{K}$  には入り

$$I(u + \varepsilon \zeta) = \int_{\Omega} |\nabla u|^2 dx + 2\varepsilon \int_{\Omega} \nabla u \nabla \zeta dx + \varepsilon^2 \int_{\Omega} |\nabla \zeta|^2 dx$$

となる. ここで  $\zeta$  は任意に選んで固定した. よって, 上式は  $\varepsilon$  についての 2 次関数と思ってよい. この 2 次関数は  $\varepsilon = 0$  の時に最小であるべきことから次式を得る:

$$\frac{d}{d\varepsilon} I(u + \varepsilon \zeta)|_{\varepsilon=0} = 0$$

これより直ちに,

$$2 \int_{\Omega} \nabla u \nabla \zeta dx = 0$$

が成り立つ。

上式を，第 1 変分方程式 (the first variation) とよぶ。  
定理にまとめると；

**Theorem 4.4.** 最小化関数ならば弱解である。

一方、この場合逆も成り立つ。そうすると次のことがいえる：

**Proposition 4.7.**  $u$  が  $-\int_{\Omega} \nabla u \nabla \zeta dx = 0 \quad \forall \zeta \in C_0^\infty(\Omega)$  を満たすことと、 $\int_{\Omega} |\nabla u|^2 dx$  の最小化関数であることは同値である。

[proof.] 弱解ならば最小化関数であることを示せばよい。

まず、 $\int_{\Omega} \nabla u \nabla \zeta dx = 0$  が  $\forall \zeta \in H_0^1(\Omega)$  に対しても成り立つことに注意する。ただし、 $H_0^1(\Omega) = W_0^{1,2}(\Omega) = \{v \in W^{1,2}(\Omega), v|_{\partial\Omega} = 0\}$  である。 $\mathcal{K}$  中の任意の関数  $v$  を選ぶと  $v - u \in H_0^1(\Omega)$  なので、 $v = u + (v - u) = u + \zeta, \zeta \in H_0^1(\Omega)$  と書ける。

$$I(v) = I(u + \zeta) - I(u) = \int_{\Omega} |\nabla \zeta|^2 dx$$

であるから、 $\zeta$  を恒等的に 0 でない関数とすると  $I(v) > I(u)$  となる。よって、 $u$  は  $I$  の minimizer である。

ここで、

$$2 \int_{\Omega} |\nabla \zeta|^2 dx = \frac{d^2}{d\varepsilon^2} I(u + \varepsilon \zeta)$$

のことを，第 2 変分とよぶ。

次に離散化した変分問題を考えてみよう。 $h := 1/N$  とおくと、

$$I(v) = \int_0^1 \left( \frac{dv}{dx} \right)^2 dx \cong \sum_{i=0}^{N-1} \left( \frac{a_{i+1} - a_i}{h} \right)^2 h = \frac{1}{h} \sum_{i=0}^{N-1} (a_{i+1} - a_i)^2$$

だから、 $\frac{\partial}{\partial a_i} I = 0 \quad (i = 1, \dots, N-1)$  は

$$2a_i - a_{i+1} - a_{i-1} = 0 \quad (a_0 = a, a_N = b)$$

となる。これは、(3.2) の数値解と同様である。

#### 4.4 Minimizer の存在

**Theorem 4.5.**  $\Omega$  が有界。 $g$  (境界条件をあたえる関数) が内部まで定義されていて  $I(g) < +\infty$  とする。このとき、 $I$  の *Minimizer* が  $\mathcal{K}$  で存在する。



$v \in \mathcal{K}$  とすると,  $v - g \in W_0^{1,2}(\Omega)$  (境界で0) である. 一般に  $f \in W_0^{1,2}(\Omega)$  に対して,  $\|f_x\|_{L^2} \geq C(\|f\|_{L^2} + \|f_x\|_{L^2})$  が成り立つような  $C > 0$  が存在する. 特に,  $\|f_x\|_{L^2} \geq C\|f\|_{L^2}$  をポアンカレの不等式とよぶ. 実際  $\Omega = (0, l)$  のとき  $f(x) = \int_0^x f_x(l) dl$  より,  $|f(x)| \leq \int_0^x |f_x(l)| dl$  だから

$$\int_{\Omega} |f(x)| dx \leq \int_{\Omega} \int_0^l |f_x| dl dx = \int_0^l dl \int_{\Omega} |f_x| dx \leq \text{diam}(\Omega) \cdot \int_{\Omega} |f_x| dx$$

同様にして

$$\begin{aligned} |f(x)|^2 &\leq \left( \int_0^x |f_x(l)| dl \right)^2 \\ &= \left[ \left( \int_0^x |f_x(l)|^2 dl \right)^{1/2} \left( \int_0^x 1^2 dl \right)^{1/2} \right]^2 \\ &= \text{diam}(\Omega) \int_0^x |f_x(l)|^2 dl \end{aligned}$$

が成り立つ. 両辺を  $\Omega$  上で積分して,

$$\int_{\Omega} |f(x)|^2 dx \leq \text{diam}^2(\Omega) \int_{\Omega} |f_x(x)|^2 dx$$

である. 次に  $I(v) \leq M$  ならば  $v - g \in W_0^{1,2}(\Omega)$  が有界である.  $\mathcal{K}_M := \{v \in \mathcal{K} : \int_{\Omega} |\nabla v|^2 dx < M\}$  とすると,  $\mathcal{K}_M \neq \emptyset$  かつ有界なので,  $\{u_j\} \subset \mathcal{K}_M$  で次を満たすものがとれる.

$$\begin{aligned} \lim_{j \rightarrow \infty} I(u_j) &= \inf_{v \in \mathcal{K}} I(v), \\ I(u_j) &\geq I(u_{j+1}) \quad \forall j \in N. \end{aligned}$$

これを  $I$  の最小化関数列 (minimizing sequence) と呼ぶ. このとき,  $\{u_j\}$  は  $W^{1,2}(\Omega)$  で有界であるから, 一般論として次のことが言える,

**Proposition 4.8.**  $\{u_j\}$  には適当な部分列  $\{u_{j_k}\}$  とある  $u \in W^{1,2}(\Omega)$  が存在して  $u_{j_k} \rightarrow u$ . (弱収束 in  $W^{1,2}(\Omega)$ )

**Proposition 4.9.**  $I$  は弱収束列に対して下半連続である. すなわち

$$\liminf_{j \rightarrow \infty} I(u_j) \geq I(u).$$

[proof.]  $\{u_j\}$  を  $W^{1,2}$  の弱収束列とする. 一般に

$$0 \leq \int_{\Omega} |\nabla(u - u_j)|^2 dx \leq \int_{\Omega} |\nabla u|^2 dx - 2 \int_{\Omega} \nabla u \nabla u_j dx + \int_{\Omega} |\nabla u_j|^2 dx$$

が成り立つ．この両辺に  $\int_{\Omega} |\nabla u|^2 dx$  を加えて

$$\int_{\Omega} |\nabla u|^2 dx \leq 2 \int_{\Omega} |\nabla u \nabla (u - u_j)|^2 dx + \int_{\Omega} |\nabla u_j|^2 dx$$

この両辺の  $\liminf_{j \rightarrow \infty}$  をとると右辺の第 1 項が 0 に収束することより

$$\int_{\Omega} |\nabla u|^2 dx \leq \liminf_{j \rightarrow \infty} \int_{\Omega} |\nabla u|^2 dx$$

(証明終わり) .

Proposition 3.5. と Proposition 3.6. を合わせると, minimizer の存在が  
いえる.

## 5 二次元楕円型方程式の離散化

### 5.1 ラプラス方程式の弱形式

ラプラス方程式の有限要素法による離散化について考える．ラプラス方程式

$$\begin{aligned} \Delta u &= 0 \quad \text{on } \Omega \subset \mathbf{R}^n \\ u &= g \quad \text{on } \partial\Omega \end{aligned}$$

を弱形式表示すると

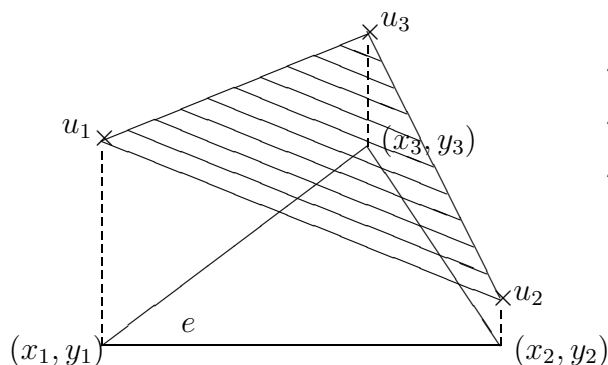
$$\begin{aligned} - \int_{\Omega} \nabla u \cdot \nabla \zeta dx &= 0 \quad \forall \zeta \in C_0^\infty(\Omega) \\ u &= g \quad \text{on } \partial\Omega \end{aligned}$$

となる．この式を出発点として離散化を行う．今回は  $\Omega \subset \mathbf{R}^2$  で話を進める．

### 5.2 有限要素分割

領域  $\Omega \subset \mathbf{R}^2$  を三角形の小領域 (有限要素または 要素という．また, 各三角形の頂点を 節点またはノードという) に分割する．このときに, 分割には不足や重なりがなく, 隣り合う有限要素は辺どうしがぴったりと合うようにする．

### 5.3 面積座標



$u_1 : (x_1, y_1)$  における  $u$  の値

$u_2 : (x_2, y_2)$  における  $u$  の値

$u_3 : (x_3, y_3)$  における  $u$  の値

ある要素  $e$  上で，関数  $u(x, y)$  を区分的 1 次関数近似したものを  $\hat{u}(x, y)$  とすると，

$$\hat{u}(x, y) = \alpha x + \beta y + \gamma \quad (5.1)$$

と書くことができる．ここで， $\alpha, \beta, \gamma$  は定数である．

そして，1つの要素  $e$  にある3つの節点の座標をそれぞれ  $(x_i, y_i)$  ( $i = 1, 2, 3$ ) とし，各節点上での関数  $u$  の値を  $u_i$  ( $i = 1, 2, 3$ ) とすると，

$$\begin{cases} u_1 = \alpha x_1 + \beta y_1 + \gamma \\ u_2 = \alpha x_2 + \beta y_2 + \gamma \\ u_3 = \alpha x_3 + \beta y_3 + \gamma \end{cases}$$

であるから，

$$\begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \begin{pmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix}$$

となる．ここで，

$$D := \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix}$$

とおくと，各要素の3節点は同一直線上にないので  $D \neq 0$  である．したがって，Cramer の公式により

$$\alpha = \frac{\begin{vmatrix} u_1 & y_1 & 1 \\ u_2 & y_2 & 1 \\ u_3 & y_3 & 1 \end{vmatrix}}{D}, \quad \beta = \frac{\begin{vmatrix} x_1 & u_1 & 1 \\ x_2 & u_2 & 1 \\ x_3 & u_3 & 1 \end{vmatrix}}{D}, \quad \gamma = \frac{\begin{vmatrix} x_1 & y_1 & u_1 \\ x_2 & y_2 & u_2 \\ x_3 & y_3 & u_3 \end{vmatrix}}{D}$$

である．そして， $(i, j, k) = (1, 2, 3), (2, 3, 1), (3, 1, 2)$  に対して，

$$a_i := \frac{y_j - y_k}{D}, \quad b_i := \frac{x_k - x_j}{D}, \quad c_i := \frac{x_j y_k - x_k y_j}{D}$$

と定義すると，

$$\begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix} = \begin{pmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix}$$

となり，

$$\alpha = \sum_{i=1}^3 a_i u_i, \quad \beta = \sum_{i=1}^3 b_i u_i, \quad \gamma = \sum_{i=1}^3 c_i u_i, \quad (5.2)$$

である．したがって，(5.1), (5.2) より

$$\hat{u}(x, y) = \sum_{i=1}^3 (a_i x + b_i y + c_i) u_i$$

となり，

$$\lambda_i(x, y) := a_i x + b_i y + c_i \quad (5.3)$$

と置けば，

$$\hat{u}(x, y) = \sum_{i=1}^3 \lambda_i u_i \quad (5.4)$$

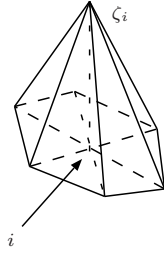
と表現できる．このとき， $(\lambda_1, \lambda_2, \lambda_3)$  を面積座標と呼ぶ．これを用いると  $\hat{u}$  の  $x, y$  での微分がノードの座標とそこでの関数値のみで表示される利点がある．すなわち

$$\begin{aligned} \frac{\partial \hat{u}}{\partial x} &= \sum_{i=1}^3 \frac{\partial \lambda_i}{\partial x} u_i = \sum_{i=1}^3 a_i u_i \\ \frac{\partial \hat{u}}{\partial y} &= \sum_{i=1}^3 \frac{\partial \lambda_i}{\partial y} u_i = \sum_{i=1}^3 b_i u_i \end{aligned}$$

などと計算される．

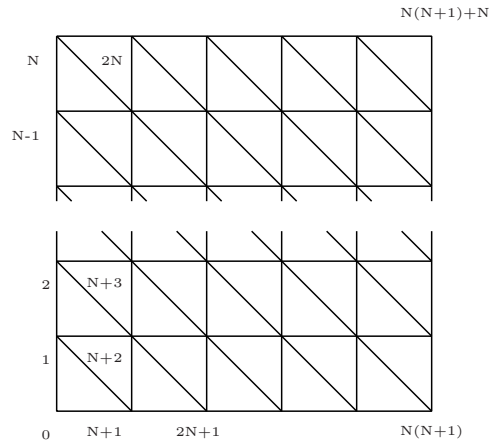
## 5.4 基底関数

基底関数は各ノード  $i$  で定める．基底関数  $\zeta_i$  は，ノード  $i$  上で値が 1 となり，近接した要素上以外の領域では常に値が 0 となる関数を指す．グラフで表すとピラミッドのような形をしている．



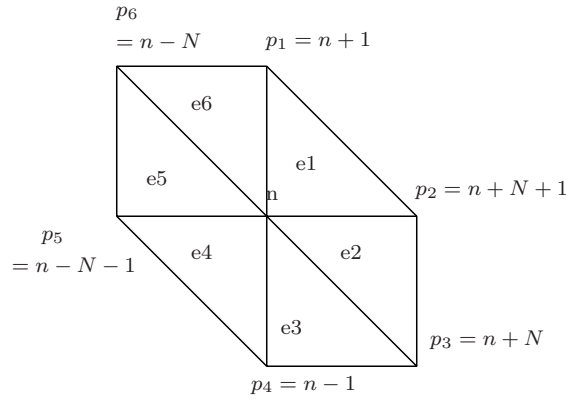
## 5.5 近似関数を用いた積分計算

正方形の領域  $\Omega$  を  $x$  軸方向に  $N$  等分,  $y$  軸方向に  $N$  等分し一様格子に分割する. 分割の幅は  $\Delta x$  とする. 各ノードには以下のように番号を付ける.



座標が  $(i \times \Delta x, j \times \Delta x)$  のノードの番号は  $i(N+1) + j$  となる.  
 $(i = 0, \dots, N, j = 0, \dots, N)$

ここで  $n = i(N+1) + j$  とする.  $n$  番目のノードが属する三角形要素で  $n$  以外のノードの番号を  $p_1, p_2, \dots, p_6$  に置き換え, 以下の様にそれぞれの三角形の領域に  $e_{n_1}, e_{n_2}, \dots, e_{n_6}$  と名前を付ける.



今、領域の  $n$  番目のノード上で値が 1 となる基底関数  $\zeta_n$  を考える.  $\zeta_n$  が常に 0 とはならない要素上で, 以下の式

$$-\int \int_{\Omega} \nabla u \cdot \nabla \zeta_n dx dy = 0 \quad (5.5)$$

を離散化してみる.  $\zeta_n \equiv 0$  となる要素上では, 積分値は 0 となるため, 考える必要は無い. 要素  $e_{n_k}$  上で  $u$  を一次関数で近似すると, (5.4) より,

$$u(x, y) = \hat{u}^{(e_{n_k})}(x, y) = \sum_{m=n, p_{\alpha}, p_{\beta}} \lambda_m^{(e_{n_k})} u_m \quad \text{in } (x, y) \in e_{n_j} \quad (5.6)$$

と書くことができる. ただし,  $\lambda_m^{(e_{n_j})}$  は要素  $e_{n_j}$  上での式 (5.3) で表される面積座標  $\lambda_m$  であり,  $u_m$  はノード  $m$  上での  $u$  の値である.  $\zeta_n$  は基底関数の定義から,

$$\zeta_n(x, y) = \begin{cases} \lambda_n^{(e_{n_1})} & (x, y) \in e_{n_1} \\ \lambda_n^{(e_{n_2})} & (x, y) \in e_{n_2} \\ \lambda_n^{(e_{n_3})} & (x, y) \in e_{n_3} \\ \lambda_n^{(e_{n_4})} & (x, y) \in e_{n_4} \\ \lambda_n^{(e_{n_5})} & (x, y) \in e_{n_5} \\ \lambda_n^{(e_{n_6})} & (x, y) \in e_{n_6} \\ 0 & \text{otherwise} \end{cases} \quad (5.7)$$

となる. 式 (5.6), 式 (5.7) から式 (5.5) を計算すると,

$$\begin{aligned}
-\int \int_{\Omega} \nabla u \cdot \nabla \zeta dx dy &= -\sum_{k=1}^6 \int \int_{e_{n_k}} \left( \nabla \left( \sum_l \lambda_l^{(e_{n_k})} u_l \right) \cdot \nabla \lambda_n^{(e_{n_k})} \right) dx dy \\
&= -\sum_{k=1}^6 \left( \int \int_{e_{n_k}} \left( \sum_l \nabla \lambda_l^{(e_{n_k})} u_l \right) \cdot \nabla \lambda_n^{(e_{n_k})} dx dy \right) \\
&= -\sum_{k=1}^6 \left( \sum_l u_l \int \int_{e_{n_k}} \nabla \lambda_l^{(e_{n_k})} \cdot \nabla \lambda_n^{(e_{n_k})} dx dy \right) \\
&= -\frac{|e|}{(\Delta x)^2} (8u_n - 2u_{p_i} - 2u_{p_2} - 2u_{p_4} - 2u_{p_5}) \\
&= -(4u_n - u_{n+1} - u_{n+N+1} - u_{n-1} - u_{n-N-1}) \quad (5.8)
\end{aligned}$$

となる. ただし  $|e|$  は要素の面積  $(\frac{1}{2}(\Delta x)^2)$  である.

ディリクレ境界上のノードの数を  $M$  個とすると, 境界上以外のノードでは, 式 (5.8) を満たすので,  $N(N+1)+1-M$  個の代数方程式が作られる.

具体例として, 以下の二次元ラプラス方程式

$$\begin{aligned}
\Delta u &= 0 \quad \text{on } \Omega = \{(x, y) \mid -1 \leq x, y \leq 1\} \\
u &= 1 \quad \text{on } \partial\Omega
\end{aligned}$$

から代数方程式を導き出してみる. 今,  $N=4$  とする. 先に説明した通りに計算を行うと,

$$\begin{pmatrix} 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 4 & 0 & 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 4 & -1 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & -1 & 4 & -1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & -1 & 4 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 & 0 & 4 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 \end{pmatrix} \begin{pmatrix} u_6 \\ u_7 \\ u_8 \\ u_{11} \\ u_{12} \\ u_{13} \\ u_{16} \\ u_{17} \\ u_{18} \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \\ 2 \\ 1 \\ 0 \\ 1 \\ 2 \\ 1 \\ 2 \end{pmatrix}$$

という行列方程式が導出される. 上記の行列は, 要素のほとんどが 0 となる. このような行列を疎行列という.

## 6 CG法

### 6.1 逐次最小化法

以下では  $N \times N$  実正定値対称行列  $A = (a_{ij})$  に対して、線形方程式

$$Ax = b \quad (6.1)$$

を考察の対象とする。

$\mathbb{R}^N$  の内積を  $(x, y)$  と表す。  $A$  が正定値対称であるので、次の 2 次形式

$$f(x) = \frac{1}{2}(x, Ax) - (x, b) \quad (6.2)$$

の最小値を与える  $x$  が (6.1) の解である。

何らかの規則により  $f(x^{k+1}) < f(x^k)$  を満たすような  $\{x^k\}$  で、 $x^k \rightarrow x$  が期待できるものをみつけ、これをある種の反復により行う方法が逐次最小化法である。

通常 of 逐次最小化法の手順は、

- (i) 次の近似解の探索方向  $p$  を決定し、
- (ii) その探索方向で  $f(x)$  が最小となる点を決め、次の近似解とする

である。残差ベクトル  $r = b - Ax$  が収束条件を満たすまでこの手順を繰り返せば良い。探索方向の決定方法により様々な方法があり得、CG法もその一種である。探索方向が決まるとすれば、解補正の大きさ  $\alpha_k$  を用いて次の step での近似解  $x^{k+1}$  が

$$x^{k+1} = x^k + \alpha_k p^k \quad (6.3)$$

と定まる。

形式的に  $f(x^{k+1})$  を計算すると

$$\begin{aligned} f(x^{k+1}) &= f(x^k + \alpha_k p^k) \\ &= \frac{1}{2}(x^k + \alpha_k p^k, A(x^k + \alpha_k p^k)) - (x^k + \alpha_k p^k, b) \\ &= \frac{1}{2}(x^k, Ax^k) + \frac{1}{2}(x^k, \alpha_k A p^k) + \frac{1}{2}(\alpha_k p^k, Ax^k) + \frac{1}{2}(\alpha_k p^k, \alpha_k A p^k) \\ &\quad - (x^k, b) - (\alpha_k p^k, b) \end{aligned}$$

ここで  $A$  の対称性により、

$$\frac{1}{2}(x^k, \alpha_k A p^k) + \frac{1}{2}(\alpha_k p^k, Ax^k) = \alpha_k (p^k, Ax^k).$$



また,

$$\alpha_k(p^k, Ax^k) - \alpha(p^k, b) = \alpha_k(p^k, Ax^k - b) = \alpha_k(p^k, r^k)$$

を考慮すると

$$f(x^{k+1}) = f(x^k) - \alpha_k(p^k, r^k) + \frac{1}{2}\alpha_k^2(p^k, Ap^k)$$

となる. ここで,  $\alpha_k$  を  $f(x^{k+1})$  が最小となるように決める. すなわち

$$\frac{\partial}{\partial \alpha_k} f(x^{k+1}) = 0 \quad (6.4)$$

とする. (6.4) は

$$-(p^k, r^k) + \alpha_k(p^k, Ap^k) = 0 \quad (6.5)$$

だから,

$$\alpha_k = \frac{(p^k, r^k)}{(p^k, Ap^k)} \quad (6.6)$$

とえらばよい. このことをふまえ, 以下に逐次最小化法のアルゴリズムを示す.

(0)  $x^0$  is an initial data,  $r^0 = b - Ax^0$ ,  $k = 0$

(1) for  $k = 0, 1, \dots$ , until  $\|r^k\|/\|b\| < \exists \epsilon$  do:

(1-1) determine  $p^k$  systematically as retrieval direction,

(1-2)  $\alpha_k = \frac{(p^k, r^k)}{(p^k, Ap^k)}$  s.t.  $f(x^{k+1})$  be minimized.

(1-2)  $x^{k+1} = x^k + \alpha_k p^k$

(1-3)  $r^{k+1} = r^k - \alpha_k Ap^k$

(1-4)  $k++$

## 6.2 共役勾配法

共役勾配法は前節の逐次最小化法の一つである. 一般に, 逐次最小化法に従えば, 現在の位置  $x_k$  は次のように定められている.

$$x^k = x^0 + \sum_{0 \leq l < k} \alpha_l p^l \quad (6.7)$$

共役勾配法では  $p^k$  を以下の様に決定する.

$$p^k = r^k + \beta_{k-1}p^{k-1} \quad (6.8)$$

ここで  $r^k$  は残差ベクトルで

$$r^k = b - Ax^k \quad (6.9)$$

$\beta_{k-1}$  は  $(p^k, Ap^{k-1}) = 0$  を満足するように決定する. すなわち

$$\beta_{k-1} = -\frac{(p^{k-1}, Ar^k)}{(p^{k-1}, Ap^{k-1})} \quad (6.10)$$

である.

このように  $p^k$  を選ぶことで, 残差ベクトル  $r^k$  は直交条件

$$(r^i, r^j) = 0 \quad (i \neq j) \quad (6.11)$$

修正ベクトル  $p^k$  は A-共役条件

$$(p^i, Ap^j) = 0 \quad (i \neq j) \quad (6.12)$$

を満足する. 残差ベクトルの直交性により共役勾配法は高々  $N$  回の反復で収束することが期待される.

また,

$$\begin{aligned} (p^k, r^k) &= (r^k + \beta_{k-1}p^{k-1}, r^k) \\ &= (r^k, r^k) + \beta_{k-1}(p^{k-1}, r^k) \\ &= (r^k, r^k) + \beta_{k-1}(p^{k-1}, r^{k-1} - \alpha_{k-1}Ap^{k-1}) \\ &= (r^k, r^k) + \beta_{k-1}((p^{k-1}, r^{k-1}) - \alpha_{k-1}(p^{k-1}, Ap^{k-1})) \\ &= (r^k, r^k) + \beta_{k-1}((p^{k-1}, r^{k-1}) - (p^{k-1}, r^{k-1})) \\ &= (r^k, r^k) \end{aligned}$$

より,

$$\alpha_k = \frac{(p^k, r^k)}{(p^k, Ap^k)} = \frac{(r^k, r^k)}{(p^k, Ap^k)} \quad (6.13)$$

となる.

さらに  $\beta_k$  は (6.10) と直交条件 (6.11) を用いることで, 以下のように計算できる.

$$\begin{aligned} (p^k, Ar^{k+1}) &= (Ap^k, r^{k+1}) \\ &= \frac{1}{\alpha_i}(r^k - r^{k+1}, r^{k+1}) \\ &= -\frac{1}{\alpha_i}(r^{k+1}, r^{k+1}) \\ &= -\frac{(p^k, Ap^k)}{(r^k, r^k)}(r^{k+1}, r^{k+1}) \end{aligned}$$

であるから,

$$\begin{aligned}\beta_k &= -\frac{(p^k, Ar^{k+1})}{(p^k, Ap^k)} \\ &= \frac{(r^{k+1}, r^{k+1})}{(r^k, r^k)}\end{aligned}$$

となる.

以上により, 共役勾配法のアルゴリズムは次のようになる:

(0)  $x^0$  is an initial guess,  $r^0 = b - Ax^0$ ,  $\beta_{k-1} = 0$ ,  $k = 0$

(1) for  $k = 0, 1, \dots$ , until  $\|r^k\|/\|b\| < \exists \epsilon$  do:

$$(1-1) \quad p^k = r^k + \beta_{k-1}p^{k-1}$$

$$(1-2) \quad \alpha_k = \frac{(r^k, r^k)}{(p^k, Ap^k)}$$

$$(1-3) \quad x^{k+1} = x^k + \alpha_k p^k$$

$$(1-4) \quad r^{k+1} = r^k - \alpha_k Ap^k$$

$$(1-5) \quad \beta_k = \frac{(r^{k+1}, r^{k+1})}{(r^k, r^k)} \text{ s.t. } (p^k, Ap^{k+1}) = 0$$

$$(1-6) \quad k++$$

**Theorem 6.1.**

$$p^k = r^k + \beta_{k-1}p^{k-1} \quad \text{s.t.} \quad (p^{k-1}, Ap^k) = 0$$

のとき

$$\begin{cases} (r^i, r^j) = 0 & (i \neq j) \\ (p^i, Ap^j) = 0 & (i \neq j) \end{cases}$$

が成り立つ.

[proof .] 以下では式 (6.8) により, 直交条件 (6.11) と A-共役条件 (6.12) が成り立つことを帰納法で証明する.

(i) 式 (6.10) より

$$(p^1, Ap^0) = 0.$$

式 (6.13) より

$$(r^1, r^0) = (r^0, r^0) - \alpha_0(Ap^0, r^0) = 0$$

となり  $i, j \leq 1$  のとき (6.11), (6.12) は満足される.

(ii)  $i, j \leq m$  で式 (6.11), (6.12) が成り立っていると仮定する.

$p^{m+1} = r^{m+1} + \beta_m p^m$  として  $j \leq m$  について  $(r^{m+1}, r^j) = 0, (p^{m+1}, Ap^j) = 0$  を示せば良い.

$j < m$  のとき

$$\begin{aligned} (r^{m+1}, r^j) &= (r^m - \alpha_m Ap^m, r^j) \\ &= -\alpha_m (Ap^m, p^j) + \alpha_m \beta_{j-1} (Ap^m, p^{j-1}) = 0 \end{aligned}$$

また,  $j = m$  のとき

$$\begin{aligned} (r^{m+1}, r^m) &= (r^m, r^m) - \alpha_m (Ap^m, r^m) = (r^m, r^m) - \alpha_m (Ap^m, p^m) \\ &= 0 \end{aligned}$$

式 (6.10) より  $(p^{m+1}, Ap^m) = 0$  であるので,  $j < m$  について検証する.

$$\begin{aligned} (p^{m+1}, Ap^j) &= (r^{m+1} + \beta_m p^m, Ap^j) = (r^{m+1}, Ap^j) \\ &= \frac{1}{\alpha_j} (r^{m+1}, r^j - r^{j+1}) = 0 \end{aligned}$$

より式 (6.11), (6.12) は  $i, j \leq m + 1$  で成り立つ (証明終わり).

### 6.3 前処理付共役勾配法

共役勾配法の収束の速さは係数行列  $A$  の固有値分布と密接に関係している. 線形方程式 (6.1) の係数行列  $A$  の固有多項式が重根を持つ場合, 共役勾配法の反復回数が減少する. これは残差ベクトル  $r$  の動く空間の次元が重根の数だけ減るからである. これを示す.

まず行列  $A$  が実対称行列であることから,  $A$  の固有値は全て実数であり, 重複を許して  $N$  個存在する. それを  $\{\lambda_i\}_{1 \leq i \leq N}$  とおき, 対応する固有ベクトルを  $\{v_i\}_{1 \leq i \leq N}$  とおく. この固有ベクトルは全て直交するように取ることが出来る:

$$(v_i, v_j) = \delta_{i,j} \quad (1 \leq i, j \leq N)$$

$k$  回目の反復における近似解  $x^k$  に対応する残差ベクトル  $r^k = b - Ax^k$  をこの固有ベクトル  $\{v_i\}_{1 \leq i \leq N}$  を用いて表現すると

$$r^k = \sum_{1 \leq i \leq N} \nu_i^k v_i. \quad (6.14)$$

ただし

$$r^0 = \sum_{1 \leq i \leq N} \mu_i v_i, \quad v_i^0 = \mu_i \quad (6.15)$$

とする．次に CG 法の関係式

$$\begin{aligned} r^{k+1} &= r^k - \alpha_k A p^k \\ p^{k+1} &= r^{k+1} + \alpha_k p^k, \quad p^0 = r^0 \end{aligned}$$

を用いて  $r^1, p^1, r^2, p^2, \dots$  を順次計算すると，

$$\begin{aligned} r^1 &= (I - \alpha_0 A) r^0 \\ p^1 &= [(1 + \beta_0) I - \alpha_0 A] r^0 \\ r^2 &= [I - (\alpha_0 + \alpha_1 + \alpha_1 \beta_0) A + \alpha_0 \alpha_1 A^2] r^0 \\ p^2 &= [(1 + \beta_1 + \beta_0 \beta_1) I - (\alpha_0 + \alpha_1 + \alpha_1 \beta_0 + \alpha_0 \beta_1) A + \alpha_0 \alpha_1 A^2] r^0 \\ r^3 &= [I - (\alpha_0 + \alpha_1 + \alpha_2 + \alpha_1 \beta_0 + \alpha_2 \beta_2 + \alpha_2 \beta_0 \beta_1) A \\ &\quad + (\alpha_0 \alpha_1 + \alpha_0 \alpha_2 + \alpha_1 \alpha_2 + \alpha_1 \alpha_2 \beta_0 + \alpha_0 \alpha_2 \beta_1) A^2 - \alpha_0 \alpha_1 \alpha_2 A^3] r^0. \\ &\dots \end{aligned}$$

すなわち  $r^k$  は  $k$  次多項式  $\phi_k(\xi)$  を用いて

$$r^k = \phi_k(A) r^0 \quad (6.16)$$

と表すことが出来る．

ところで， $x^k = A^{-1}(b - r^k)$  の関係を用いると， $A$  が対称であることに注意して，

$$\begin{aligned} f(x) &= \frac{1}{2} (x, Ax) - (x, b) \\ &= \frac{1}{2} [(r, A^{-1}r) - (b, A^{-1}b)] \end{aligned} \quad (6.17)$$

であるから， $f(x)$  を最小にする CG 法は  $F(r) = (r, A^{-1}r)$  を最小にしている．従って，CG 法の  $k$  回目の反復では  $F(r^k)$  が最小になるように  $r^k$  が，また

$$r^k = \sum_{1 \leq i \leq N} v_i^k v_i \quad (6.18)$$

となるように  $\{v_i^k\}_{1 \leq i \leq N}$  が決定される．そして，(6.15) と (6.16)，さらに

$$A v_i = \lambda_i v_i \quad (6.19)$$

より,

$$r^k = \sum_{1 \leq i \leq N} \mu_i \phi_k(\lambda_i) v_i \quad (6.20)$$

であるから,

$$v_i^k = \mu_i \phi_k(\lambda_i) \quad (6.21)$$

である. ここで,  $A$  の固有値のうち最後の  $M$  個が重複しているとする,

$$\begin{aligned} \lambda_{N-M+1} &= \lambda_{N-M+2} = \cdots = \lambda_N \\ \phi_k(\lambda_{N-M+1}) &= \phi_k(\lambda_{N-M+2}) = \cdots = \phi_k(\lambda_N) \end{aligned}$$

だから,

$$\begin{aligned} r^k &= \sum_{1 \leq i \leq N} v_i^k v_i = \sum_{1 \leq i \leq N} \mu_i \phi_k(\lambda_i) v_i \\ &= \sum_{1 \leq i \leq N-M} v_i^k v_i + \phi_k(\lambda_N) \sum_{N-M < i \leq N} \mu_i v_i \end{aligned} \quad (6.22)$$

従って  $k$  回目の反復では  $F(r^k)$  が最小になるように  $r^k$  が, また

$$r^k = \sum_{1 \leq i \leq N-M} v_i^k v_i + \phi_k(\lambda_N) \sum_{N-M < i \leq N} \mu_i v_i \quad (6.23)$$

となるように  $\{v_i^k\}_{1 \leq i \leq N-M \text{ or } i=N}$  が決定される. よって, 残差ベクトルの自由度が  $M-1$  だけ落ちていることがわかる. この為,  $A$  の固有方程式が重根を持つと CG 法の収束が速くなるのである.

以上の議論は, 幾何学的に考えれば, 固有値が縮退している場合には  $f(x)$  の等値面が  $N$  次元同心球面に近くなるために, 共役勾配法の反復回数が減少すると理解できる. これを以下の計算で簡単に説明する.

$\forall x \in \mathbf{R}$  に対し,  $A$  の固有ベクトル  $\{v_i\}_{1 \leq i \leq N}$  を用いると

$$x = \sum_{1 \leq i \leq N} \eta_i v_i. \quad (6.24)$$

次に  $f(x)$  を計算すると,

$$\begin{aligned} f(x) &= \frac{1}{2}(x, Ax) - (x, b) \\ &= \frac{1}{2} \sum_{1 \leq i \leq N} (\eta_i)^2 \lambda_i - \sum_{1 \leq i \leq N} (b, v_i) \eta_i \end{aligned} \quad (6.25)$$

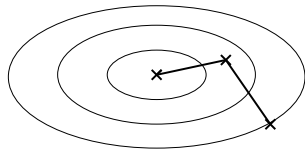
ここで、 $A$  の固有値のうち最後の  $M$  個が重複しているとすると、

$$\lambda_{N-M+1} = \lambda_{N-M+2} = \cdots = \lambda_N$$

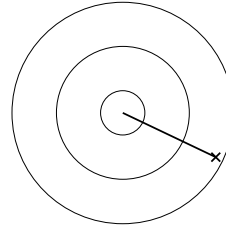
より、

$$\begin{aligned} f(x) = & \frac{1}{2} \sum_{1 \leq i \leq N-M} (\eta_i)^2 \lambda_i - \sum_{1 \leq i \leq N-M} (b, v_i) \eta_i \\ & + \frac{1}{2} \sum_{N-M < i \leq N} (\eta_i)^2 \lambda_N - \sum_{N-M < i \leq N} (b, v_i) \eta_i \end{aligned} \quad (6.26)$$

極端な場合、 $A = I$  であれば、 $A$  の固有方程式は  $N$  重根をもち、一回の反復で収束する。



固有値が縮退していない場合



固有値が縮退している場合

図 1:  $f(x)$  の等高線 (2次元の場合)

以上より、 $A$  と近い行列  $C \sim A$  を作り、その逆行列を (6.1) に用いて固有値の分布を密集させ、その新たな方程式を CG 法で求解すれば有利であると期待される。その  $C$  の選択の基準は、

- (i)  $C$  が  $A$  に良く近似されていること  
 $\Leftarrow A = C$  ならば反復無しで解が求まる
- (ii) (不完全) 分解し易いこと  
 $\Leftarrow$  前処理に時間がかかっているとは本末転倒
- (iii)  $C$  とその分解後の三角行列が、 $A$  と同程度に疎であること  
 $\Leftarrow$  メモリ消費量や演算回数で損をする訳にはいかない

である。

このような基準に基づいて適当な前処理行列  $C$  を選んだとし、この  $C$  が正定値対称であるとする、 $C$  は  $LL^T$  に Cholesky 分解できる。ここで  $L$  は下三角行列である。

$$C = LL^T \quad (6.27)$$

この  $C$  を用いて

$$\tilde{A} = L^{-1}AL^{-T}, \quad \tilde{x} = L^T x, \quad \tilde{b} = L^{-1}b$$

と定義すると, (6.1) は

$$\tilde{A}\tilde{x} = \tilde{b} \quad (6.28)$$

と同値である. 更に, 相似変換  $L^T \tilde{A} L^T = C^{-1}A (\sim I)$  より,  $\tilde{A}$  の固有値は  $C^{-1}A$  のそれと一致する.

このようにして得られた, 固有値分布が改善されたと期待される方程式 (6.28) に対して, CG 法を適用する. 特に,  $C$  を求める際に不完全 Cholesky 分解を用いるものは iCCG 法 (incomplete Cholesky decompositioned CG) と呼ばれる.

(6.28) に対して CG 法を適用し整理すると, 前処理付き CG 法のアルゴリズムは次のようになる.

(0)  $x^0$  is an initial guess,  $r^0 = b - Ax^0$ ,  $\beta_{k-1} = 0$ ,  $k = 0$

(1) for  $k = 0, 1, \dots$ , until  $\|r^k\|/\|b\| < \exists \epsilon$  do:

$$(1-1) \quad p^k = C^{-1}r^k + \beta_{k-1}p^{k-1}$$

$$(1-2) \quad \alpha_k = \frac{(r^k, C^{-1}r^k)}{(p^k, Ap^k)}$$

$$(1-3) \quad x^{k+1} = x^k + \alpha_k p^k$$

$$(1-4) \quad r^{k+1} = r^k - \alpha_k Ap^k$$

$$(1-5) \quad \beta_k = \frac{(r^{k+1}, C^{-1}r^{k+1})}{(r^k, C^{-1}r^k)}$$

$$(1-6) \quad k++$$

## 6.4 不完全分解

線型方程式の解法として, 係数行列  $A$  を下三角行列  $L = (l_{i,j})$  ( $l_{i,j} = 0, i < j$ ) と上三角行列  $U = (u_{i,j})$  ( $u_{i,j} = 0, i > j$ ) の積に分解する LU 分解法が良く知られているが, 特に係数行列が対称行列であればこの手法は Cholesky 分解法と呼ばれる.

具体的には  $L$  を単位下三角行列として LU 分解は次の演算で求められる:

$$u_{i,j} = a_{i,j} - \sum_{1 \leq k < i} l_{i,k} u_{k,j} \quad (1 \leq i \leq j \leq N)$$

$$l_{i,j} = \left[ a_{i,j} - \sum_{1 \leq k < j} l_{i,k} u_{k,j} \right] / u_{j,j} \quad (1 \leq j < i \leq N)$$



大規模疎行列の線形方程式を反復法で求解する場合，前処理として，計算する部分を適当に決める不完全 Cholesky 分解が利用される．例えば次のような集合  $G = \{(i, j) \in [1, N] \times [1, N] \mid a_{i,j} \neq 0\}$  を考える．不完全 Cholesky 分解では， $(i, j)$  が  $G$  に属さない場合，対応する  $u, l$  の計算を行わない．この  $G$  の決定には，メモリ使用量，演算回数等のバランスをとることが重要である．

## 7 定常反復法

### 7.1 Jacobi 法

ヤコビ法は， $n$  元連立一次方程式

$$Ax = b$$

を解くための解法の一つであり，反復法的一种である．反復法の中では収束が遅い部類である．

ヤコビ法は，行列  $A$  が対角優位であれば収束する．ポアソン方程式やラプラス方程式を差分法等で離散化すると，連立一次方程式  $Ax = b$  が生成されるが，この  $A$  は一般に疎行列で，対角優位性を有しており，ヤコビ法が適用できる．

以下，ヤコビ法のアルゴリズムを紹介する．いま， $A$  を以下のように3つの行列にわけると，

$$A = L + D + U$$

ここで，

- D :  $n \times n$  の対角行列
- L : 下三角行列 (対角部分は 0)
- U : 上三角行列 (対角部分は 0)

である．適当な初期値  $x^{(0)}$  から始め，以下の漸化式により反復計算を行う．

$$x^{(k+1)} = Hx^{(k)} + C$$

ここで，

$$\begin{aligned} H &= -D^{-1}(L + U) \\ C &= D^{-1}b \end{aligned}$$

である．適当な収束条件を課す必要があるが，ここではヤコビ法の1回の反復における総計算量に比べて計算量が比較的少ない次の条件

$$\frac{\sum_i |x_i^{(k+1)} - x_i^{(k)}|}{\sum_i |x_i^{(k)}|} < \epsilon$$

を採用する ( $\epsilon$  は十分小さな定数) .

具体的なアルゴリズムは以下のようになる .

```
given matrix information:  $n, a_{ij}, b_i$  ( $1 \leq i, j \leq n$ )
set initial value:  $\mathbf{x}$ 
loop = 1
while ( loop == 1 ) do:
  for  $i = 1, \dots, n$  do:
    
$$z_i = \frac{1}{a_{ii}} \left\{ b_i - \sum_{j \neq i} a_{ij} x_j \right\}$$

    sum1 = sum1 +  $|z_i|$ 
    sum2 = sum2 +  $|z_i - x_i|$ 
  end do
  if (  $\frac{\text{sum2}}{\text{sum1}} < \epsilon$  ) then
    loop = 0
  end if
   $\mathbf{x} = \mathbf{z}$ 
end do
```

## 7.2 Gauss-Seidel 法

ガウスザイデル法は,  $n$  元連立一次方程式

$$Ax = b$$

を解くための解法の一つであり, 反復法に分類される. ガウスザイデル法は収束が遅い部類の解法であるが, 収束は一般にヤコビ法よりも速い. 実用では高速な解法であるマルチグリッド法で局所的に用いられている. ガウスザイデル法は, 誤差の高周波部分 (ギザギザなところ) の収束が速いが, 一方低周波部分 (滑らかなところ) の収束が遅いという特徴がある. マルチグリッド法では, 行列を格子とみなし, 細かい格子と荒い格子を用いる. 細かいグリッドでの誤差の低周波部分が, 荒いグリッドでは相対的に高周波部分として表現されるため, ガウスザイデル法を用いると収束が速くなる.

ガウスザイデル法は, 行列  $A$  が対角優位, もしくは正定値対称であれば必ず収束する. ポアソン方程式やラプラス方程式を差分法等で離散化すると, 連立一次方程式  $Ax = b$  が生成されるが, この  $A$  は一般に疎行列で, また正定値対称性と対角優位性を有しており, ガウスザイデル法が適用できる.

以下，ガウスザイデル法のアルゴリズムを紹介する．ヤコビ法と同様に， $A$  を 3 つの行列にわけると

$$A = L + D + U$$

適当な初期値  $x^{(0)}$  から始め，以下の漸化式

$$x^{(k+1)} = Hx^{(k)} + C$$

により反復計算を行うが，ガウスザイデル法では以下のようにする，

$$H = -(D + L)^{-1}U$$

$$C = (D + L)^{-1}b$$

適当な収束条件を課す必要があるが，ここでもガウスザイデル法の 1 回の反復における総計算量に比べて計算量が比較的少ない次の条件

$$\frac{\sum_i |x_i^{(k+1)} - x_i^{(k)}|}{\sum_i |x_i^{(k)}|} < \epsilon$$

を採用する ( $\epsilon$  は十分小さな定数) ．

具体的なアルゴリズムは以下のようになる．

given matrix information:  $n, a_{ij}, b_i$  ( $1 \leq i, j \leq n$ )

set initial value:  $x$

loop = 1

while ( loop == 1 ) do:

  for  $i = 1, \dots, n$  do:

$$x_{new} = \frac{1}{a_{ii}} \left\{ b_i - \sum_{j \neq i} a_{ij} x_j \right\}$$

$$\text{sum1} = \text{sum1} + |x_{new}|$$

$$\text{sum2} = \text{sum2} + |x_{new} - x_i|$$

$$x_i = x_{new}$$

  end do

  if (  $\frac{\text{sum2}}{\text{sum1}} < \epsilon$  ) then

    loop = 0

  end if

end do

次に， $A$  が対角優位である場合にガウスザイデル法が収束することを証明する．

Theorem 7.1.  $A$  が対角優位

$$\max_i \frac{1}{|a_{ii}|} \left( \sum_{j \neq i} |a_{ij}| \right) < 1$$

ならば, ガウスザイデル法は収束する.

*Proof.*  $H$  の固有値を用いて証明する.

$$\begin{aligned} H &= -(D + L)^{-1}U \\ H\mathbf{v} &= \lambda\mathbf{v} \rightarrow -(D + L)^{-1}U\mathbf{v} = \lambda\mathbf{v} \end{aligned}$$

まとめると

$$-U\mathbf{v} = \lambda(D + L)\mathbf{v}$$

ここで

$$|v_p| = \max_i |v_i|$$

とし, 連立方程式の第  $p$  段目をかくと

$$-\sum_{j>p} a_{pj}v_j = \lambda \sum_{j<p} a_{pj}v_j + \lambda a_{pp}v_p$$

これを  $v_p$  で割ると

$$-\sum_{j>p} a_{pj}v_j/v_p = \lambda \sum_{j<p} a_{pj}v_j/v_p + \lambda a_{pp}$$

すなわち

$$\lambda a_{pp} = -\sum_{j>p} a_{pj}v_j/v_p - \lambda \sum_{j<p} a_{pj}v_j/v_p$$

ここで,  $|\lambda| \geq 1$  と仮定する.

$$\begin{aligned} |\lambda| |a_{pp}| &\leq |\lambda| \sum_{j \neq p} |a_{pj}v_j/v_p| \\ &\leq |\lambda| \sum_{j \neq p} |a_{pj}| \end{aligned}$$

よって

$$|a_{pp}| \leq \sum_{j \neq p} |a_{pj}|$$

これは対角優位に反する. よって  $|\lambda| < 1$ .

(証明終)

最後に、並列計算への応用における問題について少し触れる。  
ヤコビ法が並列化が容易なのに対し、ガウスザイデル法は一般に並列化が困難である。ヤコビ法では、1回の反復において  $x^{k+1}$  の全成分を求めるまで  $x^{k+1}$  自身の値を用いない。ガウスザイデル法では、1回の反復において  $x^{k+1}$  の各成分が計算されたときに、その値を即座に他の  $x^{k+1}$  の成分の計算に用いる。このように、ガウスザイデル法では1回の反復において各成分で依存関係が生じるため、並列化が困難となる。計算速度を落とさずにガウスザイデル法の並列化を行うには、高度なプログラミングが必要となる。

### 7.3 分散並列処理

並列計算を行う目的は、複数の CPU を用いることによって、単一 CPU の場合よりも早く計算を行うことにある。並列計算マシンは、共有メモリ型と分散メモリ型の二つに分類することが出来る。

共有メモリ型マシンは、主記憶装置を複数の CPU で共有するタイプのマシンである。一般的に、この処理系の長所は比較的コーディングが楽で、しかも良い性能が得られることであるが、短所としては計算機本体の値段が高いということが挙げられる。共有メモリ型マシンでのコーディングは、OpenMP (Open Multi Processor) や thread という規格に基づいて行う。一方分散メモリ型マシンとは、CPU とメモリの組 (ノード) をネットワークによって複数繋げたもので、いわゆる PC クラスタやグリッドがこれに相当する。分散メモリ型マシンの長所は安価にシステムを構築できることであるが、コーディングは前者に比べ容易ではない。分散メモリ型マシンでのコーディングは MPI (Message Passing Interface) や PVM (Parallel Virtual Machine) という規格に基づいて行う。

この他にも、これらのハイブリッドといえる分散共有メモリ型や共有分散メモリ型が存在する。分散共有メモリ型は、各ノードが共有メモリ型になっているもので、現在の主流である。有名な地球シミュレータもこれに相当する。共有分散メモリ型は複数の CPU がネットワークを介してメモリを共有するもので、ネットワーク負荷が増大するがコーディングが容易である。

### 7.4 内積演算の平行化

本節では  $\mathbb{R}^N$  の内積演算を例に分散メモリ型マシンへの実装を考える。 $M$  個の CPU での並列計算を行う場合、ベクトルデータ  $u = (u_1, \dots, u_N)$  と  $v = (v_1, \dots, v_N)$  をそれぞれ次のように  $M$  個のブロックに分割する。

$$u = \left( \tilde{u}_0 \mid \tilde{u}_1 \mid \dots \mid \tilde{u}_{M-1} \right), v = \left( \tilde{v}_0 \mid \tilde{v}_1 \mid \dots \mid \tilde{v}_{M-1} \right)$$

このように分割した後、各 CPU でブロック化されたベクトルの内積演算を行い、最後にそれらの和を取れば良い．従って内積演算を行うプロシージャは次のようになる:

(0)  $\gamma_m = (\tilde{u}_m, \tilde{v}_m)$  is calculated at each CPU

(1) send  $\gamma_m$  to all ranks and make summation:

$$\gamma = \sum_{m=0}^{M-1} \gamma_m \text{ [with MPI\_Allreduce]}$$

## 7.5 CG 法のパラレル化

前節に続いて CG 法の分散メモリ型マシンへの実装を考える．全  $M$  個の CPU での並列計算を行なう場合，行列  $A$  を次のように縦横それぞれ  $M$ ，合計  $M^2$  の領域に分割する．その対角に位置する小行列のみを取り出し，それを行列  $K$  とする (ブロック iCCG 法):

$$A = \begin{pmatrix} \tilde{A}_{0,0} & \tilde{A}_{0,1} & \cdots & \tilde{A}_{0,M-1} \\ \tilde{A}_{1,0} & \tilde{A}_{1,1} & \cdots & \tilde{A}_{1,M-1} \\ \cdots & \cdots & \cdots & \cdots \\ \tilde{A}_{M-1,0} & \tilde{A}_{M-1,1} & \cdots & \tilde{A}_{M-1,M-1} \end{pmatrix}, \quad K = \begin{pmatrix} \tilde{K}_0 & & & 0 \\ & \tilde{K}_1 & & \\ & & \ddots & \\ 0 & & & \tilde{K}_{M-1} \end{pmatrix},$$

$$\tilde{K}_m = \tilde{A}_{m,m}$$

これらの対角小行列  $\tilde{K}_m$  をそれぞれの CPU (番号  $m$ ) が担当する．またベクトル演算の場合も前節と同様に  $M$  分割し，CPU  $m$  が  $m$  番目の部分を担当する．対角小行列  $\tilde{K}_m$  を各小行列毎に不完全分解し ( $\tilde{C}_m$  とする)、前処理付き CG 法に利用する．従って前処理付き CG 法のプロシージャは次のようになる:

(0) make preconditioned matrix  $\tilde{C}_m$  ( $i = 1, \dots, m$ )

(1)  $x^0$  is an initial guess,  $r^0 = b - Ax^0$ ,  $\beta_{k-1} = 0$ ,  $k = 0$

$$\zeta_m^k = \left( r_m^k, \tilde{C}_m^{-1} r_m^k \right),$$

send  $\zeta_m^k$  to all ranks and make summation  $\zeta^k = \sum_m \zeta_m^k$  with MPI\\_Allreduce

(2) for  $k = 0, 1, \dots$ , until  $\|r^k\|/\|b\| < \exists \epsilon$  do:

$$(2-1) \quad p_m^k = \tilde{C}_m^{-1} r_m^k + \beta_{k-1} p_m^{k-1}$$

(2-2) send  $p_m^k$  to all ranks and build  $p^k$  from  $p_m^k$  with MPI\\_Allgatherv

$$(2-3) \quad \eta^k = (p^k, Ap^k)$$

$$(2-4) \quad \alpha_k = \zeta^k / \eta^k$$

$$(2-5) \quad x_m^{k+1} = x_m^k + \alpha_k p_m^k$$

$$(2-6) \quad r_m^{k+1} = r_m^k - \alpha_k A p_m^k$$

$$(2-7) \quad \zeta_m^{k+1} = \left( r_m^{k+1}, \tilde{C}_m^{-1} r_m^{k+1} \right)$$

$$(2-8) \quad \text{send } \zeta_m^{k+1} \text{ to all ranks and make summation} \\ \zeta^{k+1} = \sum_m \zeta_m^{k+1} \text{ with MPI\_Allreduce}$$

$$(2-9) \quad \beta_k = \zeta^{k+1} / \zeta^k$$

$$(2-10) \quad k++$$

## 8 離散勾配流法 (Discrete Morse Flow Method)

ここで、離散勾配流法を紹介する。この方法は時間を差分化し、元の問題を各時間ステップでの最小化問題で近似する。さらに最小化関数を時間補間し、時間差分幅を0に持っていくことで元の問題の弱解が得られる。

### 8.1 放物型離散勾配流法

熱方程式とを例にとって詳細に解説する。

熱方程式を滑らかな境界  $\partial\Omega$  を持つ有界領域  $\Omega \subset \mathbb{R}^m$  で考える。境界上、同時 Dirichlet 境界条件が与えられているとする。そして、初期形状  $u_0 \in H_0^1(\Omega)$  も与えられている。すると、以下の問題を解くことになる：

$$u_t(t, x) = \Delta u(t, x) \quad \text{in } Q_T = (0, T) \times \Omega, \quad (8.1)$$

$$u(t, x) = 0 \quad \text{on } (0, T) \times \partial\Omega, \quad (8.2)$$

$$u(0, x) = u_0(x) \quad \text{in } \Omega. \quad (8.3)$$

まず、時間区間  $(0, T)$  を  $N$  等分し、時間差分幅を  $h = T/N$  とおく。ここで  $u_0$  は時刻  $t = 0$  での近似解に対応する。時刻  $t = nh$  ( $n = 1, 2, \dots, N$ ) での近似解  $u_n$  を次の汎関数の  $H_0^1(\Omega)$  での最小化関数として定義する：

$$J_n(u) = \int_{\Omega} \frac{|u - u_{n-1}|^2}{2h} dx + \frac{1}{2} \int_{\Omega} |\nabla u|^2 dx. \quad (8.4)$$

この汎関数の第二項目は  $H^1(\Omega)$  での弱収束に対して下半連続であり、第一項は  $L^2(\Omega)$  で連続である。汎関数が下から有界であるから、minimizer の存在がいえる。

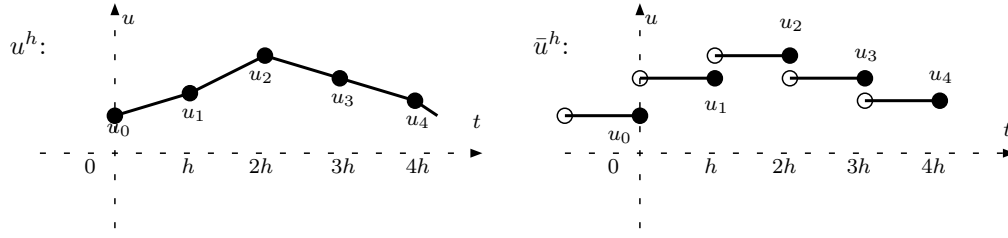


図 2: Minimizers の補間

### 8.1.1 近似解の構成

次に,  $\{u_n\}_{n=0}^N$  の最小化関数を時間補間して, 近似解  $\bar{u}^h$  と  $u^h$  を定義する (図 2 を参照):

$$\bar{u}^h(t, x) = \begin{cases} u_0(x), & t = 0 \\ u_n(x), & t \in ((n-1)h, nh], n = 1, \dots, N \end{cases} \quad (8.5)$$

$$u^h(t, x) = \begin{cases} u_0(x), & t = 0 \\ \frac{t-(n-1)h}{h}u_n(x) + \frac{nh-t}{h}u_{n-1}(x), & t \in ((n-1)h, nh], n = 1, \dots, N. \end{cases}$$

$u_n$  は汎関数  $J_n$  の最小化関数であるので,  $u_n$  での  $J_n$  の第一変分が 0 になる. したがって, 任意の  $\varphi \in H_0^1(\Omega)$  に対し, 次の関係が成り立つ.

$$\begin{aligned} 0 &= \frac{d}{d\varepsilon} J_n(u_n + \varepsilon\varphi)|_{\varepsilon=0} = \lim_{\varepsilon \rightarrow 0} \frac{J_n(u_n + \varepsilon\varphi) - J_n(u_n)}{\varepsilon} \\ &= \lim_{\varepsilon \rightarrow 0} \frac{1}{\varepsilon} \int_{\Omega} \frac{|u_n + \varepsilon\varphi - u_{n-1}|^2 - |u_n - u_{n-1}|^2}{2h} dx \\ &\quad + \lim_{\varepsilon \rightarrow 0} \frac{1}{2\varepsilon} \int_{\Omega} (|\nabla u_n + \varepsilon\nabla\varphi|^2 - |\nabla u_n|^2) dx \\ &= \lim_{\varepsilon \rightarrow 0} \int_{\Omega} \frac{(2u_n - 2u_{n-1} + \varepsilon\varphi)\varphi}{2h} dx + \lim_{\varepsilon \rightarrow 0} \frac{1}{2} \int_{\Omega} (2\nabla u_n \nabla\varphi + \varepsilon|\nabla\varphi|^2) dx \\ &= \int_{\Omega} \frac{u_n - u_{n-1}}{h} \varphi dx + \int_{\Omega} \nabla u_n \nabla\varphi dx \end{aligned} \quad (8.6)$$

$\bar{u}^h$  と  $u^h$  の定義 (8.5) を使うと, 次のように書き換えることができる.

$$\int_{\Omega} [u_t^h(t)\varphi + \nabla\bar{u}^h(t)\nabla\varphi] dx = 0 \quad \text{for a.e. } t \in (0, T) \quad \forall \varphi \in H_0^1(\Omega)$$

上式が任意の  $\varphi \in L^2(0, T; H_0^1(\Omega))$  に対して成り立つので, 時間において  $(0, T)$  で積分すると,

$$\int_0^T \int_{\Omega} [u_t^h \varphi + \nabla\bar{u}^h \nabla\varphi] dx dt = 0 \quad \forall \varphi \in L^2(0, T; H_0^1(\Omega)) \quad (8.7)$$

が得られる. これより,  $\bar{u}^h$  と  $u^h$  を近似弱解と呼ぶことにする.



### 8.1.2 近似弱解の収束

これから時間差分幅を 0 に持っていき、真の弱解を構成することを目標とする。そのためには、近似弱解の評価が必要になる。以下の補題でそれについて述べる。

**Lemma 8.1.**  $J_n$  ( $n = 1, \dots, N$ ) は (8.4) で定義した汎関数で、 $u_n$  はそれに対する  $H_0^1(\Omega)$  における最小化関数とする。関数  $\bar{u}^h$  と  $u^h$  を (8.5) で定義する。さらに、 $h \leq 1$  と仮定する。そのとき、以下の評価が成り立つ。

$$\int_0^t \|u_t^h\|_{L^2(\Omega)} dt + \|\nabla \bar{u}^h(t)\|_{L^2(\Omega)} \leq C_E \quad \text{for a.e. } t \in (0, T) \quad (8.8)$$

ただし、定数  $C_E$  が初期関数の  $H^1$  ノルムに依るが、 $h$  には依らない。

*Proof.*  $u_n$  が  $J_n$  の minimizer であることにより、

$$J_n(u_n) \leq J_n(u_{n-1})$$

であるので、

$$\int_{\Omega} \frac{|u_n - u_{n-1}|^2}{2h} dx + \frac{1}{2} \int_{\Omega} |\nabla u_n|^2 dx \leq \frac{1}{2} \int_{\Omega} |\nabla u_{n-1}|^2 dx \quad (8.9)$$

式 (8.9) の両辺を  $n = 1, \dots, k$  まで和を取ると、

$$\sum_{n=1}^k \int_{\Omega} \frac{|u_n - u_{n-1}|^2}{2h} dx + \frac{1}{2} \int_{\Omega} |\nabla u_k|^2 dx \leq \frac{1}{2} \int_{\Omega} |\nabla u_0|^2 dx \quad (8.10)$$

ここで

$$u_t^h(t) = \frac{u_n - u_{n-1}}{h} \quad \text{for } t \in ((n-1)h, nh), \quad n = 1, 2, \dots, N$$

であるので、式 (8.10) は

$$\frac{1}{2} \sum_{n=1}^k \int_{(n-1)h}^{nh} \int_{\Omega} (u_t^h)^2 dx dt + \frac{1}{2} \int_{\Omega} |\nabla u_k|^2 dx \leq \frac{1}{2} \int_{\Omega} |\nabla u_0|^2 dx \quad (8.11)$$

となり、式 (8.8) が成り立つ。  $\square$

(8.8) の評価のおかげで、Eberlein-Shmulyan の定理が適用でき、 $L^2(Q_T)$  においてある関数  $\mathbf{v}$  に弱収束する部分列  $\{\nabla \bar{u}^{h_k}\}_{k \in \mathbb{N}}$  があることが分かる。この部分列からさらに部分列を選び、 $\{u_t^{h_{k_l}}\}_{l \in \mathbb{N}}$  が  $L^2(Q_T)$  においてある関数  $U$  に収束することがいえる。表記を簡単にするため、次のように書くことにする。

$$\nabla \bar{u}^h \rightharpoonup \mathbf{v} \quad \text{in } (L^2(Q_T))^m, \quad (8.12)$$

$$u_t^h \rightharpoonup U \quad \text{in } L^2(Q_T). \quad (8.13)$$

$L^2(Q_T)$  の意味で  $v = \nabla u$  と  $U = u_t$  が成り立つような  $u \in H_0^1(Q_T)$  が存在するということを証明しなければならない。そのため、より詳細な評価が必要である。まず、 $t \in ((n-1)h, nh)$  とすると

$$\begin{aligned}
& \|\bar{u}^h - u^h\|_{L^2(Q_T)}^2 \\
&= \int_0^T \int_{\Omega} (\bar{u}^h - u^h)^2 dx dt \\
&= \sum_{n=1}^N \int_{(n-1)h}^{nh} \int_{\Omega} \left( u_n - \frac{t - (n-1)h}{h} u_n - \frac{nh - t}{h} u_{n-1} \right)^2 dx dt \\
&= \sum_{n=1}^N \int_{(n-1)h}^{nh} \int_{\Omega} \left( \frac{nh - t}{h} \right)^2 (u_n - u_{n-1})^2 dx dt \\
&= \sum_{n=1}^N \int_{\Omega} \frac{h}{3} (u_n - u_{n-1})^2 dx \\
&= \frac{h^2}{3} \int_0^T \int_{\Omega} (u_t^h)^2 dx dt \leq \frac{h^2 C_E^2}{3}
\end{aligned}$$

が得られる。つまり、

$$\|\bar{u}^h - u^h\|_{L^2(Q_T)} \leq Ch.$$

次に、

$$\begin{aligned}
& \|u^h\|_{L^2(Q_T)}^2 - \|\bar{u}^h\|_{L^2(Q_T)}^2 \\
&= \int_0^T \int_{\Omega} ((u^h)^2 - (\bar{u}^h)^2) dx dt \\
&= \sum_{n=1}^N \int_{(n-1)h}^{nh} \int_{\Omega} \left[ \left( \frac{t - (n-1)h}{h} u_n - \frac{nh - t}{h} u_{n-1} \right)^2 - u_n^2 \right] dx dt \\
&= \sum_{n=1}^N \int_{(n-1)h}^{nh} \int_{\Omega} \left[ \frac{(t - (n-1)h)^2 - h^2}{h^2} u_n^2 \right. \\
&\quad \left. + 2 \frac{(t - (n-1)h)(nh - t)}{h^2} u_n u_{n-1} + \frac{(nh - t)^2}{h^2} u_{n-1}^2 \right] dx dt \\
&= \sum_{n=1}^N \int_{\Omega} \left[ -\frac{2h}{3} u_n^2 + \frac{h}{3} u_n u_{n-1} + \frac{h}{3} u_{n-1}^2 \right] dx \\
&\leq \frac{h}{6} \sum_{n=1}^N \int_{\Omega} [-4u_n^2 + u_n^2 + u_{n-1}^2 + 2u_{n-1}^2] dx \\
&\leq \frac{h}{2} \sum_{n=1}^N \int_{\Omega} (-u_n^2 + u_{n-1}^2) dx = \frac{h}{2} \int_{\Omega} (u_0^2 - u_N^2) dx \\
&\leq \frac{h}{2} \|u_0\|_{L^2(\Omega)}^2
\end{aligned}$$

と計算できる. 同様に,

$$\|\nabla u^h\|_{L^2(Q_T)}^2 - \|\nabla \bar{u}^h\|_{L^2(Q_T)}^2 \leq \frac{h}{2} \|\nabla u_0\|_{L^2(\Omega)}^2$$

が分かる. そして Poincaré の不等式より

$$\|u^h\|_{L^2(Q_T)} \leq C_P \|\nabla u^h\|_{L^2(Q_T)} \quad \text{for all } h \in (0, 1) \quad (8.14)$$

となるような  $C_P$  が存在する. 以上の結果をまとめる.

**Lemma 8.2.** 以下の関係式が成り立つ.

$$\|\bar{u}^h - u^h\|_{L^2(Q_T)} \leq Ch, \quad (8.15)$$

$$\|u^h\|_{L^2(Q_T)}^2 \leq \|\bar{u}^h\|_{L^2(Q_T)}^2 + \frac{h}{2} \|u_0\|_{L^2(\Omega)}^2, \quad (8.16)$$

$$\|\nabla u^h\|_{L^2(Q_T)}^2 \leq \|\nabla \bar{u}^h\|_{L^2(Q_T)}^2 + \frac{h}{2} \|\nabla u_0\|_{L^2(\Omega)}^2. \quad (8.17)$$

(8.8), (8.17) と (8.14) より,  $u^h$  が  $H^1(Q_T)$  で一様有界である. ゆえに,  $H^1(Q_T)$  で弱収束し,  $L^2(Q_T)$  で強収束する部分列がある. 収束先を  $u$  と書く:

$$u^h \rightharpoonup u \quad \text{weakly in } H^1(Q_T). \quad (8.18)$$

(8.13) より,  $U = u_t$  がほとんどいたるところで成り立つ. さらに, (8.12) より, 任意の  $\varphi \in C_0^\infty(Q_T)$  に対して,

$$\int_0^T \int_\Omega \left( \frac{\partial \bar{u}^h}{\partial x_i} - \frac{\partial u^h}{\partial x_i} \right) \varphi \, dx \, dt \rightarrow \int_0^T \int_\Omega \left( \mathbf{v}_i - \frac{\partial u}{\partial x_i} \right) \varphi \, dx \, dt \quad \text{as } h \rightarrow 0+$$

が成立し, (8.15) より

$$\int_0^T \int_\Omega \left( \frac{\partial \bar{u}^h}{\partial x_i} - \frac{\partial u^h}{\partial x_i} \right) \varphi \, dx \, dt = - \int_0^T \int_\Omega (\bar{u}^h - u^h) \frac{\partial \varphi}{\partial x_i} \, dx \, dt \rightarrow 0 \quad \text{as } h \rightarrow 0+$$

がいえる. すなわち,  $Q_T$  のほとんどいたるところで  $\mathbf{v} = \nabla u$  が得られた. 以上を合わせると, 次の関係をみだす関数  $u \in H^1(Q_T)$  が存在することを示した.

$$\nabla \bar{u}^h \rightharpoonup \nabla u \quad \text{in } (L^2(Q_T))^m, \quad (8.19)$$

$$u_t^h \rightharpoonup u_t \quad \text{in } L^2(Q_T). \quad (8.20)$$

今, (8.7) において  $h \rightarrow 0+$  という極限がとれるようになった. ここで,  $\varphi \in L^2(0, T; H_0^1(\Omega))$  を任意に選ぶと

$$\int_0^T \int_\Omega \nabla \bar{u}^h \nabla \varphi \, dx \, dt \rightarrow \int_0^T \int_\Omega \nabla u \nabla \varphi \, dx \, dt \quad \text{as } h \rightarrow 0+ \quad (8.21)$$

となる. さらに,

$$\int_0^T \int_{\Omega} u_t^h \varphi \, dx \, dt \rightarrow \int_0^T \int_{\Omega} u_t \varphi \, dx \, dt \quad \text{as } h \rightarrow 0+$$

が得られる.

以下のことが分かった. すなわち  $u$  は

$$\int_0^T \int_{\Omega} (u_t \varphi + \nabla u \nabla \varphi) \, dx \, dt = 0 \quad \forall \varphi \in L^2(0, T; H_0^1(\Omega)) \quad (8.22)$$

を満たす. また,  $(\{0\} \times \Omega) \cup ([0, T] \times \partial\Omega)$  上のトレースが 0 という  $H^1(Q_T)$  の関数が  $H^1(Q_T)$  の閉部分空間であり, 弱閉集合でもあるので, (8.18) より  $u$  が初期条件 (8.3) と境界条件 (8.2) をトレースの意味で満たすことがいえる.

## 8.2 双曲型離散勾配流法

次に波動方程式を例にとって詳細に解説する. 波動方程式を滑らかな境界  $\partial\Omega$  を持つ有界領域  $\Omega \subset \mathbb{R}^m$  で考える. 境界上, 同時 Dirichlet 境界条件が与えられているとする. そして, 初期形状  $u_0 \in H_0^1(\Omega)$  と初期速度  $v_0 \in H_0^1(\Omega)$  も与えられている. すると, 以下の問題を解くことになる:

$$u_{tt}(t, x) = \Delta u(t, x) \quad \text{in } Q_T = (0, T) \times \Omega, \quad (8.23)$$

$$u(t, x) = 0 \quad \text{on } (0, T) \times \partial\Omega, \quad (8.24)$$

$$u(0, x) = u_0(x) \quad \text{in } \Omega, \quad (8.25)$$

$$u_t(0, x) = v_0(x) \quad \text{in } \Omega. \quad (8.26)$$

熱方程式のとき同様に時間区間を  $N$  等分し, さらに,  $u_1(x) = u_0(x) + hv_0(x)$  と定義する.

ここで  $u_0$  は時刻  $t = 0$  での近似解,  $u_1$  は時刻  $t = h$  での近似解に対応する. 時刻  $t = nh$  ( $n = 2, 3, \dots, N$ ) での近似解  $u_n$  を次の汎関数の  $H_0^1(\Omega)$  での最小化関数として定義する.

$$J_n(u) = \int_{\Omega} \frac{|u - 2u_{n-1} + u_{n-2}|^2}{2h^2} \, dx + \frac{1}{2} \int_{\Omega} |\nabla u|^2 \, dx \quad (8.27)$$

この汎関数の第二項目は  $H^1(\Omega)$  での弱収束に対して下半連続であり, 第一項は  $L^2(\Omega)$  で連続である. 汎関数が下から有界であるから, minimizer の存在がいえる.

### 8.2.1 近似解の構成

$u_n$  は汎関数  $J_n$  の最小化関数であるので,  $u_n$  での  $J_n$  の第一変分が 0 になる. したがって, 任意の  $\varphi \in H_0^1(\Omega)$  に対し, 次の関係が成り立つ.

$$\begin{aligned}
0 &= \frac{d}{d\varepsilon} J_n(u_n + \varepsilon\varphi)|_{\varepsilon=0} = \lim_{\varepsilon \rightarrow 0} \frac{J_n(u_n + \varepsilon\varphi) - J_n(u_n)}{\varepsilon} \\
&= \lim_{\varepsilon \rightarrow 0} \frac{1}{\varepsilon} \int_{\Omega} \frac{|u_n + \varepsilon\varphi - 2u_{n-1} + u_{n-2}|^2 - |u_n - 2u_{n-1} + u_{n-2}|^2}{2h^2} dx \\
&\quad + \lim_{\varepsilon \rightarrow 0} \frac{1}{2\varepsilon} \int_{\Omega} (|\nabla u_n + \varepsilon\nabla\varphi|^2 - |\nabla u_n|^2) dx \\
&= \lim_{\varepsilon \rightarrow 0} \int_{\Omega} \frac{(2u_n + \varepsilon\varphi - 4u_{n-1} + 2u_{n-2})\varphi}{2h^2} dx + \lim_{\varepsilon \rightarrow 0} \frac{1}{2} \int_{\Omega} (2\nabla u_n \nabla\varphi + \varepsilon|\nabla\varphi|^2) dx \\
&= \int_{\Omega} \frac{u_n - 2u_{n-1} + u_{n-2}}{h^2} \varphi dx + \int_{\Omega} \nabla u_n \nabla\varphi dx \tag{8.28}
\end{aligned}$$

$\bar{u}^h$  と  $u^h$  の定義 (8.5) を使うと, 次のように書き換えることができる:

$$\int_{\Omega} \left[ \frac{u_t^h(t) - u_t^h(t-h)}{h} \varphi + \nabla \bar{u}^h(t) \nabla \varphi \right] dx = 0 \quad \text{for a.e. } t \in (h, T) \quad \forall \varphi \in H_0^1(\Omega).$$

上式が任意の  $\varphi \in L^2(0, T; H_0^1(\Omega))$  に対して成り立つので, 時間において  $(h, T)$  で積分すると,

$$\int_h^T \int_{\Omega} \left[ \frac{u_t^h(t) - u_t^h(t-h)}{h} \varphi + \nabla \bar{u}^h \nabla \varphi \right] dx dt = 0 \quad \forall \varphi \in L^2(0, T; H_0^1(\Omega)) \tag{8.29}$$

が得られる. これより,  $\bar{u}^h$  と  $u^h$  を近似弱解と呼ぶことにする.

### 8.2.2 近似弱解の収束

これから時間差分幅を 0 に持っていき, 真の弱解を構成することを目標とする. そのためには, 近似弱解の評価が必要になる. 以下の補題でそれについて述べる.

**Lemma 8.3.**  $J_n$  ( $n = 2, \dots, N$ ) は (8.27) で定義した汎関数で,  $u_n$  はそれに対する  $H_0^1(\Omega)$  における最小化関数だとする. 関数  $\bar{u}^h$  と  $u^h$  を (8.5) で定義する. さらに,  $h \leq 1$  と仮定する. そのとき, 以下の評価が成り立つ.

$$\|u_t^h(t)\|_{L^2(\Omega)} + \|\nabla \bar{u}^h(t)\|_{L^2(\Omega)} \leq C_E \quad \text{for a.e. } t \in (0, T) \tag{8.30}$$

ただし, 定数  $C_E$  が初期関数の  $H^1$  ノルムに依るが,  $h$  には依らない.

*Proof.* この種の評価は普通, 方程式を解の時間微分でテストすることによって示される. 離散の場合は, これが (8.28) で  $\varphi := u_n - u_{n-1}$  とおくことに相当する.

$$\int_{\Omega} \frac{u_n - 2u_{n-1} + u_{n-2}}{h^2} (u_n - u_{n-1}) dx + \int_{\Omega} (\nabla u_n - \nabla u_{n-1}) \nabla u_n dx = 0$$

が分かる. ここで, 次の不等式

$$\frac{a^2}{2} - \frac{b^2}{2} \leq (a - b)a, \quad \forall a, b \in \mathbb{R}$$

を応用し,  $n = 2, 3, \dots, N$  に対し次が成り立つことが確認される.

$$\begin{aligned} \int_{\Omega} \left[ \left( \frac{u_n - u_{n-1}}{h} \right)^2 - \left( \frac{u_{n-1} - u_{n-2}}{h} \right)^2 + |\nabla u_n|^2 - |\nabla u_{n-1}|^2 \right] dx &\leq 0 \\ \int_{\Omega} \left[ \left( \frac{u_n - u_{n-1}}{h} \right)^2 + |\nabla u_n|^2 \right] dx &\leq \int_{\Omega} \left[ \left( \frac{u_{n-1} - u_{n-2}}{h} \right)^2 + |\nabla u_{n-1}|^2 \right] dx \end{aligned}$$

この不等式の  $n = 2$  から任意の自然数  $k \leq N$  までの和を取ると,

$$\begin{aligned} \int_{\Omega} \left[ \left( \frac{u_k - u_{k-1}}{h} \right)^2 + |\nabla u_k|^2 \right] dx &\leq \int_{\Omega} \left[ \left( \frac{u_1 - u_0}{h} \right)^2 + |\nabla u_1|^2 \right] dx \\ &\leq \int_{\Omega} [(v_0)^2 + |\nabla u_0 + h \nabla v_0|^2] dx \\ &\leq \int_{\Omega} [(v_0)^2 + 2|\nabla u_0|^2 + 2h^2|\nabla v_0|^2] dx \\ &\leq 2\|u_0\|_{H^1(\Omega)}^2 + 2\|v_0\|_{H^1(\Omega)}^2 \end{aligned}$$

となる.

$$u_t^h(t) = \frac{u_k - u_{k-1}}{h} \quad \text{for } t \in ((k-1)h, kh), \quad k = 1, 2, \dots, N$$

であるから, (8.30) が示された. □

(8.30) の評価のおかげで, 式 (8.12), (8.13) 同様に以下のことがいえる:

$$\nabla \bar{u}^h \rightharpoonup \mathbf{v} \quad \text{in } (L^2(Q_T))^m, \quad (8.31)$$

$$u_t^h \rightharpoonup U \quad \text{in } L^2(Q_T). \quad (8.32)$$

$L^2(Q_T)$  の意味で  $v = \nabla u$  と  $U = u_t$  が成り立つような  $u \in H_0^1(Q_T)$  が存在するということを証明しなければならない. そのため, より詳細な評価が必要である.

まず,  $t \in ((n-1)h, nh)$  とすると.

$$\begin{aligned}
\|\bar{u}^h(t) - u^h(t)\|_{L^2(\Omega)}^2 &= \int_{\Omega} (\bar{u}^h - u^h)^2 dx \\
&= \int_{\Omega} \left( u_n - \frac{t - (n-1)h}{h} u_n - \frac{nh - t}{h} u_{n-1} \right)^2 dx \\
&= \int_{\Omega} \left( \frac{nh - t}{h} \right)^2 (u_n - u_{n-1})^2 dx \\
&\leq \int_{\Omega} (u_n - u_{n-1})^2 dx = h^2 \int_{\Omega} (u_t^h)^2 dx \\
&\leq C_E^2 h^2.
\end{aligned}$$

が得られる. つまり,

$$\|\bar{u}^h(t) - u^h(t)\|_{L^2(\Omega)} \leq Ch \quad \text{for a.e. } t \in (0, T).$$

また熱方程式のときに得られた式 (8.16), (8.17) も同様に成り立つ.

以上の結果をまとめる.

**Lemma 8.4.** 以下の関係式が成り立つ.

$$\|\bar{u}^h(t) - u^h(t)\|_{L^2(\Omega)} \leq Ch \quad \text{for a.e. } t \in (0, T), \quad (8.33)$$

$$\|u^h\|_{L^2(Q_T)}^2 \leq \|\bar{u}^h\|_{L^2(Q_T)}^2 + \frac{h}{2} \|u_0\|_{L^2(\Omega)}^2, \quad (8.34)$$

$$\|\nabla u^h\|_{L^2(Q_T)}^2 \leq \|\nabla \bar{u}^h\|_{L^2(Q_T)}^2 + \frac{h}{2} \|\nabla u_0\|_{L^2(\Omega)}^2. \quad (8.35)$$

(8.30), (8.35) と (8.14) より,  $u^h$  が  $H^1(Q_T)$  で一様有界である. ゆえに,  $H^1(Q_T)$  で弱収束し,  $L^2(Q_T)$  で強収束する部分列がある. 収束先を  $u$  と書く:

$$u^h \rightharpoonup u \quad \text{weakly in } H^1(Q_T). \quad (8.36)$$

(8.32) より,  $U = u_t$  がほとんどいたるところで成り立つ. さらに, (8.31) より, 任意の  $\varphi \in C_0^\infty(Q_T)$  に対して,

$$\int_0^T \int_{\Omega} \left( \frac{\partial \bar{u}^h}{\partial x_i} - \frac{\partial u^h}{\partial x_i} \right) \varphi dx dt \rightarrow \int_0^T \int_{\Omega} \left( \mathbf{v}_i - \frac{\partial u}{\partial x_i} \right) \varphi dx dt \quad \text{as } h \rightarrow 0+,$$

が成立し, (8.33) より

$$\int_0^T \int_{\Omega} \left( \frac{\partial \bar{u}^h}{\partial x_i} - \frac{\partial u^h}{\partial x_i} \right) \varphi dx dt = - \int_0^T \int_{\Omega} (\bar{u}^h - u^h) \frac{\partial \varphi}{\partial x_i} dx dt \rightarrow 0 \quad \text{as } h \rightarrow 0+$$

がいえる. すなわち,  $Q_T$  のほとんどいたるところで  $\mathbf{v} = \nabla u$  が得られた.

以上を合わせると, 次の関係をみたす関数  $u \in H^1(Q_T)$  が存在することを示した.

$$\nabla \bar{u}^h \rightharpoonup \nabla u \quad \text{in } (L^2(Q_T))^m, \quad (8.37)$$

$$u_t^h \rightharpoonup u_t \quad \text{in } L^2(Q_T). \quad (8.38)$$

今, (8.29) において  $h \rightarrow 0+$  という極限がとれるようになった. ここで,  $\varphi$  を  $C_0^\infty([0, T]; C_0^\infty(\Omega))$  に制限する.

$$\begin{aligned} \int_h^T \int_\Omega \nabla \bar{u}^h \nabla \varphi \, dx \, dt &= \int_0^T \int_\Omega \nabla \bar{u}^h \nabla \varphi \, dx \, dt - \int_0^h \int_\Omega \nabla \bar{u}^h \nabla \varphi \, dx \, dt \\ &\rightarrow \int_0^T \int_\Omega \nabla u \nabla \varphi \, dx \, dt \quad \text{as } h \rightarrow 0+, \end{aligned} \quad (8.39)$$

ここで, (8.30) の  $\nabla \bar{u}^h$  の有界性を用いた:

$$\begin{aligned} \left| \int_0^h \int_\Omega \nabla \bar{u}^h \nabla \varphi \, dx \, dt \right| &\leq \int_0^h \left( \int_\Omega |\nabla \bar{u}^h|^2 \, dx \right)^{1/2} \left( \int_\Omega |\nabla \varphi|^2 \, dx \right)^{1/2} dt \\ &\leq \int_0^h \sqrt{C_E} C \, dt = Ch \rightarrow 0 \quad \text{as } h \rightarrow 0+. \end{aligned}$$

さらに,

$$\begin{aligned} &\int_h^T \int_\Omega \frac{u_t^h(t) - u_t^h(t-h)}{h} \varphi \, dx \, dt \\ &= \int_h^T \int_\Omega \frac{u_t^h(t)}{h} \varphi(t) \, dx \, dt - \int_0^{T-h} \int_\Omega \frac{u_t^h(t)}{h} \varphi(t+h) \, dx \, dt \\ &= \int_0^T \int_\Omega u_t^h(t) \frac{\varphi(t) - \varphi(t+h)}{h} \, dx \, dt - \int_0^h \int_\Omega \frac{u_t^h(t)}{h} \varphi(t) \, dx \, dt \\ &\quad + \int_{T-h}^T \int_\Omega \frac{u_t^h(t)}{h} \varphi(t+h) \, dx \, dt \quad (8.40) \\ &\rightarrow - \int_0^T \int_\Omega u_t \varphi_t \, dx \, dt - \int_\Omega v_0 \varphi(0) \, dx \quad \text{as } h \rightarrow 0+. \end{aligned}$$

が得られる. この収束は次の事実を使って得られた:

(i) (8.40) の最初の項では,  $u_t^h$  が弱収束し,  $(\varphi(t) - \varphi(t+h))/h$  が  $L^2(Q_T)$  で強収束する;

(ii) 第二項では,  $u_t^h = (u_1 - u_0)/h = v_0$  が  $t \in (0, h)$  に対して成り立つ;

(iii) 第三項では,  $\varphi(t+h) = 0$  が  $t \in (T-h, T)$  に対して成り立つ.

以下のことが分かった.

$$\int_0^T \int_\Omega (-u_t \varphi_t + \nabla u \nabla \varphi) \, dx \, dt - \int_\Omega v_0 \varphi(0) \, dx = 0 \quad \forall \varphi \in C_0^\infty([0, T]; C_0^\infty(\Omega)). \quad (8.41)$$



$(\{0\} \times \Omega) \cup ([0, T] \times \partial\Omega)$  上のトレースが 0 という  $H^1(Q_T)$  の関数が  $H^1(Q_T)$  の閉部分空間であり, 弱閉集合でもあるので, (8.36) より  $u$  が初期条件 (8.25) と境界条件 (8.24) をトレースの意味で満たすことが言える.

## A 熱方程式入門 (解の構成, 差分法)

### A.1 フーリエの解の数学的意味づけ

本節では次の 1 次元熱方程式を考える．本小節は [7] による．

$$\frac{\partial}{\partial t} u = \frac{\partial^2}{\partial x^2} u \quad x \in \Omega, 0 < t < \infty \quad (\text{A.1})$$

ただし,  $\Omega = (0, \pi)$  とする．

境界条件と初期条件はそれぞれ

$$u(0, t) = g_1(t), \quad u(\pi, t) = g_2(t), \quad (0 \leq t < +\infty), \quad (\text{A.2})$$

$$u(x, 0) = f(x), \quad x \in \Omega. \quad (\text{A.3})$$

を課す．ここでは簡単のため  $g_1(t) = g_2(t) \equiv 0$  とする．

まず,  $f(x)$  は,  $f(0) = f(\pi) = 0$  をみたしているとし, かつ,  $C^1$  級の函数としよう． $x \in [-\pi, 0]$  では  $f(x) = -f(-x)$  とおいて,  $f(x)$  の定義域を  $[-\pi, \pi]$  にひろげると,  $f(x)$  はそこで奇函数だから, Fourier 級数は  $\sin nx$  の項のみよりなる．すなわち,

$$f(x) = \sum_{n=1}^{\infty} c_n \sin nx$$

と, Fourier 級数で表わされる (右辺は  $f \in C^1$  ならば一様収束する．実際, 有界変動であればよい)．

形式的に,

$$u(x, t) = \sum_{n=1}^{\infty} e^{-n^2 t} c_n \sin nx \quad (\text{A.4})$$

を定義すると, これが求める解である．このことを示そう．

まず, (A.4) の級数が  $\Omega \times (0, \infty)$  の各点で収束することを示そう．

$$c_n = \frac{2}{\pi} \int_0^{\pi} f(x) \sin nx \, dx \rightarrow 0 \quad (n \rightarrow \infty)$$

Riemann-Lebesgue の定理より,  $|c_n| < M (n = 1, 2, \dots)$  であり,  $\sum e^{-n^2 t} < +\infty$  であるから  $t > 0$  で右辺の級数が収束することは明らかである．

実際,  $u(x, t)$  は  $t > 0$  で,  $(x, t)$  に関して何回でも連続的の微分可能である．このことを示す．まず,  $x$  に関して項別微分したもの

$$\sum_{n=1}^{\infty} n c_n e^{-n^2 t} \sin nx \quad (\text{A.5})$$

の各項の絶対値をとると，

$$\sum e^{-n^2 t} n |c_n| \leq M \sum_{n=1}^{\infty} n e^{-n^2 t} < +\infty$$

で評価される．よって (A.5) は一様収束するので，形式的な項別微分は正しいことになる．すなわち，

$$\frac{\partial u}{\partial x} = \sum_{n=1}^{\infty} n e^{-n^2 t} c_n \cos nx.$$

同様にして，

$$\frac{\partial^2 u}{\partial x^2} = - \sum_{n=1}^{\infty} n^2 e^{-n^2 t} c_n \sin nx.$$

$t$  に関する微分を考えよう． $\delta (> 0)$  を一つとり， $\delta \leq t < \infty$  で  $u(x, t)$  の  $t$  に関する項別微分したものは

$$\sum n^2 e^{-n^2 t} |c_n \sin nx| \leq M \sum_n n^2 e^{-\delta n^2} < +\infty$$

で  $t$  に関して，一様に評価されるから項別微分は正しい．したがって， $t > 0$  で，項別微分は正しい．すなわち，

$$\frac{\partial u}{\partial t} = - \sum n^2 e^{-n^2 t} c_n \sin nx.$$

ゆえに， $u(x, t)$  は  $t > 0$  で解であることがわかった．

同様にして， $t > 0$  で無限回微分可能な函数であることもわかる．

つぎに， $t \rightarrow +0$  のとき， $x$  に関して一様に

$$u(x, t) \rightarrow f(x)$$

となることを示そう．それには，Fourier 級数  $\sum c_n \sin nx$  が一様収束であることと， $\{e^{-n^2 t}\}_{n=1,2,\dots}$  が正の減少列であることに注意して Abel の変換より，任意の  $\varepsilon$  に対して， $N$  が定まって

$$\left| \sum_{n=N+1}^{\infty} e^{-n^2 t} c_n \sin nx \right| < \varepsilon, \quad (t \geq 0)$$

がわかる．ゆえに

$$\begin{aligned} |u(x, t) - f(x)| &\leq \left| \sum_{n=1}^N (e^{-n^2 t} - 1) c_n \sin nx \right| + \left| \sum_{n=N+1}^{\infty} e^{-n^2 t} c_n \sin nx \right| \\ &\quad + \left| \sum_{n=N+1}^{\infty} c_n \sin nx \right| \end{aligned}$$

とわけると，最後の2項は， $t$ の如何にかかわらず  $2\varepsilon$  より小で，第一項は  $t \rightarrow +0$  とともに0に近づく(証明終わり)．

Theorem 1.1. 上記の解  $u(x, t)$  が条件を満足するただ一つのものである．

今，そのようなものが，二つあったとし， $u_1(x, t)$ ， $u_2(x, t)$  としよう． $u(x, t) = u_1 - u_2$  は，初期，ならびに境界条件0の解である．そこで，

$$E(t) = \int_0^\pi u(x, t)^2 dx$$

を考えると， $t > 0$  で，

$$E'(t) = 2 \int_0^\pi u \cdot u_t dx = 2 \int_0^\pi u \cdot u_{xx} dx = -2 \int_0^\pi u_x^2 dx \leq 0 .$$

ここで， $u$  が(1)の解であることと，境界  $x = 0$ ， $x = \pi$  で  $u = 0$  となることを使った．したがって  $E(t) \geq 0$  で，かつ非増大函数であり， $\lim_{t \rightarrow +0} E(t) = 0$  であるから  $E(t) \equiv 0$ ．すなわち  $u(x, t) \equiv 0$ ．

## B クリロフ部分空間

$n$  を1より大きい自然数とする． $i \in \{1, \dots, n\}$  とし，

$$K_i(A, r^0) := \text{span}\{r^0, Ar^0, \dots, A^{i-1}r^0\}$$

と定義する．

Lemma B.1.

$$p^j, r^j \in K_{j+1}(A, r^0), \quad j = 0, \dots, n-1$$

*Proof.* 帰納法による．

$j = 0$  のときは， $p^0 = r^0 \in K_1(A, r^0)$  なので，定理は成り立つ．

次に  $j < i$  ならば定理が成り立つと仮定しよう．すると， $j = i$  のとき，

$$r^i = r^{i-1} + \alpha_{i-1} A p^{i-1}$$

であるが， $r^{i-1} \in K_i(A, r^0)$  かつ  $A p^{i-1} \in A K_i(A, r^0) \subset K_{i+1}(A, r^0)$  なので， $r^i \in K_{i+1}(A, r^0)$  である．また，

$$p^i = r^i + \sum_{l=1}^{i-1} \beta_l^i p^l$$

であるが， $\beta_l^i p^l \in K_i(A, r^0)$  なので結局， $p^i \in K_{i+1}(A, r^0)$  □

**Corollary B.2.**  $r^i = 0$  かつ  $r^j \neq 0$  ( $j = 0, \dots, i-1$ ) のとき

$$K_i(A, r^0) = \text{span}\{r^0, r^1, \dots, r^{i-1}\} = \text{span}\{p^0, r^1, \dots, p^{i-1}\}$$

*Proof.* Lemma B.1 より

$$\begin{aligned} \text{span}\{r^0, r^1, \dots, r^{i-1}\} &\subset K_i(A, r^0), \\ \text{span}\{p^0, r^1, \dots, p^{i-1}\} &\subset K_i(A, r^0). \end{aligned}$$

さらに,

$$(r^k, r^l) = \begin{cases} 0 & \text{if } k \neq l, \\ |r^k|^2 & \text{if } k = l. \end{cases}$$

だから,

$$\dim K_i(A, r^0) = \dim \text{span}\{r^0, \dots, r^{i-1}\} = i.$$

したがって

$$K_i(A, r^0) = \text{span}\{r^0, \dots, r^{i-1}\}.$$

よって, 任意の  $k \in \{0, \dots, i-1\}$  に対して, ある  $\gamma_l^k \in \mathbf{R}$  が存在して

$$p^k = r^k + \sum_{l=1}^{k-1} \beta_l^k p^l = r^k + \sum_{l=1}^{k-1} \gamma_l^k r^l$$

となる. これは  $\{p^0, \dots, p^{i-1}\}$  が一次独立であることを意味する. ゆえに

$$\dim K_i(A, r^0) = \dim \text{span}\{p^0, \dots, p^{i-1}\} = i.$$

したがって

$$K_i(A, r^0) = \text{span}\{p^0, \dots, p^{i-1}\}.$$

□

**Proposition B.3.**  $r^i = 0$ ,  $r^j \neq 0$  ( $j = 0, 1, \dots, i-1$ ) ならば,

$$(r^k, Ap^l) = 0 \quad \forall k, \forall l \in \{0, 1, \dots, i-1\}$$

が成り立つ.

*Proof.*  $r^k \in K_{k+1}(A, r^0)$  かつ,  $k \neq j$  ならば  $(r^k, r^j) = 0$  であることに注意すると, Corollary B.2 より, 任意の  $\xi \in K_k(A, r^0)$  に対して  $(r^k, \xi) = 0$  となる.

また,

$$Ap^l \in AK_{l+1}(A, r^0) \subset K_{l+2}(A, r^0)$$

である.

よって,  $k \geq l + 2$  を仮定すれば,

$$Ap^l \subset K_{l+2}(A, r^0) \subset K_k(A, r^0)$$

に注意して,

$$(r^k, Ap^l) = 0$$

を得る.

□

Corollary B.4.

$$\begin{aligned} p^k &= r^k + \sum_{l=1}^{k-1} \beta_l^k p^l && \left( \beta_l^k = -\frac{(r^k, Ap^l)}{(p^l, Ap^l)} \right) \\ &= r^k + \beta_{k-1}^k p^{k-1} \end{aligned}$$

## C MPIプログラミング

以降では, 分散メモリ型マシンでのコーディングとして, MPI の一種である MPICH を用いた方法を紹介する.

### C.1 並列計算の初期化

MPICH を用いて並列計算を行なう場合, 全ての MPICH のルーチンを呼び出す前に必ず `MPI_Init()` が呼び出されなければならない. MPICH の終了時には `MPI_Finalize()` が呼び出されなければならない.

また, 全ての CPU 数を確認する為のルーチン

```
MPI_Comm_size(comm, nprocs, ierror)
integer comm, nprocs, ierror
```

が用意されている. `comm` は, 通常 `MPI_COMM_WORLD` を指定しておけば良い.

他にも, 各 CPU の番号 (ランク) を確認する為のルーチン

```
MPI_Comm_rank(comm, myrank, ierror)
integer comm, myrank, ierror
```

が用意されている. こちらも通常, `comm` として `MPI_COMM_WORLD` を指定しておけば良い.

```
*****
```

```
c MPICH の初期化と終了のみを行なう
```

```
program initFinal
implicit none
include 'mpif.h'

integer nprocs, myrank, ierror

call MPI_Init(ierror)
call MPI_Comm_size(MPI_COMM_WORLD, nprocs, ierror)
call MPI_Comm_rank(MPI_COMM_WORLD, myrank, ierror)

call MPI_Finalize(ierror)
end
```

```
*****
```

## C.2 hello MPI world

実際に並列処理したい部分は、前小節で挙げたプログラム中の `MPI_Init()` と `MPI_Finalize()` の間に記述すれば良い。以下は、各 CPU が共通のコメント “hello MPI world” と全ノード数及びランクをを標準出力に出力するプログラムである。

```
*****
```

```
c hello world with MPICH
```

```
program hello
include 'mpif.h'

integer nprocs, myrank, ierror

call MPI_Init(ierror)
call MPI_Comm_size(MPI_COMM_WORLD, nprocs, ierror)
call MPI_Comm_rank(MPI_COMM_WORLD, myrank, ierror)

print *, "hello MPI + Fortan world!! [on the rank",
&      myrank, " /", nprocs, "]"
call MPI_Finalize(ierror)
```

end

\*\*\*\*\*

### C.3 1 対 1 通信

MPICH ではメッセージ通信の為に 100 以上のルーチンが用意されている。その中でも、1 対 1 のメッセージ通信を行う最も基本的なルーチン MPI\_Send() と MPI\_Recv() を以下に示す。

MPI\_Send(sendBuf, size, datatype, destRank, tag, comm, ierror)

|                  |  |
|------------------|--|
| ***** sendBuf    | 送るデータの先頭 (型は任意)                            |
| integer size     | 送るデータのサイズ                                  |
| integer datatype | 送るデータの型を表すマクロ<br>(MPI_DOUBLE_PRECISION など) |
| integer destRank | 送り先のランク                                    |
| integer tag      | メッセージタグ<br>(対応する MPI_Recv() の tag と同じ値)    |
| integer comm     | MPI_COMM_WORLD でよい                         |
| integer ierror   |  |

MPI\_Recv(recvBuf, size, datatype, srcRank, tag, comm, status, ierror)

|                                 |   |
|---------------------------------|---|
| ***** recvBuf                   | 受け取るデータを格納する先頭 (型は任意)                   |
| integer size                    | 受け取るデータのサイズ                             |
| integer datatype                | 受け取るデータの型を表すマクロ                         |
| integer srcRank                 | 送り主のランク                                 |
| integer tag                     | メッセージタグ<br>(対応する MPI_Send() の tag と同じ値) |
| integer status(MPI_STATUS_SIZE) |   |
| integer comm                    | MPI_COMM_WORLD でよい                      |

次の例では 2 CPU 間でデータ通信を行なっている。

\*\*\*\*\*

c send & receive data among 2 CPUs

```
program sendReceive02
include 'mpif.h'

integer nprocs, myrank, ierror
integer srcRank, dstRank, tag
```



```

integer send, recv, size
integer status(MPI_STATUS_SIZE)
call MPI_Init(ierr)
call MPI_Comm_size(MPI_COMM_WORLD, nprocs, ierr)
call MPI_Comm_rank(MPI_COMM_WORLD, myrank, ierr)

size = 1
tag = 1000
if (myrank .eq. 0) then
    srcRank = myrank + 1
    dstRank = myrank + 1
    send = myrank
    recv = myrank
    print *, " [on rank ", myrank, " before] send = ",
&         send, " recv = ", recv
    call MPI_Send(send, size, MPI_INTEGER, dstRank, tag,
&         MPI_COMM_WORLD, ierr)
    call MPI_Recv(recv, size, MPI_INTEGER, srcRank, tag,
&         MPI_COMM_WORLD, status, ierr)
    print *, " [on rank ", myrank, " after] send = ",
&         send, " recv = ", recv
elseif (myrank .eq. 1) then
    srcRank = myrank - 1
    dstRank = myrank - 1
    send = myrank
    recv = myrank
    print *, " [on rank ", myrank, " before] send = ",
&         send, " recv = ", recv
    call MPI_Recv(recv, size, MPI_INTEGER, srcRank, tag,
&         MPI_COMM_WORLD, status, ierr)
    call MPI_Send(send, size, MPI_INTEGER, dstRank, tag,
&         MPI_COMM_WORLD, ierr)
    print *, " [on rank ", myrank, " after] send = ",
&         send, " recv = ", recv
endif

call MPI_Finalize(ierr)
end

```

\*\*\*\*\*

## 参考文献

- [1] H. W. Alt - L. A. Caffarelli, “*Existence and regularity for a minimum problem with free boundary*”, J. Reine Angew. Math., **325** (1981), 105-144.
- [2] H. Berestycki - L. A. Caffarelli - L. Nirenberg, “*Uniform estimates for regularization of free boundary problems*, in Analysis and Partial Differential Equation, Marcel Dekker, New York, 1990.
- [3] H. Imai - K. Kikuchi - K. Nakane - S. Omata - T. Tachikawa, “*A numerical approach to the asymptotic behavior of solutions of a one-dimensional hyperbolic free boundary problems*”, JJIAM **18** No.1(2001) 43-58.
- [4] K. Kikuchi - S. Omata, “*A free boundary problem for a one dimensional hyperbolic equation*”, Adv. Math. Sci. Appl. **9** No.2 (1999) 775-786.
- [5] S. Omata, “*A Numerical treatment of film motion with freeboundary*”, Adv. Math. Sci. Appl., **14**, (2004) 129-137.
- [6] T. Nagasawa - K. Nakane - S. Omata, “*Numerical Computations for motion of vortices governed by a Hyperbolic Ginzburg-Landau System*”, Non-linear Anal. **51** (2002) No.1 Ser A: Theory Methods, 67-77.
- [7] 溝畑 茂      偏微分方程式論      岩波書店
- [8] 戸川 隼人      共役勾配法      教育出版