

# 計算物理学および演習

## 偏微分方程式の数値解法 1

畝山多加志

### 1 境界値問題と初期値問題

流体力学や電磁気学、量子力学では基礎方程式は場の量に対する偏微分方程式として表現される。このような対象を計算機で取り扱う場合には常微分方程式とは異なる手法が必要となる。

偏微分方程式の例として単純な 3 次元空間で、時間について 1 階、空間について 2 階の以下の拡散方程式を考える。

$$\frac{\partial u(\mathbf{r}, t)}{\partial t} = \nabla^2 u(\mathbf{r}, t) = \frac{\partial^2 u(\mathbf{r}, t)}{\partial x^2} + \frac{\partial^2 u(\mathbf{r}, t)}{\partial y^2} + \frac{\partial^2 u(\mathbf{r}, t)}{\partial z^2} \quad (1)$$

ここで  $\mathbf{r}, t$  は 3 次元空間中の位置および時間を表し、 $u(\mathbf{r}, t)$  は温度場や濃度場のような量である。時間・空間等については全て無次元化されているものとする。偏微分方程式に対しては初期条件と境界条件が必要である。与えられた初期条件・境界条件のもとで偏微分方程式を解いて  $u(\mathbf{r}, t)$  を求める、というのがここで考える問題である。

偏微分方程式を扱う際、時間発展ではなく十分時間が経過したあとの定常状態が重要となる場面がある。(拡散方程式であれば、時間変化よりもむしろ最終的にどのような場が実現されるかだけがわかればいい、という場面に相当する。) そのような場合、時間発展を解くのではなく、時間変化をゼロとした微分方程式を解けば十分である。例として挙げた拡散方程式の場合では  $\partial u(\mathbf{r}, t)/\partial t = 0$  を代入して得られる偏微分方程式 (Laplace 方程式)

$$\nabla^2 u(\mathbf{r}) = 0 \quad (2)$$

を与えられた境界条件のもとで解く、という問題を考えることになる。ただし、時間依存性を持たないため  $u(\mathbf{r}, t)$  を  $u(\mathbf{r})$  とした。Laplace 方程式は物理ではよく用いられる方程式である。類似した偏微分方程式として、Poisson 方程式もよく用いられる。

$$\nabla^2 u(\mathbf{r}) = \phi(\mathbf{r}) \quad (3)$$

ただし、 $\phi(\mathbf{r})$  は湧き出しに相当する場の量 (電磁気の場合には電荷分布) である。これらはほとんど同じ方程式なので、まとめて Laplace-Poisson 方程式などと呼ばれる。

このような問題は境界値問題と呼ばれる。一般に空間の次元が 2 以上の場合にはそれぞれの方向の微分の項が存在するため、時間に依存しなくてもやはり偏微分方程式として扱わねばならない。最初に示した拡散方程式のように時間変化をともなう問題は初期値問題とよばれ、境界値問題とは区別して考えることが多い。これは時間と空間の役割が異なるためである。通常、時間は  $t \geq 0$  の範囲で考え、どれだけ大きくできるが、空間は境界条件によって決められた特定の領域を考えるため、有限となっている。また、境界条件は円の上で値が一定、境界で法線方向の微分がゼロ、といったような形で与えられることもあり、 $t = 0$  の時点での値をするだけの初期条件とはずいぶん異なる。境界値問題と初期値問題ではそれぞれ問題の性質の違いを反映して異なる数値スキームが用いられる。今回は主に境界値問題のための数値スキームを考える。

## 2 空間の離散化と微分の差分化

### 2.1 離散化

これから偏微分方程式に対する各種数値解法を考えていくが、まず初めに計算機で偏微分方程式を取り扱うための離散化と微分の差分化から考える。

上記の 3 次元空間の問題より簡単な 1 次元空間の場合を考える。このとき、位置を  $x$  とすれば拡散方程式は

$$\frac{\partial u(x,t)}{\partial t} = \frac{\partial^2 u(x,t)}{\partial x^2} \quad (4)$$

とできる。また、 $0 \leq x \leq L$  の領域のみを考えるものとし、初期条件と境界条件がそれぞれ

$$u(x,0) = v(x), \quad u(0,t) = w_0, \quad u(L,t) = w_L \quad (5)$$

で与えられているとする。

常微分方程式を扱う際に、計算機では無限小や連続を扱えないために時間を離散化し、ある時間刻みで与えられる離散時間のみについて考えた。偏微分方程式の場合にも同様に考え、時間と空間をともに離散化する。

$$t_i = hi \quad (i = 0, 1, 2, \dots) \quad (6)$$

$$x_j = \Delta j \quad (j = 0, 1, 2, \dots, N) \quad (7)$$

ここで、 $h$  は時間刻み、 $\Delta$  は空間刻みを表す。また、空間については  $0 \leq x \leq L$  であるから、空間の刻みは  $N$  番目の離散化点が

$$x_N = \Delta N = L \quad (8)$$

となるように取っておくことにする。 $\Delta, h$  がともに十分小さい量であれば、連続的な時間と空間を十分によく表現できるはずである。離散化した時間と空間を使うと場は

$$u_{j,i} = u(x_j, t_i) \quad (i = 0, 1, 2, \dots, \quad j = 0, 1, 2, \dots, N) \quad (9)$$

のように離散化できる。あとは、何らかの形で時間と空間についての偏微分を差分で近似してやれば偏微分方程式を計算機で扱える形へと変形でき、数値的に解くことができるようになる。

### 2.2 差分化

時間については基本的に常微分方程式の場合とあまり変わらないとみなして Euler 法等を用いて差分化することが考えられる。すなわち、もとの偏微分方程式の左辺は

$$\left. \frac{\partial u(x,t)}{\partial t} \right|_{x_j, t_i} \approx \frac{u_{j,i+1} - u_{j,i}}{h} \quad (10)$$

のような形で近似できるはずである。また、このような差分スキームの近似の精度は Taylor 展開から得られる。より高次の差分スキームは時間についての Taylor 展開の高次項を取り入れることで得られる。(このあたりは基本的には常微分方程式の際の考え方と同じである。)

空間についての偏微分は 2 階となっている。常微分方程式の場合と同様に考えれば、新しい変数を導入して 1 階の偏微分に変換し、その後でそれぞれを Euler 法的に扱えばよさそうである。つまり、1 階の偏微分を

$$\left. \frac{\partial u(x,t)}{\partial x} \right|_{x_j, t_i} \approx \frac{u_{j+1,i} - u_{j,i}}{\Delta} \quad (11)$$

のように近似することが考えられる (図 1(a))。しかしながら、空間については時間と異なり境界 ( $j = 0$  と  $j = N$ ) が決まっている。片方の境界だけが特別ということではなく、正方向と負方向もどちらかが特別ということはない。そのため、偏微分に対する

$$\left. \frac{\partial u(x,t)}{\partial x} \right|_{x_j, t_i} \approx \frac{u_{j,i} - u_{j-1,i}}{\Delta} \quad (12)$$

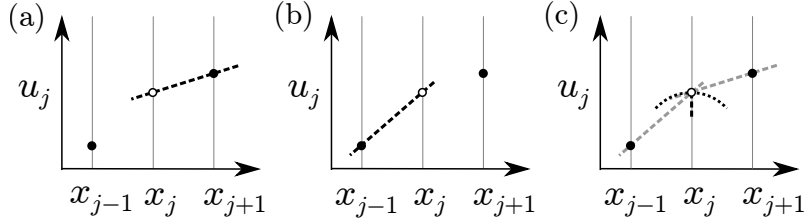


図 1: 関数  $u(x)$  の 2 階微分の差分化。(a) (b)  $x$  について正と負の方向で 1 階微分を差分化する。(c) (a),(b) の 1 階微分を組み合わせて、考えている点  $x_j$  について対称な 2 階微分の差分とする (中心差分)。

という近似でもよさそうである (図 1(b))。安直にはこれらの差分を 2 回施せば 2 階の偏微分の近似とできるはずである。しかし、どちらの近似も一方向に偏ったような形となっている。対称性の観点から考えると、常微分方程式の Euler 法のような考え方は好ましくないとと言える。

そのため、空間については陽的な差分や陰的な差分の代わりに別の形の差分を考える。最も標準的に用いられる中心差分と呼ばれる差分では

$$\left. \frac{\partial^2 u(x, t)}{\partial x^2} \right|_{x_j, t_i} \approx \frac{u_{j+1, i} - 2u_{j, i} + u_{j-1, i}}{\Delta^2} \quad (13)$$

という形で 2 階の偏微分を近似する (図 1(c))。これは  $\Delta \rightarrow 0$  の極限で 2 階微分に帰着する。この中心差分では  $j$  に対して  $j+1$  と  $j-1$  が対称な形となっており、空間に特殊な方向がないという性質からは好ましい形であると言える。また、

$$\frac{u_{j+1, i} - 2u_{j, i} + u_{j-1, i}}{\Delta^2} = \frac{1}{\Delta} \left( \frac{u_{j+1, i} - u_{j, i}}{\Delta} - \frac{u_{j, i} - u_{j-1, i}}{\Delta} \right) \quad (14)$$

と変形できることから、差分を 2 回取った形であり、先ほど考えた正方向と負方向に偏った差分を組み合わせて差分を取った形となっていることがわかる。

Taylor 展開を考えれば

$$u_{j\pm 1, i} = u_{j, i} \pm \Delta \left. \frac{\partial u}{\partial x} \right|_{x_j, t_i} + \frac{\Delta^2}{2} \left. \frac{\partial^2 u}{\partial x^2} \right|_{x_j, t_i} \pm \frac{\Delta^3}{6} \left. \frac{\partial^3 u}{\partial x^3} \right|_{x_j, t_i} + O(\Delta^4) \quad (15)$$

より、

$$u_{j+1, i} + u_{j-1, i} = 2u_{j, i} + \Delta^2 \left. \frac{\partial^2 u}{\partial x^2} \right|_{x_j, t_i} + O(\Delta^4) \quad (16)$$

とできる。これを变形すれば

$$\left. \frac{\partial^2 u}{\partial x^2} \right|_{x_j, t_i} = \frac{u_{j+1, i} - 2u_{j, i} + u_{j-1, i}}{\Delta^2} + O(\Delta^2) \quad (17)$$

となる。 $O(\Delta^2)$  の項を無視すると中心差分が得られる。

以上の時間と空間についての差分化を組み合わせれば偏微分方程式を差分のみで表現することができる。常微分方程式の場合と同様、Taylor 展開の高次項をどこまで取り入れられるか、物理的な条件をうまく満たせるか等が異なる種々の数値スキームが存在する。

### 3 1 次元境界値問題

境界値問題の数値スキームについて考える。まず、最も簡単な 1 次元空間内の Laplace-Poisson 方程式の場合を考える。このとき、Laplace-Poisson 方程式

$$\frac{\partial^2 u(x)}{\partial x^2} = \phi(x) \quad (18)$$

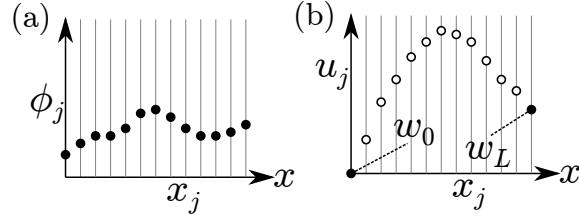


図 2: 1次元境界値問題の離散化のイメージ。(a) 関数  $\phi(x)$  を離散化した  $\phi_j$  が与えられている。(b) 関数  $u(x)$  を離散化した  $u_j$  に対して、境界  $x = 0, L$  における値  $w_0, w_L$  が与えられている。残りの部分の  $u_j$  (白い丸) を  $w_0, w_L$  と  $\phi_j$  (黒い丸) から求める。

を境界条件

$$u(0) = w_0, \quad u(L) = w_L \quad (19)$$

のもとで解けば必要な情報が得られることになる。

1次元空間を考える場合には変数は1つしかないため、偏微分方程式を常微分方程式に置き換えることが可能である。そのため、一見常微分方程式と同様に Euler 法等のスキームで解けるように見えるかもしれない。しかし、常微分方程式を解く場合と違い、境界条件が  $x = 0$  と  $x = L$  という異なる2点で指定されているため、 $x = 0$  の値からスタートして  $x$  を  $\Delta$  ずつ大きくする、という解き方を直接行うことはできない。

### 3.1 直接法

既に述べたように、空間に付いての微分は対称性のよい差分で近似するのがよいと考えられる。 $u(x), \phi(x)$  を離散化し、偏微分を中心差分で置き換える ( $O(\Delta^2)$  の項を無視する) と

$$\frac{u_{j+1} - 2u_j + u_{j-1}}{\Delta^2} = \phi_j \quad (20)$$

となる。境界条件は

$$u_0 = w_0, \quad u_N = w_L \quad (21)$$

である (図 2)。

離散化した式を変形して扱いやすい形にできないか考える。まず、両辺に  $\Delta^2$  をかけてやれば

$$u_{j+1} - 2u_j + u_{j-1} = \Delta^2 \phi_j \quad (22)$$

この式は境界の  $j = 0, N$  を除いた  $j = 1, 2, \dots, N-1$  に対して成立するので

$$\begin{aligned} u_2 - 2u_1 + u_0 &= \Delta^2 \phi_1 \\ u_3 - 2u_2 + u_1 &= \Delta^2 \phi_2 \\ u_4 - 2u_3 + u_2 &= \Delta^2 \phi_3 \\ &\vdots \\ u_{N-1} - 2u_{N-2} + u_{N-3} &= \Delta^2 \phi_{N-2} \\ u_N - 2u_{N-1} + u_{N-2} &= \Delta^2 \phi_{N-1} \end{aligned} \quad (23)$$

となる。 $N-1$  個の式があるが、行列を使ってまとめてしまえば

$$\begin{bmatrix} -2 & 1 & 0 & 0 & \dots & 0 \\ 1 & -2 & 1 & 0 & \dots & 0 \\ 0 & 1 & -2 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & 0 & \dots & -2 \end{bmatrix} \cdot \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_{N-2} \\ u_{N-1} \end{bmatrix} + \begin{bmatrix} w_0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ w_L \end{bmatrix} = \begin{bmatrix} \Delta^2 \phi_1 \\ \Delta^2 \phi_2 \\ \Delta^2 \phi_3 \\ \vdots \\ \Delta^2 \phi_{N-2} \\ \Delta^2 \phi_{N-1} \end{bmatrix} \quad (24)$$

のように書ける。さらに、

$$A = \begin{bmatrix} -2 & 1 & 0 & 0 & \dots & 0 \\ 1 & -2 & 1 & 0 & \dots & 0 \\ 0 & 1 & -2 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & 0 & \dots & -2 \end{bmatrix}, \quad u = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_{N-2} \\ u_{N-1} \end{bmatrix}, \quad b = \begin{bmatrix} \Delta^2 \phi_1 - w_0 \\ \Delta^2 \phi_2 \\ \Delta^2 \phi_3 \\ \vdots \\ \Delta^2 \phi_{N-2} \\ \Delta^2 \phi_{N-1} - w_L \end{bmatrix} \quad (25)$$

としてやれば Laplace-Poisson 方程式は最終的に

$$A \cdot u = b \quad (26)$$

という単純な形に帰着する。 $A$  はもともと微分演算子を行列の形で表現したものとみなすことができ、Laplacian 行列などと呼ばれることもある。

得られた方程式の解は行列  $A$  の逆行列  $A^{-1}$  を使えば

$$u = A^{-1} \cdot b \quad (27)$$

である。あとは、 $A^{-1} \cdot b$  を数値的に計算すれば解が得られる。このように解を直接的に求めるスキームは直接法と呼ばれる。 $A^{-1}$  を求めるだけだから簡単のように見えるかもしれないが、 $A$  は  $(N-1) \times (N-1)$  の行列であり、一般にこのようなサイズの大きな逆行列の数値的な計算は計算量が大きくなることが知られている。幸い、逆行列ではなく  $A^{-1} \cdot b$  を計算する場合はあまり計算量が大きくならないので、直接法では  $A^{-1} \cdot b$  を計算する。

一般的な行列に対して数値的に  $A^{-1} \cdot b$  を計算する方法はあるのだが<sup>1</sup>、いま考えている行列  $A$  は単純な構造をしているため、行列  $A$  の性質をうまく使うと  $A^{-1} \cdot b$  を簡単に求めることができる。行列  $A$  は対角部分とその上下のみに要素を持つ、いわゆる三重対角行列である。三重対角行列  $A$  に対して  $A^{-1} \cdot b$  は Gauss の消去法を用いれば比較的簡単に解くことができる。数値的に  $A^{-1} \cdot b$  を求める方法として Thomas 法がある。 $A \cdot u = b$  の 1 行目を  $1/2$  倍して 2 行目に足す。

$$\begin{bmatrix} -2 & 1 & 0 & \dots & 0 \\ 0 & -2 + 1/2 & -1 & \dots & 0 \\ 0 & 1 & -2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & -2 \end{bmatrix} \cdot \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_{N-1} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 + b_1/2 \\ b_3 \\ \vdots \\ b_{N-1} \end{bmatrix} \quad (28)$$

同様に 2 行目を  $2/3$  倍して 3 行目に足す。

$$\begin{bmatrix} -2 & 1 & 0 & \dots & 0 \\ 0 & -2 + 1/2 & -1 & \dots & 0 \\ 0 & 0 & -2 + 2/3 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & -2 \end{bmatrix} \cdot \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_{N-1} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 + b_1/2 \\ b_3 + (2/3)(b_2 + b_1/2) \\ \vdots \\ b_{N-1} \end{bmatrix} \quad (29)$$

このような手順を繰り返せば対角より下側は全てゼロとなる。具体的な手順としては対角要素  $d'_j$  とベクトル  $b'_j$  について  $d'_1 = -2$ ,  $b'_1 = b_1$  および  $j = 2, 3, \dots, N-1$  に対して

$$d'_j = -2 - 1/d'_{j-1}, \quad b'_j = b_j - b'_{j-1}/d'_{j-1} \quad (30)$$

<sup>1</sup>一般には行列  $A$  の LU 分解や Cholesky 分解 (対称行列のとき) を使う。これらの分解はもとの行列を対角成分の下側の要素がゼロの行列と対角要素の上側がゼロの行列の積に分解する。分解した行列を使って  $A^{-1} \cdot b$  を数値的に求める。

として、

$$\begin{bmatrix} d'_1 & 1 & 0 & \dots & 0 \\ 0 & d'_2 & -1 & \dots & 0 \\ 0 & 0 & d'_3 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & d'_{N-1} \end{bmatrix} \cdot \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_{N-1} \end{bmatrix} = \begin{bmatrix} b'_1 \\ b'_2 \\ b'_3 \\ \vdots \\ b'_{N-1} \end{bmatrix} \quad (31)$$

とする。この手順は前進消去と呼ばれる。左辺の行列は対角成分の下側を持たないため、下から順に  $u_j$  を計算できる。 $N-1$  行目を  $1/d'_{N-1}$  倍すれば

$$\begin{bmatrix} d'_1 & \dots & 0 & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \dots & d'_{N-3} & 1 & 0 \\ 0 & \dots & 0 & d'_{N-2} & 1 \\ 0 & \dots & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} u_1 \\ \vdots \\ u_{N-3} \\ u_{N-2} \\ u_{N-1} \end{bmatrix} = \begin{bmatrix} b'_1 \\ \vdots \\ b'_{N-3} \\ b'_{N-2} \\ b'_{N-1}/d'_{N-1} \end{bmatrix} \quad (32)$$

続けて  $N-2$  行目から  $N-1$  行目を引いて  $1/d'_{N-2}$  倍すれば

$$\begin{bmatrix} d'_1 & \dots & 0 & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \dots & d'_{N-3} & 1 & 0 \\ 0 & \dots & 0 & 1 & 0 \\ 0 & \dots & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} u_1 \\ \vdots \\ u_{N-3} \\ u_{N-2} \\ u_{N-1} \end{bmatrix} = \begin{bmatrix} b'_1 \\ \vdots \\ b'_{N-3} \\ (b'_{N-2} - b'_{N-1}/d'_{N-1})/d'_{N-2} \\ b'_{N-1}/d'_{N-1} \end{bmatrix} \quad (33)$$

のようになり、これを繰り返せば対角要素をすべて 1、他を 0 にできる。解は  $u_{N-1} = b'_{N-1}/d'_{N-1}$  および  $j = 1, 2, \dots, N-2$  に対して

$$u_j = (b'_j - u_{j+1})/d'_j \quad (34)$$

となる。この手続きは後退代入と呼ばれる。

### 3.2 反復法

前節の Thomas 法で直接的に  $A^{-1} \cdot b$  を求めるのはあまり効率的な方法ではないこともある。直接法ではない数値スキームとして、何らかの繰り返し処理を行って逐次的に  $A^{-1} \cdot b$  の近似解を求め、十分な近似精度に達したら数値的な解とする方法がある。このような方法は反復法と呼ばれる。

$u_j$  にある適当な初期値が設定されたとする。この初期値は差分した偏微分方程式を満たさないで、当然ながら求める解ではない。値を少し変えて偏微分方程式を満たすような  $u_j$  に近づけることができれば、同様の処理を繰り返していくことでやがて解に収束すると考えられる。満たすべき方程式

$$u_{j+1} - 2u_j + u_{j-1} = \Delta^2 \phi_j \quad (35)$$

を少し変形すると

$$u_j = \frac{u_{j+1} + u_{j-1} - \Delta^2 \phi_j}{2} \quad (36)$$

とできる。 $u_j$  を解に近い新しい値  $u_j^*$  に更新する方法として最も単純なものは

$$u_j^* = \frac{u_{j+1} + u_{j-1} - \Delta^2 \phi_j}{2} \quad (37)$$

とすることである<sup>2</sup>。 $u_j$  が解になっていれば更新後の  $u_j^*$  は  $u_j$  と一致する。解になっていなければより解に近い値となる。このような数値スキームは Jacobi 法と呼ばれる。図 3 に

<sup>2</sup>この更新は後に扱う初期値問題で可能な限り長い時間刻み  $h$  を設定した場合の時間発展と同じ形となっている。時間が十分経てば定常状態が実現されるはずなので、その意味では最も単純かつ自然な更新ということになる。

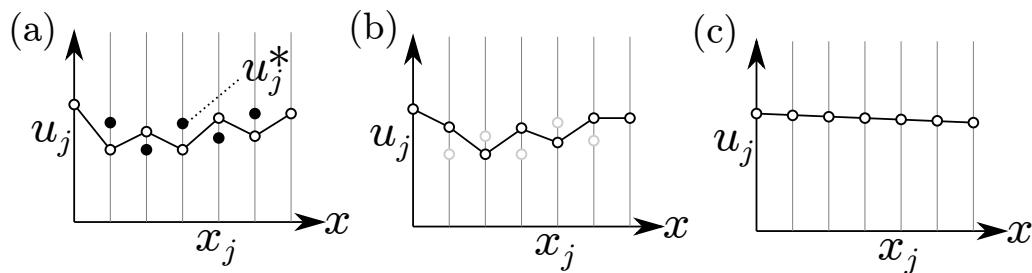


図 3: Jacobi 法による境界値問題の解法のイメージ。(a) ある  $u_j$  (白い丸) が与えられたとして、更新後の値  $u_j^*$  (黒い丸) を計算する。(b)  $u_j$  の値を更新すると、もとの値よりも解に近くなる。(c) 更新を繰り返せばやがて求める解となる。

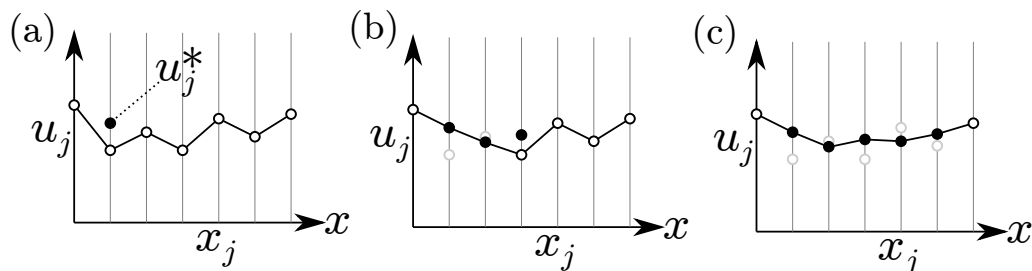


図 4: Gauss-Seidel 法による境界値問題の解法のイメージ。(a) まず 1 点について  $u_j$  (白い丸) から更新後の値  $u_j^*$  (黒い丸) を計算し更新する。(b) 更新した値を使いつつ、 $u_j$  の値を更新を続けていく。(c) 全ての点について 1 回ずつ更新が終わったところ。Jacobi 法で 1 回更新した結果より解に近い。

Jacobi 法のイメージを示す。計算機では極限や無限大を扱うことはできないので、誤差に対して許容できる値の閾値  $\epsilon$  を与え、誤差が許容限界内に収まるまで、すなわち

$$\max_j |u_j^* - u_j| \leq \epsilon \quad (38)$$

を満たすまで更新を繰り返す。

緩和法の精度は与えた誤差の許容限界で決まる。解を得るために必要な繰り返し計算の回数は離散化点数  $N$  と誤差の許容限界に依存し、 $N$  が大きい場合に Jacobi 法では必要な繰り返しの回数が非常に大きくなり、効率的ではないことがわかっている。つまり、Jacobi 法は単純であるが、効率の面から考えるとあまりよくない。より効率のいい方法として Gauss-Seidel 法がある。Gauss-Seidel 法は値の更新を

$$u_j^* = \frac{u_{j+1} + u_{j-1}^* - \Delta^2 \phi_j}{2} \quad (39)$$

のように行う。Jacobi 法によく似ているが、Jacobi 法の右辺の  $u_{j-1}$  が Gauss-Seidel 法では  $u_{j-1}^*$  に変わっている。Jacobi 法ではすべての  $j$  に対して  $u_j$  から  $u_j^*$  を計算しているが、Gauss-Seidel 法では  $j$  に対して  $u_j^*$  を求める際に既に更新をすませた値  $u_k^*$  ( $k = 0, 1, \dots, j-1$ ) を使っている (図 4)。Jacobi 法のほうが対称性が高く適切なように思えるかもしれないが、Gauss-Seidel 法では更新がかかった値を次々と次の更新に反映させるため、より解に近づきやすくなっており、結果として必要な繰り返し回数が少なくなることがわかっている。境界値問題ではあくまでも偏微分方程式を満たす解が得られればいので、解に到達するまでの手順は効率的でさえあれば対称性は低くても構わないのである。

Gauss-Seidel 法よりさらに効率が良く、繰り返し回数が少なくなるスキームとして逐次加速緩和 (successive overrelaxation, SOR) 法と呼ばれるスキームがある。Gauss-Seidel 法は既に更新された値を使っているので Jacobi 法より効率が上がっているが、しかしそれでも更新自体は何度も繰り返す必要がある。何度も似たような更新を繰り返すのであれば、1 回の更新で少し多めに緩和をさせてしまえばいい、というのが SOR 法の基本的な考え方で

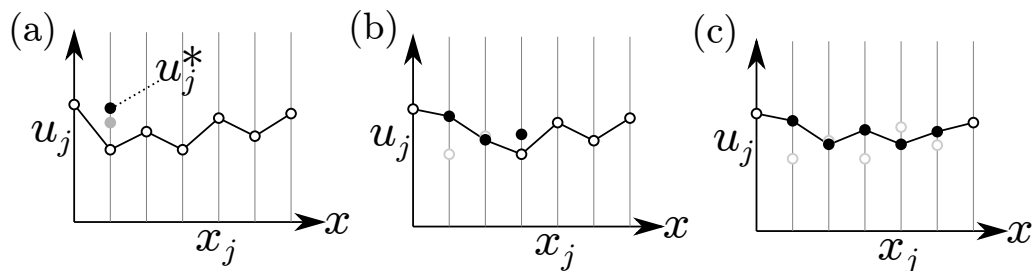


図 5: SOR 法による境界値問題の解法のイメージ。(a) Gauss-Seidel 法と同様に 1 点について  $u_j$  (白い丸) から更新後の値  $u_j^*$  (黒い丸) を計算し更新する。Gauss-Seidel 法より少し更新による変化が大きい。(b) 更新した値を使いつつ、 $u_j$  の値を更新を続けていく。(c) 全ての点について 1 回ずつ更新が終わったところ。Gauss-Seidel 法と比べて過剰な更新が行われているため、効率が良くなっている。

ある。SOR 法では緩和パラメータ  $\omega$  を導入し、更新を

$$u_j^* = \omega \frac{u_{j+1} + u_{j-1}^* - \Delta^2 \phi_j}{2} + (1 - \omega)u_j \quad (40)$$

のように行う。この更新は  $0 < \omega < 2$  の範囲でうまく働き、 $\omega = 1$  のとき Gauss-Seidel 法になる。 $1 < \omega < 2$  のとき、更新される値は Gauss-Seidel 法と比べると緩和が過剰に行われたものとなる。適切な  $\omega$  の値が選ばれれば Gauss-Seidel 法よりも少ない回数の繰り返しの効率的に解が得られる (図 5)。ただし、 $\omega$  が大きすぎると逆に効率は悪くなってしまふ。 $\omega$  の最適値は理論的に求めることができるのだが、経験的に 1 より少し大きい値 (例えば 1.2 ~ 1.4 くらい) を入れて、それなりにうまく動けばそれでよしとされることが多い。SOR 法はスキームが単純なのでプログラムを簡単に作ることができ、それなりに良い効率で動くのが大きな利点である<sup>3</sup>。

<sup>3</sup>SOR 法より効率がいい緩和法が色々開発されているのだが、どうしてもスキームがある程度以上に複雑になってしまう。複雑になるのでここでは扱わないが、前処理付き共役勾配法と呼ばれる方法は非常に効率がよく、大規模な問題を解く際によく用いられている。