

9 境界条件のある場合の微分方程式の解法

微分方程式に対して Euler 法や Runge-Kutta 法は、初期条件 (座標や時刻の一方の端点) と微分が与えられた問題に対して、微分方程式を積分して (変化を追いかけて) 解を求める方法である。これに対して、電場の境界値問題や、量子力学における束縛状態の問題等、座標の両方の端点に制限がある場合には別の方法が適している場合がある。

本章では、その中で広く使われている Relaxation 法について説明する。

9.1 境界値問題に対する差分方程式

簡単のため、次の 2 階常微分方程式

$$\frac{d^2}{dx^2}\phi(x) + g(x) = 0 \quad (9.1)$$

を考える。解 $\phi(x)$ は区間 $[x_0, x_N]$ で定義され、境界条件は、

$$\phi(x_0) = \phi_0, \quad \phi(x_N) = \phi_N \quad (9.2)$$

で与えられているものとする。Euler 法で説明した通り、1 階微分方程式で、1 階微分を差分式で置き換えるのは良くない近似である。しかしながら、2 階微分のみが現われる場合には、以下で見るように 2 階微分を差分式で置き換えるのは悪くない近似である。

区間 $[x_0, x_N]$ が間隔 h で N 個に分割されているとする。 $x_{i\pm 1} = x_i \pm h$ での値は、

$$\phi(x_{i\pm 1}) = \phi(x_i) \pm h \frac{d}{dx}\phi(x_i) + \frac{h^2}{2} \frac{d^2}{dx^2}\phi(x_i) \pm \frac{h^3}{6} \frac{d^3}{dx^3}\phi(x_i) + \mathcal{O}(h^4) \quad (9.3)$$

であるので、2 階微分は差分式、

$$\frac{d^2}{dx^2}\phi(x_i) \approx \frac{\phi(x_{i+1}) - 2\phi(x_i) + \phi(x_{i-1}))}{h^2} \quad (9.4)$$

で近似する事ができる。ここで、Taylor 展開の奇数回微分の符号が \pm であるので、この近似式は h^4 以上の項を無視した事に等しい (Euler 法の場合は、 h^2 以上の項を無視した) 事に注意せよ。従って式 (9.1) に対応する差分方程式は、

$$\phi(x_{i+1}) - 2\phi(x_i) + \phi(x_{i-1}) + h^2 g(x_i) = 0 \quad (9.5)$$

となる。

ここで境界条件から、 i の取り得る値は 1 から $N-1$ の範囲であり、 $i=0$ および N に対しては境界条件が適用される。これらの事をまとめて行列の形で表現すると、

$$\begin{pmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ -1 & 2 & -1 & 0 & \cdots & 0 \\ 0 & -1 & 2 & -1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & -1 & 2 & -1 \\ 0 & \cdots & \cdots & \cdots & 0 & 1 \end{pmatrix} \begin{pmatrix} \phi(x_0) \\ \phi(x_1) \\ \phi(x_2) \\ \vdots \\ \phi(x_{N-1}) \\ \phi(x_N) \end{pmatrix} = \begin{pmatrix} \phi_0 \\ h^2 g(x_1) \\ h^2 g(x_2) \\ \vdots \\ h^2 g(x_{N-1}) \\ \phi_N \end{pmatrix} \quad (9.6)$$

となる。後の便宜のため、式 (9.6) を

$$AY = b \quad (9.7)$$

と記号的に表し、各々の要素を A_{ij} , Y_i , b_j で参照する。

さて、問題が行列の演算に帰着したので、 A の逆行列 A^{-1} を計算して、

$$Y = A^{-1}b \quad (9.8)$$

から Y 、すなわち $\phi(x_i)$ を求めればよいと思うかも知れない。しかしながら A は見れば分かる通り、対角成分とその両端以外の要素が 0 という希薄な行列 (3 重対角行列) である¹。このような行列を逆行列を後述の消去法などにより数値計算することは無駄が多い。特に N が大きな場合は、計算機の演算速度やメモリ容量によっては、本来解けるはずの問題が解けなくなるといった事態も生じる。このような問題を扱うには、繰り返しによる逐次近似的解法が効率的であり、Relaxation 法は実用的に広く使われている方法の 1 つである。

9.2 Relaxation 法

詳細は他の数値計算の教科書に譲るが、Relaxation 法の解法の一般形は、 k 番目の近似解 $Y^{(k)}$ に対して、より精度の高い解 $Y^{(k+1)}$ を

$$Y_i^{(k+1)} = Y_i^{(k)} + \frac{\omega}{A_{ii}} \left[b_i - \sum_{j=1}^{i-1} A_{ij} Y_j^{(k+1)} - \sum_{j=i}^N A_{ij} Y_j^{(k)} \right] \quad (9.9)$$

によって逐次求めて行くというものである²。右辺第二項は、近似解 $Y_i^{(k)}$ を真の解に近付けるための補正とみなせる。その係数 ω は加速パラメータと呼ばれ、問題に応じて適切な値を選ぶ事により解の収束を早めるために使われる。例えば、 $Y_i^{(k)}$ に対する補正が常に同符号、すなわち一方からのみ真の解に漸近的に近づく場合は、 $1 < \omega < 2$ とすると収束の速さが加速される。

特に $\omega = 1$ の時は、式 (9.9) は Gauss-Seidel 法と呼ばれる有名な行列の解法に帰着する。問題の性質が良く分からない場合は、取り合えず $\omega = 1$ とするのが無難である。

今考えている問題では、行列は 3 重対角行列であるので、式 (9.9) に対応して用いるべき式は、

$$\phi^{(k+1)}(x_i) = \phi^{(k)}(x_i) + \frac{\omega}{2} \left[h^2 g(x_i) + \phi^{(k+1)}(x_{i-1}) + \phi^{(k)}(x_{i+1}) - 2\phi^{(k)}(x_i) \right] \quad (9.10)$$

と簡単化される (ただし、 $i = 1, \dots, N-1$)。

9.3 静電場の境界値問題

時間に依存しない Maxwell の方程式は

$$\begin{aligned} \nabla \cdot \mathbf{E} &= \rho(\mathbf{x}) \\ \nabla \times \mathbf{E} &= 0 \end{aligned} \quad (9.11)$$

¹一般に 0 の多い行列を疎行列という。

² A の対角成分 A_{ii} が 0 にならないように、あらかじめ行を入れ換える事は一般には可能である。

与えられる (簡単のため、 ϵ を省略する)。第二式、すなわち「渦なし」の条件から、電場は静電ポテンシャル $\phi(x)$ の勾配によって

$$\mathbf{E} = -\nabla\phi(\mathbf{x}) \quad (9.12)$$

と表す事ができる。これを第一式に代入すると、Laplace-Poisson 方程式

$$\nabla^2\phi(\mathbf{x}) = -\rho(\mathbf{x}) \quad (9.13)$$

を得る。

簡単のため二次元の場合を考えると、方程式は

$$\frac{\partial^2}{\partial x^2}\phi + \frac{\partial^2}{\partial y^2}\phi = -\rho(x, y) \quad (9.14)$$

となる。ここで、一辺の長さが h の正方形格子を考え、 x 座標の添字を i 、 y 座標の添字を j とすると、

$$\begin{aligned} \frac{\partial^2}{\partial x^2}\phi &\approx \frac{\phi(x_{i+1}, y_j) - 2\phi(x_i, y_j) + \phi(x_{i-1}, y_j))}{h^2} \\ \frac{\partial^2}{\partial y^2}\phi &\approx \frac{\phi(x_i, y_{j+1}) - 2\phi(x_i, y_j) + \phi(x_i, y_{j-1}))}{h^2} \end{aligned} \quad (9.15)$$

と差分で近似することができるので、最終的に差分方程式

$$\phi(x_{i+1}, y_j) + \phi(x_{i-1}, y_j) + \phi(x_i, y_{j+1}) + \phi(x_i, y_{j-1}) - 4\phi(x_i, y_j) = -h^2\rho(x_i, y_j) \quad (9.16)$$

を得る。特に電荷が存在しない ($\rho = 0$) の場合は、

$$\phi(x_i, y_j) = \frac{\phi(x_{i+1}, y_j) + \phi(x_{i-1}, y_j) + \phi(x_i, y_{j+1}) + \phi(x_i, y_{j-1}))}{4} \quad (9.17)$$

となり、各格子点での値が、その周辺の4点のポテンシャル値の平均になる。

Relaxation 法の漸化式は

$$\begin{aligned} \phi^{(k+1)}(x_i, y_j) &= \phi^{(k)}(x_i, y_j) \\ &+ \omega \left[\frac{\phi^{(k)}(x_{i+1}, y_j) + \phi^{(k)}(x_{i-1}, y_j) + \phi^{(k)}(x_i, y_{j+1}) + \phi^{(k)}(x_i, y_{j-1}))}{4} \right. \\ &\quad \left. - \phi^{(k)}(x_i, y_j) + \frac{h^2}{4}\rho(x_i, y_j) \right] \end{aligned} \quad (9.18)$$

とまとめられる。

9.4 四角い境界の例

練習

原点 $(x, y) = (0, 0)$ にある点電荷 q 、すなわち電荷分布

$$\rho(x, y) = \begin{cases} q & x = y = 0 \\ 0 & \text{それ以外} \end{cases} \quad (9.19)$$

がつくり出す静電場のポテンシャルを、境界条件

$$\phi(\pm 1, \pm 1) = 0 \quad \text{復号任意} \quad (9.20)$$

の下で求めよ。

9.5 サンプルプログラム

サンプルプログラムは以下の通り。計算精度を考えて、実数は倍精度を用いている。

```

program laplace
  implicit none
c constants:
  integer N,LOOP
  parameter(N = 50)           ! 区間の分割数
  parameter(LOOP = 200)      ! Relaxation 法繰り返し数
  real*8 H,W
  parameter(H = 1.0D0/N)     ! 刻み幅 (領域は [-0.5,0.5])
  parameter(W = 1.8D0)      ! 加速パラメータ
  integer LUNOUT,ERR
  parameter(LUNOUT=11,ERR=-1) ! 出力機番及びエラー識別子
c local:
  real*8 rho(0:N,0:N)       ! 2次元電荷分布
  real*8 phi(0:N,0:N)       ! 2次元ポテンシャル
  real*8 corr,cmax          ! 補正值とその最大値
  integer status            ! ファイル操作の状態
  integer i,j,k
c begin:
  do i=0,N
    do j=0,N
      rho(i,j)=0.0D0
      phi(i,j)=0.0D0
    end do
  end do
  rho(N/2,N/2) = 1.0D4 * H**2 ! 原点 (N/2,N,2) に点電荷*H^2
c Relaxation 法によるポテンシャル計算
  do k=0,LOOP
    cmax=0.0D0
    do i=1,N-1
      do j=1,N-1
        corr = w*(
&          (phi(i-1,j)+phi(i,j-1)+phi(i+1,j)+phi(i,j+1))/4.0D0
&          -phi(i,j)+rho(i,j)/4.0D0)
        if (abs(corr).gt.cmax) cmax = abs(corr)
        phi(i,j) = phi(i,j) + corr
      end do
    end do
    if (mod(k,10).eq.0) then
      write(*,'(a,I3,a,E10.3)') ! 収束の度合を確認

```

```

&          'k = ',k,', Correction = ',cmax
      endif
    end do
c 計算結果をファイルに出力
1  continue
  call file_open('output',LUNOUT,status)
  if (status.eq.ERR) goto 1
  do j=0,N
    write(LUNOUT,'(2F7.3,E12.3)')
&      (-0.5D0+i*H,-0.5D0+j*H,phi(i,j),i=0,N)
    write(LUNOUT,*) ! 結果を gnuplot の splot で読み込む為に必要な空行
  end do
  stop
end

```

9.6 実行と結果

サンプルプログラムは、

```
/home/teacher/z6wt01in/SAMPLE/laplace.f
```

として置いてある。実行ファイル laplace を

```
f77 -o laplace laplace.f file_open.f
```

により生成し、実行すると、

```

% laplace ↵
k =   0, Correction = 0.180e+01
k =  10, Correction = 0.154e+00
k =  20, Correction = 0.386e-01
...
k = 200, Correction = 0.156e-04
file name : laplace.dat ↵

```

と k が 10 毎での補正の値が表示され、収束の度合を確認する事ができる。最後に計算結果を出力するファイル名を聞いてくるので、適当なファイル名を入力すればよい。

二次元のデータを鳥瞰図や等高線で表現するには gnuplot を使うのが便利である。計算結果が laplace.dat というファイルに保存されているとする。以下のように gnuplot を起動し、splot コマンドを使うと図 9.1 のようにデータが鳥瞰図で表示される。

```

% gnuplot ↵
...(色々なメッセージが表示される)
gnuplot> set view 60,40,1,2 ↵
gnuplot> set pm3d at b ↵
gnuplot> splot "laplace.dat" with lines ↵

```

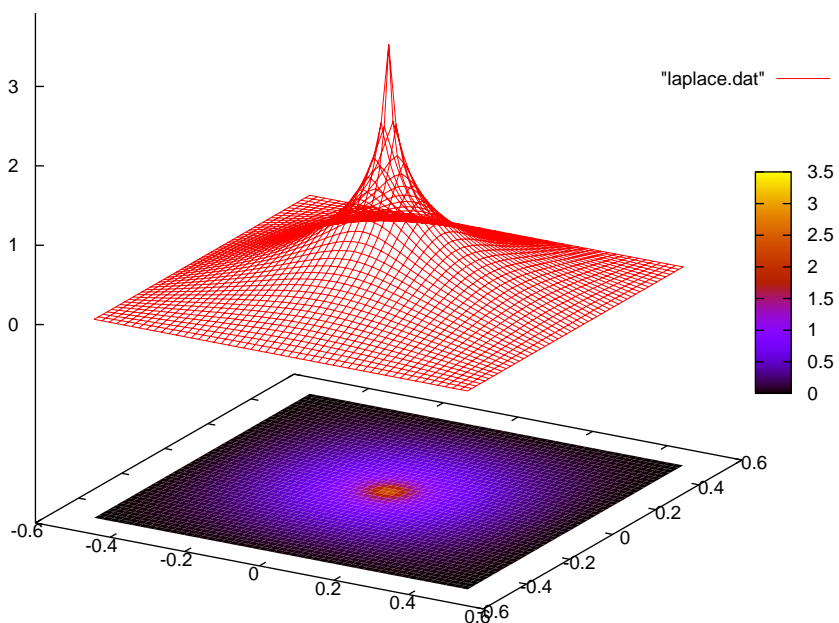


図 9.1: 原点に置かれた点電荷が作る静電場ポテンシャル。

`set view` コマンドは視点の指定である。色々試してみて、引数の意味を推察してみたい。
プリンタに出力するには、

```
gnuplot> set terminal postscript 
... (メッセージが表示される)
gnuplot> set output "laplace.ps" 
gnuplot> splot "laplace.dat" with lines 
```

とすると、`laplace.ps` というファイルが生成されるので、他の端末ウィンドウから

```
lp -d bps1-ps laplace.ps
```

とすればよい。第二講義室にはプリンタが2台あり、`bps1-ps` は `bps2-ps` でもよい。

課題：静電ポテンシャルのシミュレーション

他の色々な電荷分布と境界条件の場合 (例えば、正方形の一边がある電位に保たれていて、外は接地されている場合) について、静電場ポテンシャルを計算して図示せよ。