差分グリッドを用いた微分方程式の数値計算（0）
2 階微分方程式

SHIMURA Masato
jcd02773@nifty.com

2009 年 12 月 8 日

目次

概要

微分方程式の楽しみの一つはいきなりシュミレーションが出きることにある。蛹から孵ったのは蝶か蛾かをグラフを描きながら鑑賞できる。

[*1]

1　差分グリッドと微分方程式

差分はディジタル積分である。積分は微分の逆計算であるアナログが中心であったが、コンピュテーショナル · グリッドを用いた微分方程式、偏微分方程式の数値解法が活用されている。

数値解法には次のような手順が必要である。

- 関数のスクリプトの作成
- 差分マトリクスの作成
- 被説明変数ベクトルの作成
- 境界条件の設定
- システム方程式の計算
- グラフの作成

1.1　テーラー展開と差分

テーラー展開から前進差分、後退差分、中心化差分の式が得られる。
テーラー展開

$$f(x) = f(a) + \frac{x-a}{1!}f^{'}(a) + \frac{(x-a)^2}{2!}f^{''}(a) + \frac{(x-a)^3}{3!}f^{'''}(a) + \cdots$$

$h = x - a$ と置く

$$f(x) = f(a) + \frac{h}{1!}f^{'}(a) + \frac{h^2}{2!}f^{''}(a) + \frac{h^3}{3!}f^{'''}(a) + \cdots$$

$f(x+h), f(x-h)$ を導出する

$$f(x+h) = f(x) + f^{'}(x)h + f^{''}(x)\frac{h^2}{2} + f^{'''}(x)\frac{h^3}{6}\cdots$$

$$f(x-h) = f(x) - f^{'}(x)h + f^{''}(x)\frac{h^2}{2} - f^{'''}(x)\frac{h^3}{6}\cdots$$

[*1] 本稿では J 式表記法を併用している。

$-1 \ \rightarrow\ \_1$

$\frac{1}{32} \ \rightarrow\ 1r32$

h が十分小さいものとして大胆に h^2 以降を無視すると次の差分の式が得られる

$$
\begin{aligned}
\text{前進差分} \quad f'(x) &= \frac{df}{dx} = \frac{1}{h}\left(f(x+h) - f(x)\right) \\
\text{後退差分} \quad f'(x) &= \frac{df}{dx} = \frac{1}{h}\left(f(x) - f(x-h)\right) \\
\text{中心化差分} \quad f'(x) &= \frac{df}{dx} = \frac{1}{2h}\left(f(x+h) - f(x-h)\right)
\end{aligned}
$$

精度の良い中心差分が多く用いられるが、前進差分、後退差分も用いられる。

$$
f'(x) = \frac{df}{dx} = \frac{1}{2h}\left(f(x+h) - f(x-h)\right)
$$

中心化 2 階差分も求める

$$
f''(x) = \frac{d^2 f}{dx^2} = \frac{1}{h^2}\left(f(x+h) - 2f(x) + f(x-h)\right)
$$

1.2 境界条件

境界条件には出港、入港ともに桟橋にしっかり付ける [ディリクレ条件] と片方は仮桟橋や艀の [ノイマン · ロビン] 条件がある。境界条件により、コンピュテーショナル · グリッドの組み方も若干異なる。

境界条件　微分時に無明の彼方に消えた関数の定数項が積分定数として黒衣をまとい復活する。オイラー法やルンゲ · クッタ法は関数と初期条件を与えて右側はフリーで積分で数値解を求める。

これに対し、座標の始端と終端に制限がある場合は差分格子を用いた解法が適している。

ディリクレ境界条件　ディリクレ境界条件は両端に実数をとる。（初期値と終端値）

ノイマン境界条件　始点と勾配などのパラメータを与える

境界条件の設定　境界条件は物理公式のシュミレーションでは計算法と合わせ詳細に検討が必要であるがここでは体験にとどめる

■ディリクレ　*Johan Peter Gustav Le jenue Dirichlet* (1805-1859)

家族はベルギーの出身で 父はドイツで郵便局長の職にあった。ボン大学を卒業。後年ベルリン、ゲッチンゲン大学で教える。メンデルスゾーンの妹と結婚したことでも知られる。
かのマッハの教えを受けた。弟子にクロネッカやリプシッツがいる。
フェルマーの定理 (n=5) の証明をルジャンドルと競い、n ＝ 14 の証明を行った。

2　2 階微分方程式と差分グリッド

2.1　ディリクレ条件での基本公式

2 階微分方程式

$y'' = p(x)y' + q(x)y + r(x)$,　$x \in [a, b]$

ディリクレ条件

$y_((a) = \alpha,\ \ y(b) = \beta$

コンピュテーショナル · グリッドに載せる 2 階微分方程式

$$\{y'' = p(x)y' + q(x)y + r(x)\}|_x = x_i \quad (1 \leq i \leq N-1)$$

中心化 2 階差分の式

$$\frac{y_{i+1} - 2w_i + y_{i-1}}{h^2} + O(h^2) = p_i \frac{y_{i+1} - y_{i-1}}{2h} + q_i y_i + r_i + O(h^2)$$

p_i, q_i, r_i を用い、式の $y \to w$ への変更 (エグザクトから推計へ)

$$\frac{w_{i+1} - 2w_i + w_{i-1}}{h^2} = p_i \frac{w_{i+1} - w_{i-1}}{2h} + q_i w_i + r_i$$

境界条件　$y(x0) = \alpha,\ y(x_N) = \beta$ の 2 式を加え $y \to w$ に変更

2 階微分方程式の基本差分公式

$$\begin{aligned} w_0 &= \alpha \\ \frac{w_{i+1} - 2w_i + w_{i-1}}{h^2} = p_i \frac{w_{i+1} - w_{i-1}}{2h} + q_i w_i + r_i, \quad 1 \leq i &\leq N-1 \\ w_N &= \beta \end{aligned}$$

2.2　差分グリッド

- x_0, x_N は境界グリッドポイント (既知)
- $x_1, x_2, x_3, \cdots, x_{N-1}$ は内部グリッドポイント（未知数）
- $x_i = a + ih$
  $h = \dfrac{b-a}{h}$,　h はステップサイズまたはメッシュサイズ
- 線形計算 A**w=b** を用いる

2.2.1 マトリクス · フォーム

基本式を次に変形する (計算版型）

$$\left(-1-\frac{h}{2}p_i\right)w_{i-1}+(2+h^2q_i)w_i+\left(-1+\frac{h}{2}p_i\right)w_{i+1}=-h^2r_i$$

w_0, w_1, w_2 に置き換え

$$\left(-1-\frac{h}{2}p_1\right)w_0+(2+h^2q_1)w_1+\left(-1+\frac{h}{2}p_1\right)w_2=-h^2r_i$$

$w_0=\alpha$ に置き換え順序を変更する

$$(2+h^2q_1)w_1+\left(-1+\frac{h}{2}p_1\right)w_2=-h^2r_i+\left(1+\frac{h}{2}p_1\right)\alpha$$

$i=N-1$ を用いて置き換え、最初の計算判型から最終判型を得る

$$\left(-1-\frac{h}{2}p_{N-1}\right)w_{N-2}+(2+h^2q_{N-1})w_{N-1}=-h^2r_{N-1}+\left(1-\frac{h}{2}p_{N-1}\right)\beta$$

線形式

$w_1, w_2, w_3, \cdots, w_{N-1}$ は次の線形式から計算できる

$$A\mathbf{w}\mathbf{=}\mathbf{b}$$

差分グリッドに展開

$$A=\begin{bmatrix}
d_1 & u_1 & & & & & & & & \\
l_2 & d_2 & u_2 & & & & & & & \\
 & l_3 & d_3 & u_3 & & & & & & \\
 & & \cdots & \cdots & \cdots & & & & & \\
 & & & \cdots & \cdots & \cdots & & & & \\
 & & & & \cdots & \cdots & \cdots & & & \\
 & & & & & & l_{N-3} & d_{N-3} & u_{N-3} & \\
 & & & & & & & l_{N-2} & d_{N-2} & u_{N-2} \\
 & & & & & & & & l_{N-1} & d_{N-1}
\end{bmatrix}$$

with

$d_i = 2+h^2q_i$

$u_i = -1+\dfrac{h}{2}p_i$

$$l_i = -1 - \frac{h}{2} p_i$$

$$w = \begin{bmatrix} w_1 \\ w_2 \\ \cdot \\ \cdot \\ \cdot \\ w_{N-2} \\ w_{N-1} \end{bmatrix} \quad and \quad b = \begin{bmatrix} -h^2 r_1 + (1 + \frac{h}{2} p_1)\alpha \\ -h^2 r_2 \\ \cdots \\ \cdots \\ \cdots \\ h^2 r_{N-2} \\ -h^2 r_{N-1} + (1 - \frac{h}{2} p_{N-1})\beta \end{bmatrix}$$

A は, l_i, d_i, u_i に h や p_i, q_i が入った数値マトリクスになる。

2.3　もう一つの差分グリッド

こちらの方が良く用いられる

基本差分公式

$$\begin{aligned} w_0 &= \alpha \\ \left(-1 - \frac{h}{2} p_i\right) w_{i-1} + (2 + h^2 q_i) w_i + \left(-1 + \frac{h}{2} p_i\right) w_{i+1} &= -h^2 r_i, \quad 1 \leq i \leq N-1 \\ w_N &= \beta \end{aligned}$$

$A\mathbf{w} = \mathbf{b}$

A は $(N+1)(N+1)$ の三角対重行列

$$A = \begin{bmatrix} 1 & 0 & & & & & & & & \\ d_1 & u_1 & & & & & & & & \\ l_2 & d_2 & u_2 & & & & & & & \\ & l_3 & d_3 & u_3 & & & & & & \\ & & \cdots & \cdots & \cdots & & & & & \\ & & & \cdots & \cdots & \cdots & & & & \\ & & & & \cdots & \cdots & \cdots & & & \\ & & & & & & l_{N-3} & d_{N-3} & u_{N-3} & \\ & & & & & & & l_{N-2} & d_{N-2} & u_{N-2} \\ & & & & & & & & l_{N-1} & d_{N-1} \\ & & & & & & & & 0 & 1 \end{bmatrix}$$

$$w = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \cdot \\ \cdot \\ \cdot \\ w_{N-2} \\ w_{N-1} \\ w_N \end{bmatrix} \quad \text{and} \quad b = \begin{bmatrix} \alpha \\ -h^2 r_1 + (1 + \frac{h}{2} p_1)\alpha \\ -h^2 r_2 \\ \cdots \\ \cdots \\ \cdots \\ h^2 r_{N-2} \\ -h^2 r_{N-1} + (1 - \frac{h}{2} p_{N-1})\beta \\ \beta \end{bmatrix}$$

2.4 Worked Example

2.4.1 Demonstration Problem 2

Bradie Example 8.2

EXAMPLE .

$$\begin{array}{rl} formula & u'' = -(x+1)u' + 2u + (1-x^2)e^{-x} \\ boundary\ condition & u(0) = 1, u(1) = 0 \\ h & \frac{1}{4} \\ x_i & x_i = \dfrac{i}{4} \end{array}$$

計算用雛形 .

$$\left(-1 - \frac{h}{2} p_i\right) w_{i-1} + (2 + h^2 q_i) w_i + \left(-1 + \frac{h}{2} p_i\right) w_{i+1} = -h^2 r_i$$

境界条件 .

$$
\begin{array}{ccccccccc}
\bullet & \rightharpoonup & \bullet & \rightharpoonup & \bullet & \rightharpoonup & \bullet & \rightharpoonup & \bullet \\
 & h & & h & & h & & h & \\
x_0 = 0 & & x_1 = .25 & & x_2 = 0.5 & & x_3 = 0.75 & & x_4 = 1 \\
-1 & & & & & & & & 0 \\
 & \nwarrow & & \mathit{Dirichlet} & \mathit{boundary} & \mathit{condition} & & \nearrow & \\
\hline
-1 & & \mathit{unknown} & & \mathit{unknown} & & \mathit{unknown} & & 0
\end{array}
$$

関数のセット .

$$
\begin{aligned}
&p_1 = -(x_i + 1) = -(1 + \frac{i}{4}) \\
&q_i = 2 \\
&r_i = (1 - x_i^2)e^{-x_i} = (1 - (\frac{i}{4})^2)e^{-\frac{i}{4}}
\end{aligned}
$$

- i.y を関数側で定義している。

```
NB. Demonstration problem 2
NB. u''=-(x+1)u'+2u+(1-x^2)e^-x
NB. h=1r4,N=4,u(0,1)=-1,0
fp2=: 3 : '->:( i. >:  y) % y' NB. p'
fq2=: 2:                        NB. q
fr2=: 3 : '(1 - y0^2 )* ^ - y0=. ( i. >: y) % y' NB. r
```

- 左引数は関数とする。
- 関数を引数とする Script は副詞型（1 ： 0) で記述する
  `(fp2;fq2;fr2)  calc_mat_2nddif 1r4;4;_1 0`
- `i. # y` は関数側で記述するように統一した。
- N は $x_0, x_1, \cdots, x_n, x_{n+1}$ まで 0 オリジンで一つ多く取る。（Newman,Robin で境界条件計算のため必要となる）

差分マトリクスと線形フォーム .

- Dirichlet の境界条件の箇所をを差分マトリクスの最初と最後の行に 1 で指定する。
- 従属変数行列の最初と最後に-1,0 で数値を与える

$$\begin{vmatrix} 1 & 0 & & & \\ -27r32 & 17r8 & -37r32 & 0 & 0 \\ 0 & -13r16 & 17r8 & -19r16 & 0 \\ & 0 & -25r32 & 17r8 & -39r32 \\ & & & 0 & 1 \end{vmatrix} \begin{vmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \\ w_4 \end{vmatrix} = \begin{vmatrix} -1 \\ -0.0456329 \\ -0.0284311 \\ -0.0129163 \\ 0 \end{vmatrix}$$

計算例 .

x (fp2;fq2;fr2) ボックスで関数を与える

y (h;N; ディリクレ境界条件) ex. 1r4;4;_1 0

```
 (fp2;fq2;fr2)  calc_mat_2nddif 1r4;4;_1 0

+--------------------------------+---------------------+
|     1      0      0      0      0|        _1          _1|
|_27r32   17r8 _37r32      0      0|_0.0456329   _0.582559|
|     0 _13r16   17r8 _19r16      0|_0.0284311   _0.301452|
|     0      0 _25r32   17r8 _39r32|_0.0129163   _0.116906|
|     0      0      0      0      1|         0           0|
+--------------------------------+---------------------+
```

プロット .

plot 用に $h = 1r10$ とする。$\Delta x = 0.1$ になる。

```
]a=.(fp2;fq2;fr2)  calc_mat_2nddif 1r10;10;_1 0
'line,marker'plot (0.1*>:i.11 );{:"1 ; {: a
```

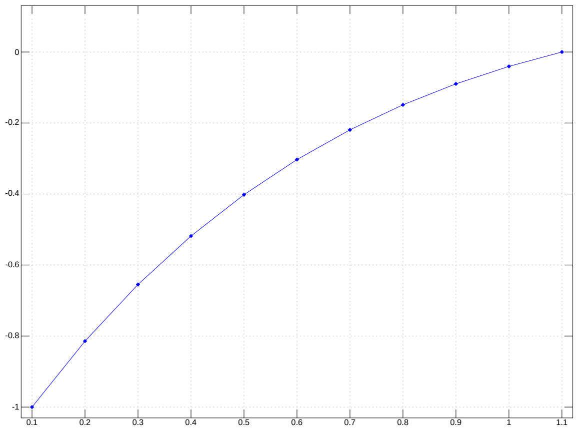

図 1

2.4.2　Demonstration Problem 1

Bradie Example 8.1

EXAMPLE　.

$$
\begin{array}{rl}
formula & -u'' + \pi^2 u = 2\pi^2 sin(\pi x) \\
boundary\ condition & u(0) = u(1) = 0 \\
h & \dfrac{1}{4} \\
N & x_i = \dfrac{1}{4}
\end{array}
$$

関数の定義　.

$$p(x) = 0$$
$$q(x) = \pi^2$$
$$r(x) = -2\pi^2 sin(\pi x)$$

```
NB. Bradie P668
NB. Dirichlet boundary condition
NB. Demonstration proberm 1
fp1=: 0:                                                   NB. p
fq1=: 3 : '1p2*1'   NB. * 1 is alternate of 1p2: NB. q
fr1=: 3 : '_2p2 * 1 o. 1p1 * ( i. >: y) % y'        NB. r
```

0:　0 を返す関数

1p2*1　引数を取らないように 1 を掛けて打ち止め

1 o.　円関数　*sin*

計算例　.

```
  clean L:0 (fp1;fq1;fr1) calc_mat_2nddif 1r4;4;0 0
+----------------------------+-----------------+
| 1       0       0       0  0|       0        0|
|_1 2.61685      _1       0  0|0.872358 0.725371|
| 0      _1 2.61685      _1  0|  1.2337  1.02583|
| 0       0      _1 2.61685 _1|0.872358 0.725371|
| 0       0       0       0  1|       0        0|
+----------------------------+-----------------+
         diff  mat                  wi     ans
```

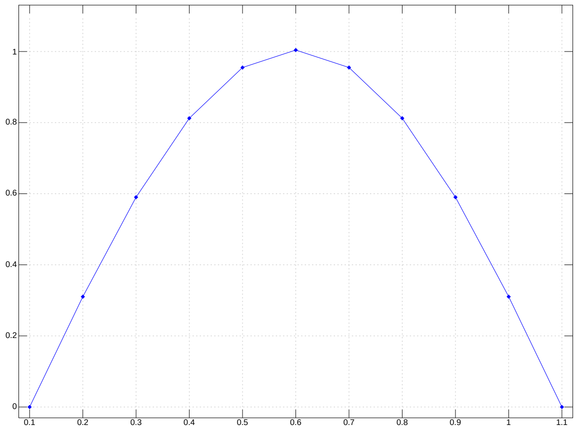

図 2

plot

2.5 Script と解説

- 複数の関数をボックスで引数に取る
- 任意の関数引数に取ることができる副詞型 (1 : 0) とする

```
NB. Usage: (fp2;fq2;fr2)  calc_mat_deriv 1r4;4;_1 0
calc_mat_2nddif =: 1 : 0
```

- 関数は左引数とし u で受ける。
- $L:0$ で関数のボックスの中の 3 の関数 `fp2;fq2;fr2` を同時に計算する

```
f=: u L:0  IX
```

- l_i, d_i, u_i を計算する。w_{i-1}, w_i, w_{i+1} となる

```
X0=:_1- (-: H) * ;{. f
X1=: 2+ (H^2) * ;1{f
X2=: _1+ (-: H) * ;{. f
```

- w_{i-1}, w_i, w_{i+1} はスカラやベクトルが混在する。これをマトリクスに整形する。(相当複雑）

```
XX=: IX adjust_length_sub X0;X1;X2

   4 adjust_length_sub 1 ; 2; 3 4 5 6
1 2 3
1 2 4
1 2 5
1 2 6
```

- 複数の 0 列を加える
- `n |."0 1` で各行を任意に捻ることができる
- ランク `" 0 1` の作用で差分型に捻る。

```
  0 _1 _2 _3 |."0 1 ] i. 4 4
 0  1  2  3
 7  4  5  6
10 11  8  9
13 14 15 12
```

```
M0=. }.}: |: (|: XX), ;("1),.(IX-2)#<IX#0 NB. drop top(a1x) and last(aNx) line
MX=.  (1,  IX#0),M0,(IX#0),1              NB. add top and last line
M1=. ((0,- i.<:  IX),0)|."0 1 MX          NB. twisted
```

- *Dirichlet* 用にマトリクスの上に 1 を左端に付加し、他は 0 とする一行を、最下に 1 を右端とし他は 0 とする一行を加える

  `MX=.  (1,  IX#0),M0,(IX#0),1`
- 従属変数 y を計算する。最初と最後の行はディリクレ条件の数値に入れ替える

```
Y0=. -(H^2)* ;{:f  NB. y
YX=. ({.U),(}. }: Y0),{:U
```

- この関数の本体。線形計算のみ `YX %. M1`

  `M1;YX,.  YX %.  M1` NB. calc differential equation

3　ノイマン · ロビン境界条件と差分グリッド

■カール · ノイマン　*Carl Neumann*(1832-1925) 物理学者フランツ · ノイマンの息子。父はケニッヒス · ベルク大学の教授を 50 年務めた。ドイツ騎士団が拓いた東プロセインのこの町はカントが活躍し、ヒルベルトも生まれている。
父フランツが数学者ヤコビと開催した数学 · 物理ゼミナールはケニッヒスベルク学派と呼ばれ、ドイツを担ったキルヒホフ、ヘッセ、C. ノイマンなど多くの学者がでた。

$$
\begin{array}{lll}
\textit{Dirichlet}\ 条件 & u(0,t) = u(l,t) = 0 & のように境界で\ u\ の値が与えられる \\
\textit{Neumann}\ 条件 & \dfrac{\partial u}{\partial x}(0,t) = \dfrac{\partial u}{\partial x}(l,t) = 0 & のように境界で\dfrac{\partial u}{\partial x}の値が与えられる \\
\textit{Robin}\ 条件 & \dfrac{\partial u}{\partial x} + hu = T & のように境界において外界への熱などの「放散」が起こっているとき
\end{array}
$$

■左が Newman/Robin **境界条件の図**　.

$$
\begin{array}{ccccccccc}
\bullet & \overset{\rightharpoonup}{h} & \bullet & \overset{\rightharpoonup}{h} & \bullet & \overset{\rightharpoonup}{h} & \bullet & \overset{\rightharpoonup}{h} & \bullet \\
x_f & & x_0 = \alpha & & x_1 & & x_2 & & x_3 \\
偽装境界 & & \hookrightarrow & & Computa & tional & domain & \longrightarrow &
\end{array}
$$

$x0 = \alpha$: ノイマン · ロビン境界条件

ノイマン境界条件[*2]

$$
\begin{aligned}
y'(a) &= \alpha \\
y'(b) &= \beta
\end{aligned}
$$

ロビン境界条件

$$\alpha_1 y(a) + \alpha_2 y'(a) = \alpha_3$$

$$\beta_1 y(a) + \beta_2 y'(a) = \beta_3$$

この式から始める

[*2] ロビン境界条件のスペシャルケースである

$$y'' = p(x)y' + q(x)y + r(x), \quad x \in [a, b]$$

computational　template

$$\left(-1 - \frac{h}{2}p_i\right)w_{i-1} + (2 + h^2 q_i)w_i + \left(-1 + \frac{h}{2}p_i\right)w_{i+1} = -h^2 r_i$$

仮想ノード w_F を入れた $x = x_0$ を含む微分方程式

$$\left(-1 - \frac{h}{2}p_0\right)w_f + (2 + h^2 q_0)w_0 + \left(-1 + \frac{h}{2}p_0\right)w_1 = -h^2 r_0$$

ロビン条件を全て求める

$$\alpha_1 y(a) + \alpha_2 y'(a) = \alpha_3 \Longrightarrow \alpha_1 w_0 + \alpha_2 \frac{w_1 - w_f}{2h} = \alpha_3$$

w_f を求める

$$w_f = w_1 \frac{2h}{\alpha_2}(\alpha_3 - \alpha_1 w_0)$$

$x = \alpha$: としたロビン境界条件での差分式

$$\left[2 + h^2 q_0 \quad -(2 + hp_0)h\frac{\alpha_1}{\alpha_2}\right]w_0 - 2w_1 = -h^2 r_0 - (2 + hp_0)h\frac{\alpha_3}{\alpha_2}$$

ノイマン境界条件 $\alpha_1 = 0$ での差分式

$$(2 + h^2 q_0)w_0 - 2w_1 = -h^2 r_0 - (2 + hp_0)h\alpha$$

ロビン境界条件 $x = b$ での差分式

$$-2w_{N-1} + \left[2 + h^2 q_N \quad +(2 - hp_N)h\frac{\beta_1}{\beta_2}\right]w_N = -h^2 r_N + (2 - hp_N)h\frac{\beta_3}{\beta_2}$$

ノイマン境界条件 $x = b$ での差分式

$$-2w_{N-1} + (2 + h^2 q_N)w_N = -h^2 r_N - (2 - hp_N)h\beta$$

マトリクスフォーム

$$A\mathbf{w} = \mathbf{b}$$

A は $(N+1)(N+1)$ の三角対重行列

$$A = \begin{bmatrix} a_{11} & a_{12} & & & & & & \\ l_1 & d_1 & u_1 & & & & & \\ & l_2 & d_2 & u_2 & & & & \\ & & \cdots & \cdots & \cdots & & & \\ & & & \cdots & \cdots & \cdots & & \\ & & & & \cdots & \cdots & \cdots & \\ & & & & & l_{N-1} & d_{N-1} & u_{N-1} \\ & & & & & & a_{N+1,N} & a_{N+1,N+1} \end{bmatrix}$$

$$b = \begin{bmatrix} b_1 \\ -h^2 r_1 \\ -h^2 r_2 \\ \cdots \\ \cdots \\ \cdots \\ h^2 r_{N-1} \\ b_{N+1} \end{bmatrix}$$

$$\begin{aligned} d_i &= \quad 2 + h^2 q_i \\ u_1 &= \quad -1 + \frac{h}{2} p_i \\ l_1 &= \quad -1 - \frac{h}{2} p_i \end{aligned}$$

4　境界条件による差分グリッドの一覧

2 階微分方程式の差分グリッド

ディリクレ、ノイマン、ロビンの境界条件を整理し、条件を指定してシステム方程式を組み上げる。

$y'' = p(x)y' + q(x)y + r(x)$ ， $x \in [a,b]$

$$\begin{array}{ccc} \textit{Dirichlet} & y(a) = \alpha & y(b) = \beta \\ \textit{Neumann} & y'(a) = \alpha & y'(b) = \beta \\ \textit{Robin} & \alpha_1 y(a) + \alpha_2 y'(a) = \alpha_3 & \beta_1 y(b) + \beta_2 y'(b) = \beta_3 \end{array}$$

$$A = \begin{bmatrix} a_{11} & a_{12} & & & & & & \\ l_1 & d_1 & u_1 & & & & & \\ & l_2 & d_2 & u_2 & & & & \\ & & \cdots & \cdots & \cdots & & & \\ & & & \cdots & \cdots & \cdots & & \\ & & & & \cdots & \cdots & \cdots & \\ & & & & & l_{N-1} & d_{N-1} & u_{N-1} \\ & & & & & & a_{N+1,N} & a_{N+1,N+1} \end{bmatrix}$$

$$b = \begin{bmatrix} b_1 \\ -h^2 r_1 \\ -h^2 r_2 \\ \cdots \\ \cdots \\ \cdots \\ h^2 r_{N-1} \\ b_{N+1} \end{bmatrix}$$

$$\begin{aligned} d_i &= \quad 2 + h^2 q_i \\ u_1 &= \quad -1 + \frac{h}{2} p_i \\ l_1 &= \quad -1 - \frac{h}{2} p_i \end{aligned}$$

境界条件の組み合わせは次の 9 通りである。(この記号で入力する）

$$\begin{array}{ccc} DD & DN & DR \\ ND & NN & NR \\ RD & RN & RR \end{array}$$
$$D = Dirichlrt,\ N = Newman,\ R = Robin$$

- a_{11}

$$a_{11} = \begin{cases} 1 & Dirichlet\ BC \quad x = a \\ d_0 & Neumann\ BC \quad x = a \\ d_0 + 2hl_0\alpha_1/\alpha_2 & Robin\ BC \quad x = a \end{cases}$$

- a_{12}

$$a_{12} = \begin{cases} 0 & Dirichlet\ BC \quad x = a \\ -2 & othewise \end{cases}$$

- $a_{N+1,N+1}$

$$a_{N+1,N+1} = \begin{cases} 1 & Dirichlet\ BC & x = b \\ d_N & Neumann\ BC & x = b \\ d_N - 2hu_N\beta_1/\beta_2 & Robin\ BC & x = b \end{cases}$$

- $a_{N+1,N}$

$$a_{N+1,N} = \begin{cases} 0 & Dirichlet\ BC & x = b \\ -2 & othewise \end{cases}$$

- b_1

$$b_1 = \begin{cases} \alpha & Dirichlet\ BC & x = a \\ -h^2r_0 + 2hl_0\alpha & Neumann\ BC & x = a \\ -h^2r_0 + 2hl_0\alpha_3/\alpha_2 & Robin\ BC & x = a \end{cases}$$

- b_{N+1}

$$b_{N+1} = \begin{cases} \beta & Dirichlet\ BC & x = b \\ -h^2r_N - 2hu_N\beta & Neumann\ BC & x = b \\ -h^2r_N - 2hu_N\beta_3/\beta_2 & Robin\ BC & x = b \end{cases}$$

4.1 Worked Example

Bradie Example 8.3

$u'' + u = sin(3x),\ \ x \in [0, \frac{\pi}{2}]$

Robin 境界条件 $x = 0$

$u(0) + u'(0) = -1$

Newman　境界条件 $x = \frac{\pi}{2}$

$u'(\frac{\pi}{2}) = 1$

unifurm partition $[0, \frac{\pi}{2}]$

$h = \frac{i\pi}{8},\ for\ i = 0, 1, 2, 3, 4$

$p_i = 0$

$q_i = 1$

$r_i = sin(\frac{3i\pi}{8})$

Robin 境界条件 $x = 0,\ \alpha_1 = \alpha_2 = 1,\ \alpha_3 = -1$

Newman 境界条件 $x = \frac{\pi}{2},\ \beta = 1$

$$\begin{bmatrix} d-\frac{\pi}{4} & -2 & & & \\ -1 & d & -1 & & \\ & -1 & d & -1 & \\ & & -1 & d & -1 \\ & & & -2 & d \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} = \begin{bmatrix} \frac{\pi}{4} \\ (\frac{\pi}{8})^2 sin(\frac{3\pi}{8}) \\ (\frac{\pi}{8})^2 sin(\frac{6\pi}{8}) \\ (\frac{\pi}{8})^2 sin(\frac{9\pi}{8}) \\ (\frac{\pi}{8})^2 + \frac{\pi}{4} \end{bmatrix}$$

$$d = 2 - \left(\frac{\pi}{8}\right)^2$$

```
   (fp3;fq3;fr3) calc_2nddiff_all 1r8p1;4;  1 1 _1 ; 1 0 0;'RN'
+----------------------------------------+--------------------+
|1.06039       _2        0        0       0| 0.785398   _1.02367|
|      _1 1.84579       _1        0       0|_0.142474  _0.935445|
|       0       _1 1.84579       _1       0|_0.109045  _0.560486|
|       0        0       _1 1.84579      _1|0.0590146 0.00995176|
|       0        0        0       _2 1.84579| 0.939611    0.51984|
+----------------------------------------+--------------------+
                                               y          (ANS)
```

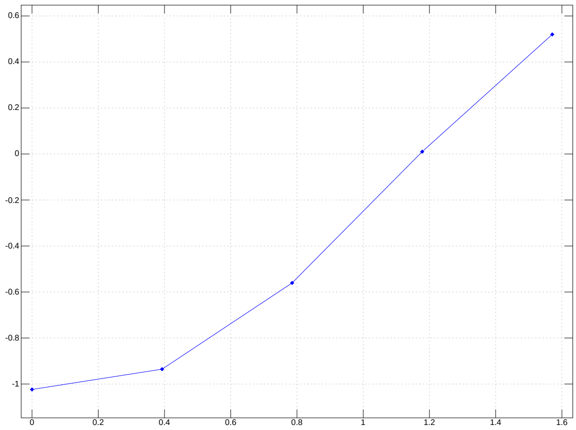

図 3

ディリクレ条件での Example 8.1 のテスト (OK)

```
 clean L:0  (fp1;fq1;fr1)  calc_2nddiff_all  1r4;4;0 0 0; 0 0 0;'DD'
+----------------------------+----------------+
| 1        0        0        0  0|        0        0|
|_1 2.61685       _1        0  0|0.872358 0.725371|
```

```
| 0       _1 2.61685       _1  0|  1.2337  1.02583|
| 0       0       _1 2.61685 _1|0.872358 0.725371|
| 0       0       0       0  1|       0        0|
+-----------------------------+-----------------+
                                         ANS
```

5 References

Brain Bradie [A friendly Introduction to Numeric Analsis] Pearson 2006

付録 A Script

```
calc_mat_2nddif =: 1 : 0
NB. Usage: (fp2;fq2;fr2)  calc_mat_2nddif 1r4;4;_1 0
NB. another // mat_lp (1r10; 10;_1 0)
'H IX U'=. y   NB. h ; x(i.n); u(0,1)
f=. u L:0  IX  NB. add N+1 is done at fpx side
X0=._1- (-: H) * ;{. f
X1=. 2+ (H^2) * ;1{f
NB.  if. 1=# X1 do. X1=: IX # X1 end.
X2=. _1+ (-: H) * ;{. f
 NB. below is expand singleton(because error occure at 1,.1 2 3)
XX=. IX adjust_length_sub X0;X1;X2
NB. - y---------------------------
Y0=. -(H^2)* ;{:f  NB. y
YX=. ({.U),(}. }: Y0),{:U
NB. except top(y0) and lasy(yn)line and append Dirichlet boundary condition
NB. ----make extended diff box--------------
M0=. }.}: |: (|: XX), ;("1),.(IX-2)#<IX#0 NB. drop top(a1x) and last(aNx) line
MX=.  (1,  IX#0),M0,(IX#0),1              NB. add top and last line
M1=. ((0,- i.<:  IX),0)|."0 1 MX          NB. twisted
NB. ---calc matrix -----------
M1;YX,.  YX %.  M1       NB. calc differential equation
)

adjust_length_sub=: 4 : 0
```

```
NB. expand to same length
NB. alternative expand_box
NB. y is X0;X1;X2
NB.x is IX
LEN=.  ; # L:0 y
INDEX=. +/ INDEX0=. LEN e. >: x
select. INDEX
 case. 3 do. XX=.|: ;("1) ,. y      NB. all vector
 case. 0 do. XX=.|: ;("1),. (>:x) # L:0 y NB. all scaller
 fcase. do.
  TMP=. (>: x) # L:0 (-. INDEX0) #  y NB. expand scallor
  TMP=.(INDEX0 # y), TMP        NB. marge
  IND=. (indt_sub INDEX0) i. 0 1 2     NB. expand index
  XX=. |: ;("1),. IND {  TMP      NB.  back to origin order
end.
XX
)

NB. find_index
indt_sub=: 3 : '(I. y),I.-.y'



NB. ------Newman/Robin boundary condition--------
NB. Bradie Example 8.3
 fp3=: 0:
 fq3=: _1
 fr3=: 3 : ' 1 o. 3r8p1 *  i. >:  y' NB. sin

 NB. alpha1=alpha2=1 , alpha3=_1
 NB. beta=1

 calc_2nddiff_all =: 1 : 0
NB.  clean L:0 (fp1;fq1;fr1)  cnr 1r4;4;0 0 0; 0 0 0;'DD' OK
NB.  (fp3;fq3;fr3) cnr 1r8p1;4; 1 1 _1 ; 1 0 0;'RN'
'H IX ALPHA BETA TYPE'=: y
NB. h ; x(i.n); 6 disimal(alpha 1 2 3 beta 1 2 3)
```

```
NB. TYPE is 'DD', DN' ,'DR'.......'RN','RR' NB. 9type
f=: u L:0 IX
NB. x
X0=:_1- (-: H) * ;{. f  NB. Li
X1=: 2+ (H^2) * ;1{f    NB. Di
X2=: _1+ (-: H) * ;{. f NB. Ui
 NB. below is expand singleton(because error occure at 1,.1 2 3)
XX=: IX adjust_length_sub X0;X1;X2
NB. - calc x and y---------------------------
Y0=: -(H^2) * ;{: f   NB. {: f is r //-h^2*r
NB. -h^2* r0 <----- body of y NB. over item is last
BX0=: ({. Y0) + +: H * {. X0   NB. (-2h^2 r0) + 2h* l_0 <-- top b1
BXN=: ({: Y0) - +: H * {: X2 NB. (-2h^2*r_n) - 2h* u_n <--bn
NB. ------------------------------------------
NB. ----matrix-calc a11 a12 and b1------------
NB. A11=. A12=. ANL=. ANR=. B1=. BN=. 0 NB. reset
IND0=: I. 'DNR' e. {. TYPE     NB. top line of a,b
NB. Newman
   if.  0= ;IND0   do.  NB. Dirichlet
      A11=. 1
      A12=. 0
      B1=. {. ALPHA
   end.
   if.  1 = ; IND0  do.
      A11=.  {. X1                 NB. keypoint = d0
      A12=. _2                 NB. neighbore
      B1=. BX0 * {. ALPHA
   end.     NB. alpha
   NB. Robin
   if. 2 = ; IND0  do.
      A11=. ({.X1) + +: H * X0 * %/ 2{. ALPHA NB. alpha1/alpha2
      A12=. _2
      B1=. BX0 * %/  2 1 { ALPHA  NB. alpha3/alpha2
    NB. Dirichlet
    end.
NB. ----matrix-calc an-1 an2 and bn--------------------------
IND1=: I. 'DNR' e. {: TYPE  NB. last line of a,b
```

```
NB. Newman
    if. 0=  ; IND1 do.    NB. Dirichlet
       ANR=. 1
       ANL=. 0
       BN=. {. BETA
    end.
     if.  1 = ; IND1  do.
      ANR=. {: X1
      ANL=. _2
      BN=. BXN * {. BETA
    end.
   NB. Robin
     if. 2=;IND1   do.
      ANR=. ({: X1) - +: H * ({: X2) * ({. BETA) % 1{. BETA  NB. beta1/beta2
      ANL=. _2
      BN=. BXN * %/  2 1 { BETA NB. beta 3/beta2
    end.
NB. ---------------------------------------------
YX=: B1,(}.}:Y0),BN   NB. drop last//   NB. y main
NB. ----make extended diff box--------------
M0=:}. }: |: (|: XX), ;("1),.(IX-2)#<IX#0 NB. except top and last line
NB. MX=:  (A1,_2,(<:IX)#0),M0,((<:IX)#0), _2,AN
MX=:(A11,A12,(<:IX)#0) , M0, ((<:IX)#0),ANL,ANR
RIND=: (0,(-&i. <: IX),0) NB. rotate index
M1=: RIND |.("0 1) MX     NB. twist except top&last
NB. --main = calc matrix -----------
M1 ; YX,.YX %. M1       NB. main = calc differential equation
)
```