

# 地球惑星科学実習 B-3: 偏微分方程式の数値解法

岩山 隆寛<sup>\*</sup>

2015 年 7 月 3, 10, 17, 24 日

## 1 はじめに

地球惑星科学の諸現象は、偏微分方程式の形で書かれることが多い。線形の偏微分方程式であれば多くの場合、解析的に（手計算で）解くことができるが、非線形の偏微分方程式や、境界条件が複雑な場合には、解析的に解くことは困難である。このようなときには計算機を用いて、その偏微分方程式を解く。ここでは、偏微分方程式の数値解法の入門として、解析的に解くことができるが、拡散方程式や波動方程式などの簡単な偏微分方程式を計算機を用いて解いてみる。

本実習では、簡単化のために空間 1 次元とし、以下の偏微分方程式を取り扱う。

### 1. 拡散方程式:

$$\frac{\partial u}{\partial t} = \nu \frac{\partial^2 u}{\partial x^2}, \quad (1)$$

ここで、 $\nu$  は正の定数である。

### 2. 線形移流方程式:

$$\frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} = 0, \quad (2)$$

ここで、 $c$  は定数である。

---

<sup>\*</sup> 神戸大学 大学院理学研究科 惑星学専攻. e-mail: iwayama@kobe-u.ac.jp

<sup>†</sup> 本実習の資料の作成に際して、名古屋工業大学 大学院工学研究科 渡邊威 准教授と神戸大学 大学院理学研究科研究員 村上真也 博士の協力を頂きました。

3. 線形移流拡散方程式:

$$\frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2}, \quad (3)$$

ここで,  $c$  は定数,  $\nu$  は正の定数である.

4. Burgers 方程式:

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

ここで,  $\nu$  は正の定数である.

これらの方程式は, 流体力学の基礎方程式である (1次元の) Navier–Stokes 方程式,

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = -\frac{1}{\rho} \frac{\partial p}{\partial x} + \nu \frac{\partial^2 u}{\partial x^2}, \quad (5)$$

を想定している. ここで,  $\nu$  は正の定数である.

## 2 gnuplot による png ファイルの作成と, アニメーションの作り方

偏微分方程式の解は, 物理量の空間分布が時間発展するので, 解をアニメーションで表示すると見やすいであろう. ここでは, 与えられた関数を gnuplot を使用して png ファイルを作り, そのファイルから (パラパラ漫画式の) アニメーションを作成する方法を解説する. ここで解説する方法は, 計算結果からアニメーションを作成する方法にも応用可能である.

### 2.1 スクリプトファイルを利用した png ファイルの作成方法

gnuplot を利用して多くの図を作成するときに, gnuplot のコマンドをいちいち打つことは効率が悪い. gnuplot のコマンドをファイル (スクリプトファイルと呼ばれる) に記述しておき, それを gnuplot で読み込んで図を作成する方法を解説する.

1. test.plt という名前のスクリプトファイルを作成する.
2. 次の内容を, 適当なエディタを使用して, test.plt に記述する.

```
set xrange[0:2*pi]
set yrange[-1:1]
set xlabel 'x'
set ylabel 'u(x,t)'
plot sin(x) title 'sin(x)'
set term png
set output 'sincurve.png'
replot
```

3. gnuplot を起動して、以下のコマンドを打ち、 $\sin(x)$  が画面に表示され、さらに sincurve.png ファイルができ、そのファイルに表示された図と同じ図 (図 1 参照) が書き込まれていることを確認する。

```
gnuplot> cd 'test.plt を保存したディレクト名'
```

```
gnuplot> load 'test.plt'
```

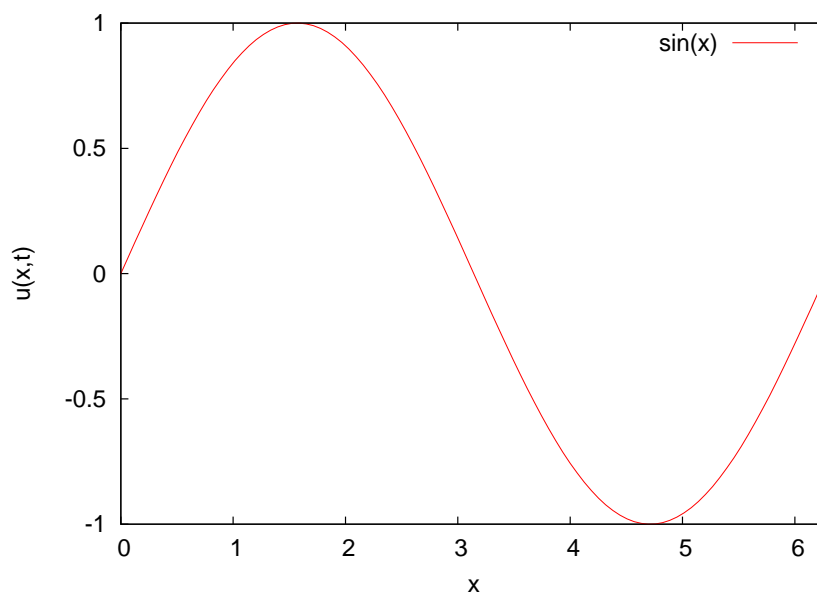


図 1  $\sin(x)$  のグラフ.

## 2.2 gif アニメーションの作成方法：その1

1. 以下の内容のスクリプトファイル (test2.plt) を作成し, gunplot で実行しなさい.\*<sup>1</sup>

```
set xrange[0:2*pi]
set yrange[-1:1]
set xlabel 'x'
set ylabel 'u(x,t)'
set term png
set size 1, 1

set output 'sincurve_000.png'
plot sin(x) title 't=0'
set output 'sincurve_001.png'
plot sin(x+pi/8.) title 't=1'
set output 'sincurve_002.png'
plot sin(x+2.*pi/8.) title 't=2'
.
.
.
set output 'sincurve_015.png'
plot sin(x+15.*pi/8.) title 't=15'
```

2. コマンドプロンプトから ImageMagic の convert コマンドを使用して, png ファイルを結合して gif アニメーションを作成する.

```
> convert sincurve_???.png sincurve.gif
```

3. sincurve.gif を適当なブラウザで開くと, ファイル名順に結合された gif アニメーションを見ることができる.

---

\*<sup>1</sup> eps ファイルでアニメーションを作ると, 前の時刻の画像が透けて見えるため, png ファイルでアニメーションを作ることにした.

## 2.3 gif アニメーションの作成方法：その2

fortran によって計算結果をファイルに書き出し, そのファイルに書かれているデータを gnuplot を用いて作図し, png ファイルとして保存する. さらに, convert コマンドを用いて, png ファイルを結合してアニメーションを作成する方法の例を示す.

1. 次の fortran プログラム (test\_1.f90) を解読しなさい.

! サンプルプログラム

! test\_1.f90

! 作成者 渡邊威 (名古屋工業大学 工学研究科 シミュレーション工学専攻)

! 改変: 2015.06.23 岩山隆寛

!-----

```
implicit none
integer n, i, j, it, nsteps, nout
real*8 t, h, tout, pi, nu, Lx, l, a
parameter (pi=3.141592653589d0, Lx=2.0d0*pi, nu=2.0d-3)
parameter (n=256, t=40.0d0, h=1.0d-2, tout=1.0d+0)
```

```
real*8 u(0:n+1), st(n,0:int(t/tout))
```

```
l=Lx/dble(n)
nsteps=int(t/h)
nout=int(tout/h)
it=0
```

! initial condition

```
do i=1,n
  u(i)=dsin(dble(i)*l)
enddo
```

! 初期条件の保存

```
do i=1,n
  st(i,it)=u(i)
enddo
```

! 時間発展

```
do j=1,nsteps
  do i=1,n
    u(i)=dsin(dble(i)*l+dble(j)*pi/dble(nsteps))
  enddo
```

! データの保存

```
if (mod(j,nout)==0) then
  it=it+1
  do i=1,n
    st(i,it)=u(i)
  enddo
endif
```

```
enddo
```

! ファイルへの書き出し

```
do it=0, int(t/tout)
  do i=1,n
    write(it+100,'(2f10.3)') i*l, st(i,it)
  enddo
enddo
```

```
stop
```

```
end
```

2. test\_1.f90 を実行し, fort.100~fort.140 の 41 個のファイルができていることを確認しなさい.

```
> gfortran test_1.f90
> ./a.out
```

3. gnuplot のスクリプトファイル (test\_1.plt)<sup>\*2</sup>を実行して, fort.100~fort.140 のファイルの中身を作図しなさい. (mov100.png~mov140.png ができる.)

---

<sup>\*2</sup> これは以下のディレクトリにあるので, 各自自分の作業しているディレクトリにコピーして使用してください.

```
gnuplot> load 'test_1.plt'
```

4. convert コマンドを打って, mov100.png~mov140.png を結合して一つのファイル move.gif を作成する.

```
> convert mov???.png move.gif
```

## 2.4 課題

test\_1.f90 を自分で改良して, diffusion\_theory.f とし, 無限領域内の 1 次元拡散方程式の基本解,

$$\frac{1}{\sqrt{4\pi\nu(t+t_0)}} \exp\left\{-\frac{x^2}{4\nu(t+t_0)}\right\} \quad (6)$$

の発展をアニメーションで示しなさい. 拡散係数  $\nu$  や時間間隔, 等は自分で適当な値を設定しなさい.

作図した関数に関する説明と gif ファイルを添付ファイルとして e-mail で iwayama@kobe-u.ac.jp 宛に送信してください.

## 3 拡散方程式の数値解法

1 次元拡散方程式,

$$\frac{\partial u}{\partial t} = \nu \frac{\partial^2 u}{\partial x^2}, \quad (7)$$

を数値的に解く. ここで,  $\nu$  は正の定数である.

### 3.1 解説

- 時間と空間の離散化:

拡散方程式 (7) を解く領域は,  $0 \leq x \leq L$  とする. この区間を  $N$  等分する. そこで, 座標変数  $x$  を正の整数  $i$  を用いて,

$$x = x_i = i\Delta x, \quad \Delta x = \frac{L}{N}, \quad (8)$$

---

\home3\iwayama\exp\_15\pde\exp\_1\test\_1.plt

と表せる. また, 時間も数値計算では離散化して扱うので,  $j$  を正の整数として,

$$t = t_j = j\Delta t, \quad (9)$$

と表す. (7) の解は,  $x, t$  に依存する量なので, したがって, 数値計算では (7) の解は,  $i, j$  に依存することになる:

$$u(x, y) \rightarrow u(x_i, t_j) \quad (10)$$

- 空間微分の表現の仕方:

(7) は時間と空間に関する微分を含むが, 数値計算では微分は近似的にしか取り扱えない. ここでは空間微分の近似の仕方について述べる.

$x_i$  における  $\partial^2 u / \partial x^2$  を  $x_{i-1}, x_i, x_{i+1}$  の3点における  $u$  の値を使って表現してみる.  $\Delta x$  が微小のとき,  $u(x_{i\pm 1}, t_j)$  を Taylor 展開すると

$$u(x_{i\pm 1}, t_j) = u(x_i, t_j) \pm \frac{\partial u}{\partial x} \Delta x + \frac{1}{2} \frac{\partial^2 u}{\partial x^2} (\Delta x)^2 + \mathcal{O}(\Delta x^3), \quad (11)$$

となる. ここで,  $\mathcal{O}(\Delta x)^3$  は  $(\Delta x)^3$  の大きさの項を表す. したがって, 上式を  $\partial^2 u / \partial x^2$  について解いて

$$\frac{\partial^2 u}{\partial x^2} = \frac{u(x_{i+1}, t_j) + u(x_{i-1}, t_j) - 2u(x_i, t_j)}{(\Delta x)^2} + \mathcal{O}(\Delta x^2) \quad (12)$$

を得る.\*3 以上から,  $(\Delta x)^2$  の精度で

$$\frac{\partial^2 u}{\partial x^2} = \frac{u(x_{i+1}, t_j) + u(x_{i-1}, t_j) - 2u(x_i, t_j)}{(\Delta x)^2} \quad (13)$$

と近似する.

- 時間微分の表現の仕方:

時間微分に関しては, 常微分方程式の解法の際に採用した Euler 法を使用する. このとき,  $\Delta t$  の1次の精度で,

$$\frac{\partial u}{\partial t} = \frac{u(x_i, t_{j+1}) - u(x_i, t_j)}{\Delta t} \quad (14)$$

と近似する.

---

\*3  $\mathcal{O}(\Delta x)$  項は相殺されて, 誤差は  $\mathcal{O}(\Delta x)^2$  程度の大きさになることに注意.



- まとめ:

以上の議論から, 拡散方程式 (7) を数値的に解く場合, 空間微分に関しては 2 次精度, 時間微分に関しては 1 次精度で

$$\frac{u(x_i, t_{j+1}) - u(x_i, t_j)}{\Delta t} = \nu \frac{u(x_{i+1}, t_j) + u(x_{i-1}, t_j) - 2u(x_i, t_j)}{(\Delta x)^2}, \quad (15)$$

もしくは,

$$u(x_i, t_{j+1}) = u(x_i, t_j) + \nu \frac{\Delta t}{(\Delta x)^2} \{u(x_{i+1}, t_j) + u(x_{i-1}, t_j) - 2u(x_i, t_j)\} \quad (16)$$

によって, 時刻  $t_j$  における  $u$  の分布から,  $t_{j+1}$  における  $u$  の分布が得られる.

## 3.2 課題

(16) に従って, 拡散方程式を数値的に解きなさい. ここで, 境界条件と初期条件は以下のとおりとする.

境界条件: 周期境界条件,

$$u(x_0, t_j) = u(x_N, t_j), \quad (17)$$

とする.

初期条件: 領域の中心を分布の中心とした Gauss 分布,

$$u(x_i, t_0) = \exp \left[ -a \left( x - \frac{L}{2} \right)^2 \right] \quad (18)$$

とする.

- 拡散方程式を数値シミュレーションするためのサンプルプログラム (diffusion\_sample.f) を完成させる. 時間発展の様子をアニメーションにより観測する. gnuplot のスクリプトファイルは, test\_diffusion.plt を使用しなさい.
- 結果が妥当であるかどうかを検証するために, シミュレーションで得られた結果と解析解を比較する. 解析解を計算するプログラムは, diffusion\_theory.f に用意されている.
- パラメータ  $\nu$  を色々な値に変えてみてシミュレーションを行う. 結果をアニメーションにより観察し, 結果の意味をよく吟味する.

## 参考文献

- 川上 一郎, 「数值計算」, 第 6 章, 岩波書店, 1989 年.