数値計算/シミュレーションハンドブック

Ver. 1.0.0

勉強会参加メンバー

2016 年 8 月 21 日

目次

記号のリスト

d　次元

h_t　時間のステップ幅

h_x　空間のステップ幅

第 1 章

本稿に関して

1.1 本稿の目的

しばしば耳にする「数値計算」や「数値シミュレーション」,「仮想実験」などは一度は少し踏み込んだところまで触っておきたいもの．そのような簡単な理由から，分野問わず幅広い知見から上記項目をまとめていくことを目的にしている．特に，専門的になりすぎず初学者がとっつきやすいノート (ハンドブックと名付けた理由) になればと思っている．ネット上にはたくさんの知識が転がっている．特に，同じような目的からか wiki なりまとめなりのサイトも増え始めている．これらネットに転がっている知識もこのノートに集約できるなら最高の幸せでる．いつの日かこのノートのおかげで数値計算に入り込むことができたという人が現れてくれたら嬉しいな．なお，本稿のデザインは及びから拝借している．この場を持って感謝申し上げる．

1.2 ルール

短いコードなら pdf に書き込み可.

1.3 索引の書き方

索引は index で可能.

1.4 C/C++ から gnuplot を呼び出す

C/C++ でプログラムを組んでいる場合に，ソースコード内に gnuplot を呼び出すコマンドを記述しておけば，いちいち実行とは別に gnuplot を開いて実行データのプロットを行う手間が省ける．

1.4.1 Linux での方法

Linux で作成中の場合，`popen` 関数[*1]で gnuplot を呼び出すことができる．

ソースコード 1.1　C++ のコード内から gnuplot を呼び出し $\sin x$ を描く

```
1 #include <iostream>
2 #include <cstdio>
3 using namespace std;
4
5 int main()
6 {
```

[*1] 外部コマンドをプログラム内で使用できる一つの関数．終わりには `pclose` 関数で閉じないといけない．

```
7           FILE *fp = popen("gnuplot", "w");
8           if (fp == NULL){
9                   return 1;
10          }
11          fputs("set␣mouse\n", fp);
12          fputs("plot␣sin(x)\n", fp);
13          fflush(fp);
14          cin.get();
15          pclose(fp);
16          return 0;
17  }
```

説明

1-3 行目: iostream は読み込みや書き出しを行う関数が入っているライブラリ．C 言語での stdio.h に対応している．また，cstdio は FILE ポインタや fputs 関数などが入っているライブラリである．3 行目は名前空間の宣言であるが無視して構わない．

7 行目: FILE 型で fp というファイルを用意する．そこに，popen 関数を用いて gnuplot と書き込む ("w")．記号 *はポインタを表すが別の記事を参照せよ．

8-17 行目: もし fp が確保できないならそこでお終い．確保できるなら，fputs 関数で ( ) 内の文字列を fp に書き出す．一つ目はマウスで操作できるようにする宣言文で，二つ目は $\sin x$ をプロットする宣言文である．fflush 関数により FILE ポインタ fp のバッファに格納されているデータを吐き出させる．最後に pclose で fp を閉じれば良い．

1.4.2 Windows での場合

Windows (Visual C++) を使う場合は，マウスが自動的に有効になっているので宣言する必要はない．また，popen 関数と pclos 関数の代わりに_popen 関数と_pclose 関数を使う．

ソースコード 1.2　Visual C++ のコード内から gnuplot を呼び出し $\sin x$ を描く

```
1   #include <iostream>
2   #include <cstdio>
3   using namespace std;
4
5   int main()
6   {
7           FILE *fp = _popen("pgnuplot.exe", "w");
8           if (fp == NULL){
9                   return 1;
10          }else{
11                  fputs("plot␣sin(x)\n", fp);
12                  fflush(fp);
13                  cin.get();
14                  _pclose(fp);
15                  return 0;
16          }
17  }
```

第 2 章

常微分方程式の数値計算

偏微分方程式の数値計算において

2.1 問題設定

常微分方程式

$$\frac{du(t)}{dt} = f(t, u(t)) \quad (t > 0)$$

について考えよう.

2.1.1 Runge-Kutta 法

第 3 章

偏微分方程式の差分法

もっとも基礎的な数値スキームである有限差分法 (Finite Difference Method; FDM) を用いて，いくつかの有名な偏微分方程式を計算する．

3.1 移流方程式

線形の (スカラー) 移流方程式[*1]は

$$\partial_t u + \nu \cdot \nabla u = f \quad \text{in } (0,\infty)\times\Omega \tag{3.1}$$

によって記述される．ここで，$u : \Omega \to \mathbb{R}$ は未知関数，$\nu \in \mathbb{R}^d$ はゼロでない定数ベクトルであり，$f = f(t,x) : (0,\infty)\times\Omega \to \mathbb{R}$ は既知の関数ある．もし $\Omega = \mathbb{R}^d$ ならば，解は

$$u(t,x) = u(0, x - t\nu) + \int_0^t f(s, x + (s-t)\nu)ds \quad (x \in \mathbb{R}^d, t \geq 0)$$

とかける ([1, Section 2.1.2])

3.1.1 離散化

3.2 Poisson 方程式

3.3 熱方程式

線形熱方程式の境界値問題は
以下の記号を用いる．

- $\delta*$ $(* = t, x, y)$：時間及び空間刻み幅
- $(t_n, x_i, y_j) := (n\delta t, i\delta x, j\delta y)$, $(n, i, j = 0, 1, 2, \cdots)$：時空間の格子点
- $u^n_{i,j} := u(t_n, x_i, y_j)$：時空間の格子点上での未知数の値

3.3.1 1 次元の問題

[*1] [English?]

1 次元の線形熱方程式に対する周期境界値問題を考える:

$$\begin{cases} u_t(t,x) - cu_{xx}(t,x) = f(t,x) & (t>0,\ 0<x<1) \\ u(0,x) = g(x) & (0 \le x \le 1) \\ u(t,0) = u(t,1) & (t \ge 0). \end{cases} \tag{3.2}$$

陽解法

内部の方程式は,

$$\frac{u_i^{n+1} - u_i^n}{\delta t} - c\frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{(\delta x)^2} = 0$$

すなわち,

$$u_i^{n+1} = \lambda u_{i+1}^n + (1-2\lambda)u_i^n + \lambda u_{i-1}^n$$

と分解される．ここで，$\lambda := c\delta t/(\delta x)^2$.

3.3.2　2 次元の問題

3.4 波動方程式

以下の記号を用いる．

- $\delta *$ $(* = t, x, y)$：時間及び空間刻み幅
- $(t_n, x_i, y_j) := (n\delta t, i\delta x, j\delta y)$, $(n, i, j = 0, 1, 2, \cdots)$：時空間の格子点
- $u_{i,j}^n := u(t_n, x_i, y_j)$：時空間の格子点上での未知数の値
-

3.4.1　1 次元の問題

次の波動方程式に対する Dirichlet 境界値問題 (固定境界値問題) を数値計算することを考える．

$$\begin{cases} \partial_{tt}u(t,x) - c^2\partial_{xx}u(t,x) = 0 & \text{for } (t,x) \in (0,\infty)\times(0,1), \\ u(0,x) = \phi(x), \quad \partial_t u(0,x) = \psi(x) & \text{for } x \in [0,1], \\ u(t,0) = u(t,1) = 0 & \text{for } t \in [0,\infty). \end{cases} \tag{3.3}$$

陽解法

内部の方程式は,

$$\frac{u_i^{n+1} - 2u_i^n + u_i^{n-1}}{(\delta t)^2} - c^2\frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{(\delta x)^2} = 0$$

すなわち,

$$u_i^{n+1} = \lambda u_{i+1}^n + 2(1-\lambda)u_i^n + \lambda u_{i-1}^n - u_i^{n-1}$$

と分解される．ここで，$\lambda := (c\delta t/\delta x)^2$.

初期条件は，$u_i^0 = \Phi_i$ 及び

$$u(\delta t, x) \simeq u(0,x) + \delta t\partial_t u(0,x) + \frac{(\delta t)^2}{2}\partial_{tt}u(0,x)$$

より

$$u_i^1 = \Phi_i + \Psi_i\delta t + \frac{\lambda}{2}(\Phi_{i+1} - 2\Phi_i + \Phi_{i-1})$$

で決まる．ここで，$\Phi_i := \phi(x_i)$ 及び $\Psi_i := \psi(x_i)$ である．なお，境界条件は $u_0^n = u_N^n = 0$ で決まる．

3.4.2 2 次元の問題

正方領域上の波動方程式に対する Dirichlet 境界値問題 (固定境界値問題) を数値計算することを考える.

$$\begin{cases} \partial_{tt}u(t,x,y) - c^2\Delta u(t,x,y) = 0 & \text{for } (t,x,y) \in (0,\infty)\times(0,1)\times(0,1), \\ u(0,x,y) = \phi(x,y), \quad \partial_t u(0,x,y) = \psi(x,y) & \text{for } (x,y) \in [0,1]\times[0,1], \\ u(t,x,0) = u(t,0,y) = 0 & \text{for } (t,x,y) \in [0,\infty)\times[0,1]\times[0,1]. \end{cases} \tag{3.4}$$

3.4.3 陽解法

内部の方程式は

$$\frac{u_{i,j}^{n+1} - 2u_{i,j}^{n} + u_{i,j}^{n-1}}{(h_t)^2} - c^2\left(\frac{u_{i+1,j}^{n} - 2u_{i,j}^{n} + u_{i-1,j}^{n}}{(h_x)^2} + \frac{u_{i,j+1}^{n} - 2u_{i,j}^{n} + u_{i,j-1}^{n}}{(h_y)^2}\right) = 0$$

すなわち,

$$f$$

3.5 いくつかの応用

3.5.1 熱移流方程式

3.5.2 分散波動方程式

参考文献

[1] L. C. Evans, *Partial Differential Equations*, Partial Differential Equations, in the series Graduate studies in mathematics, Second Edition v. 19, American Math. Society, 2010.

第 4 章

偏微分方程式の有限要素法

第 5 章

その他の偏微分方程式

5.1 Hamilton-Jacobi 方程式

5.2 自由境界問題

第 6 章

モンテカルロシミュレーション

第 7 章

セルオートマトン