

差分法の勉強の手引き

桂田 祐史

2015年12月11日, 2016年4月23日

現象数理学科 Mac (Xcode の C コンパイラと、GLSC と、cglsc コマンドが使える) を持っている人向けの説明。

1 参考資料

勉強する場合、以下の資料をすべて印刷することを勧める。

(2) のプログラムに関しては、自分の Mac にダウンロードすること (ブラウザで Ctrl-クリックして、名前をつけて保存する)。

(1) 桂田「発展系の数値解析」¹

差分法入門。熱方程式の初期値境界値問題 (今のテキストの第5課の問題) を題材にして説明してある。

(2) 現象数理学科 Mac の cglsc コマンドでコンパイル出来るプログラム

(a) `heat1d-e-glsc.c`²

同次 Dirichlet 条件 $u(0, t) = u(1, t) = 0$ の場合、陽解法のプログラム (もっとも基本的、短いので読んで理解しやすい)。「発展系の数値解析」3節

(b) `heat1d-i-glsc.c`³

同次 Dirichlet 条件 $u(0, t) = u(1, t) = 0$ の場合、陰解法 (θ 法) のプログラム。「発展系の数値解析」5節 (3)

(c) `heat1n-i-glsc.c`⁴

同次 Neumann 条件 $u_x(0, t) = u_x(1, t) = 0$ の場合、陰解法 (θ 法) のプログラム。「熱方程式に対する差分法 I」1章

(3) 桂田「熱方程式に対する差分法 I」第1章⁵

同次 Dirichlet 境界条件 $u(0, t) = u(1, t) = 0$ 以外の境界条件の問題を解くためにどうすれば良いか、解説してある。

(4) 桂田「発展系の数値解析の続き」⁶

連立1次方程式を解くための LU 分解の説明を詳しく書いてある。

¹<http://nalab.mind.meiji.ac.jp/~mk/labo/text/heat-fdm-0.pdf>

²<http://nalab.mind.meiji.ac.jp/~mk/program/fdm/heat1d-e-glsc.c>

³<http://nalab.mind.meiji.ac.jp/~mk/program/fdm/heat1d-i-glsc.c>

⁴<http://nalab.mind.meiji.ac.jp/~mk/program/fdm/heat1n-i-glsc.c>

⁵<http://nalab.mind.meiji.ac.jp/~mk/labo/text/heat-fdm-1.pdf>

⁶<http://nalab.mind.meiji.ac.jp/~mk/labo/text/heat-fdm-0-add.pdf>

2 勉強の進め方

- §2 の (1) 「発展系の数値解析」 の 1, 2, 3 節を読む。
- §2 の (2) (a) `heat1d-e-glsc.c` は 3 節の差分方程式を計算して、熱方程式の初期値境界値問題 (テキストの 5 課のものに相当) を解くプログラムである。コンパイル&実行してみる (もしこれが出来なかった場合は、Mac を持って桂田のところに来ること、出来るようにしてあげます)。これを解読する (確かに 3 節で説明してある計算をしていることを読み取る) こと。
- 「発展系の数値解析」 の 4 節を、コンピューターで再現しながら読む。そのためには、§2 の (2) の (b) `heat1d-i-glsc.c`, (c) `heat1n-i-glsc.c` を用いる。
 - Dirichlet 条件の問題を解くには `heat1d-i-glsc.c`
 - Neumann 条件の問題を解くには `heat1n-i-glsc.c`
 - 入力パラメーターは、「発展系の数値解析」 図の中に書いてあるので難しくはないはず。
- 「発展系の数値解析」 の 5 節を読む。θ 法というのを説明してあるが、それを実装したプログラムが `heat1d-i-glsc.c` である。プログラムを読んで、そのことを理解すること。
- `heat1d-i-glsc.c` の中で、連立 1 次方程式 (「発展系の数値解析」 の p. 16 式 (46)) を解くのに、`trilu()`, `trisol()` という関数を使っているが、これは係数行列を LU 分解して解いている。LU 分解とは何かについては⁷、§2 の (4) 「発展系の数値解析の続き」に書いてあるが、(短時間に自力で理解するのは大変と思われるので) とりあえず「これで連立 1 次方程式を解いているらしい」でパスしても構わない。
- Neumann 境界条件 $u_x(0, t) = u_x(1, t) = 0$ の問題をどう解くかは、「発展系の数値解析」には書いてない。§2 の (3) 「熱方程式に対する差分法 I」 第 1 章に詳しく書いてある。`heat1n-i-glsc.c` はそれを実装したものである。輪講のテキスト [1] 第 7 課の問題の境界条件は、 $u(0, t) = 0$, $u(1, t) + \gamma u_x(1, t) = 0$ で、この第 2 の方程式をどう扱うかも「熱方程式に対する差分法 I」 第 1 章には書いてある。これを解くためのプログラムは自分で作る必要があるが、`heat1d-i-glsc.c` をたたき台にすれば、比較的容易である (分かっしまえば数行書き換えるだけ)。

⁷線形代数のシラバスに書いてあるようだけど、「知らない」という人が多いようなので…

3 サンプル・プログラムのコンパイルと実行

コンパイルと実行の例

```
[takebe:~/work] mk% cglsc heat1d-i-glsc.c
[takebe:~/work] mk% ./heat1d-i-glsc
入力して下さい : nfunc(1..5)=1
入力して下さい :  $\theta=0.5$ 
入力して下さい : N=100
入力して下さい :  $\lambda=0.5$ 
時間の刻み幅  $\tau = 5e-05$  になりました。
入力して下さい : 最終時刻 Tmax=1
入力して下さい : グラフ書き換え時間間隔 ( $\Delta t$ )=0.01
T= 0.0000e+00
  I      u(i)      I      u(i)      I      u(i)      I      u(i)      I      u(i)
  0 0.0000e+00  1 1.0000e-02  2 2.0000e-02  3 3.0000e-02  4 4.0000e-02
  5 5.0000e-02  6 6.0000e-02  7 7.0000e-02  8 8.0000e-02  9 9.0000e-02
 10 1.0000e-01 11 1.1000e-01 12 1.2000e-01 13 1.3000e-01 14 1.4000e-01
 15 1.5000e-01 16 1.6000e-01 17 1.7000e-01 18 1.8000e-01 19 1.9000e-01
 20 2.0000e-01 21 2.1000e-01 22 2.2000e-01 23 2.3000e-01 24 2.4000e-01
 25 2.5000e-01 26 2.6000e-01 27 2.7000e-01 28 2.8000e-01 29 2.9000e-01
 30 3.0000e-01 31 3.1000e-01 32 3.2000e-01 33 3.3000e-01 34 3.4000e-01
 35 3.5000e-01 36 3.6000e-01 37 3.7000e-01 38 3.8000e-01 39 3.9000e-01
 40 4.0000e-01 41 4.1000e-01 42 4.2000e-01 43 4.3000e-01 44 4.4000e-01
 45 4.5000e-01 46 4.6000e-01 47 4.7000e-01 48 4.8000e-01 49 4.9000e-01
 50 5.0000e-01 51 4.9000e-01 52 4.8000e-01 53 4.7000e-01 54 4.6000e-01
 55 4.5000e-01 56 4.4000e-01 57 4.3000e-01 58 4.2000e-01 59 4.1000e-01
 60 4.0000e-01 61 3.9000e-01 62 3.8000e-01 63 3.7000e-01 64 3.6000e-01
 65 3.5000e-01 66 3.4000e-01 67 3.3000e-01 68 3.2000e-01 69 3.1000e-01
 70 3.0000e-01 71 2.9000e-01 72 2.8000e-01 73 2.7000e-01 74 2.6000e-01
 75 2.5000e-01 76 2.4000e-01 77 2.3000e-01 78 2.2000e-01 79 2.1000e-01
 80 2.0000e-01 81 1.9000e-01 82 1.8000e-01 83 1.7000e-01 84 1.6000e-01
 85 1.5000e-01 86 1.4000e-01 87 1.3000e-01 88 1.2000e-01 89 1.1000e-01
 90 1.0000e-01 91 9.0000e-02 92 8.0000e-02 93 7.0000e-02 94 6.0000e-02
 95 5.0000e-02 96 4.0000e-02 97 3.0000e-02 98 2.0000e-02 99 1.0000e-02
100 0.0000e+00
マウスでウィンドウをクリックして下さい。
[takebe:~/work] mk%
```

以下、入力するパラメーターの説明

- `nfunc` は5つ用意してある初期値のうち、どれを選ぶか、番号で指定する。
- θ , N , λ については、「発展系の数値解析」の中で説明してある。
- `Tmax` は時間をどこまで計算するか、 t の最大値というつもり。
- 計算自体は、時間の刻み幅 τ ごとに行っている。 τ はとても小さくなりがちなので、毎回グラフを描いていると、画面が“真っ黒”になる。そこでグラフ自体は別に時間間隔を指定するようにした。その時間間隔が Δt である。
- 今回のプログラムでは、特に理由がなければ、`Tmax` は1, Δt は0.01 で良いだろう。

このプログラムでは、グラフを重ね描きしているが、プログラム中の `int erase_always = 0;` を `int erase_always = 1;` に書き換えると、毎回画面をクリアしてからグラフを描くようになる。

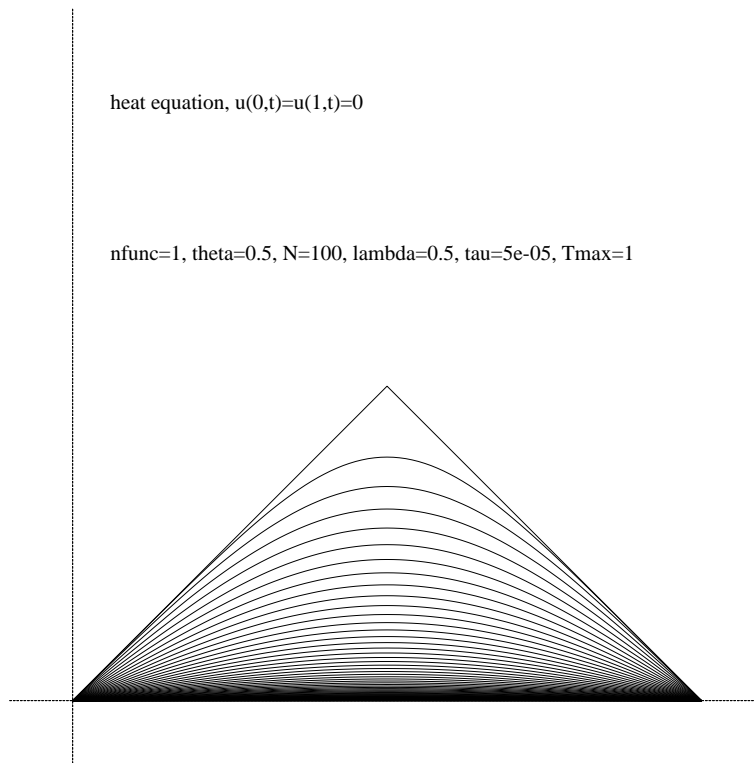


図 1: 同次 Dirichlet 境界条件の場合

参考文献

- [1] Farlow, S. J.: *Partial Differential Equations for Scientists and Engineers*, John Wiley & Sons, Inc (1982), 邦訳: スタンリー・ファーロウ 著, 入理 正夫・入理 由美 訳, 偏微分方程式, 朝倉書店 (1996).