時間発展する問題を解く

桂田 祐史

2015 年 12 月 8 日, 2015 年 12 月 21 日

1 前回の講義のメモ: 熱方程式に対する有限要素法

1.1 準備 — Poisson 方程式を思い出す

まず Poisson 方程式の弱形式とプログラム例を示す。

$$\Omega = (0,1)\times(0,1), \quad \Gamma := \partial\Omega,$$
$$\Gamma_1 = \{(0,y) \mid 0 \le y \le 1\} \cup \{(x,0) \mid 0 \le x \le 1\}, \quad \Gamma_2 := \Gamma \setminus \Gamma_1.$$

$f\colon \Omega \to \mathbb{R}$, $g_1\colon \Gamma_1 \to \mathbb{R}$, $g_2\colon \Gamma_2 \to \mathbb{R}$ が与えられたとして、次の境界値問題を考える。

$$\begin{aligned}
-\triangle u(x,y) &= f(x,y) \quad ((x,y)\in\Omega),\\
u(x,y) &= g_1(x,y) \quad ((x,y)\in\Gamma_1),\\
\frac{\partial u}{\partial n}(x,y) &= g_2(x,y) \quad ((x,y)\in\Gamma_2).
\end{aligned}$$

Find $u \in X_{g_1}$ s.t.

$$\langle u, v\rangle - (f,v) - [g_2, v] = 0 \quad (v \in X).$$

ここで

$$X_{g_1} := \left\{ w \in H^1(\Omega) \,\middle|\, w = g_1 \quad \text{on } \Gamma_1 \right\}, \quad X := \left\{ v \in H^1(\Omega) \,\middle|\, v = 0 \quad \text{on } \Gamma_1 \right\}.$$

poisson.edp

```
// 菊地文雄, 有限要素法概説, サイエンス社
int m=10;
mesh Th=square(m,m);
plot(Th,wait=true);
fespace Vh(Th,P1);
Vh u,v;
func f=1;
func g1=0;
func g2=0;
solve Poisson(u,v)=
  int2d(Th)(dx(u)*dx(v)+dy(u)*dy(v))
 -int2d(Th)(f*v)
 -int1d(Th,2,3)(g2*v)
 +on(1,4,u=g1);
plot(u,wait=1,ps="kikuchi.ps");
```

square() は辺が座標軸に平行な長方形領域を三角形分割する関数である。下の辺、右の辺、上の辺、左の辺の順に 1, 2, 3, 4 というラベルをつける。

int1d(Th,2,3) はラベル 2, 3 の上の線積分である。

自分で境界を定義してから、囲まれる領域を分割するプログラムは、「FreeFem++ の紹介」[1] を見よ。

1.2　熱方程式に対する 後退 Euler 法

$u^n = u(\cdot, t_n)$ とするとき、u_t を後退差分近似して

$$\frac{1}{\tau}\left(u^n - u^{n-1}, v\right) + \langle u^n, v\rangle - (f, v) - [g_2, v] = 0.$$

あるいは

$$\left(u^{n+1} - u^n, v\right) + \tau\langle u^{n+1}, v\rangle - \tau(f, v) - \tau[g_2, v] = 0.$$

(f, g_2 が時間依存している場合は、f^{n+1}, g_2^{n+1} のようにすべきである。)

heatB.edp

```
// 菊地文雄，有限要素法概説，サイエンス社の Poisson 方程式の問題の非定常版
int m=10;
real Tmax=10, tau=0.01, t;
func f=1;
func g1=0;
func g2=0;
func u0=sin(pi*x)*sin(pi*y);
mesh Th=square(m,m);
plot(Th,wait=true);
fespace Vh(Th,P1);
Vh u=u0,uold,v;
problem heat(u,v)=
  int2d(Th)(u*v+tau*(dx(u)*dx(v)+dy(u)*dy(v)))
 -int2d(Th)(tau*f*v)
 -int1d(Th,2,3)(tau*g2*v)
 -int2d(Th)(uold*v)
 +on(1,4,u=g1);
for (real t=0;t<Tmax;t+=tau) {
  uold=u;
  heat;
  plot(u,wait=0);
}
plot(u,wait=1);
```

1.3　熱方程式に対する θ 法

FreeFem++ のマニュアルに §9.5.1 Mathematica Theory on Time Difference Approximations という項があり、θ 法のサンプル・プログラムが載っている。(相変わらず凝った例である。シンプル

[1] http://nalab.mind.meiji.ac.jp/~mk/labo/text/welcome-to-freefem-2012/node8.html

なのを載せれば良いのに…)

前項と同じ熱方程式の場合に説明する。

$$u^{n+\theta} := \theta u^{n+1} + (1-\theta)u^n$$

とおいて (もし f や g_2 が時間依存する場合は f や、g_2 も同様に処理すべき)、

$$\frac{1}{\tau}\left(u^{n+1}-u^n, v\right) + \langle u^{n+\theta}, v\rangle - (f, v) - [g_2, v] = 0.$$

2 本日の実習

1. 前節の 2 つのプログラムを入力&実行し、熱方程式版の最終結果 ($t = \mathrm{T}_{\max}$) が、Poisson 方程式の結果とほぼ同じであることを確認せよ。
2. θ 法のプログラム `heatT.edp` を作成せよ。
3. 安定性について実験的に調べよ。(差分法では、$1/2 \le \theta \le 1$ の場合は無条件安定、$0 \le \theta < 1/2$ の場合は $0 < \lambda \le \dfrac{1}{2(1-2\theta)}$ が安定のための必要十分条件であった。ただし $\lambda = \dfrac{\tau}{h_x^2} + \dfrac{\tau}{h_y^2}$.)
4. (もし出来れば) 厳密解が分かる問題を選び、誤差を調べよ。

FreeFem++ には、マニュアルもあるが、簡単な使い方は、桂田 [1], [2] で分かると思う。
理論の理解には、齊藤 [3] がお勧め。

3 おまけ

波動方程式の初期値境界値問題 (とりあえず太鼓) を解くプログラムを作れ。

(1) $$\frac{1}{c^2}u_{tt} = \triangle u \quad ((x,y,t) \in \Omega \times (0,\infty)),$$

(2) $$u(x,y,t) = 0 \quad ((x,y,t) \in \partial\Omega \times (0,\infty)),$$

(3) $$u(x,y,0) = \phi(x,y), \quad u_t(x,y,0) = \psi(x,y) \quad ((x,y) \in \overline{\Omega}).$$

$$\left(\frac{u^{n+1}-2u^n+u^{n-1}}{\tau^2}, v\right) = c^2 \langle u^n, v\rangle$$

u^1 については、次を参考にすること。

(1) Taylor 展開で 1 次近似

$$u(x,y,\tau) \fallingdotseq u(x,y,0) + u_t(x,y,0)\tau = \phi(x,y) + \tau\psi(x,y).$$

(2) Taylor 展開で 2 次近似

$$\begin{aligned} u(x,y,\tau) &\fallingdotseq u(x,y,0) + u_t(x,y,0)\tau + \frac{u_{tt}(x,y,0)}{2}\tau^2 \\ &= u(x,y,0) + u_t(x,y,0)\tau + \frac{\tau^2}{2}\cdot c^2 \triangle u(x,y,0) \\ &= \phi(x,y) + \psi(x,y)\tau + \frac{(c\tau)^2}{2}\triangle \phi(x,y). \end{aligned}$$

4　おまけの追加

差分法の安定性の話をする際に、従来サンプル・プログラムは C+GLSC で記述したものを紹介していたが、そういう環境を持っていない学生がいたので、以前冗談半分に書いた FreeFem++ 言語による、1 次元熱方程式の初期値境界値問題のプログラムを紹介した。

heat1d-e-freefem.edp

```
// heat1d-e-freefem.edp
// 実行: FreeFem++ heat1d-e-freefem.edp
// N, x が大域的な識別子を隠すと警告が出る (つまり名前が衝突する)。

int i, N=50, n, nMax;
real h = 1.0 / N, lambda = 0.5, Tmax = 1.0;
real tau = lambda * h^2;
real[int] x(N+1);
real[int] u(N+1);
real[int] newu(N+1);

cout << "h= " << h << endl;
cout << "lambda= " << lambda << endl;
cout << "tau= " << tau << endl;

func real f(real x) {
  if (x < 0.5)
    return x;
  else
    return 1 - x;
}

for (i = 0; i <= N; i++) {
  x[i] = i * h;
  u[i] = f(x[i]);
}
plot([x,u], bb=[[-0.1,-0.1],[1.1,1.1]],aspectratio=true, wait=true);

nMax = rint(Tmax / tau);
cout << "nMax =" << nMax << endl;
for (n = 1; n <= nMax; n++) {
  for (i = 1; i < N; i++)
    newu[i] = (1.0 - 2.0 * lambda) * u[i] + lambda * (u[i+1] + u[i-1]);
  // cout << newu << endl;
  newu[0] = newu[N] = 0;
  u = newu;
  plot([x, u], bb=[[-0.1,-0.1],[1.1,1.1]],aspectratio=true);
}
```

参考文献

[1] 桂田祐史：FreeFEM++ の紹介, http://nalab.mind.meiji.ac.jp/~mk/labo/text/welcome-to-freefem-2012/ (2007～).

[2] 桂田祐史：FreeFEM++ ノート, http://nalab.mind.meiji.ac.jp/~mk/labo/text/freefem-note.pdf (2012～).

[3] 齊藤宣一：熱方程式に対する有限要素法と誤差解析, 今は公開していないみたい。僕の学生で読みたい人は相談。 (2006).