応用数値解析特論 第 10 回

～発展系の数値解析 (2)～

かつらだ　まさし
桂田 祐史

https://m-katsurada.sakura.ne.jp/ana2022/

2023 年 6 月 27 日

目次 I

1. 本日の講義内容、連絡事項、レポート課題
2. 発展系の数値解析 (続き)
   - 熱方程式に対する有限要素法 (続き)
     - 実習課題 (再提示)
   - 波動方程式の初期値境界値問題
     - 例題と弱形式
     - 太鼓の振動のプログラム
3. 参考文献

本日の講義内容、連絡事項、レポート課題)

- 発展系の続き。波動方程式を解くプログラムを紹介します。
- レポート課題は、自分で決めた問題を有限要素法で解く、というものにしようと考えている。テーマ決めには、マニュアルである Hecht [1] や、大塚・高石 [2] などを参考にすると良い。内容について質疑応答。

8.2.6 実習課題 (再提示)

1. 2 つのプログラム (`poisson-kikuchi-square.edp`, `heatB.edp`) を入手&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. ある程度分割を細かくして、`init=` の指定の効果を調べよ。
(指定しないと遅く、指定しないで `solver=CG` とすると少し速くなるが、CG 法にせず直接法の系統で `init=` を指定した方が速い (ようである)。)
実行時間は `time` コマンドで計測できる (`time FreeFem++ heatB.edp`)。
4. 安定性について実験的に調べよ。(長方形領域における差分法では、$1/2 \le \theta \le 1$ 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 $\lambda = \dfrac{\Delta t}{\Delta x^2} + \dfrac{\Delta t}{\Delta y^2}$.
⋯ 有限要素法の場合は、このような簡単な判定条件は得られないが、θ が 1 に近い時、0 に近い時、Δt を変えて、安定に計算出来るかどうか試してみる。)
5. (もし出来れば) 厳密解が分かる問題を選び、誤差を調べよ。
6. 自分が選んだ問題 (領域などを変える) で数値実験してみよ。

実習: まずはじめに

熱方程式に対して、時間微分を θ 法で離散化したプログラムを作って、安定性の条件を調べてみよう。

heatB.edp を叩き台にして heatT.edp を作る

```
curl -O https://m-katsurada.sakura.ne.jp/program/fem/heatB.edp
cp heatB.edp heatT.edp
```

方針

θ 法では、$0 \le \theta \le 1$ を満たす θ を取って、微分方程式 $\dfrac{\partial u}{\partial t}(\cdot, t_n) = \triangle u(\cdot, t_n) + f$ を

$$\frac{1}{\Delta t}\left(u^{n+1} - u^n\right) = \triangle u^{n+\theta} + f \tag{1}$$

で置き換えるわけである。ここで

$$u^{n+\theta} := \theta u^{n+1} + (1-\theta)u^n. \tag{2}$$

(2) を (1) に代入して

$$\frac{1}{\Delta t}\left(u^{n+1} - u^n\right) = \theta \triangle u^{n+1} + (1-\theta)\triangle u^n + f. \tag{3}$$

これから弱形式を導ける。

後退 Euler 法のコードを読む

$$\text{(4)} \qquad \frac{1}{\Delta t}\left(u^{n+1} - u^n\right) = \triangle u^{n+1}.$$

弱形式は以下の通り。

$$\text{(5)} \qquad \left(u^{n+1}, v\right) - (u^n, v) + \Delta t\langle u^{n+1}, v\rangle - \Delta t(f, v) - \Delta t[g_2, v] = 0.$$

heatB.edp 内の problem

```
problem heat(u,v,init=n)=
  int2d(Th)(u*v)-int2d(Th)(uold*v)
 +int2d(Th)(tau*(dx(u)*dx(v)+dy(u)*dy(v)))
 -int2d(Th)(tau*f*v)-int1d(Th,2,3)(tau*g2*v)
 +on(1,4,u=g1);
```

このコードが (5) の実現であることは容易に理解できるであろう。

FreeFem++ の文法についての注意 (文句)

FreeFem++ の文法は制限が強く、自分で弱形式をコーディングしようとすると、色々な文法エラーに遭遇する。

- `int2d(Th)(u*v)-int2d(Th)(uold*v)` の部分を `int2d(Th)((u-uold)*v)` や `int2d(Th)(u*v-uold*v)` に置き換えることは出来ない。
- 一方で `int2d(Th)(u*v)` と `int2d(Th)(tau*(dx(u)*dx(v)+dy(u)*dy(v)))` を `int2d(Th)(u*v+tau*(dx(u)*dx(v)+dy(u)*dy(v)))` のように 1 つにまとめることはできる。
- `定数*int2d(Th)(...)` という式が扱えないので、`int2d(Th)(tau*(dx(u)*dx(v)+dy(u)*dy(v)))` の定数部分を `tau*int2d(Th)((dx(u)*dx(v)+dy(u)*dy(v)))` のようにくくり出すことは出来ない。

積分は複数の `int2d()()` にバラす、積分の前にある定数は被積分関数に入れてしまう、という方針が良いだろう (と個人的に考えている)。

θ 法のコードを作る

弱形式は次のようになる。

$$(6)\quad \frac{1}{\Delta t}\left(u^{n+1}-u^n, v\right)+\langle u^{n+\theta}, v\rangle-(f, v)-[g_2, v]=0, \quad u^{n+\theta}:=\theta u^{n+1}+(1-\theta)u^n.$$

すなわち

$$(7)\quad \left(u^{n+1}, v\right)-(u^n, v)+\Delta t\theta\langle u^{n+1}, v\rangle+\Delta t(1-\theta)\langle u^n, v\rangle-\Delta t(f, v)-\Delta t[g_2, v]=0.$$

次のコードがこの弱形式を実現したものであることは理解できるであろう。

コード例

```
problem heat(u,v,init=i)=
  int2d(Th)(u*v)-int2d(Th)(uold*v)
 +int2d(Th)(theta*tau*(dx(u)*dx(v)+dy(u)*dy(v)))
 +int2d(Th)((1-theta)*tau*(dx(uold)*dx(v)+dy(uold)*dy(v)))
 -int2d(Th)(tau*f*v)-int1d(Th,2,3)(tau*g2*v)
 +on(1,4,u=g1);
```

もちろん、もっとばらばらにしても良い。

動作確認してみよう。

安定性条件の実験

1 次元熱方程式を差分法で解いたとき

- $\theta = 1$ (後退 Euler 法) であれば、最大値ノルムの意味で無条件に安定であった。
- ノルムによっては、(もっと弱い条件) $\theta \geq 1/2$ であれば無条件に安定であった。
- $0 < \theta < 1/2$ のとき、時間刻みが十分小 ($\Delta t \leq \dfrac{(\Delta x)^2}{2(1-2\theta)}$) でないと不安定になる。

heatT.edp で $\theta = 1, 1/2, 1/4$ の場合に、時間刻みを大きくしても安定性が保たれるか調べる。

パラメーターがキーボード入力出来るように改造する

```
real Tmax=10, tau=0.01, t, theta=1;// theta を加える。Tmax も大きくする。

// コメント・アウトすると、m, tau, theta が実行時に変更できるようになる
// cout << "m dt theta: "; cin >> m >> tau >> theta;
// cout << "m=" << m << ", tau=" << tau << ", theta=" << theta << endl;
mesh Th=square(m,m);
...
```

波動方程式の初期値境界値問題を解くプログラムを作ってみよう。

(8a) $$\frac{1}{c^2}u_{tt} = \triangle u \quad ((x,y,t) \in \Omega \times (0,\infty)),$$

(8b) $$u(x,y,t) = 0 \quad ((x,y,t) \in \partial\Omega \times (0,\infty)),$$

(8c) $$u(x,y,0) = \phi(x,y), \quad u_t(x,y,0) = \psi(x,y) \quad ((x,y) \in \overline{\Omega}).$$

$u^n = u(\cdot, t_n)$ として

(9) $$\left(\frac{u^{n+1} - 2u^n + u^{n-1}}{\Delta t^2}, v\right) = c^2 (\triangle u^n, v) = -c^2 \langle u^n, v \rangle \quad (v \in \hat{X}).$$

$u^0 = \phi$. u^1 については、複数の手段が考えられる。

1. Taylor 展開で 1 次近似 (以下のサンプル・プログラムで採用)
$$u(x,y,\Delta t) \fallingdotseq u(x,y,0) + u_t(x,y,0)\Delta t = \phi(x,y) + \psi(x,y)\Delta t.$$

2. Taylor 展開で 2 次近似
$$\begin{aligned} u(x,y,\Delta t) &\fallingdotseq u(x,y,0) + u_t(x,y,0)\Delta t + \frac{u_{tt}(x,y,0)}{2}\Delta t^2 \\ &= u(x,y,0) + u_t(x,y,0)\Delta t + \frac{\Delta t^2}{2} \cdot c^2 \triangle u(x,y,0) \\ &= \phi(x,y) + \psi(x,y)\Delta t + \frac{(c\Delta t)^2}{2}\triangle\phi(x,y). \end{aligned}$$

8.3.2 太鼓の振動のプログラム

サンプル・プログラムを用意してある。円盤領域における同次 Dirichlet 境界条件なので、太鼓の膜の振動現象のモデルと解釈できる (理想化した場合は厳密解が Bessel 関数で表示出来る)。

ターミナルでこうして入手

```
curl -O https://m-katsurada.sakura.ne.jp/program/fem/taiko.edp
```

`plot()` で `,dim=3` とした場合、描画範囲どうやって指定するかが分からず、苦し紛れに定数関数 (`top`, `bottom`) を描くようにしている。誰か解決策を見つけた人、教えて下さい。

8.3.2 太鼓の振動のプログラム

```
// taiko.edp
border Gamma(t=0,2*pi) { x=cos(t); y=sin(t); }
mesh Th=buildmesh(Gamma(40));
plot(Th,wait=true);
fespace Vh(Th,P1);
Vh newu,u,oldu,v,top,bottom;
func phi=1-x^2-y^2; func psi=0;
func topv=1.2; func bottomv=-1.2;
top=topv; bottom=bottomv;
real tau=0.01,Tmax=10;
real [int] levels =-1.5:0.1:1.5;
u=phi;
newu=u+tau*psi;
problem wave(newu,v)=
  int2d(Th)(newu*v)+int2d(Th)(-2*u*v+oldu*v+tau^2*(dx(u)*dx(v)+dy(u)*dy(v)))
 +on(Gamma,newu=0);
for (real t=0; t<Tmax; t+=tau) {
    oldu = u;
    u = newu;
    wave;
    plot(top,bottom,u,dim=3,value=1,fill=0,viso=levels,wait=0);
}
plot(top,bottom,u,dim=3,value=1,fill=0,viso=levels,wait=1);
```

以前のやり残し　有限要素解の誤差を見る

https://m-katsurada.sakura.ne.jp/lecture/ouyousuuchikaisekitokuron-2023/ANA07_0530_handout.pdf#page=57

参考文献

[1] Hecht, F.: Freefem++, https://doc.freefem.org/pdf/FreeFEM-documentation.pdf, 以前は http://www3.freefem.org/ff++/ftp/freefem++doc.pdf にあった。 (??).

[2] 大塚厚二, 高石武史：有限要素法で学ぶ現象と数理 — FreeFem++数理思考プログラミング —, 共立出版 (2014), https://sites.google.com/a/comfos.org/comfos/ffempp というサポート WWW サイトがある．Maruzen eBook に入っているので、https://elib.maruzen.co.jp/elib/html/BookDetail/Id/3000018545 でアクセス出来る.