応用数値解析特論 第 10 回

～発展問題 (2)～

桂田 祐史 (かつらだ まさし)

http://nalab.mind.meiji.ac.jp/~mk/ana/

2020 年 11 月 30 日

目次

本日の内容

- 前回に引き続き発展系 (時間に依存して系が変化する系) の話。ようやく有限要素法で解く話になる。サンプル・プログラムを提示して解説する。あまり深い話は出来ない。

本日の内容

- 前回に引き続き発展系 (時間に依存して系が変化する系) の話。ようやく有限要素法で解く話になる。サンプル・プログラムを提示して解説する。あまり深い話は出来ない。
- 予定より進行がやや遅れている (脇道にそれ過ぎたかもしれない)。当初では、最終回に各自のレポートの説明をしてもらう予定であったが、今後の授業の進行によっては授業中に時間が確保できるか分からない (一応その予定にしておくが、どうするか、12 月 14 日の授業までに決定する)。

本日の内容

- 前回に引き続き発展系 (時間に依存して系が変化する系) の話。ようやく有限要素法で解く話になる。サンプル・プログラムを提示して解説する。あまり深い話は出来ない。
- 予定より進行がやや遅れている (脇道にそれ過ぎたかもしれない)。当初では、最終回に各自のレポートの説明をしてもらう予定であったが、今後の授業の進行によっては授業中に時間が確保できるか分からない (一応その予定にしておくが、どうするか、12 月 14 日の授業までに決定する)。
- レポート課題を出す。次のスライドで説明する。

レポート課題

2 つの課題を出す。

レポート課題

2 つの課題を出す。

レポート課題 A　本日の実習課題 (スライド 12) についてレポートせよ。締め切りは 12 月 19 日 (土曜) 18:00.
形式は A4 サイズの PDF, 提出は Oh-o! Meiji で行う。
これは指示が具体的なので取り組みやすいと思われる。この課題については締め切り後に簡単な解説を行う。

レポート課題

2 つの課題を出す。

レポート課題 A 本日の実習課題 (スライド 12) についてレポートせよ。締め切りは 12 月 19 日 (土曜) 18:00.
形式は A4 サイズの PDF, 提出は Oh-o! Meiji で行う。
これは指示が具体的なので取り組みやすいと思われる。この課題については締め切り後に簡単な解説を行う。

レポート課題 B 有限要素法や変分法に関係する自由課題。数学的な分析と数値実験の少なくともどちらかがあること。例えば、自分が興味ある現象の数理モデルについて説明し、それを有限要素法によってシミュレーションして得た結果を分析する、という内容で構わない。
締め切りは 2021 年 1 月 15 日 (金曜) 18:00.
形式は A4 サイズの PDF, 提出は Oh-o! Meiji で行う。

熱方程式 (内部で熱が発生する or 熱が吸収される) の初期値境界値問題

$$
\begin{aligned}
&(1a) \qquad u_t(x,t) = \triangle u(x,t) + f(x) \quad ((x,t) \in \Omega \times (0,\infty)),\\
&(1b) \qquad u(x,t) = g_1(x) \quad ((x,t) \in \Gamma_1 \times (0,\infty)),\\
&(1c) \qquad \frac{\partial u}{\partial \boldsymbol{n}}(x,t) = g_2(x) \quad ((x,t) \in \Gamma_2 \times (0,\infty)),\\
&(1d) \qquad u(x,0) = u_0(x) \quad (x \in \overline{\Omega})
\end{aligned}
$$

を考える。

熱方程式 (内部で熱が発生する or 熱が吸収される) の初期値境界値問題

$$
\begin{aligned}
&\text{(1a)} \qquad u_t(x,t) = \triangle u(x,t) + f(x) \quad ((x,t) \in \Omega \times (0,\infty)),\\
&\text{(1b)} \qquad u(x,t) = g_1(x) \quad ((x,t) \in \Gamma_1 \times (0,\infty)),\\
&\text{(1c)} \qquad \frac{\partial u}{\partial \boldsymbol{n}}(x,t) = g_2(x) \quad ((x,t) \in \Gamma_2 \times (0,\infty)),\\
&\text{(1d)} \qquad u(x,0) = u_0(x) \quad (x \in \overline{\Omega})
\end{aligned}
$$

を考える。

多くの設定は、これまで扱ってきた Poisson 方程式の境界値問題に準じる。

熱方程式 (内部で熱が発生する or 熱が吸収される) の初期値境界値問題

$$
\begin{aligned}
&\text{(1a)} && u_t(x,t) = \triangle u(x,t) + f(x) \quad ((x,t) \in \Omega \times (0,\infty)),\\
&\text{(1b)} && u(x,t) = g_1(x) \quad ((x,t) \in \Gamma_1 \times (0,\infty)),\\
&\text{(1c)} && \frac{\partial u}{\partial \boldsymbol{n}}(x,t) = g_2(x) \quad ((x,t) \in \Gamma_2 \times (0,\infty)),\\
&\text{(1d)} && u(x,0) = u_0(x) \quad (x \in \overline{\Omega})
\end{aligned}
$$

を考える。

多くの設定は、これまで扱ってきた Poisson 方程式の境界値問題に準じる。

Ω は $\mathbb{R}^2$ の有界領域で、その境界 $\Gamma := \partial\Omega$ は $\Gamma = \Gamma_1 \cup \Gamma_2$, $\Gamma_1 \cap \Gamma_2 = \emptyset$ と分割されているとする。$\boldsymbol{n}$ は Γ_2 上の点 x における外向き単位法線ベクトルである。また $u\colon \overline{\Omega} \times [0,\infty) \to \mathbb{R}$ は未知関数であり、それ以外の $u_0\colon \overline{\Omega} \to \mathbb{R}$, $f\colon \Omega \to \mathbb{R}$, $g_1\colon \Gamma_1 \to \mathbb{R}$, $g_2\colon \Gamma_2 \to \mathbb{R}$ は既知関数とする。

熱方程式 (内部で熱が発生する or 熱が吸収される) の初期値境界値問題

$$u_t(x,t) = \triangle u(x,t) + f(x) \quad ((x,t) \in \Omega \times (0,\infty)), \tag{1a}$$

$$u(x,t) = g_1(x) \quad ((x,t) \in \Gamma_1 \times (0,\infty)), \tag{1b}$$

$$\frac{\partial u}{\partial \boldsymbol{n}}(x,t) = g_2(x) \quad ((x,t) \in \Gamma_2 \times (0,\infty)), \tag{1c}$$

$$u(x,0) = u_0(x) \quad (x \in \overline{\Omega}) \tag{1d}$$

を考える。

多くの設定は、これまで扱ってきた Poisson 方程式の境界値問題に準じる。

Ω は $\mathbb{R}^2$ の有界領域で、その境界 $\Gamma := \partial\Omega$ は $\Gamma = \Gamma_1 \cup \Gamma_2$, $\Gamma_1 \cap \Gamma_2 = \emptyset$ と分割されているとする。$\boldsymbol{n}$ は Γ_2 上の点 x における外向き単位法線ベクトルである。また $u\colon \overline{\Omega} \times [0,\infty) \to \mathbb{R}$ は未知関数であり、それ以外の $u_0\colon \overline{\Omega} \to \mathbb{R}$, $f\colon \Omega \to \mathbb{R}$, $g_1\colon \Gamma_1 \to \mathbb{R}$, $g_2\colon \Gamma_2 \to \mathbb{R}$ は既知関数とする。

$f = f(x, t)$, $g_1 = g_1(x, t)$, $g_2 = g_2(x, t)$ と時間依存している場合の問題を解くのも難しくない。

9.2.2 解法の方針

時間微分については差分法で近似し、空間微分については有限要素法で近似する。つまり前節の最後に書いたように、まず

(2a) $$\frac{u^{n+1}-u^n}{\Delta t}=\triangle u^{n+1}+f \quad (\text{後退 Euler 法の場合}),$$

あるいは

(2b) $$\frac{u^{n+1}-u^n}{\Delta t}=\triangle[(1-\theta)u^n+\theta u^{n+1}]+f \quad (\theta\ \text{法の場合})$$

と時刻について差分近似してから、各時刻 t_{n+1} で (1b), (1c) と合わせて、Ω における境界値問題とみなす。すなわち、境界条件は

(3a) $$u^{n+1}=g_1 \quad (\text{on } \Gamma_1),$$

(3b) $$\frac{\partial u^{n+1}}{\partial \boldsymbol{n}}=g_2 \quad (\text{on } \Gamma_2).$$

9.2.2 解法の方針

時間微分については差分法で近似し、空間微分については有限要素法で近似する。つまり前節の最後に書いたように、まず

(2a) $$\frac{u^{n+1}-u^n}{\Delta t} = \triangle u^{n+1} + f \quad (\text{後退 Euler 法の場合}),$$

あるいは

(2b) $$\frac{u^{n+1}-u^n}{\Delta t} = \triangle[(1-\theta)u^n + \theta u^{n+1}] + f \quad (\theta\ \text{法の場合})$$

と時刻について差分近似してから、各時刻 t_{n+1} で (1b), (1c) と合わせて、Ω における境界値問題とみなす。すなわち、境界条件は

(3a) $$u^{n+1} = g_1 \quad (\text{on } \Gamma_1),$$

(3b) $$\frac{\partial u^{n+1}}{\partial \boldsymbol{n}} = g_2 \quad (\text{on } \Gamma_2).$$

(u^n を既知とすると、$-\triangle u^{n+1} + cu^{n+1} = F$ という形をしていて、Poisson 方程式ではないが、同様に解くことが出来る)。

9.2.2 解法の方針

時間微分については差分法で近似し、空間微分については有限要素法で近似する。つまり前節の最後に書いたように、まず

(2a) $$\frac{u^{n+1}-u^n}{\Delta t}=\triangle u^{n+1}+f \quad (\text{後退 Euler 法の場合}),$$

あるいは

(2b) $$\frac{u^{n+1}-u^n}{\Delta t}=\triangle[(1-\theta)u^n+\theta u^{n+1}]+f \quad (\theta\text{ 法の場合})$$

と時刻について差分近似してから、各時刻 t_{n+1} で (1b), (1c) と合わせて、Ω における境界値問題とみなす。すなわち、境界条件は

(3a) $$u^{n+1}=g_1 \quad (\text{on } \Gamma_1),$$

(3b) $$\frac{\partial u^{n+1}}{\partial \boldsymbol{n}}=g_2 \quad (\text{on } \Gamma_2).$$

(u^n を既知とすると、$-\triangle u^{n+1}+cu^{n+1}=F$ という形をしていて、Poisson 方程式ではないが、同様に解くことが出来る)。

以下、内積の記号 $\langle\cdot,\cdot\rangle$, $(\cdot,\cdot)$, $[\cdot,\cdot]$ や、関数空間 $\hat{X}_{g_1}$, $\hat{X}$ は Poisson 方程式のときのものを利用する。

9.2.3 熱方程式に対する前進 Euler 法

一応、時刻についての導関数 $\partial u/\partial t$ を前進差分近似した、前進 Euler 法についても述べておく。

9.2.3 熱方程式に対する前進 Euler 法

一応、時刻についての導関数 $\partial u/\partial t$ を前進差分近似した、前進 Euler 法についても述べておく。

弱形式は

$$\left(\frac{u^{n+1}-u^n}{\Delta t}, v\right) - \langle u^n, v\rangle - (f, v) - [g_2, v] \quad (v \in \hat{X}). \tag{4}$$

すなわち

$$\left(u^{n+1}, v\right) - (u^n, v) + \Delta t\langle u^n, v\rangle - \Delta t(f, v) - \Delta t[g_2, v] = 0 \quad (v \in \hat{X}) \tag{5}$$

となる。

9.2.3 熱方程式に対する前進 Euler 法

一応、時刻についての導関数 $\partial u/\partial t$ を前進差分近似した、前進 Euler 法についても述べておく。

弱形式は

(4) $$\left(\frac{u^{n+1}-u^n}{\Delta t},v\right)-\langle u^n,v\rangle-(f,v)-[g_2,v]\quad (v\in\hat{X}).$$

すなわち

(5) $$\left(u^{n+1},v\right)-(u^n,v)+\Delta t\langle u^n,v\rangle-\Delta t(f,v)-\Delta t[g_2,v]=0\quad (v\in\hat{X})$$

となる。

これは私が不勉強なのかもしれないが、この方法を使うプログラムは見たことがない。差分法の場合と違って陽解法でないので (つまり u^{n+1} を求めるのに、結局は連立 1 次方程式を解く必要がある) メリットがないからだろうか (と考えている)。安定性を調べるのは意味があるので、数値実験してみても良いだろう。

9.2.4 熱方程式に対する後退 Euler 法

まず後退 Euler 法のプログラムを紹介しよう。弱形式は

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

すなわち

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

9.2.4 熱方程式に対する後退 Euler 法

まず後退 Euler 法のプログラムを紹介しよう。弱形式は

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

すなわち

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

サンプル・プログラムを用意してある。

ターミナルではこうして入手

```
curl -O http://nalab.mind.meiji.ac.jp/~mk/program/fem/heatB.edp
```

次の次のスライドに載せてある。

9.2.4 熱方程式に対する後退 Euler 法

まず後退 Euler 法のプログラムを紹介しよう。弱形式は

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

すなわち

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

サンプル・プログラムを用意してある。

ターミナルではこうして入手

```
curl -O http://nalab.mind.meiji.ac.jp/~mk/program/fem/heatB.edp
```

次の次のスライドに載せてある。

ループの制御変数を `i` として、`problem` に `,init=i` と書き足すのがミソ。最初は `i` が 0 であるので `init` は `false`、それ以降は $\mathtt{i} \neq 0$ であるので `init` は `true`, と指示するのが工夫 (そうしないと毎ステップで行列を再構成してしまう)。連立 1 次方程式の係数行列が時刻 (したがって n) に依らないことに注意しよう。

```
// heatB.edp --- 熱方程式を後退 Euler 法 (陰解法) で解く
// http://nalab.mind.meiji.ac.jp/~mk/program/fem/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 i,m=10;
real Tmax=10, tau=0.01, t;
mesh Th=square(m,m);
plot(Th,wait=true);
fespace Vh(Th,P1);
func f=1; func g1=0; func g2=0;
func u0=sin(pi*x)*sin(pi*y);
Vh u=u0, uold, v;
plot(u,cmm="t=0",wait=1);
problem heat(u,v,init=i)=
  int2d(Th)(u*v+tau*(dx(u)*dx(v)+dy(u)*dy(v)))-int2d(Th)(uold*v)
 -int2d(Th)(tau*f*v)-int1d(Th,2,3)(tau*g2*v)
 +on(1,4,u=g1);
for (i=0;i<Tmax/tau;i++) {
  uold=u;
  t=(i+1)*tau;
  heat;
  plot(u,cmm="t="+t,wait=0); // ps="heat"+i+".ps" として保存することも可能
}
plot(u,wait=1);
```

付録: solve と problem にかかわるパラメーター

FreeFem++ ドキュメンテーションの §3.3.13 (Hecht [1]) から。

付録: solve と problem にかかわるパラメーター

FreeFem++ ドキュメンテーションの §3.3.13 (Hecht [1]) から。

- `solver=` には LU, CG, Crout, Cholesky, GMRES, sparsesolver, UMFPACK が指定できる (最初の 5 つはアルゴリズムの名前, 説明省略)。
  デフォールトでは sparsesolver で、それは他の sparsesolver が定義されていなければ UMFPACK に等しい。それは他の直接法のソルバーが使えない場合は LU にセットされる。
  行列のメモリーへの格納の仕方は、solver により決まる。LU の場合は非対称なスカイライン (説明省略)、Crout の場合は対称なスカイライン、Cholesky の場合は正値対称なスカイライン、CG の場合は正値対称な疎行列、その他 (GMRES, sparsesolver, UMFPACK) では疎行列。

付録: solve と problem にかかわるパラメーター

FreeFem++ ドキュメンテーションの §3.3.13 (Hecht [1]) から。

- `solver=` には LU, CG, Crout, Cholesky, GMRES, sparsesolver, UMFPACK が指定できる (最初の 5 つはアルゴリズムの名前, 説明省略)。
  デフォールトでは sparsesolver で、それは他の sparsesolver が定義されていなければ UMFPACK に等しい。それは他の直接法のソルバーが使えない場合は LU にセットされる。
  行列のメモリーへの格納の仕方は、solver により決まる。LU の場合は非対称なスカイライン (説明省略)、Crout の場合は対称なスカイライン、Cholesky の場合は正値対称なスカイライン、CG の場合は正値対称な疎行列、その他 (GMRES, sparsesolver, UMFPACK) では疎行列。
- `init=`論理型の式
  `false` または 0 のとき、行列が再構成 (reconstruct) される、とある。初期化されているかどうか、という意味か？

付録: solve と problem にかかわるパラメーター

FreeFem++ ドキュメンテーションの §3.3.13 (Hecht [1]) から。

- solver= には LU, CG, Crout, Cholesky, GMRES, sparsesolver, UMFPACK が指定できる (最初の 5 つはアルゴリズムの名前, 説明省略)。
  デフォールトでは sparsesolver で、それは他の sparsesolver が定義されていなければ UMFPACK に等しい。それは他の直接法のソルバーが使えない場合は LU にセットされる。
  行列のメモリーへの格納の仕方は、solver により決まる。LU の場合は非対称なスカイライン (説明省略)、Crout の場合は対称なスカイライン、Cholesky の場合は正値対称なスカイライン、CG の場合は正値対称な疎行列、その他 (GMRES, sparsesolver, UMFPACK) では疎行列。
- init=論理型の式
  false または 0 のとき、行列が再構成 (reconstruct) される、とある。初期化されているかどうか、という意味か？
- eps=実数型の式
  反復法の停止則を指定する。
  $\varepsilon < 0$ の場合は $\|Ax - b\| < |\varepsilon|$, $\varepsilon > 0$ の場合は $\|Ax - b\| < \frac{|\varepsilon|}{\|Ax_0 - b\|}$
  (と書いてあるけれど、$\frac{\|Ax-b\|}{\|Ax_0-b\|} < |\varepsilon|$ の間違いではないかな？)

(連立 1 次方程式のアルゴリズムを学んだことがないと、少し分かりにくいかもしれない…)

9.2.5 熱方程式に対する θ 法

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

とおいて

$$\frac{u^{n+1}-u^n}{\Delta t} = \triangle u^{n+\theta} + f \quad (\text{in } \Omega), \tag{6a}$$

$$u^{n+1} = g_1 \quad (\text{on } \Gamma_1), \tag{6b}$$

$$\frac{\partial u^{n+1}}{\partial n} = g_2 \quad (\text{on } \Gamma_2). \tag{6c}$$

(もし f や g_2 が時間依存する場合は、f や g_2 は $t = t_{n+1}$ のときの値を用いる。)

9.2.5 熱方程式に対する θ 法

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

とおいて

$$\text{(6a)} \quad \frac{u^{n+1}-u^n}{\Delta t} = \triangle u^{n+\theta} + f \quad (\text{in } \Omega),$$

$$\text{(6b)} \quad u^{n+1} = g_1 \quad (\text{on } \Gamma_1),$$

$$\text{(6c)} \quad \frac{\partial u^{n+1}}{\partial n} = g_2 \quad (\text{on } \Gamma_2).$$

(もし f や g_2 が時間依存する場合は、f や g_2 は $t = t_{n+1}$ のときの値を用いる。)
弱形式は

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

すなわち

$$\text{(8)} \quad \left(u^{n+1}, v\right) - (u^n, v) + \Delta t\theta\langle u^{n+1}, v\rangle + \Delta t(1-\theta)\langle u^n, v\rangle - \Delta t(f, v) - \Delta t[g_2, v] = 0.$$

9.2.5 熱方程式に対する θ 法

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

とおいて

$$\text{(6a)} \quad \frac{u^{n+1}-u^n}{\Delta t} = \triangle u^{n+\theta} + f \quad (\text{in } \Omega),$$

$$\text{(6b)} \quad u^{n+1} = g_1 \quad (\text{on } \Gamma_1),$$

$$\text{(6c)} \quad \frac{\partial u^{n+1}}{\partial n} = g_2 \quad (\text{on } \Gamma_2).$$

(もし f や g_2 が時間依存する場合は、f や g_2 は $t = t_{n+1}$ のときの値を用いる。)
弱形式は

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

すなわち

$$\text{(8)} \quad \left(u^{n+1}, v\right) - (u^n, v) + \Delta t\theta\langle u^{n+1}, v\rangle + \Delta t(1-\theta)\langle u^n, v\rangle - \Delta t(f, v) - \Delta t[g_2, v] = 0.$$

記憶用には (7) の方が短くて便利、プログラムを書くには (8) の方が便利であろう。

9.2.6 実習課題

9.2.6 実習課題

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。

9.2.6 実習課題

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。

9.2.6 実習課題

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 ($t = \mathrm{T}_{\mathrm{max}}$) 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`)。

9.2.6 実習課題

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。)

9.2.6 実習課題

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。

9.2.6 実習課題

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 ($t = \mathrm{T}_{\mathrm{max}}$) 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。

9.2.7 その他

時間発展問題の理論的解析について、日本語で読めるテキストはほとんどない (Poisson 方程式の解析より一段以上高度である)。

齊藤 [2] は貴重である。(齊藤 [3] というのもある)。

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。

(9a) $$\frac{1}{c^2}u_{tt} = \triangle u \quad ((x,y,t) \in \Omega \times (0,\infty)),$$

(9b) $$u(x,y,t) = 0 \quad ((x,y,t) \in \partial\Omega \times (0,\infty)),$$

(9c) $$u(x,y,0) = \phi(x,y), \quad u_t(x,y,0) = \psi(x,y) \quad ((x,y) \in \overline{\Omega}).$$

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

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。

(9a) $$\frac{1}{c^2}u_{tt} = \triangle u \quad ((x,y,t) \in \Omega \times (0,\infty)),$$

(9b) $$u(x,y,t) = 0 \quad ((x,y,t) \in \partial\Omega \times (0,\infty)),$$

(9c) $$u(x,y,0) = \phi(x,y), \quad u_t(x,y,0) = \psi(x,y) \quad ((x,y) \in \overline{\Omega}).$$

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

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 (以下のサンプル・プログラムで採用)

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

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。

(9a) $$\frac{1}{c^2}u_{tt} = \triangle u \quad ((x,y,t) \in \Omega \times (0,\infty)),$$

(9b) $$u(x,y,t) = 0 \quad ((x,y,t) \in \partial\Omega \times (0,\infty)),$$

(9c) $$u(x,y,0) = \phi(x,y), \quad u_t(x,y,0) = \psi(x,y) \quad ((x,y) \in \overline{\Omega}).$$

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

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

9.3.2 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)。

9.3.2 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)。

ターミナルではこうして入手

```
curl -O http://nalab.mind.meiji.ac.jp/~mk/program/fem/taiko.edp
```

9.3.2 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)。

ターミナルではこうして入手

```
curl -O http://nalab.mind.meiji.ac.jp/~mk/program/fem/taiko.edp
```

`plot()` で $\boxed{\texttt{,dim=3}}$ 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。

9.3.2 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);
```

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] 齊藤宣一：熱方程式に対する有限要素法と誤差解析, 東大数理科学研究科の「応用数理特別講義 III」(2004 June) の講義ノートの縮小版。今は公開していないみたい。(2006 年 3 月 29 日).

[3] 齊藤宣一：発展方程式の数値解析 — 最大値原理, 解析半群と有限要素法,
http://www.infsup.jp/saito/materials/110827tsukuba1.pdf (2011/8/27).