FreeFem++ 実習の手引き

桂田 祐史

2024 年 6 月 11 日, 2024 年 6 月 18 日

1 参考になる情報

“マニュアル” PDF は

/Applications/FreeFem++.app/Contents/ff-4.14/share/FreeFEM/FreeFEM-documentation.pdf

にある。この PDF を検索する、というのが 1 つの手段である。

例えばホームディレクトリにリンクを貼るには

```
ln -s /Applications/FreeFem++.app/Contents/ff-4.14/share/FreeFEM/FreeFEM-documentation.pdf ~
```

唯一の書籍である大塚・高石 [?] は現在品切れだが、明治大学の学生は、Maruzen eBook でアクセスできる。

https://elib.maruzen.co.jp/elib/html/BookDetail/Id/3000018545

ネット上にそれなりの情報がある。
拙作

- 「FreeFem の紹介」https://m-katsurada.sakura.ne.jp/labo/text/welcome-to-freefem/
- 「FreeFem++ノート」https://m-katsurada.sakura.ne.jp/labo/text/freefem-note/
  文法メモのようなこと — かなりの部分は授業で説明済み

応用数理学会のチュートリアル (英語だけれど、マニュアルよりきちんとしている)

- Suzuki, A.: Finite element programming by FreeFem++ –intermediate course, 日本応用数理学会「産業における応用数理」研究部会のソフトウェアセミナー「FreeFem++ による有限要素プログラミング – 中級編 –」(2016/2/11-12) の配布資料で、https://www.ljll.math.upmc.fr/~suzukia/FreeFempp-tutorial-JSIAM2016/ から入手できる (2016).
- Suzuki, A.: Finite element programming by FreeFem++ –advanced course, 日本応用数理学会「産業における応用数理」研究部会のソフトウェアセミナー「FreeFem++ による有限要素プログラミング – 中級編 –」(2016/6/4-5) の配布資料で、https://www.ljll.math.upmc.fr/~suzukia/FreeFempp-tutorial-JSIAM2016b/ から入手できる (2016).

2 プログラム作成実習

多くの場合、類似した問題のプログラムを探してみるのが実際的であろう。

2.1 Laplace 方程式を解いてみよう

次のような Laplace 方程式の境界値問題を考えよう。

$$\Omega = \left\{(x,y)\in\mathbb{R}^2 \,\middle|\, x^2+y^2<1\right\},$$

$$\begin{aligned}
\gamma_1 &: (x,y) = (\cos\theta,\sin\theta) \quad (\theta\in[0,\pi/6]),\\
\gamma_2 &: (x,y) = (\cos\theta,\sin\theta) \quad (\theta\in[\pi/6,\pi]),\\
\gamma_3 &: (x,y) = (\cos\theta,\sin\theta) \quad (\theta\in[\pi,4\pi/3]),\\
\gamma_4 &: (x,y) = (\cos\theta,\sin\theta) \quad (\theta\in[4\pi/3,2\pi]).
\end{aligned}$$

$v_n\colon \partial\Omega\to\mathbb{R}$ を

$$v_n := \begin{cases} -2 & (\text{on } \gamma_1)\\ 1 & (\text{on } \gamma_3)\\ 0 & (\text{on } \gamma_2\cup\gamma_4)\end{cases}$$

で与える。

次の Laplace 方程式の境界値問題を考える。

$$\triangle u = 0 \quad \text{in } \Omega,\qquad \frac{\partial u}{\partial n} = v_n \quad \text{on } \partial\Omega$$

これは実は 2 次元非圧縮ポテンシャル流の速度ポテンシャル u を求める問題である。速度場を $\boldsymbol{v}$ とすると、境界上で

$$\frac{\partial u}{\partial n} = \boldsymbol{v}\cdot\boldsymbol{n}$$

が成り立つことが知られているので、

$$\boldsymbol{v}\cdot\boldsymbol{n} = \begin{cases} -2 & (\text{on } \gamma_1)\\ 1 & (\text{on } \gamma_3)\\ 0 & (\text{on } \gamma_2\cup\gamma_4)\end{cases}$$

が成り立っている、ということである。曲線 C と法線ベクトル $\boldsymbol{n}$ に対して線積分

$$\int_C \boldsymbol{v}\cdot\boldsymbol{n}\,ds$$

は、単位時間に C を超えて流れる流体の量 (面積) を意味する。

上の境界値は、

$$(\heartsuit)\qquad \int_{\partial\Omega}\frac{\partial u}{\partial n}\,ds = 0$$

を満たすように選んでいる。実はこの条件は、境界値問題の解が存在するための必要十分条件である。実際、必要性は次のようにして分かる:

$$\int_{\partial\Omega}\frac{\partial u}{\partial n}\,ds = \int_{\partial\Omega}\boldsymbol{v}\cdot\boldsymbol{n}\,ds = \int_\Omega \operatorname{div}\boldsymbol{v}\,d\boldsymbol{x} = \int_\Omega 0\,d\boldsymbol{x} = 0.$$

この問題を解くプログラムを 1 から作成しても良いが、Laplace 方程式や Poisson 方程式のプログラムを探して、それを書き換えることでプログラムを作成してみよう。

(Poisson 方程式 $-\triangle u = f$ で $f=0$ とすると、Laplace 方程式であることに注意。)

Poisson 方程式用のプログラム `poisson-kikuchi.edp` は公開してある。それを叩き台にしてみる。

```
curl -O https://m-katsurada.sakura.ne.jp/program/fem/poisson-kikuchi.edp
cp poisson-kikuchi.edp my-laplace.edp
```

この `my-laplace.edp` を書き換えて行こう。

これは Poisson 方程式の Dirichlet-Neumann 境界値問題を解くプログラムであるが、$f=0$ とすれば Laplace 方程式となる。

+on(Gamma1,) を削除すると Dirichlet 境界条件はなくすことができる。

$\displaystyle\int_{\gamma_1\cup\gamma_2\cup\gamma_3\cup\gamma_4} v_n v\,ds$ は、各パートにわけて与えればよい。例えば`-int1d(Th,gamma1)(g21*v)-int1d(Th,ga`
とする。γ_2 と γ_4 では $v_n=0$ なので、積分を計算する必要はない。γ_1, γ_3 での g_2 を g21, g23 で与えることにした。

解 u の等高線は、いわゆる等ポテンシャル線であるが、その gradient は速度場 $\boldsymbol{v}$ なので、それを描くと良いかもしれない。

```
// ベクトル場の表示
Vh u1,u2;
u1=dx(u);
u2=dy(u);
plot([u1,u2],wait=1);
```

解答例

```
// my-laplace.edp
// https://m-katsurada.sakura.ne.jp/program/fem/poisson-kikuchi.edp を元に
した
// 菊地文雄，有限要素法概説，サイエンス社

border gamma1(t=0,pi/6)      { x=cos(t); y=sin(t);  }
border gamma2(t=pi/6,pi)     { x=cos(t); y=sin(t); }
border gamma3(t=pi,4*pi/3)   { x=cos(t); y=sin(t);  }
border gamma4(t=4*pi/3,2*pi) { x=cos(t); y=sin(t); }
int m=10;
mesh Th = buildmesh(gamma1(m)+gamma2(5*m)+gamma3(2*m)+gamma4(4*m));
plot(Th,wait=true,ps="Th.eps");
savemesh(Th,"Th.msh");
fespace Vh(Th,P1);
Vh u,v;
func f=0;
func g21=-2;
func g23=1;
solve Poisson(u,v)=
  int2d(Th)(dx(u)*dx(v)+dy(u)*dy(v))
 -int2d(Th)(f*v)
 -int1d(Th,gamma1)(g21*v)
 -int1d(Th,gamma3)(g23*v);
plot(u,wait=1,ps="my-laplace.eps");

//3 次元鳥瞰図
//real [int] levels =0.0:0.01:1.0;
//plot(u,dim=3,viso=levels,fill=true,wait=true);

// ベクトル場の表示
Vh u1,u2;
u1=dx(u);
u2=dy(u);
plot([u1,u2],wait=1);
```

余談　また、いわゆる流線を描くために、流れ関数 ψ を求めて、その等高線として描くこともできる。流れ関数の境界上の値 $\psi|_{\partial\Omega}$ は、v_n から線積分でもとまる:

$$\psi(\boldsymbol{x}) = \int_{C_{\boldsymbol{x}}} v_n\, ds.$$

これを使えば、Laplace 方程式の Dirichlet 境界値問題の解として求まる。

2.2　熱方程式の θ 法のプログラムを書いてみよう

前回講義の資料

https://m-katsurada.sakura.ne.jp/lecture/ouyousuuchikaisekitokuron-2024/ANA07_0604_handout.pdf#page=27

に取り組んでみよう。