

応用数値解析特論メモ No. 3 (書きかけ)

桂田 祐史

2026年4月28日, 2026年4月29日

1 FreeFEM に触れてみる

この講義は、前半のうち板書コーギが主体になるけれど (PDF 資料 <https://m-katsurada.sakura.ne.jp/ana2026/nonopen/fem.pdf> の解説をする)、こういうことが出来るようになる、というのを知っておこう。

注意 FreeFEM は以前は FreeFem++ と呼ばれていたが、今では FreeFEM と呼ぶことになったらしい (でも、今でもコマンドは FreeFem++ だな)。

1.1 手短な紹介

FreeFEM は、2次元, 3次元領域における偏微分方程式の問題を有限要素法で解くための、一種の PSE (problem solving environment) である。考えている問題の適切な弱形式が分かっているならば (弱形式の求め方はこれから講義するのである)、簡単なプログラムでシミュレーションできる。

パリ第6大学 J. L. Lions 研究所の Frédéric Hecht, Oliver Pironneau, A. Le Hyaric, 広島国際学院大学の 大塚厚二氏らによって開発された。(なんとと言ってもメインの開発者は Hecht であるが、つい最近一線を退いた?) フリーソフトウェアとして提供されている。

1.2 入手

「FreeFEM — An open-source PDE Solver using the Finite Element Method」¹ が総本山。Download² から入手できる。2026年4月20日の最新版は 4.16 であるが、現在 Mac 版の提供がされていないので、Mac を使う人は 4.15 を使うと良い。

- Apple Silicon Mac では、<https://github.com/FreeFem/FreeFem-sources/releases/download/v4.15/FreeFEM-v4.15-Apple-Silicon-15.4.dmg> を
- Intel Mac では、<https://github.com/FreeFem/FreeFem-sources/releases/download/v4.15/FreeFEM-v4.15-Intel-0.dmg> を

入手する。

ターミナルから FreeFEM-v4.15-Apple-Silicon-15.4.dmg を入手

```
curl -OL https://github.com/FreeFem/FreeFem-sources/releases/download/v4.15/FreeFEM-v4.15-Apple-Silicon-15.4.dmg
```

¹<https://freefem.org/>

²<https://github.com/FreeFem/FreeFem-sources/releases>

1.3 インストール

Intel Mac で、FreeFEM-v4.15-Intel-0.dmg をインストールする手順については、

「FreeFem++ 4.15 をインストールしてみた」

<https://m-katsurada.sakura.ne.jp/knowhow-2024/node67.html>

を参考にして下さい。

Apple Silicon Mac で FreeFEM-v4.15-Apple-Silicon-15.4.dmg をインストールする手順については、以下の説明を見て下さい。実は FreeFEM-v4.15-Apple-Silicon-15.4.dmg は、厳密には v4.15 ではなく、v4.15.1 であるらしく、上の「FreeFem++ 4.15 をインストールしてみた」はまったく参考になりません。

FreeFEM-v4.15-Apple-Silicon-15.4.dmg を (例えば Finder でダブルクリックして) 開いたあと、ターミナルで次のようにする。

```
cd /Volumes/freefem4.5.1
bash ./Install-app.sh
```

パスワードを尋ねられるので入力する。画面に表示されないが、パスワードを入力した後に `enter` キーをタイプする

これはエラーになる。その原因が知りたければ、「FreeFEM v4.15 (実は v4.15.1) でトラブル」³ を見て下さい。以下では原因の説明は省き、インストール手順のみ説明します。

この後、ターミナルで次の2つのどちらかを実行する。

対策1 (末尾のドット `.` は忘れずに)

```
sudo mkdir -p /opt/homebrew/opt/gcc/lib/gcc/current/
cd /opt/homebrew/opt/gcc/lib/gcc/current/
sudo ln -s /Applications/FreeFem++.app/Contents/ff-4.15.1/gnu/libgfortran.5.dylib .
sudo ln -s /Applications/FreeFem++.app/Contents/ff-4.15.1/gnu/libquadmath.0.dylib .
```

(FreeFem++.app は、Homebrew というものでインストールされたファイルを参照しているので、それを強引に用意する、というもの。Homebrew を使う人は副作用が生じないように注意が必要である。)

対策2

```
cd -
curl -O https://nalab.mind.meiji.ac.jp/~mk/misc/20260421/fix-freefem-rpaths.sh
chmod +x fix-freefem-rpaths.sh
sudo ./fix-freefem-rpaths.sh /Applications/FreeFem++.app
```

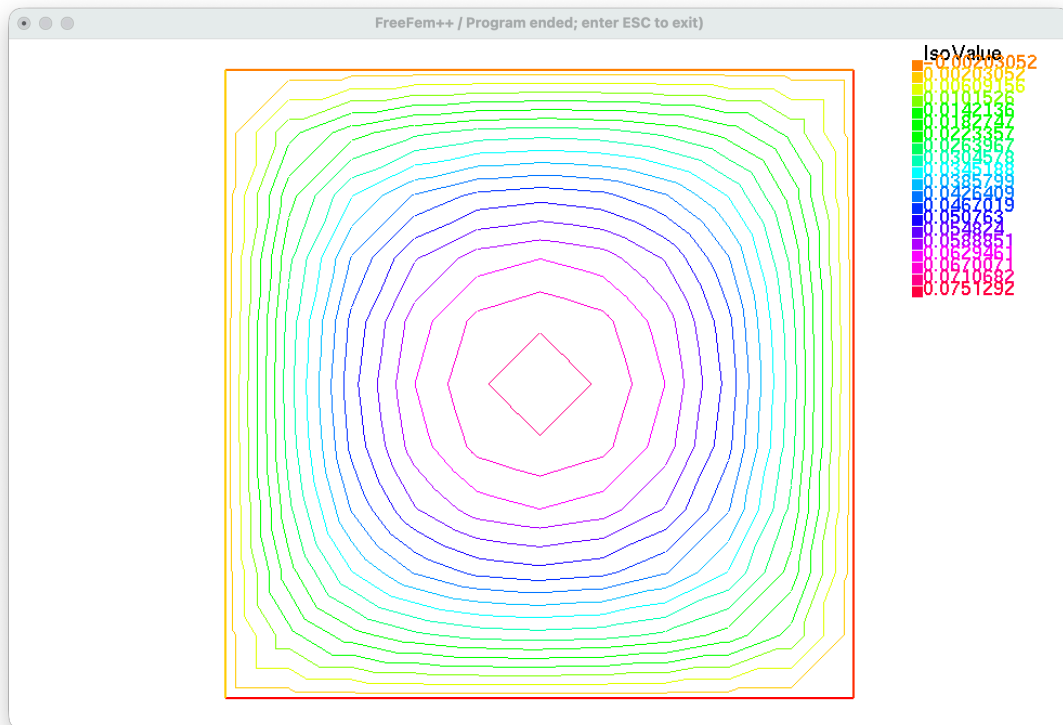
(FreeFEM のパッケージで用意されているファイルを参照するように、FreeFem++.app を修正する。)

動作チェックをしよう。新しいターミナルを出して

サンプル・プログラム Laplace.edp を実行してみる

```
cp -pr /Applications/FreeFem++.app/Contents/ff-4.15.1/share/FreeFEM/4.15/examples .
cd examples/tutorial/
FreeFem++ Laplace.edp
```

³<https://m-katsurada.sakura.ne.jp/knowhow-2026/node3.html>



このウィンドウが出れば正常にインストールできた可能性が高い。

1.4 情報入手

Documentation として、Hecht [1] がある。マニュアルというよりは、膨大な事例集と呼ぶのが適切な感じがする。(インストールすると、

`/Applications/FreeFem++.app/Contents/ff-4.15.1/share/FreeFEM/FreeFEM-documentation.pdf`

に置かれているはず。)

大塚・高石 [2] という日本語テキストがある (現在品切れのようだが、Maruzen eBook で読むことが出来る)。

自分でプログラムを書く場合は、上記の資料を参考にすると良い。FreeFEM の文法については、桂田 [3] というメモも参考になるだろう。

1.5 試しにサンプル・プログラムを実行する

桂田 [4] に色々なサンプル・プログラムがある。以下、それを動かしてみよう。

1.5.1 Poisson 方程式

桂田 [4] の該当箇所

<https://m-katsurada.sakura.ne.jp/labo/text/welcome-to-freefem/node5.html>

$$\Omega := \{(x, y) \in \mathbb{R}^2 \mid x^2 + y^2 < 1\}$$

$$(1) \quad -\Delta u = f \quad (\text{in } \Omega), \quad u = 0 \quad (\text{on } \partial\Omega).$$

ただし

$$f(x, y) = xy \quad ((x, y) \in \Omega)$$

とする。

ターミナルでこうすれば実行できる

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

plot() コマンドに wait=true があると、そこで一時停止する。前に進めるためには、`enter` キーを叩く。最後は `esc` 入力で終了する。

FreeFEM のプログラムには、.edp という拡張子をつけることになっている。フランス語で偏微分方程式を équations aux dérivées partielles というのでその頭文字を取ったのだとか。(英語では partial differential equation を PDE と訳すことがよくある。)

version 4.15 から Markdown 形式のファイルも入力できるようになったそうである。

境界値問題 (1) の弱形式は

$$\int_{\Omega} (u_x v_x + u_y v_y) dx dy = \int_{\Omega} f v dx dy \quad (v \in X)$$

である ($f = 0$ の Laplace 方程式の場合、前回講義の Dirichlet 原理のところで一瞬顔を見せた式である。)。弱形式については、今後の講義で詳しく解説する。次のプログラムの 20 行目にこの式が「書いてある」。

19-21 行で一つの命令であるが、何とこれで有限要素解が求まり、以下は結果の表示をしているだけである。

テキスト・エディターで poisson.edp を編集してから、FreeFem++ poisson.edp を再実行してみると良い。

poisson.edp

```
1 // poisson.edp
2 // https://m-katsurada.sakura.ne.jp/program/fem/poisson.edp
3 // https://m-katsurada.sakura.ne.jp/labo/text/welcome-to-freefem/node4.html
4
5 // 境界の定義 (単位円), いわゆる正の向き
6 border Gamma(t=0,2*pi) { x=cos(t); y=sin(t); }
7 // 三角形要素分割を生成 (境界を 50 に分割)
8 mesh Th = buildmesh(Gamma(50));
9 plot(Th,wait=true); // plot(Th,wait=true,ps="Th.eps");
10 // 有限要素空間は P1 (区分的1次多項式) 要素
11 fespace Vh(Th,P1);
12 Vh u,v;
13 // Poisson 方程式  $-\Delta u=f$  の右辺
14 func f = x*y;
15 // 現在時刻をメモ
16 real start = clock();
17 // 問題を解く
18 solve Poisson(u,v)
19 = int2d(Th) (dx(u)*dx(v)+dy(u)*dy(v))-int2d(Th) (f*v)
20 +on(Gamma,u=0);
21 // 可視化 (等高線)
22 plot(u,wait=true);
23 // 塗りつぶし
24 real [int] levels =-0.012:0.001:0.012;
25 //plot(u,viso=levels,fill=true,wait=true);
26 // 可視化 (3次元) --- マウスで使って動かせる
27 //plot(u,dim=3,wait=true);
28 plot(u,dim=3,viso=levels,fill=true,wait=true);
29 // 計算時間を表示
30 cout << " CPU time= " << clock() - start << endl;
```

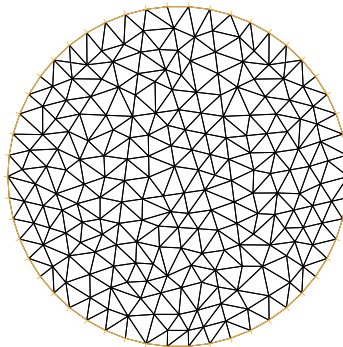


図 1: 三角形分割

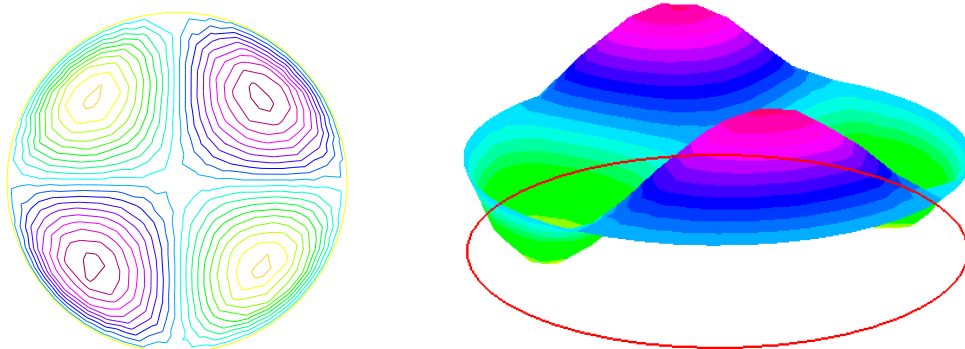


図 2: 等高線と鳥瞰図

1.5.2 熱伝導方程式

桂田 [4] の該当箇所

<https://m-katsurada.sakura.ne.jp/labo/text/welcome-to-freefem/node11.html>

領域の内部で熱が発生する (or 熱が吸収される) 場合の熱方程式の初期値境界値問題を解いてみよう。

$$\begin{aligned}\Omega &:= \{(x, y) \in \mathbb{R}^2 \mid 0 < x < 1, 0 < y < 1\} \\ \Gamma_1 &= \{(0, y) \mid 0 \leq y \leq 1\} \cup \{(x, 0) \mid 0 \leq x \leq 1\}, \\ \Gamma_2 &= \{(1, y) \mid 0 \leq y \leq 1\} \cup \{(x, 1) \mid 0 \leq x \leq 1\}.\end{aligned}$$

つまり Ω は正方形領域、 Γ_1 は左と下の辺を合わせたもの、 Γ_2 は右と上の辺を合わせたものである。

$$\begin{aligned}(2a) \quad & u_t(x, y, t) = \Delta u(x, y, t) + f(x, y) \quad ((x, y, t) \in \Omega \times (0, \infty)), \\ (2b) \quad & u(x, y, t) = g_1(x, y) \quad ((x, y, t) \in \Gamma_1 \times (0, \infty)), \\ (2c) \quad & \frac{\partial u}{\partial n}(x, y, t) = g_2(x, y) \quad ((x, y, t) \in \Gamma_2 \times (0, \infty)), \\ (2d) \quad & u(x, y, 0) = u_0(x, y) \quad ((x, y) \in \bar{\Omega})\end{aligned}$$

以下のプログラムでは

$$f(x, y) = 1, \quad g_1(x, y) = 0, \quad g_2(x, y) = 0, \quad u_0(x, y) = \sin(\pi x) \sin(\pi y)$$

としてある。

heatB.edp

```
1 // heatB.edp --- 熱方程式を後退 Euler 法 (陰解法) で解く
2 // https://m-katsurada.sakura.ne.jp/program/fem/heatB.edp
3 // 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
4 int i,m=10;
5 real Tmax=1, tau=0.01, t;
6 // 次の2行の行頭の // を削除すると、m, tau, theta が実行時に入力できる
7 // cout << "m dt theta: "; cin >> m >> tau >> theta;
8 // cout << "m=" << m << ", tau=" << tau << ", theta=" << theta << endl;
9 mesh Th=square(m,m);
10 plot(Th,wait=true);
11 fespace Vh(Th,P1);
12 func f=1; func g1=0; func g2=0;
13 func u0=sin(pi*x)*sin(pi*y);
14 Vh u=u0, uold, v;
15 plot(u,cmm="t=0",wait=1);
16 problem heat(u,v,init=i)=
17   int2d(Th)(u*v)+int2d(Th)(tau*(dx(u)*dx(v)+dy(u)*dy(v)))-int2d(Th)(uold*v)
18   -int2d(Th)(tau*f*v)-int1d(Th,2,3)(tau*g2*v)
19   +on(1,4,u=g1);
20 for (i=0;i<Tmax/tau;i++) {
21   uold=u;
22   t=(i+1)*tau;
23   heat;
24   plot(u,cmm="t="+t,wait=0); // ps="heat"+i+".ps" として保存することも可能
25 }
26 plot(u,wait=1);
```

ターミナルでこうすれば実行できる

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

時間発展する系なので、動画で見るのが分かりやすいであろう。

<https://m-katsurada.sakura.ne.jp/ana2026/heat.mp4>

定常解 v に収束することが分かるが、実は、Poisson 方程式の境界値問題

$$-\Delta v = f \quad (\text{in } \Omega), \quad v = g_1 \quad (\text{on } \Gamma_1), \quad \frac{\partial v}{\partial n} = g_2 \quad (\text{on } \Gamma_2)$$

の解となっている (<https://m-katsurada.sakura.ne.jp/labo/text/welcome-to-freefem/node8.html>)。

1.5.3 一様流の中に置かれた円柱のまわりの流れ

桂田 [4] の該当箇所

<https://m-katsurada.sakura.ne.jp/labo/text/welcome-to-freefem/node12.html>

日本応用数学会の「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)」で鈴木厚先生が解説された資料を拝借している。

このセミナーの資料は (セミナー当時とは違って) 以下から入手できる。

<https://www.ljll.fr/~suzukia/FreeFempp-tutorial-JSIAM2016/>

応用数値解析の過去の講義で、このプログラムを解説している。そのときのスライド資料

https://m-katsurada.sakura.ne.jp/ana2022/ANA12_1219_handout.pdf#page=6

(方程式等はこの資料を見て下さい。)

ターミナルでこうすれば実行できる

```
curl -O https://www.ljll.fr/~suzukia/FreeFempp-tutorial-JSIAM2016/EDP/NS-cylinder.edp
FreeFem++ NS-cylinder.edp
```

(流線を表示していることが多いが、時々速度場の矢印と圧力の等高線を表示する。)

これも時間発展する系なので、動画で見てみよう。

- <https://m-katsurada.sakura.ne.jp/ana2026/NSstream.mp4> 流線
- <https://m-katsurada.sakura.ne.jp/ana2026/NSvp.mp4> 速度場と圧力場

2 Poisson 方程式に対する Ritz-Galerkin 法

「有限要素法への入門」

<https://m-katsurada.sakura.ne.jp/ana2026/nonopen/fem.pdf>

に従い講義する (パスワードはシラバスの補足に書いてあります)。

2026/4/21 は、問題 (V) を書いたところまで (§2.3 に入ったところ)。

参考文献

- [1] Hecht, F.: Freefem++ (Documentation), <https://github.com/FreeFem/FreeFem-doc/raw/pdf/FreeFEM-documentation.pdf>, <http://www3.freefem.org/ff++/ftp/freefem++doc.pdf>.
- [2] 大塚厚二, 高石武史: 有限要素法で学ぶ現象と数理 — FreeFem++ 数値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.
- [3] 桂田祐史: FreeFEM++ ノート, <https://m-katsurada.sakura.ne.jp/labo/text/freefem-note.pdf> (2012~).
- [4] 桂田祐史: FreeFEM++ の紹介, <https://m-katsurada.sakura.ne.jp/labo/text/welcome-to-freefem/> (2007~).