応用数値解析特論 第 11 回

～Navier-Stokes 方程式に対する有限要素法 (2)～
定常 Navier-Stokes 方程式に対する Newton 法

かつらだ まさし
桂田 祐史
http://nalab.mind.meiji.ac.jp/~mk/ana2021/

2021 年 12 月 6 日

目次

本日の講義内容、連絡事項

- 定常 Navier-Stokes 方程式の境界値問題を解く 1 つのアルゴリズムを解説する。キーワードは Newton 法である。
- Newton 法の参考資料
  - 「非線形方程式、特に代数方程式の数値解法」 (桂田 [?])
  - 「多次元 Newton 法の例」
    http://nalab.mind.meiji.ac.jp/~mk/labo/text/Newton/
    $-u''(x) = u(x)^2$ $(x \in (0,1))$, $u(0) = u(1) = 0$ を差分法で解くプログラムの紹介。

11.4 定常 Navier-Stokes 方程式に対する Newton 法

11.4.1 例題プログラムの紹介

前回、定常 Stokes 方程式を有限要素法で解いた。今回は定常 Navier-Stokes 方程式を有限要素法で解いてみよう。

例題プログラムは FreeFem++ 4.9 に付属する

`/usr/local/ff++/share/FreeFEM/4.9/examples/examples/NSNewton.edp`
(2012 January, Hecht)

である。

ターミナルでこうすればコピーできる (4.9 インストールした人向け)

```
cp -p /usr/local/ff++/share/FreeFEM/4.9/examples/examples/NSNewton.edp .
```

テキスト・エディターで開いて中身を読んでみよう。

これはマニュアル (Hecht [?]) の §2.12 に掲載されているプログラムとほぼ同じである (本来完全に同じにすべきであろう)。

11.4.2 領域とその三角形分割

Ω は $C_e + B_{eb} + B_{eo} + B_{eu} - C_c$ で囲まれる領域とする。ただし、$R = 5,\ L = 15$,

$$
\begin{aligned}
C_e &: (x, y) = (R\cos t, R\sin t) \quad (t \in [\pi/2, 3\pi/2]),\\
C_c &: (x, y) = ((\cos t)/2, (\sin t)/2) \quad (t \in [0, 2\pi]),\\
B_{eb} &: (x, y) = (Lt^{1.2}, -R) \quad (t \in [0, 1]),\\
-B_{eu} &: (x, y) = (Lt^{1.2}, R) \quad (t \in [0, 1]),\\
B_{eo} &: (x, y) = (L, t) \quad (t \in [-R, R]).
\end{aligned}
$$

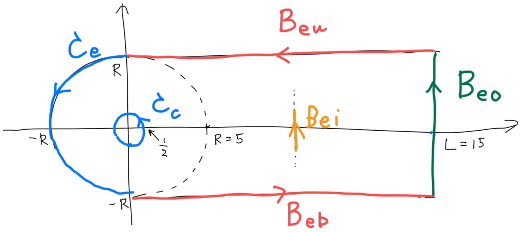

図 1: 領域 Ω ($\partial\Omega = C_e + B_{eb} + B_{eo} + B_{eu} - C_c$)

11.4.2 領域とその三角形分割

$$\Gamma_1 := -C_c + C_e + B_{eb} + B_{eu}, \quad \Gamma_2 := B_{eo}$$

とおく。$\Gamma_1 \cup \Gamma_2 = \partial\Omega$ である (境界条件の種類に応じた分割)。

B_{eb} は $(x, y) = (Lt, R)$ $(t \in [0, 1])$ と書いても同じことであるが、FreeFem++ では t について等分することを想定してプログラムを書いたため、この式を採用しているのであろう。(こういう工夫が実際に効果があるのかどうかは分からない。)

B_{eu} はプログラムでは

```
border beu(tt=1, 0){real t=tt^1.2; x=t*L; y=R; label=1;}
```

として与えられている。FreeFem++ では、`tt=1,0` と書くことで、パラメーターが 1 から 0 まで減少する曲線を表すことが出来る、ということであろう (こういうことが出来ることは、このプログラムを見て初めて知った)。

11.4.2 領域とその三角形分割

三角形分割をする際に

$$B_{ei}: (x, y) = (L/2, t) \quad (t \in [-R/4, R/4])$$

という曲線 (線分) も用いている。

```
Th=buildmesh(cc(-40)+ce(20)+beb(15)+beu(15)+beo(8)+bei(10));
```

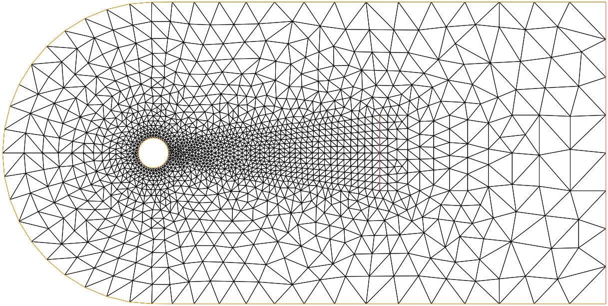

図 2: 領域の三角形分割. 中央部に B_{ei} (朱色の線分) が見える

11.4.3 方程式

このプログラムが解く問題の方程式は (境界条件が、マニュアルにも、プログラムの注釈にも明記されていないが (真面目にやってほしい)$\cdots$)、多分、次のものである。

(1a) $$(\boldsymbol{u}\cdot\nabla)\boldsymbol{u}-\nu\triangle\boldsymbol{u}+\nabla p=0 \quad \text{in } \Omega,$$

(1b) $$\nabla\cdot\boldsymbol{u}=0 \quad \text{in } \Omega,$$

(1c) $$\boldsymbol{u}=\boldsymbol{U} \qquad \text{on } \Gamma_1\setminus C_c,$$

(1d) $$\boldsymbol{u}=\boldsymbol{0} \qquad \text{on } C_c,$$

(1e) $$-p\boldsymbol{n}+\nu\frac{\partial\boldsymbol{u}}{\partial\boldsymbol{n}}=\boldsymbol{0} \quad \text{on } B_{eo}.$$

(判断の根拠: 1 というラベルをつけた Γ_1 で $\delta\boldsymbol{u}=\boldsymbol{0}$ としていることから、そこで Dirichlet 境界条件をつけているのだろう。Newton 法の初期値 $\boldsymbol{u}_0$ は

$$\boldsymbol{u}_0=\begin{cases}(1,0)^\top & (x^2+y^2>2)\\ (0,0)^\top & (x^2+y^2\le 2)\end{cases}$$

であったから、$\delta\boldsymbol{u}$ も Γ_1 では $\boldsymbol{0}$ である。)

(1e) は自然境界条件である (弱形式には explicit に現れない — Poisson 方程式の問題における同次 Neumann 境界条件のようなもの)。

一様流 ($\boldsymbol{U}:=(1,0)^\top$) の中に置かれた円柱の周りの流れを求めよという問題である。

前回スライドp. 15「もう 1 つの弱形式」と類似点が多い。見比べてみよう。

11.4.4 問題点と対処の方針

問題になることと、それへの対処について、次の 2 点を指摘しておく。

(a) Navier-Stokes 方程式は非線形方程式であるので、一度に解を求めるのが難しい。Newton 法を用いて解くことにする (反復計算で近似解を求める)。

(b) 動粘性係数 $\nu = \frac{1}{200}$ は素朴な数値計算にとっては、かなり小さい ("円柱" の直径 1 であるから、これを代表的な長さ、$\|\boldsymbol{U}\| = 1$ を代表的な速さに選ぶと、ν の逆数が Reynolds 数 R_e に相当するので、$R_e = 200$ ということになり、これが実はかなり大きい、ということである)。Newton 法の初期値は真の解に十分近いものを選ばないと、反復が収束しないかもしれない。そのため、大きめの ν についての解を求め、それを小さい ν の問題の Newton 法の初期値とすることで、解きたい ν に近づいていく。

11.4.5 Newton 法の復習

方程式 $f(u) = 0$ の近似解を求めるための Newton 法とは、まず適当な初期値 u_0 を選び、漸化式

$$u_{i+1} = u_i - f'(u_i)^{-1} f(u_i) \tag{2}$$

により $\{u_i\}$ を定め、十分大きな i に対して、u_i を近似解に採用する、というものである。

(2) は、$f(u) = 0$ を $u = u_i$ で線形近似した

$$f'(u_i)(u - u_i) + f(u_i) = 0$$

の解を u_{i+1} とする、ということである。

$w_i := f'(u_i)^{-1} f(u_i)$ とおくと、w_i は $f'(u_i) w_i = f(u_i)$ の解であることに注意すると、次のアリゴリズムを得る (FreeFem++ のマニュアルでは、F の微分は DF と表している)。

Find $u \in V$ such that $f(u) = 0$ where $f\colon V \to V$.

1. choose $u_0 \in V$
2. `for (i=0; i<niter; i=i+1)`
    1. Solve $f'(u_i) w_i = f(u_i)$
    2. $u_{i+1} := u_i - w_i$
       break if $\|w_i\| < \varepsilon$　(修正量が小さくなったら反復を停止する)

11.4.6 我々の問題での f と f' は？

$f = 0$ が (1a) –(1e) と一致するようにするには

$$f(\boldsymbol{u}, p) := \begin{pmatrix} (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} - \nu\triangle\boldsymbol{u} + \nabla p \\ \nabla\cdot u \\ \boldsymbol{u}\big|_{\Gamma_1} - \boldsymbol{u}_b \\ -p\boldsymbol{n} + \nu\frac{\partial\boldsymbol{u}}{\partial\boldsymbol{n}}\Big|_{B_{eo}} \end{pmatrix}$$

とすれば良いだろう。ここで $\boldsymbol{u}_b$ は次式で定まる Γ_1 上の関数である。

$$\boldsymbol{u}_b := \begin{cases} (1,0)^\top & (\text{on } \Gamma_1 \setminus C_c) \\ (0,0)^\top & (\text{on } C_c). \end{cases}$$

11.4.6 我々の問題での f と f' は？　(続き)

$$
\begin{aligned}
&f(\boldsymbol{u}+\delta\boldsymbol{u}, p+\delta p) - f(\boldsymbol{u}, p)\\
&= \begin{pmatrix}
((\boldsymbol{u}+\delta\boldsymbol{u})\cdot\nabla)\,\boldsymbol{u} - \nu\triangle(\boldsymbol{u}+\delta\boldsymbol{u}) + \nabla(p+\delta p) - ((\boldsymbol{u}\cdot\nabla)\,\boldsymbol{u} - \nu\triangle\boldsymbol{u} + \nabla p)\\
\nabla\cdot(\boldsymbol{u}+\delta\boldsymbol{u}) - \nabla\cdot\boldsymbol{u}\\
(\boldsymbol{u}+\delta\boldsymbol{u})|_{\Gamma_1} - \boldsymbol{u}_b - \left(\boldsymbol{u}|_{\Gamma_1} - \boldsymbol{u}_b\right)\\
-(p+\delta p)\boldsymbol{n} + \nu\left.\frac{\partial(\boldsymbol{u}+\delta\boldsymbol{u})}{\partial\boldsymbol{n}}\right|_{B_{eo}} - \left(-p\boldsymbol{n} + \nu\left.\frac{\partial\boldsymbol{u}}{\partial\boldsymbol{n}}\right|_{B_{eo}}\right)
\end{pmatrix}\\
&= \begin{pmatrix}
(\delta\boldsymbol{u}\cdot\nabla)\,\boldsymbol{u} + (\boldsymbol{u}\cdot\nabla)\delta\boldsymbol{u} - \nu\triangle(\delta\boldsymbol{u}) + \nabla(\delta p) + \delta\boldsymbol{u}\cdot\nabla(\delta\boldsymbol{u})\\
\nabla\cdot(\delta\boldsymbol{u})\\
\delta\boldsymbol{u}|_{\Gamma_1}\\
-(\delta p)\boldsymbol{n} + \nu\left.\frac{\partial(\delta\boldsymbol{u})}{\partial\boldsymbol{n}}\right|_{B_{eo}}
\end{pmatrix}.
\end{aligned}
$$

であるから

$$
f'(\boldsymbol{u}, p)\begin{pmatrix}\delta\boldsymbol{u}\\ \delta p\end{pmatrix} = \begin{pmatrix}
(\delta\boldsymbol{u}\cdot\nabla)\,\boldsymbol{u} + (\boldsymbol{u}\cdot\nabla)\delta\boldsymbol{u} - \nu\triangle(\delta\boldsymbol{u}) + \nabla(\delta p)\\
\nabla\cdot(\delta\boldsymbol{u})\\
\delta\boldsymbol{u}|_{\Gamma_1}\\
-(\delta p)\boldsymbol{n} + \nu\left.\frac{\partial(\delta\boldsymbol{u})}{\partial\boldsymbol{n}}\right|_{B_{eo}}
\end{pmatrix}.
$$

(書くと手間だが、要するに 2 次式であるから、慣れると結果がすぐに分かる。)

11.4.7 Newton 反復の弱形式

$f'(\boldsymbol{u}_i, p_i) \begin{pmatrix} \delta\boldsymbol{u} \\ \delta p \end{pmatrix} = f(\boldsymbol{u}_i, p_i)$ を具体的に書くと

$$\text{(3a)}\quad (\delta\boldsymbol{u}\cdot\nabla)\boldsymbol{u}_i + (\boldsymbol{u}_i\cdot\nabla)\,\delta\boldsymbol{u} - \nu\triangle(\delta\boldsymbol{u}) + \nabla(\delta p) = (\boldsymbol{u}_i\cdot\nabla)\,\boldsymbol{u}_i - \nu\triangle\boldsymbol{u}_i + \nabla p_i \quad \text{in } \Omega,$$

$$\text{(3b)}\quad \nabla\cdot\delta\boldsymbol{u} = \nabla\cdot\boldsymbol{u}_i \quad \text{in } \Omega,$$

$$\text{(3c)}\quad \delta\boldsymbol{u} = 0 \quad \text{on } \Gamma_1,$$

$$\text{(3d)}\quad -(\delta p)\boldsymbol{n} + \nu\frac{\partial(\delta\boldsymbol{u})}{\partial\boldsymbol{n}} = \boldsymbol{0} \quad \text{on } B_{eo}.$$

当たり前のことであるが、$\delta\boldsymbol{u}$, δp についての線形方程式である。

最初の 2 つに試験関数 (それぞれ $\boldsymbol{v}$ と q) をかけて積分すると

$$\int_\Omega \left[((\delta\boldsymbol{u}\cdot\nabla)\boldsymbol{u}_i)\cdot\boldsymbol{v} + ((\boldsymbol{u}_i\cdot\nabla)\,\delta\boldsymbol{u})\cdot\boldsymbol{v} - \nu\triangle(\delta\boldsymbol{u})\cdot\boldsymbol{v} + \nabla(\delta p)\cdot\boldsymbol{v}\right] d\boldsymbol{x},$$

$$= \int_\Omega ((\boldsymbol{u}_i\cdot\nabla)\,\boldsymbol{u}_i - \nu\triangle\boldsymbol{u}_i + \nabla p_i)\cdot\boldsymbol{v}\; d\boldsymbol{x}$$

$$\int_\Omega q\,(\nabla\cdot(\delta\boldsymbol{u}))\,d\boldsymbol{x} = 0.$$

それぞれ部分積分して (境界積分を消去するのに、(3d) を用いる)

(4a)
$$\int_\Omega \left[((\delta \boldsymbol{u}\cdot\nabla)\boldsymbol{u}_i)\cdot\boldsymbol{v} + ((\boldsymbol{u}_i\cdot\nabla)\,\delta\boldsymbol{u})\cdot\boldsymbol{v} - \nu\nabla(\delta\boldsymbol{u}) : \nabla\boldsymbol{v} - \delta p(\nabla\cdot\boldsymbol{v})\right] d\boldsymbol{x},$$
$$= \int_\Omega ((\boldsymbol{u}_i\cdot\nabla)\,\boldsymbol{u}_i - \nu\triangle\boldsymbol{u}_i + \nabla p_i)\cdot\boldsymbol{v}\; d\boldsymbol{x}$$

(4b)
$$\int_\Omega q\,(\nabla\cdot(\delta\boldsymbol{u}))\,d\boldsymbol{x} = 0.$$

これが弱形式である。

11.4.8 プログラム中の弱形式解読

次のようなマクロが定義されている。

```
Grad(u1,u2)  [dx(u1),dy(u1),dx(u2),dy(u2)]
```

これは $\nabla \boldsymbol{u}$ を 4 次元ベクトルにしたものである。これから

```
Grad(du1,du2)'*Grad(v1,v2)
```

は

```
dx(du1)*dx(v1)+dy(du1)*dy(v1)+dx(du2)*dx(v2)+dy(du2)*dy(v2)
```

となる。これは $\nabla(\delta \boldsymbol{u}) : \nabla \boldsymbol{v}$ である。

注　FreeFem++ では、' は転置を表す (MATLAB でも使われる、古い記法の利用)。

ベクトルの内積

```
real[int] a=[1,2,3,4];
real[int] b=[-4,-3,-2,-1];
cout << a'*b << endl;
```

これで -20 という値が表示される。

一方

```
UgradV(u1,u2,v1,v2) [[u1,u2]'*[dx(v1),dy(v1)],[u1,u2]'*[dx(v2),dy(v2)]]
```

は

$$\begin{pmatrix} \texttt{u1*dx(v1)+u2*dy(v1)} \\ \texttt{u1*dx(v2)+u2*dy(v2)} \end{pmatrix},$$

すなわち

$$(\boldsymbol{u}\cdot\nabla)\,\boldsymbol{v} = \left(u_1\frac{\partial}{\partial x_1} + u_2\frac{\partial}{\partial x_2}\right)\begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} u_1\frac{\partial v_1}{\partial x_1} + u_2\frac{\partial v_1}{\partial x_2} \\ u_1\frac{\partial v_2}{\partial x_1} + u_2\frac{\partial v_2}{\partial x_2} \end{pmatrix}$$

を意味する。ゆえに

```
UgradV(du1,du2,u1,u2)'*[v1,v2]
```

は $((\delta\boldsymbol{u}\cdot\nabla)\,\boldsymbol{u})\cdot\boldsymbol{v}$ を、

```
UgradV(u1,u2,du1,du2)'*[v1,v2]
```

は $((\boldsymbol{u}\cdot\nabla)\,(\delta\boldsymbol{u}))\cdot\boldsymbol{v}$ を計算する。

11.4.8 プログラム中の弱形式解読 (続き)

```
int2d(Th)(
          nu*(Grad(du1,du2)'*Grad(v1,v2))
        + UgradV(du1,du2, u1, u2)'*[v1,v2]
        + UgradV( u1, u2,du1,du2)'*[v1,v2]
        - div(du1,du2)*q - div(v1,v2)*dp
        - 1e-8*dp*q // stabilization term
)
```

は次式を意味する。

$$\int_\Omega \left[\nu\nabla(\delta\boldsymbol{u}) : \nabla\boldsymbol{v} + ((\delta\boldsymbol{u}\cdot\nabla)\,\boldsymbol{u})\cdot\boldsymbol{v} + ((\boldsymbol{u}\cdot\nabla)\,(\delta\boldsymbol{u}))\cdot\boldsymbol{v} - q\nabla\cdot(\delta\boldsymbol{u}) - \delta p\nabla\cdot\boldsymbol{v} - 10^{-8}(\delta p)q\right] d\boldsymbol{x}$$

```
- int2d(Th)(
          nu*(Grad(u1,u2)'*Grad(v1,v2))
        + UgradV(u1,u2, u1, u2)'*[v1,v2]
        - div(u1,u2)*q - div(v1,v2)*p
        - 1e-8*p*q
         )
```

は次式を意味する。

$$-\int_\Omega \left[\nu\nabla\boldsymbol{u}_i : \nabla\boldsymbol{v} + ((\boldsymbol{u}_i\cdot\nabla)\,\boldsymbol{u}_i)\cdot\boldsymbol{v} - q\nabla\cdot\boldsymbol{u}_i - p_i\nabla\cdot\boldsymbol{v} - 10^{-8}p_i q\right] d\boldsymbol{x}$$

11.4.9 初期値の選択

最初に $\nu = \dfrac{1}{50}$ について、

$$\boldsymbol{u}_0(x,y) = \begin{cases} (1,0)^\top & (x^2+y^2>2) \\ (0,0)^\top & (x^2+y^2\le 2) \end{cases}$$

を Newton 法の初期値として Navier-Stokes 方程式の解を求め、それを ν を小さくした Navier-Stokes 方程式に対する Newton 法の初期値としている。

最終的には、$\nu = \dfrac{1}{200}$ の場合の Navier-Stokes 方程式の解を求めている。

Newton 法の収束が早ければ、ν をより速く小さくしたり、Newton 法が収束しなければ、ν を少し (戻して) 大きくしたり、細かい制御を行っている。余裕があればプログラムを読んでみよう。

それが筋が通ったものであるか分からないが「そこまでやるか」と感心した。こういうプログラムを公開してもらえるのは、大変ありがたいことである。

11.4.10 余談: Oseen 方程式

`NSNewton.edp` の弱形式を定義しているところで `solve Oseen()` と書いてある。

Oseen(オゼーン) 方程式は、Navier-Stokes 方程式を一様な定常流の周りで線形近似したものである。

無限遠で小さいが 0 ではない、一様な速度 $U\boldsymbol{e}_x = (U, 0, 0)^\top$ を持つ流れを扱いたい。

$$\boldsymbol{v} := \boldsymbol{u} - U\boldsymbol{e}_x$$

とおく ($\boldsymbol{v}$ は $\delta\boldsymbol{u}$ と書くのが良いかもしれない)。このとき

$$(\boldsymbol{u}\cdot\nabla)\,\boldsymbol{u} = \left((U+v_1)\frac{\partial}{\partial x_1} + v_2\frac{\partial}{\partial x_2} + v_3\frac{\partial}{\partial x_3}\right)\begin{pmatrix} U+v_1 \\ v_2 \\ v_3 \end{pmatrix} = U\frac{\partial \boldsymbol{v}}{\partial x_1} + (\boldsymbol{v}\cdot\nabla)\boldsymbol{v} \fallingdotseq U\frac{\partial \boldsymbol{v}}{\partial x_1}.$$

そこで近似方程式として

$$\frac{\partial \boldsymbol{v}}{\partial t} + U\frac{\partial \boldsymbol{v}}{\partial x_1} = -\frac{1}{\rho}\nabla p + \nu\triangle\boldsymbol{v} \tag{5}$$

の採用が考えられる。この方程式が普通 Oseen 方程式と呼ばれる (と私は習った)。

Wikipedia 等を見ると、もう少し意味を広げて解釈されることもあるようである。しかし、Hecht のプログラムで、Oseen と言っているのは、かなりの拡張解釈であろう。

“Oseen 近似” という検索語でヒットした玉田 [?] は、Oseen 方程式が研究された経緯が分かる解説である (と私は感じた)。

補足

(工事中)
　説明がイマイチのところがあり、一部は式レベルでミスをしている気がする。時間が取れしだい直す。今は要点のみ以下に示す。

　例えば、A が線形作用素、a が双線形作用素とするとき、

$$f(\boldsymbol{u}) = \begin{pmatrix} A\boldsymbol{u} - \boldsymbol{b} \\ a(\boldsymbol{u}, \boldsymbol{u}) \end{pmatrix}$$

の微分は

$$f'(\boldsymbol{u})\boldsymbol{v} = \begin{pmatrix} A\boldsymbol{v} \\ a(\boldsymbol{u}, \boldsymbol{v}) + a(\boldsymbol{v}, \boldsymbol{u}) \end{pmatrix}.$$

　これを言っておくと、計算の見通しが良くなる。定理に格上げしておいてそれを使って議論すると、説明がかなり短縮されそうである。

参考文献