Numerical Methods for Fluid Dynamics: TD3

Emmanuel DORMY (dormy@dma.ens.fr)

Feb 2024

Forewords

Download TD3.tar, and then type :

```
> tar xvf TD3.tar
> cd TD3
> ipython --pylab
```

1 The Leray projector

Let us consider in a domain $[0, \pi] \times [0, \pi]$ the flow defined as $\mathbf{u} = u\,\mathbf{e}_x + v\,\mathbf{e}_y$, with

$$u = 2\,\sin(x)\,\cos(y)\,, \qquad v = 0\,. \tag{1}$$

We want to project $\mathbf{u}$ to a solenoidal vector field using the Leray projector

$$\mathbb{P}(\mathbf{u}) = \mathbf{u} - \boldsymbol{\nabla}\phi\,, \qquad \Delta\phi = \boldsymbol{\nabla}\cdot\mathbf{u}\,.$$

1. Verify that $\boldsymbol{\nabla}\cdot\mathbf{u} \neq 0$ et $\boldsymbol{\nabla}\cdot\mathbb{P}(\mathbf{u}) = 0$.
2. What are the boundary conditions on $\mathbf{u}\cdot\mathbf{n}$?
3. We want the same boundary conditions to be met by $\mathbb{P}(\mathbf{u})$ than by $\mathbf{u}$, what are then the boundary conditions on ϕ ?
4. Compute ϕ and then $\mathbb{P}(\mathbf{u})$.
5. Plot $\mathbb{P}(\mathbf{u})$ using *Python* (the *streamplot* function can be used).

2 The lid-driven cavity

Our goal will be to write a *Python* to solve for the time-dependent Stokes equations in a lid-driven cavity :

$$\partial_t \mathbf{u} = -\boldsymbol{\nabla} p + \Delta \mathbf{u}\,, \tag{2}$$

$$\boldsymbol{\nabla} \cdot \mathbf{u} = 0\,, \tag{3}$$

$$\mathbf{u}(0 \leq x \leq 1\,, \quad 0 \leq y \leq 1\,, \quad t = 0) = \mathbf{0}\,, \tag{4}$$

$$\mathbf{u} = \mathbf{g} \quad \text{on} \quad \partial\Omega \quad \text{for} \quad t > 0\,. \tag{5}$$

where

$$\mathbf{g} = \begin{cases} \mathbf{0} & \text{for} \quad (0 \leq x \leq 1,\, y = 0)\,,\ (x = 0,\, 0 \leq y < 1)\,,\ (x = 1,\, 0 \leq y < 1)\,, \\ \mathbf{e}_x & \text{for} \quad (0 \leq x \leq 1,\, y = 1)\,, \end{cases} \tag{6}$$

and $(\mathbf{e}_x, \mathbf{e}_y)$ is an orthogonal normed basis of the plan.

For the spatial discretisation of this problem, we will rely on centered second order finite difference schemes. The laplacian will thus be discretised using a 5 points stencil. Regarding the temporal discretisation, we will adopt an explicit Euler scheme of 1st order. To handle incompressibility, we will use a splitting method.

Starting with a field $\mathbf{u}^n$, we want to construct $\mathbf{u}^{n+1}$ by solving for the system

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = \Delta \mathbf{u}^n\,, \tag{7}$$

$$\mathbf{u}^* = \mathbf{g} + \Delta t\, \boldsymbol{\nabla} p^{n+1} \qquad \text{on} \qquad \partial\Omega\,, \tag{8}$$

$$\Delta p^{n+1} = \frac{1}{\Delta t} \boldsymbol{\nabla} \cdot \mathbf{u}^*\,, \tag{9}$$

$$\partial_n p^{n+1} = 0 \qquad \text{on} \qquad \partial\Omega\,, \tag{10}$$

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^*}{\Delta t} = -\boldsymbol{\nabla} p^{n+1}\,. \tag{11}$$

1. Show that this system implies $\boldsymbol{\nabla} \cdot \mathbf{u}^{n+1} = 0, \qquad \mathbf{u}^{n+1} = \mathbf{g}$ on $\partial\Omega$ and

   $$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} = -\boldsymbol{\nabla} p^{n+1} + \Delta \mathbf{u}^n\,.$$

   Consider the coupling between the above equations.

2. Starting with vanishing fields at $t = 0$ for $\mathbf{u}^0$ and p^0. The supplied *Python* code solves the system for $\mathbf{u}^*$:

   $$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = \Delta \mathbf{u}^n\,, \tag{12}$$

   with

   $$\mathbf{u}^* = \mathbf{g} \qquad \text{on} \qquad \partial\Omega\,. \tag{13}$$

You may later try to modify it with

$$\mathbf{u}^* = \mathbf{g} + \Delta t\,\boldsymbol{\nabla} p^n \qquad \text{on} \qquad \partial\Omega\,. \tag{14}$$

What is the key difference between equation (13), equation (14) and equation (8) ?

Plot the vector field $\mathbf{u}^*$.

Knowing $\mathbf{u}^*$, we now consider the system

$$\Delta p^{n+1} = \frac{1}{\Delta t}\,\boldsymbol{\nabla}\cdot\mathbf{u}^*\,, \tag{15}$$

$$\partial_n p^{n+1} = 0\,. \tag{16}$$

Compute $\boldsymbol{\nabla}\cdot\mathbf{u}^*/\Delta t$, then compute p^{n+1}.

3. By studying the code, understand how the Dirichlet and Neumann conditions are impleemented.

DM3 : Moffatt vortices in a rectangular cavity

1. We want to represent the streamlines for the velocity field. What equation needs to be satisfied by the streamfunction ψ for the flow ? What are the boundary conditions for ψ ? Modifying the supplied code (and using *PoissD*), compute and represent the streamlines for the flow.
2. Use this code to investigate elongated domaines with aspect ratio between 3 and 10, what do you observe ?