AC247 - Homework #1

Matheus C. Fernandes

September 16, 2015

Problem 1: Derive the Navier-Stokes equations for incompressible flows

From: Lecture 2

To begin this derivation, we must first obtain Cauchy's equation of motion. Let's begin by recalling Newton's second law applied to the momentum of a fluid element. Namely we want to balance the time rate of change of momentum in a control volume, with the rate of momentum inflow and outflow as well as the forces acting on the control volume. Which for the x-component can be described by

$$\rho\frac{\partial u}{\partial t}(\Delta x\Delta y\Delta z) = -\rho\left(u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}+w\frac{\partial u}{\partial z}\right)(\Delta x\Delta y\Delta z)+\rho g_x(\Delta x\Delta y\Delta z)+\left(\frac{\partial T_{xx}}{\partial x}+\frac{\partial T_{yx}}{\partial y}+\frac{\partial T_{zx}}{\partial z}\right)(\Delta x\Delta y\Delta z).$$

The first term (LHS) is the accumulation term, the second term is the flow term, the third term is the gravitational force term, and the last term is the shear forces term. Now, we can do some canceling out and re-arranging to obtain:

$$\rho\left(\frac{\partial u_i}{\partial t}+\nabla u_i\right)=\rho g_i+\frac{\partial T_{ij}}{\partial x_j}\Rightarrow\boxed{\rho\frac{Du_i}{Dt}=\rho g_i+\frac{\partial T_{ij}}{\partial x_j}.}\tag{1}$$

This is called the cauchy momentum equations, where T_{ij} are the elements of the stress tensor. The stress tensor for the purposes of fluid dynamics is defined as

$$T_{ij}=-p\delta_{ij}+\mu\left(\frac{\partial u_j}{\partial x_i}+\frac{\partial u_i}{\partial x_j}\right)\tag{2}$$

and is known to be symmetric i.e. $T_{ij}=T_{ji}$ for a Newtonian viscous fluid recalling that the stress component t_i on a surface element with normal $\boldsymbol{n}$ may be written as

$$t_i=T_{ij}n_j.$$

Furthermore, by definition, we consider the material derivative, relating the lagrangian reference frame to the eulerian reference frame as the operator:

$$\frac{D}{Dt}\overset{\text{def}}{=}\frac{\partial}{\partial t}+\boldsymbol{u}\cdot\boldsymbol{\nabla}$$

Plugging in eq. (2) into eq. (1) we obtain:

$$\rho\frac{D\boldsymbol{u}}{Dt}=-\frac{\partial p}{\partial x_i}+\mu\frac{\partial}{\partial x_i}\left(\frac{\partial u_j}{\partial x_j}\right)+\mu\frac{\partial^2 u_i}{\partial x_j^2}+\rho g_i\tag{3}$$

For an incompressible fluid, we know that the volume of the fluid at any given time cannot change, hence the continuity equation

$$\frac{\partial\rho}{\partial t}+\boldsymbol{\nabla}\cdot(\rho\boldsymbol{u})=0$$

becomes,

$$\frac{\partial u_x}{\partial x}+\frac{\partial u_y}{\partial y}+\frac{\partial u_z}{\partial z}=0\rightarrow\boldsymbol{\nabla}\cdot\boldsymbol{u}=0.$$

Thus eq. (3) can ultimately be simplified to

$$\rho\frac{D\boldsymbol{u}}{Dt}=-\boldsymbol{\nabla}p+\rho g+\mu\boldsymbol{\nabla}^2\boldsymbol{u}.$$

Whence, the final Navier-Stokes equations for an incompressible fluid are:

$$\boxed{\begin{gathered}\rho\frac{D\boldsymbol{u}}{Dt} = -\nabla p + \rho\boldsymbol{g} + \mu\nabla^2\boldsymbol{u},\\ \nabla\cdot\boldsymbol{u} = 0\end{gathered}}$$

Problem 2: Derive the analytical expression of the Poiseuille Flow in 2D (known as flow in Hele-Shaw cell), show that the nonlinear terms are identically zero

From: Lecture 2

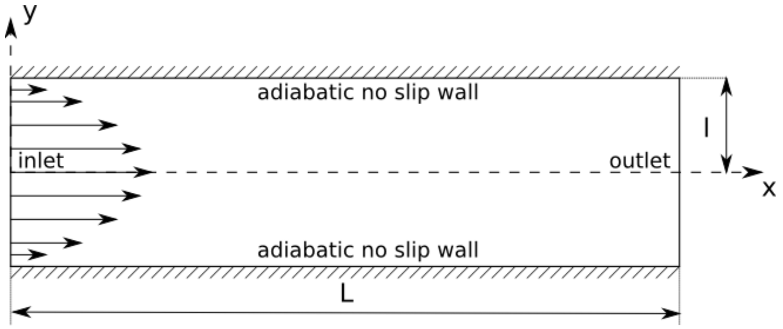


Figure 1: Schematic of 2D Poiseuille flow

Considering the above problem in steady state, we can drop entirely the time dependence in the equations. Furthermore, we note certain symmetry constraints on the velocity vector

$$u_y = 0, \tag{4a}$$

$$u_z = 0, \tag{4b}$$

$$\frac{\partial u_x}{\partial x} = 0, \tag{4c}$$

whence we solve for $u_x(y)$ which is a function of only one variable. Now we know that the Navier-Stokes equations become only one equation solving for the x-component of the velocity given by:

$$\rho u\frac{du}{dx} = -\frac{dp}{dx} + \rho g_x + \mu\frac{d^2u}{dy^2}, \text{where because of symmetry } \frac{\partial p}{\partial y} = \frac{\partial p}{\partial z} = 0.$$

Because of eq. (4c), and the fact that we are ignoring body forces we can simplify to

$$\frac{dp}{dx} = \mu\frac{d^2u}{dy^2}. \tag{5}$$

Here we have $f(x) = h(y)$, and to solve this we can treat each side as a constant and solve them independently. We know that

$$\frac{dp}{dx} = C_1$$

and

$$p = C_1x + C_2.$$

We further know the boundary conditions for the inlet and the outlet are

$$p(x=0) = P_i, \text{ and } p(x=L) = P_o,$$

respectively. Pugging the boundary conditions into our equation we get that:

$$C_1 = -(P_i - P_o)/L = -\Delta P/L$$

$$C_2 = P_i,$$

so that the LHS of the equation becomes

$$p = P_i - x\Delta P/L$$

The RHS of eq. (5) is a linear second order ODE, which we must have two boundary conditions to solve for. The following are the conditions for the walls that are assumed to be no-slip:

$$u(y = I) = 0 \tag{6a}$$

$$u(y = -I) = 0. \tag{6b}$$

So we plug in what we know about p into the LHS of eq. (5) and find that the ODE which we need to solve is

$$\frac{d^2u}{dy^2} = \frac{-\Delta P}{L\mu}. \tag{7}$$

So we solve it by integrating to obtain:

$$\frac{du}{dy} = \frac{-\Delta P}{L\mu} + C_3$$

$$u = \frac{-\Delta P y^2}{L\mu} + C_3 y + C_4$$

Plugging in the BC described in eq. (6) we obtain the parabolic velocity profile and linear pressure as

$$\boxed{\begin{gathered} u_x(y) = \frac{\Delta P}{2\mu L}\left[I^2 - y^2\right] \\ p(x) = P_i - x\Delta P/L \end{gathered}}$$

It is easy to show that the non-linear terms are identically zero. Assume the term

$$\boldsymbol{u}\cdot\nabla\boldsymbol{u} = \begin{bmatrix} u_x \\ u_y \end{bmatrix} \cdot \begin{bmatrix} \frac{\partial u_x}{\partial x} & \frac{\partial u_y}{\partial x} \\ \frac{\partial u_x}{\partial y} & \frac{\partial u_y}{\partial y} \end{bmatrix} = \begin{bmatrix} u_x \cancelto{0}{\frac{\partial u_x}{\partial x}} + \frac{\partial u_x}{\partial y} \cancelto{0}{u_y} \\ u_x \cancelto{0}{\frac{\partial u_y}{\partial x}} + \cancelto{0}{u_y \frac{\partial u_y}{\partial y}} \end{bmatrix}$$

Thus, making the entire term go to zero.

Problem 3

September 16, 2015

1 Finite Differece Code for Hele-Shaw Cell aka. Poiseuille Flow in 2D

1.1 Importing Libraries

```
In [39]: %matplotlib inline

         import numpy as np
         import matplotlib.pyplot as plt
         import math
         from pylab import *
```

1.2 Parameters

Assuming the incompressible Navier-Stokes Equation:

$$\frac{\partial u_x}{\partial t} = -\frac{1}{\rho}\frac{\partial p}{\partial x} + \nu\frac{\partial^2 u_x}{\partial y^2},$$

we know from the nature of the problem that $\frac{\partial p}{\partial x} = \frac{\Delta p}{L}$. Namely the above becomes:

$$\frac{\partial u_x}{\partial t} = -\frac{1}{\rho}\frac{\Delta p}{L} + \nu\frac{\partial^2 u_x}{\partial y^2}.$$

Where we define the above parameters below. For simplicity let

$$\xi = \frac{1}{\rho}\frac{\Delta p}{L}$$

such that

$$\frac{\partial u_x}{\partial t} = -\xi + \nu\frac{\partial^2 u_x}{\partial y^2}.$$

```
In [40]: rho= 1000 #[kg/m^3]
         deltaP=100000 #[Pa]
         L=15 #length of tube in [m], assuming that the flow is 1D, though
         nu= 1 # [m^2/s]
         D=2 # upward dimension of tube, analogous to diameter of circular tube
         deltaT= 0.5 # time step in [s]
         lengthT=60 #duration of time in [s]
         deltaY= 0.01 # [m], discretization in space
         u0=0; #[m/s] BC at bottom face, no slip
         uD=0; #[m/s] BC at top face, no slip
         def uI(y,D): return y*(D-y) # initial condition function in [m/s]
         xi=(1.0/rho)*(deltaP/L) # parameter for the first term of the diff eq
```

1.3 Finite Difference Code Implementation

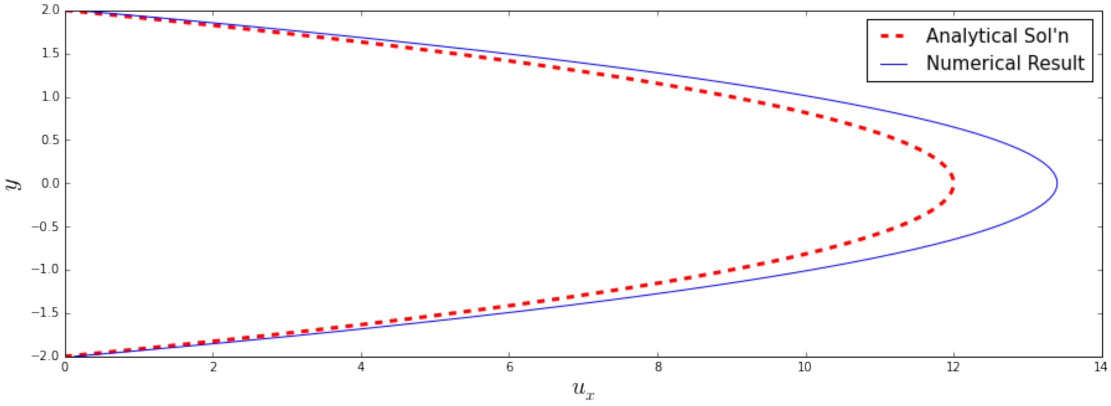
Here we iplement a finite difference code for the above equations with the above parameters.

The code is implemented using a center difference discretization in space.

```
In [41]: #calculating using finite difference
         y=np.linspace(-D,D,2*D/deltaY)
         t=np.linspace(0,lengthT,lengthT/deltaT)
         A=np.zeros((len(y),len(y)))
         for ii in range(0,len(y)):
             A[ii,ii]=2
             if ii!=0:
                 A[ii,ii-1]=-1
                 A[ii-1,ii]=-1
         A=A*(deltaY**-2)*nu
         ff=np.ones((len(y)))*xi
         u=np.linalg.solve(A,ff)
```

```
In [42]: #Calculating the analytical values
         u_comp=(deltaP/(2*nu*rho*L))*(D**2-y**2)
```

```
In [43]: fig = plt.figure(figsize=(15,5))
         plt.plot(u_comp,y,'--r',label='Analytical Sol\'n',linewidth=3)
         plt.plot(u,y,label='Numerical Result')
         plt.xlabel('$u_x$',fontsize=20)
         plt.ylabel('$y$',fontsize=20)
         plt.legend(fontsize=15)
         plt.show()
```



```
In [ ]:
```

```
In [ ]:
```