



Modeling Creeping Flow

Ethan Rowbury

Introduction

Theory:

Navier-Stokes Equations

Continuity Equation

$$\nabla \cdot \vec{V} = 0$$

Momentum Equations

$$\rho \frac{D\vec{V}}{Dt} = -\nabla p + \rho \vec{g} + \mu \nabla^2 \vec{V}$$

Total derivative

=

$$\rho \left[\frac{\partial V}{\partial t} + (\vec{V} \cdot \nabla) V \right]$$

Change of velocity
with time

Convective term

Pressure gradient

Fluid flows in the
direction of largest
change in pressure.

Body force term

External forces, that
act on the fluid
(gravitational force
or electromagnetic).

Diffusion term

For a Newtonian
fluid, viscosity
operates as a
diffusion of
momentum.

$$\mu \nabla^2 \mathbf{u} - \nabla p + \mathbf{f} = \mathbf{0}$$

$$\nabla \cdot \mathbf{u} = 0$$

Introduction

Visual:



[UNM Physics and Astronomy](https://www.youtube.com/watch?v=p08_KITKP50&t=51s)

https://www.youtube.com/watch?v=p08_KITKP50&t=51s

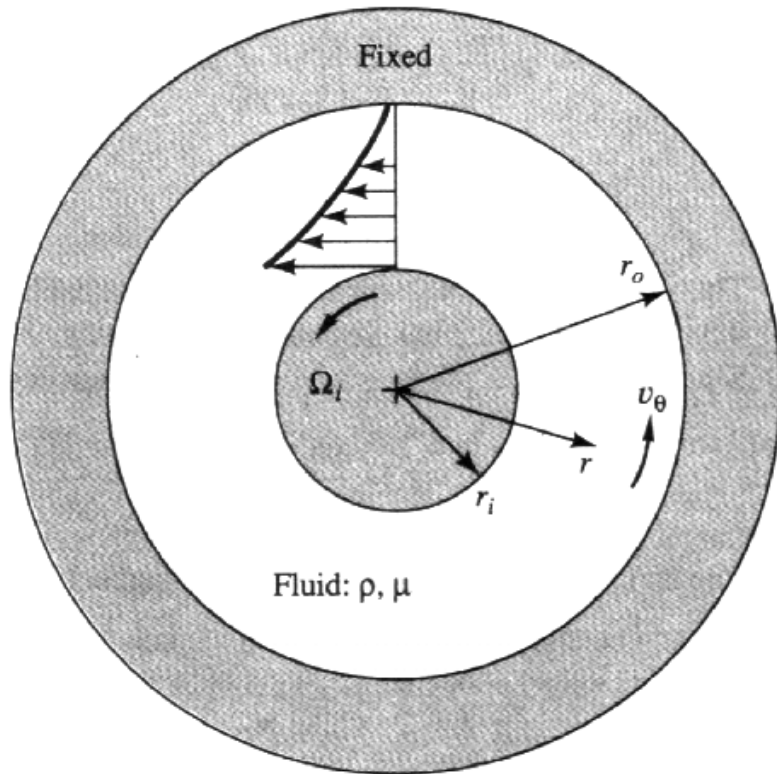
Objectives

- Model creeping flow using Python
- Build prototype to replicate



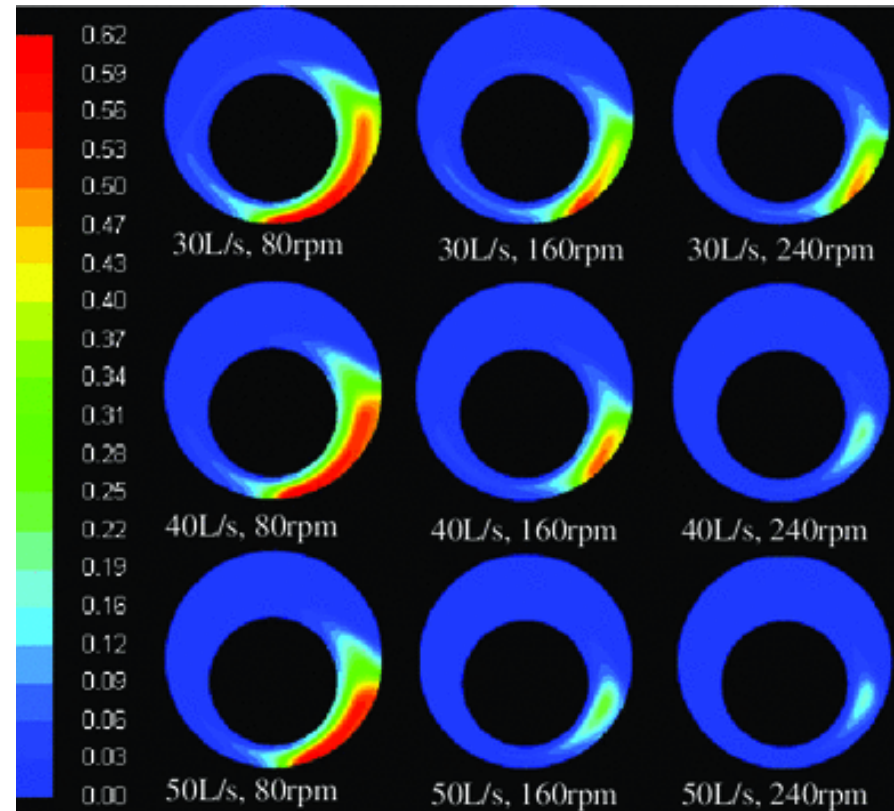
Methods

Velocity Profile



https://www.researchgate.net/figure/Circular-Couette-Flow-theoretical-model-of-pressure-gradient-caused-between-rotating_fig13_3902274

Position mapping



<https://link.springer.com/article/10.1007/s13202-014-0118-x>

Methods

Container

Fluid



<https://www.fishersci.ca/shop/products/clear-graduated-ac-round-polypropylene-pe-foam-ptfe-lined-cap-attached/02911458a>

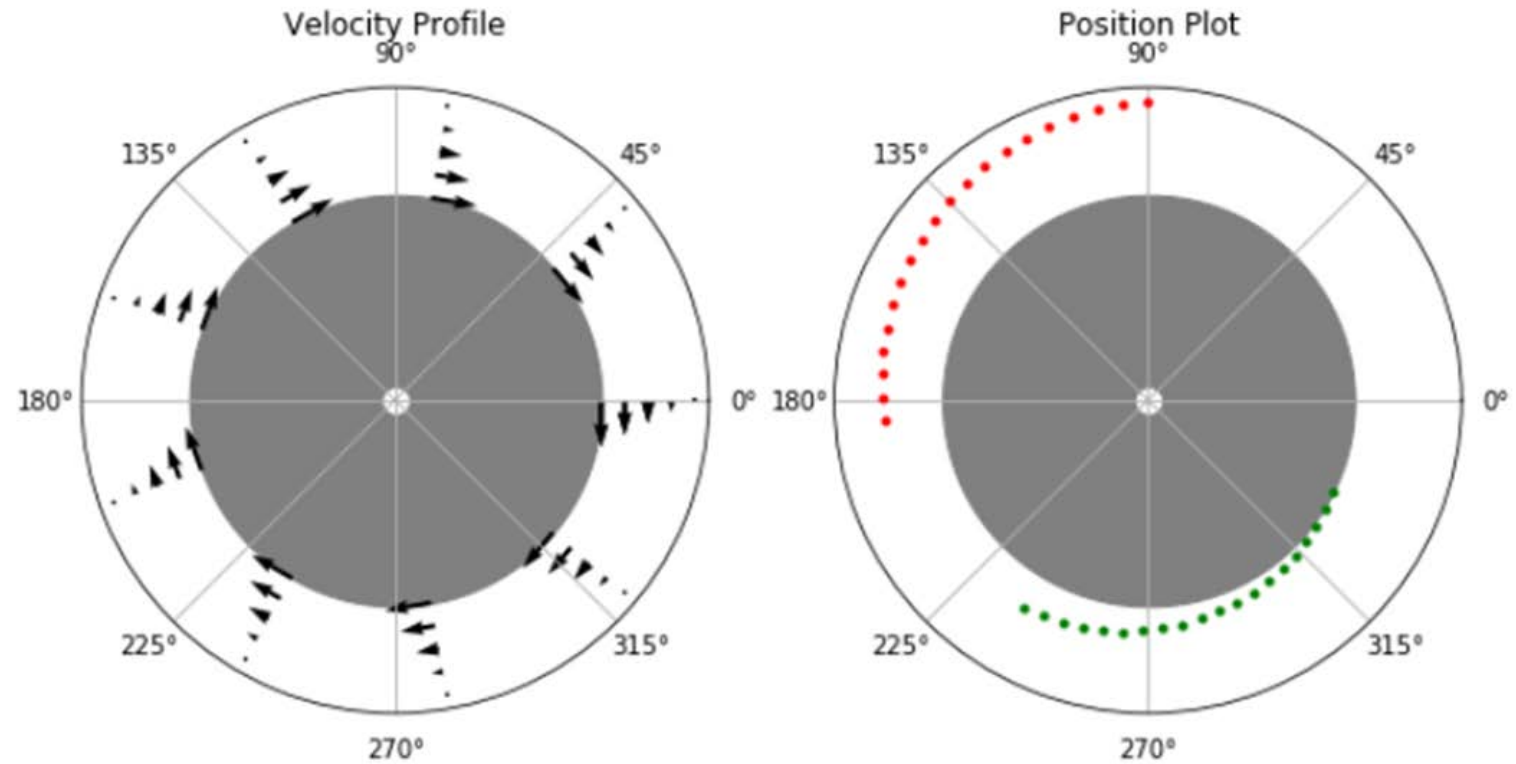


<https://www.pinterest.com/pin/331225747565361193/?lp=true>

Results

Unique and insightful plots

Animated rotation



Results

Prototype achievements and shortcomings:

- Precise dye placement
- Smearing





Discussion

Python Model

- Animation
- Slider bar

Annulus Prototype

- Increase volume and stability
- Bubbles
- More data

Questions





Acknowledgments

- My wife, Beth
- Nathan Bevan
- Joseph Rich
- Nicholas Rollins
- Dr. Doug Tree

Example Code

```
# time loop
Nt = 1200      # number of steps
for t in range(Nt+1):

    # framework for plots
    fig1, (plot1, plot2) = plt.subplots(1, 2, subplot_kw=dict(projection = 'polar'), figsize = [10,10])
    # sets up a figure of subplots
    plot1.set_title("Velocity Profile", va='bottom')      # plot1 is velocity
    plot2.set_title("Position Plot", va='bottom')        # plot2 is position

    thetas2 = np.linspace(0, 2*np.pi, 100) # theta values for center circle

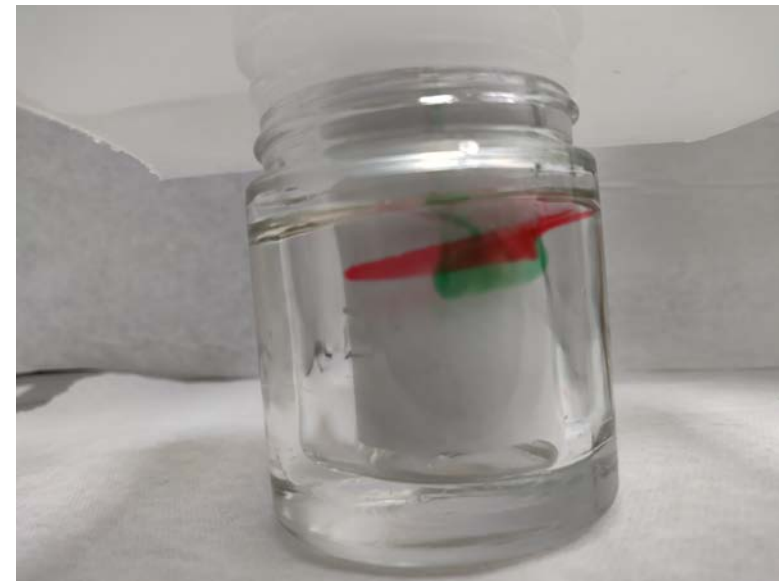
    plot1.fill_between(thetas2, 0, k*R, color='gray') # fills in center circle
    plot2.fill_between(thetas2, 0, k*R, color='gray') # fills in center circle

    plot1.set_rticks([])      # no radial ticks
    plot2.set_rticks([])      # no radial ticks
    plot2.set_rmax(R)         # insures the plots are on the same scale

    # velocity plot
    U = np.sin(t*np.pi*40/Nt) # Linear velocity of inner circle as a function of time

    radii = np.linspace(k*R,R,5)      # r values for meshgrid
    thetas = np.linspace(0, 2*np.pi, 10) # theta values for meshgrid
    theta, r = np.meshgrid(thetas, radii)
```

More Annulus Pictures



Inconclusive Data

