Justine John A Serdoncillo
AEM 8209 - Fluid Mechanics 2

24 February 2023
Problem set 2

1a) find a complex potential function for a flow having exactly 3 stagnation points at the locations $(1,0), \left(-\frac{1}{2}, \frac{\sqrt{3}}{2}\right), \left(-\frac{1}{2}, \frac{-\sqrt{3}}{2}\right)$

Stagnation point means $u, v = 0$. using $W = u - iv = 0$ at stagnation point

then the complex potential function should satisfy

$$W = (z - z_0)(z - z_1)(z - z_2)$$ where z_0, z_1 & z_2 are the stagnation points

$$z_0 = 1, \quad z_1 = -\frac{1}{2} + \frac{\sqrt{3}}{2}i, \quad z_2 = -\frac{1}{2} - \frac{\sqrt{3}}{2}i = z_1^*$$

$$W = (z-1)(z-z_1)(z-z_1^*) = (z-1)(z^2 \underbrace{- zz_1 - zz_1^*}_{-z(z_1+z_1^*) = -z(2(-\frac{1}{2})) = z} + z_1 z_1^*)$$

$$z_1 z_1^* = \sqrt{\left(-\frac{1}{2}\right)^2 + \left(\frac{\sqrt{3}}{2}\right)^2} = \sqrt{\frac{1}{4} + \frac{3}{4}} = 1$$

$$= (z-1)(z^2+z+1) = z^3 + \cancel{z^2} + \cancel{z} - \cancel{z^2} - \cancel{z} - 1$$

$$W = z^3 - 1$$

$$W = \frac{dF}{dz} \Rightarrow \int W dz = \int dF \Rightarrow F = \int z^3 - 1 \, dz = \frac{z^4}{4} - z$$

$$\boxed{F(z) = \frac{z^4}{4} - z}$$

b) express the real velocity potential $\phi(x,y)$ & $\psi(x,y)$ in terms of x & y

$$F(z) = \frac{z^4}{4} - z = \phi(x,y) + i\psi(x,y), \quad z = x+iy$$

$$\frac{(x+iy)^4}{4} - (x+iy) = \frac{x^4 + 4ix^3y + 6(i)^2x^2y^2 + 4(i)^3xy^3 + (i)^4y^4}{4} - x - iy$$

$$= \left(\frac{x^4 - 6x^2y^2 + y^4}{4} - x\right) + i\left(\frac{4x^3y - 4xy^3}{4} - y\right)$$

$$\boxed{\phi(x,y) = \frac{x^4 - 6x^2y^2 + y^4 - 4x}{4} \qquad \psi(x,y) = x^3y - xy^3 - y}$$

c) plotted in the attached code & plots

d) Write a question appropriate for an undergraduate fluid mechanics exam.

Given $\psi(x,y) = x^3y - xy^3 - y$ &

$$u = \frac{\partial\psi}{\partial y} \quad \& \quad v = \frac{-\partial\psi}{\partial x}$$

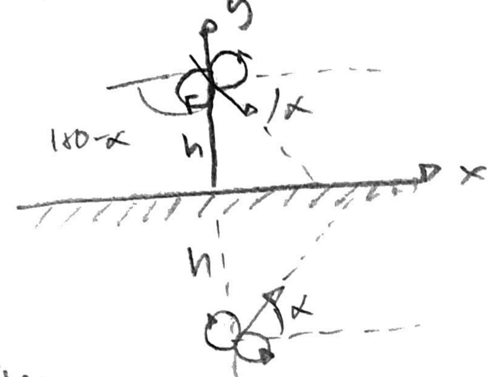
solve for $\phi(x,y)$ where

$$\frac{\partial\phi}{\partial x} = u \quad \& \quad \frac{\partial\phi}{\partial y} = v$$

2. A dipole of strength $\mu = 1$ is placed a distance $h = 1$ away from an infinite flat plate such that its axis is oriented at an angle $\alpha = \frac{\pi}{3}$ w/ respect to the horizontal. Assume that the fluid above the wall is inviscid and has density $\rho = 1$

---

a. Find complex potential function for this flow.



in order to generate an infinite flat plate, the method of images must be used. An additional dipole is generated below the plate & completely mirroring the dipole.

For a dipole $F(z) = \frac{\mu}{z}$

to rotate

$z \to uz$ where $u = re^{-i\theta}$, θ is rotate

to translate

$z \to z - z_0$ where z_0 is translation

then

$r_{1,2} = 1 \quad \theta_1 = \frac{2\pi}{3} \quad \theta_2 = \frac{-2\pi}{3}$

$z_{01} = 1i \quad z_{02} = -1i$

$$F(z) = \frac{e^{i\frac{2\pi}{3}}\mu}{z - 1i} + \frac{e^{-i\frac{2\pi}{3}}\mu}{z + 1i} = \mu\left(\frac{\cos\left(\frac{2\pi}{3}\right) + i\sin\left(\frac{2\pi}{3}\right)}{z - 1i} + \frac{\cos\left(\frac{-2\pi}{3}\right) + i\sin\left(\frac{-2\pi}{3}\right)}{z + 1i}\right)$$

$$= \mu\left(\frac{-\frac{1}{2} + \frac{\sqrt{3}}{2}i}{z - 1i} + \frac{-\frac{1}{2} - \frac{\sqrt{3}}{2}i}{z + 1i}\right) = \mu\left(\frac{\left(-\frac{1}{2} + \frac{\sqrt{3}}{2}i\right)(z + 1i) + \left(-\frac{1}{2} - \frac{\sqrt{3}}{2}i\right)(z - 1i)}{(z - 1i)(z + 1i)}\right) = \mu\left[\frac{\left(-\frac{z}{2} - \cancel{\frac{i}{2}} + \cancel{\frac{\sqrt{3}iz}{2}} - \frac{\sqrt{3}}{2}\right) + \left(-\frac{z}{2} \cancel{+ \frac{i}{2}} - \frac{\sqrt{3}iz}{2}\right)}{z^2 + 1}\right.$$

$$= \frac{2\mu\left(-\frac{z}{2} - \frac{\sqrt{3}}{2}\right)}{z^2 + 1} = \boxed{\frac{-\mu\left(\sqrt{3} + z\right)}{z^2 + 1} = F(z)}$$

b) find location of stagnation points

Stagnation pts occur at $u, v = 0$ $\qquad w = u - iv = 0$

$$w = \frac{dF}{dz} = \frac{d}{dz}\left[\frac{-\mu(\sqrt{3}+z)}{z^2+1}\right] = \frac{(z^2+1)(-\mu) + (\mu\sqrt{3}+\mu z)(2z)}{(z^2+1)^2} = 0$$

$$-\cancel{\mu} z^2 - \cancel{\mu} + \cancel{\mu} 2\sqrt{3} z + 2\cancel{\mu} z^2 = 0 \Rightarrow z^2 + 2\sqrt{3} z - 1 = 0$$

$$z = \frac{-2\sqrt{3} \pm \sqrt{(2\sqrt{3})^2 - 4(1)(-1)}}{2} = \frac{-2\sqrt{3} \pm \sqrt{12+4}}{2} = \sqrt{3} \pm \frac{4}{2} = -\sqrt{3} \pm 2$$

$$\boxed{\text{Stagnation points at } z = -\sqrt{3}+2,\ z = -\sqrt{3}-2}$$

c) Assume $P_{below\,plate} = P_\infty$, find F_{net} acting on the plate

$$F_x - iF_y = -\rho\pi\,[\Sigma]$$

$$F(z)\ \text{dipole} = O\left(\frac{1}{z}\right)$$

$$w = \frac{dF}{dz}\ \text{dipole} = \frac{d}{dz}\left(O\left(\frac{1}{z}\right)\right) = O\left(\frac{1}{z^2}\right)$$

use $\mathrm{Res}_f(z_1) = \frac{1}{(m-1)!}\frac{d^{m-1}}{dz^{m-1}}\left[(z-z_1)^m f(z)\right]\Big|_{z=z_1}$ $\qquad$ #didn't really use this explicitly

for a general case $\qquad \theta_1 = \pi-\alpha \qquad \theta_2 = \pi+\alpha$

$$F(z) = \frac{e^{i(\pi-\alpha)}}{z-1i} + \frac{e^{i(\pi+\alpha)}}{z+1i} = \frac{\cos(\pi-\alpha)+i\sin(\pi-\alpha)}{z-1i} + \frac{\cos(\pi+\alpha)+i\sin(\pi+\alpha)}{z+1i}$$

let $c = \cos(\alpha)$, $s = \sin(\alpha)$

$\cos(\pi-\alpha) = -\cos(\alpha) \qquad \sin(\pi-\alpha) = \sin(\alpha)$

$\cos(\pi+\alpha) = -\cos(-\alpha) = -\cos(\alpha) \qquad \sin(\pi+\alpha) = -\sin(\alpha)$

$$F(z) = \frac{-c+is}{z-1i} + \frac{-c-is}{z+1i}$$

$$w = \frac{dF(z)}{dz} = -\left(\frac{-c+is}{(z-1i)^2}\right) + -\left(\frac{(-c-is)}{(z+1i)^2}\right) = \frac{c-is}{(z-1i)^2} + \frac{c+is}{(z+1i)^2}$$

<u>Blasius Thm</u>

$$F_x - iF_y = \frac{i\rho}{2}\oint w^2 dz$$

<u>Laurent Series</u>

$$w^2 = f(z) = \sum_{n=-\infty}^{\infty} a_n (z-z_j)^n$$

<u>Residue theorem</u>

$$\oint f(z)dz = 2\pi i \sum_{j=1}^{m} a_{-1}$$

using the following theorems just need to find the a_{-1} for the singularity at $z=-1i$ which is in the control volume

$\frac{a_{-1}}{z+1i}$ want

$$w^2 = \underbrace{\frac{(c-is)^2}{(z-1i)^4}}_{\text{no } \frac{a_{-1}}{z+1i} \text{ term}} + \underbrace{\frac{(c+is)^2}{(z+1i)^4}}_{A} + \underbrace{\frac{2(c-is)(c+is)}{(z-1i)^2(z+1i)^2}}_{B}$$

$$A = \frac{c^2+2cis+s^2}{(z+1i)^4} = \frac{1+2cis}{(z+1i)^4}$$ can't decompose further, no $\frac{a_{-1}}{z+1i}$ term

$$B = \frac{2(c+is)(c-is)}{(z-1i)^2(z+1i)^2} = \frac{2(c^2+s^2)}{(z-1i)^2(z+1i)^2} = \frac{2}{(z-1i)^2(z+1i)^2}$$

use partial fraction decomposition

$$\frac{2}{(z-1i)^2(z+1i)^2} = \frac{A}{(z-1i)} + \frac{B}{(z-1i)^2} + \frac{C}{(z+1i)} + \frac{D}{(z+1i)^2} \Rightarrow 2 = A(z+1i)^2(z-1i) + B(z+i)^2 + C(z+1i)(z-1i)^2 + D(z-i)^2$$

if $z=i$: $2 = B(2i)^2 \Rightarrow 2 = -4B \rightarrow B = \frac{-1}{2}$

$z=-i$: $2 = D(-2i)^2 \Rightarrow 2 = -4D \Rightarrow D = -1/2$

$z=0$: $2 = A(i)^2(-i) + B(i)^2 + c(i)(-i)^2 + D(-i)^2 = Ai - B - Ci - D = Ai - Ci + 1$

(A) $1 = (A-C)i \Rightarrow A - C = \frac{1}{i} = -i$

$z=2i$: $2 = A(3i)^2(i) + B(3i)^2 + C(3i)(i)^2 + D(i)^2 = -9Ai - 9B - 3Ci - D = 2$

$-9Ai - 3Ci + \cancel{5} = \cancel{2} \Rightarrow 3 = 9Ai + 3Ci \Rightarrow 1 = 3Ai + Ci$ (D)

add (A) & (D) $2 = 4Ai$ $\quad A = \frac{1}{2i} = -i/2$ $\quad C = i/2$ want this cuz $\frac{a_{-1}}{z+i}$ term

$$F_x - iF_y = -\rho\pi[\Sigma] = -\rho\pi\frac{i}{2}$$

$F_x = 0 \quad F_y = \frac{\rho\pi}{2}$

$$\boxed{F_{net} = \frac{\rho\pi}{2} \text{ in the } +y \text{ direction}}$$

d) How does the net force on the plate depend on the angle α? Find the net force on the plate $F_{net}(\alpha)$ as a function of α, where $0 \le \alpha \le 2\pi$

---

in Part c) I used a general case for α not just $\alpha = \pi/3$

it was seen than $F_{net} \neq f(\alpha)$ but was constant. This is as expected because the singularity did not change at all

$$\boxed{F_{net}(\alpha) = \frac{\rho\pi}{2} \text{ in } +y \text{ direction for } 0 \le \alpha \le 2\pi}$$

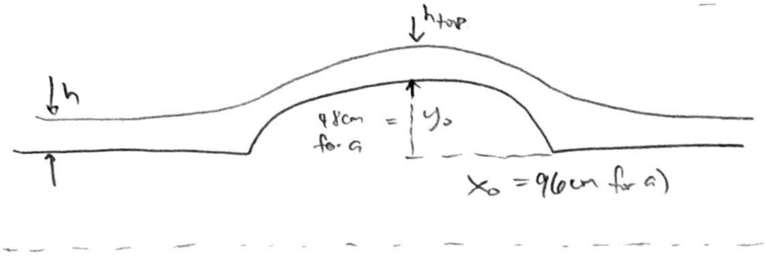
3) Consider a greenhouse in the shape of a semi-elliptical cylinder of width 192 cm and height 48 cm.

a) find an expression for h_{top} in terms of x_0, y_0, h

b) given $h = 17$ cm, find h_{top}

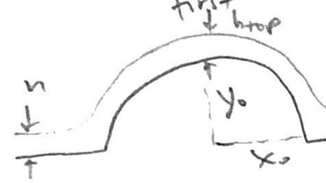
c) now given $x_0 = 48$ cm and $y_0 = 96$ cm find h_{top} given $h = 17$ cm, compare w/ b)

c) plot streamlines of b) & c) & Attempt to use BEM solver to compute streamlines & compare



a) Solving for a semi-regular cylider first

$x_0 = y_0 = a$



this is a flow over a circular cylinder with and use a joukowski transform from ζ to z

$$z = \zeta + \frac{b^2}{\zeta}$$

for the circular cylider

$$F(z) = z + \frac{a^2}{z} \qquad z = re^{i\theta} = r\cos\theta + i\sin\theta$$

$$= re^{i\theta} + \frac{a^2}{r}e^{-i\theta}$$

$$= (r\cos\theta + ir\sin\theta) + \frac{a^2}{r}(\cos\theta - i\sin\theta)$$

$$= \underbrace{\left(r\cos\theta + a^2 r\cos\theta\right)}_{\phi} + i\underbrace{\left(r\sin\theta - \frac{a^2}{r}\sin\theta\right)}_{\psi}$$

$$\psi = r\sin\theta - \frac{a^2\sin\theta}{r} = y - \frac{a^2\sin\theta}{r}$$

$$\psi(r \to \infty, y = h, \theta \to \pi) = h$$

h_{top} point occurs on the same streamline as above

$$\psi\left(r = h_{top}, \theta = \frac{\pi}{2}\right) = h$$

$$h_{top}\cancel{\sin\left(\frac{\pi}{2}\right)} - \frac{a^2}{h_{top_\zeta}}\cancel{\sin\left(\frac{\pi}{2}\right)} = h$$

$$h_{top_\zeta}^2 - a^2 - h_{top_\zeta}h = 0$$

use quadratic eqn, use only positive value

$$h_{top_\zeta} = \frac{h + \sqrt{h^2 + 4a^2}}{2}$$

$$F(z) = z + \frac{a^2}{z}$$

now need to get a in terms of x_0 & y_0

for ellipse case $\quad x_0 = a + \frac{b^2}{a} \quad y_0 = a - \frac{b^2}{a}$

$$x_0 + y_0 = 2a \quad \Rightarrow \quad a = \frac{x_0 + y_0}{2}$$

$$h_{top_\zeta} = \frac{h}{2} + \sqrt{\frac{h^2}{4} + \cancel{\frac{4}{4}}\left(\frac{x_0+y_0}{2}\right)^2}$$

$$h_{top_\zeta} = \frac{h}{2} + \sqrt{\left(\frac{h}{2}\right)^2 + \left(\frac{x_0+y_0}{2}\right)^2}$$

now need to use the joukowski transform

$$h_{top\,z} = h_{top_\zeta}i + \frac{b^2}{h_{top_\zeta}i}$$

$$x_0 - y_0 = \frac{2b^2}{a} \qquad b = \sqrt{\frac{(x_0 - y_0)a}{2}}$$

$$b = \sqrt{\frac{(x_0-y_0)\left(\frac{x_0+y_0}{2}\right)}{2}} = \frac{\sqrt{x_0^2 - y_0^2}}{2}$$

$$h_{top\,z} = \frac{h}{2} + \sqrt{\left(\frac{h}{2}\right)^2 + (x_0+y_0)^2} + \frac{\frac{x_0^2 - y_0^2}{4}}{\frac{h}{2} + \sqrt{\left(\frac{h}{2}\right)^2 + (x_0+y_0)^2}\,i} \cdot \frac{i}{i}$$

$$h_{top} = h_{top\,z} - y_0$$

$$\boxed{h_{top} = \underbrace{\frac{h}{2} + \sqrt{\left(\frac{h}{2}\right)^2 + \left(\frac{x_0+y_0}{2}\right)^2}}_{1} \; \frac{-(x_0^2-y_0^2)/4}{\underbrace{\frac{h}{2} + \sqrt{\left(\frac{h}{2}\right)^2 + \left(\frac{x_0+y_0}{2}\right)^2}}_{2}} - \underbrace{y_0}_{1}}$$

b) plug in $h = 17\text{cm}$

$X_0 = 96\text{cm}, \; y_0 = 48\text{cm}$

$$h_{top} = \frac{17}{2} + \sqrt{\left(\frac{17}{2}\right)^2 + \left(\frac{96+48}{2}\right)^2} \; \frac{-\left((96^2-48)^2/4\right)}{\frac{17}{2} + \sqrt{\left(\frac{17}{2}\right)^2 + \left(\frac{96+48}{2}\right)^2}} - 48$$

$= 11.6667$

$$\boxed{h_{top} = 11.67\text{cm}}$$

c) new case plug in $h = 17\text{cm}$

$X_0 = 48\text{cm} \quad y_0 = 96\text{cm}$

$$h_{tp} = \frac{17}{2} + \sqrt{\left(\frac{17}{2}\right)^2 + \left(\frac{96+48}{2}\right)^2} \; \frac{-(48^2-96^2)/4}{\frac{17}{2} + \sqrt{\left(\frac{17}{2}\right)^2 + \left(\frac{96+48}{2}\right)^2}} - 96$$

$= 6.3333$

$$\boxed{h_{top} = 6.333\text{cm}}$$

$h_{topB}(1) = h_{topC}(1) = \square$

~~let~~ $\Delta = 48, \; 2\Delta = 96, \; \square = h_{top\,y}$

$(X_0)^2 - (y_0)^2$ for $h_{topB} = (2\Delta)^2 - \Delta^2 = 3\Delta^2$

$h_{topC} = \Delta^2 - (2\Delta)^2 = -3\Delta^2$

$h_{topB} + h_{topC}$ [illegible]

$=$ [illegible] Δ

$$\boxed{h_{top}B \text{ is } > h_{top}C}$$

$h_{topB} + h_{topC}$

$$= \left[\frac{h}{2} + \square - \cancel{\frac{(3\Delta^2)/4}{\frac{h}{2}+\square}} - \Delta\right] + \left[\frac{h}{2} + \square - \cancel{\frac{(-3\Delta^2)/4}{\frac{h}{2}+\square}} - 2\Delta\right]$$

$= h + 2\square - 3\Delta$

nothing special

d) plots in python

also BEM method

Homework 2

Number 1 c)

After determining the necessary complex potential function for this flow, the streamlines and potential lines are plotted as seen in the figure below. The solid lines correspond to the streamlines while the dashed, translucent lines correspond to the potential lines. The red points are the stagnation points and it can be seen that it does lie on the following required locations. The black arrows are the quiver plot for the velocity flow.

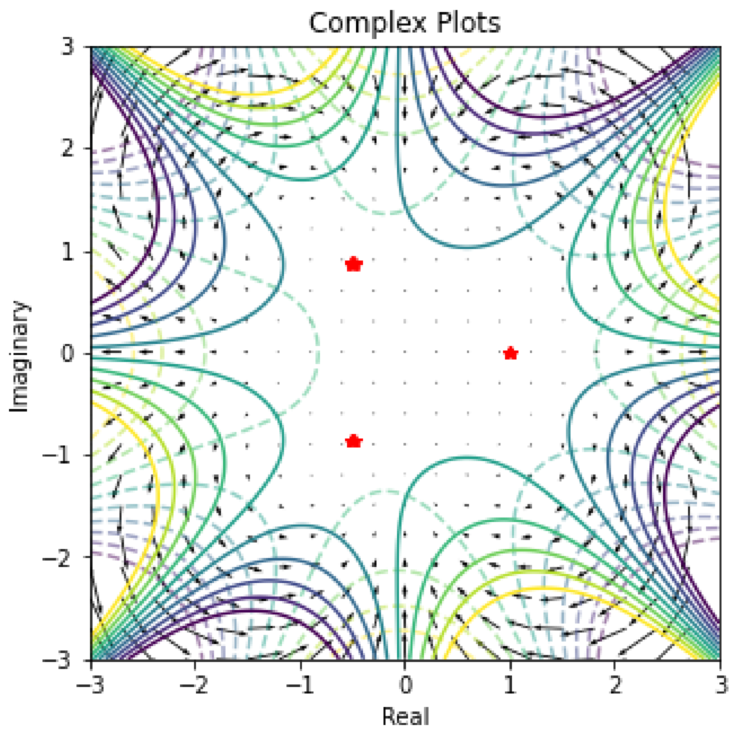


Figure 1: Complex Potential Flow for 3 Stagnation Points

Number 2 a)

After determining the necessary complex potential function for this flow, the streamlines are plotted as seen in the figure below. Similarly, the potential lines are also plotted as dashed translucent lines. It can be seen that the flow is acting on a flat plate and this is done using the method of images.

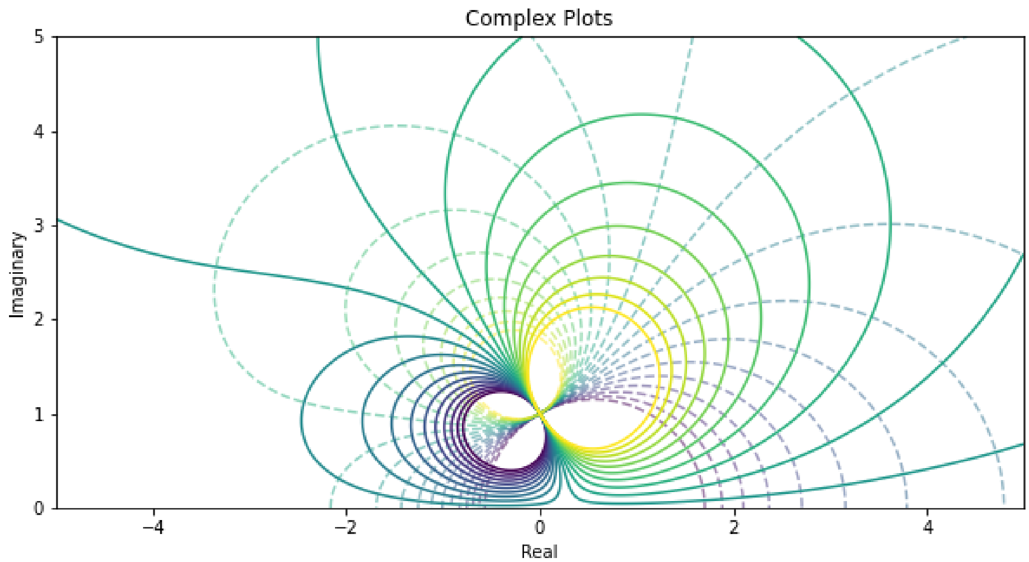


Figure 2: Complex Potential Flow for Dipole on Infinite Flat Plate

Number 3 d

The following streamlines are plotted for the two elliptical flows. The blue dots correspond to points that evaluate to the 0 streamline, while the cyan lines are with the 17cm streamline. It can be seen that at x=0, the cyan line is at around 60 which corresponds to the similar htop calculated of 11.67cm. The same can be said for the Prized Rose problem showing a smaller htop of around 3.33cm.

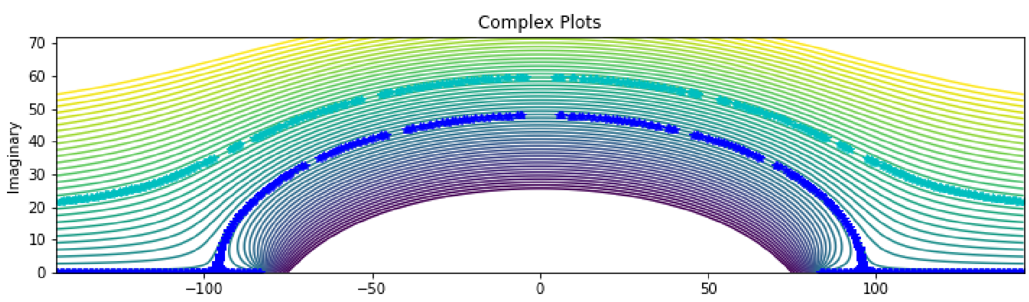


Figure 3: Streamlines for Elliptic Greenhouse

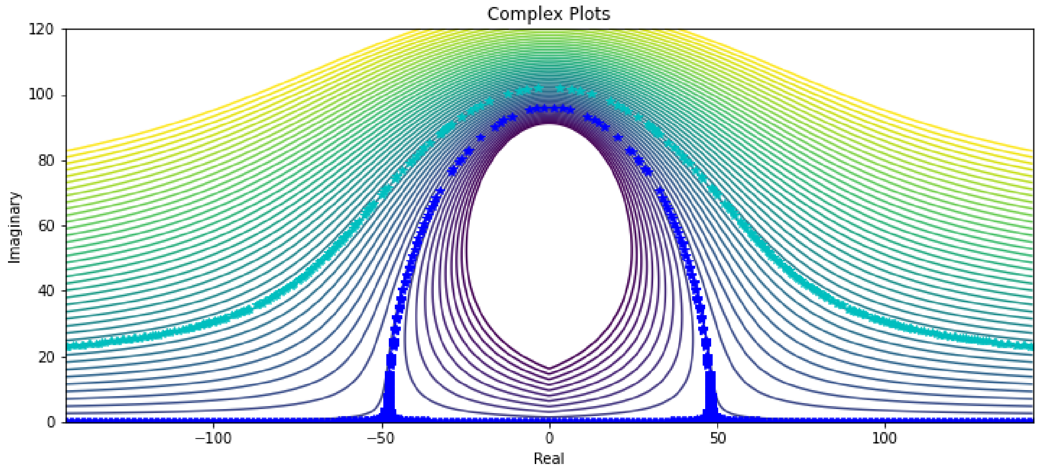


Figure 4: Streamlines for Elliptic Prized Rose

Using the BEM solver from the previous homework, the following figures are created as shown below.

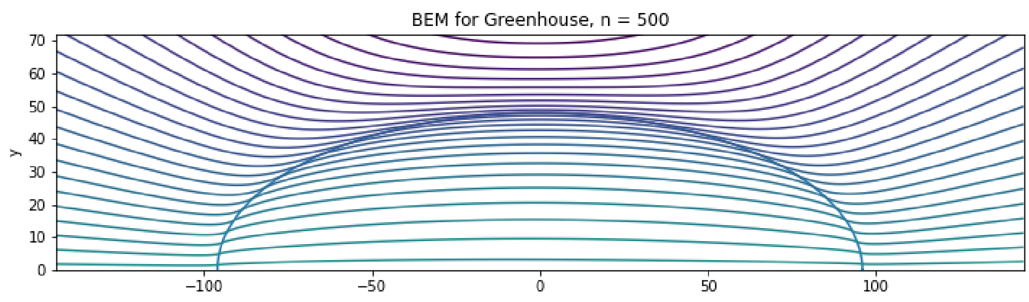


Figure 5: Streamlines from BEM Solver for Greenhouse

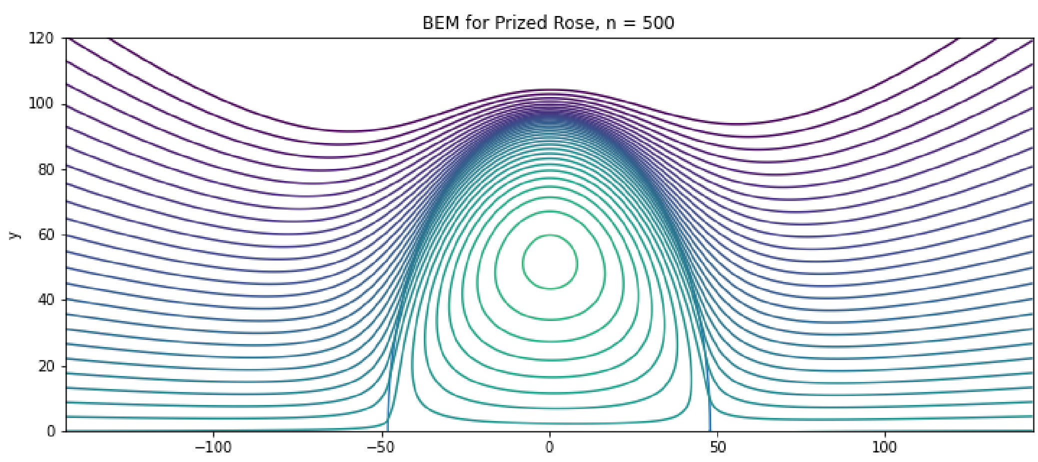


Figure 6: Streamlines from BEM Solver for Prized Rose

It can be seen that there is a clear difference between the analytical solution and the BEM method. It can be seen that from using the BEM method, the streamlines are somewhat going up instead of going down. At close streamlines to the surface this may be an accurate representation, however, close to infinity, it doesn't show a proper uniform flow.

Appendix)

Python Code for Problems 1,2,3

```
# Imported from CFD Project
import numpy as np
import time
from scipy.stats import norm
import matplotlib.pyplot as plt
#################
def HW4(problem='blah',let='a',num=10,num2=None,
        plotzero=False,plotstag=False,func='nope',s=5,pot=True,stream=False):
    g = 500
    X = np.linspace(-s,s,g)
    Y = np.linspace(-s,s,g)

    if problem == 'Problem 1':
        fig, ax = plt.subplots(figsize=(5,5))
    elif problem == 'Problem 2':
        fig, ax = plt.subplots(figsize=(10,5))
    elif problem == 'Problem 3':
        if let == 'b':
            fig, ax = plt.subplots(figsize=(12,3))
        if let == 'c':
            fig, ax = plt.subplots(figsize=(12,5))
    if problem == 'blah':
        fig, ax = plt.subplots(figsize=(20,20))
    #ax.grid()
    ax.set_title('Complex Plots')
    ax.set_xlabel('Real')
    ax.set_ylabel('Imaginary')
    ax.set_xlim([-s,s])
    ax.set_ylim([-s,s])

    if problem == 'Problem 1':
        pass
    elif problem == 'Problem 2':
        ax.set_xlim([-s,s])
        ax.set_ylim([ 0,s])
    elif problem == 'Problem 3':
        if let == 'b':
            X = np.linspace(-144,144,g)
            Y = np.linspace(   0, 72,g)
            ax.set_xlim([-144,144])
            ax.set_ylim([   0, 72])
        if let == 'c':
            X = np.linspace(-144,144,g)
            Y = np.linspace(   0,120,g)
            ax.set_xlim([-144,144])
            ax.set_ylim([   0,120])

    x,y = np.meshgrid(X,Y)
    r = np.arctan2(y,x)
    t = np.sqrt(x**2+y**2)
    z = x + 1j*y
    ''' Input Function Here '''

    if problem == 'Problem 1':
        Fz = z**4/4 - z
        w = z**3 - 1
        u = w.real
        v = w.imag
    elif problem == 'Problem 2':
        M = 1
        Fz = -M*(np.sqrt(3)+z) / (z**2+1)
        w = (z**2+2*np.sqrt(3)*z-1)/ (z**2+1)**2
        u = w.real
        v = w.imag
    elif problem == 'Problem 3':
        if let == 'b':
            a = 72
```

```
            b = 24 * np.sqrt(3)
        if let == 'c':
            a = 72
            b = 24 * np.sqrt(3) * 1j
        l = z/2 + np.sign(z.real) * np.sqrt((z/2)**2-b**2)
        Fz = l + a**2/l
    else:
        Fz = -M*(np.sqrt(3)+z) / (z**2+1)

    phi = Fz.real
    psi = Fz.imag

    if func == 'lmao':
        if pot == True:
            mean = np.mean(phi)
            std = np.std(phi)
            levels = np.linspace(norm(mean,std).ppf(0.1), norm(mean,std).ppf(0.9), num)
            ax.contour(x, y, phi, levels=levels, linestyles='dashed',alpha=0.5)
        mean = np.mean(psi)
        std = np.std(psi)
        levels = np.linspace(norm(mean,std).ppf(0.1), norm(mean,std).ppf(0.9), num)
        ax.contour(x, y, psi, levels=levels)
    else:
        if pot == True:
            ax.contour(x, y, phi, num, linestyles='dashed',alpha=0.5)
        ax.contour(x, y, psi, num)

    #### Plot Zero Streamline ####
    if plotzero == True:
        zero = np.argwhere(np.around(psi,1) == 0)
        zeros = np.zeros((2,zero.shape[0]))
        for i in range(zero.shape[0]):
            zeros[0,i] = x[zero[i,0],zero[i,1]]
            zeros[1,i] = y[zero[i,0],zero[i,1]]
        ax.plot(zeros[0],zeros[1],'b*')

    if stream != False:
        zero = np.argwhere(np.around(psi,1) == stream)
        zeros = np.zeros((2,zero.shape[0]))
        for i in range(zero.shape[0]):
            zeros[0,i] = x[zero[i,0],zero[i,1]]
            zeros[1,i] = y[zero[i,0],zero[i,1]]
        ax.plot(zeros[0],zeros[1],'c*')

    if 'u' and 'v' in locals():
        mag = np.sqrt(u**2+v**2)
        #### Plot Stagnation Points ####
        if plotstag == True:
            zero = np.argwhere(np.around(mag,1) == 0)
            zeros = np.zeros((2,zero.shape[0]))
            for i in range(zero.shape[0]):
                zeros[0,i] = x[zero[i,0],zero[i,1]]
                zeros[1,i] = y[zero[i,0],zero[i,1]]
            ax.plot(zeros[0],zeros[1],'r*')
        if num2 != None:
            ll = int(g/num2)
            skip = (slice(None, None, ll), slice(None, None, ll))
            ax.quiver(x[skip],y[skip],u[skip],v[skip])
    if problem == 'Problem 1':
        name = 'first'
    elif problem == 'Problem 2':
        name = 'second'
    elif problem == 'Problem 3':
        if let == 'b':
            name = 'third'
        if let == 'c':
            name = 'fourth'
    if 'name' in locals():
        gg = 'images/' + name + '.png'
        plt.savefig(gg)

if __name__ == "__main__":
```

```
    #HW4('Problem 1',func='lmao',num=10,plotstag=True,num2=20,s=3)
    HW4('Problem 2',func='lmao',num=20,s=5,pot=True)# num2=50, plotstag=True # weird stagnations stuff
    HW4('Problem 3','b',func='lmao',pot=False,num=50,plotzero=True,stream=17) # good now tbh
    HW4('Problem 3','c',func='lmao',pot=False,num=50,plotzero=True,stream=17) # also good now tbh
    #HW4(func='lmao',pot=False,num=50,s=5,num2=51)
```

Python Code for Problem 3 BEM Solver

```
# -*- coding: utf-8 -*-
"""
Created on Fri Feb 10 10:06:00 2023

@author: jjser
"""
# Imported from CFD Project
import numpy as np
import matplotlib.pyplot as plt
import math
from numpy import linalg as la
from numpy import asarray
import pandas
import time
from scipy.stats import norm

#################
def G(r):
    return np.log(r)/(2*np.pi)
def dGdn(R,N,r):
    return R @ N / (2*np.pi*r)

def prob(a,b,n,num):
    # 'a' is the length of the major axis
    # 'b' is the length of the minor axis
    # 'n' is the number of elements in the boundary element method
    # 'num' is the number of streamlines
    if a > b:
        shape = 'Greenhouse'
        prob = 1
        fig0, ax0 = plt.subplots(figsize=(12,3))
    elif b > a:
        shape = 'Prized Rose'
        prob = 2
        fig0, ax0 = plt.subplots(figsize=(12,5))
    else:
        shape = 'bro wrong problem'
        prob = False

    dtheta = 2*np.pi / n
    x0 = np.zeros(n+1)
    y0 = np.zeros(n+1)
    theta = np.zeros(n+1)

    # Creating the actual circle first
    for i in range(n+1):
        theta[i] = i * dtheta
        x0[i] = a*np.cos(theta[i])
        y0[i] = b*np.sin(theta[i])
    # Applying BEM
    A = np.zeros((n,n))
    B = np.zeros((n,n))
    area = np.zeros(n)
    cx = np.zeros(n)
    cy = np.zeros(n)
    N = np.zeros((n,2))

    #fig0, ax0 = plt.subplots(figsize=(5,5))
    #fig0, ax0 = plt.subplots()
    title = 'BEM for ' + shape + ', n = ' + str(n)
    ax0.set_title(title)
    ax0.set_xlabel('x')
    ax0.set_ylabel('y')
    s = 1.5*max(a,b)
    if a > b:
        ax0.set_xlim([-144,144])
        ax0.set_ylim([   0, 72])
    elif b > a:
        ax0.set_xlim([-144,144])
        ax0.set_ylim([   0,120])
```

```python

    for i in range(n):
        area[i] = math.sqrt((x0[i+1]-x0[i])**2+(y0[i+1]-y0[i])**2)
        cx[i] = 0.5 *  (x0[i+1]+x0[i])
        cy[i] = 0.5 *  (y0[i+1]+y0[i])
        N[i][0] = -(y0[i+1]-y0[i])
        N[i][1] =  (x0[i+1]-x0[i])
        N[i] = N[i] / la.norm(N[i])
    for i in range(n):
        for j in range(n):
            if i==j:
                B[i,j] = 0
                A[i,j] = -0.5
            else:
                R = [cx[j]-cx[i],cy[j]-cy[i]]
                r = la.norm(R)
                R = R/r
                A[i,j] = dGdn(R,N[j],r) * area[j]
                B[i,j] = G(r) * area[j]
    ax0.plot(x0,y0)

    # Solve for psi1 and dpsi1dn
    psi1 = np.zeros(n)
    for i in range(n):
        psi1[i] = -y0[i]
    dpsi1dn = la.inv(B) @ A @ psi1.T

    # Plotting the Cylinder
    g = 501
    x = np.linspace(-s,s,g)
    y = np.linspace(-s,s,g)
    X,Y = np.meshgrid(x,y)
    psi = np.zeros((g,g))

    # Streamfunction Contours
    for k in range(n):
        rx = cx[k] - X
        ry = cy[k] - Y
        rmag = np.sqrt(rx**2+ry**2)
        psi += (psi1[k]*(rx*N[k][0]+ry*N[k][1])/(2*np.pi*rmag)-G(rmag)*dpsi1dn[k])*area[k]
    psi += Y

    mean = np.mean(psi)
    std = np.std(psi)
    levels = np.linspace(norm(mean,std).ppf(0.1), norm(mean,std).ppf(0.9), num)
    ax0.contour(x, y, psi, levels=levels)

    if prob != False:
        name = 'images/BEM' + str(prob) + '.png'
        fig0.savefig(name)
    '''
    zero = np.argwhere(np.around(psi,1) == 0)
    zeros = np.zeros((2,zero.shape[0]))
    for i in range(zero.shape[0]):
        zeros[0,i] = X[zero[i,0],zero[i,1]]
        zeros[1,i] = Y[zero[i,0],zero[i,1]]
    ax0.plot(zeros[0],zeros[1],'b*')

    zero = np.argwhere(np.around(psi,1) == 17)
    zeros = np.zeros((2,zero.shape[0]))
    for i in range(zero.shape[0]):
        zeros[0,i] = X[zero[i,0],zero[i,1]]
        zeros[1,i] = Y[zero[i,0],zero[i,1]]
    ax0.plot(zeros[0],zeros[1],'c*')
    '''

##### BEM for a 4:1 ellipse
#prob(4,1,500,15)
prob(96,48,500,50)
prob(48,96,500,50)
start_time = time.time()
print("--- %10s seconds ---" % np.round((time.time() - start_time),4))
```