"

Bryce Lindsey - MAE5943 HW8 (Homann flow)

```
This document summarizes the HW8 assignment for MAE5943
(problem statement seen below)

submitted by:
    Bryce Lindsey
    bryce.lindsey@okstate.edu
```

HW8 Problem Statement :

Stagnation point flow problem towards an axisymmetric body such as a sphere or a cone can be represented using the ordinary differential equation (ODE) below:

$$\varphi''' + 2\varphi\varphi'' - (\varphi')^2 + 1 = 0$$

Boundary Conditions:

$$\varphi_{(\eta=0)} = 0$$

$$\varphi'_{(\eta=0)} = 0$$

$$\varphi'_{(\eta\to\infty)} \to 0$$

Solve the Homann flow problem using the numerical methods discussed in the lecture.

- Investigate the effect of η_{max} and N_{max}.
    - Assume that N_{max}=40 and compare the solutions for η_{max}=[5, 10, 20].
    - Assume η_{max}=10 and compare the solution for N_{max}=[10, 20, 40, 80].
- Discuss your observations and submit your findings in a technical report format.
- On the same plot, compare the solutions of Hiemenz and Homann flows for η_{max}=10 and N_{max}=40.
- Include your code in the appendix.

Homann Flow

Homann flow is an approach to solve a flow approaching a radially-symmetric body, where a stagnation point exists, by splitting the analysis into the inviscid outer region and the viscous boundary layer region. Homann flow is typically considered:

- Laminar
- Axisymmetric
- Steady
- Incompressible

We showed in lecture that Homann flow can be represented by the following ODE using a similarity solution:

$$\phi''' + 2\phi\phi'' - (\phi)'^2 + 1 = 0$$

In this ODE, the parameter φ' corresponds to the velocity profile at a particular location along the body in the flow. φ' is a function of η, which is a non-dimensional term which can be associated with a location on the body (if desired).

Boundary Conditions:

$$\phi'(0) = 0$$

$$\phi(0) = 0$$

$$\phi'(\eta \to \infty) = 1$$

(I believe there is a typo in the problem statement for Φ')

Tranformation:

with φ' = h and φ = p, and using Thomas' algorithm, this ODE can be *numerically* solved.

Homann flow function

My Homann flow function is defined in the block below

Inputs

- *ηmax* = Maximum value of η (treat as infinity boundary condition)
- *N* = number of discrete points throughout η to use
- *h_itermax* = Maximum amount of iterations for h to converge (given the error inputs). This prevents a "run away" while loop during h convergence.
- *ϵProfile* = Each iteration of "h" will compute the profile error between the current h and the last h. Make ϵProfile small enough to have confidence in the final profile of h.
- *ϵBC* = Computes the error of the last 2 points in the h vector (at the BCs). These 2 points can be thought of as h[∞]-h[∞-1] and the difference between the two should be very small after the convergence of h occurs (h changes very little at the BCs)

Output

- *η* = Output values of η.
- *h* = Is refined throughout the h convergence loops and is equivalent to the primary output parameter we want, φ' (which corresponds to a local velocity profile).

BL_homann

Computes the Homann flow transformed solution

```
BL_homann(ηmax, N, h_itermax, ϵProfile, ϵBC)
```

If the arguments are missing, it will use the default values.

```
BL_homann(ηmax=10, N=50, h_itermax=50, ϵProfile=1e-6, ϵBC=1e-6)
```

```
"""
Computes the Homann flow transformed solution

    BL_homann(ηmax, N, h_itermax, ϵProfile, ϵBC)

If the arguments are missing, it will use the default values.

    BL_homann(ηmax=10, N=50, h_itermax=50, ϵProfile=1e-6, ϵBC=1e-6)
```

```
"""
function BL_homann(ηmax=10, N=50, h_itermax=50, ϵProfile=1e-6, ϵBC=1e-6)
        Δη = ηmax/N
        Δη² = Δη^2

        iter = 0
        errorProfile = 1.
        errorBC = 1.

        A = zeros(N+1)
        B = zeros(N+1)
        C = zeros(N+1)
        D = zeros(N+1)

        G = zeros(N+1)
        H = zeros(N+1)

        p = zeros(N+1)
        h = zeros(N+1)
        η = zeros(N+1)

        η = [(i-1)*Δη for i=1:N+1]

        h[1] = 0.0
        h[N+1] = 1.0
        p[1] = 0.0

        G[1] = 0.0
        H[1] = 0.0

        h = [(i-1)/N for i=1:N+1]

        for i=2:N+1
            p[i] = p[i-1] + (h[i] + h[i-1]*Δη/2)
        end

        hstart = time()
        println("################################ \nBeginning h-convergence loop... \n")
        println("h iteration   h profile error   convergence of h(η -> ∞)")
        println("---------------------------------------")

        while (ϵProfile<=errorProfile || ϵBC <= errorBC) && iter<h_itermax

            A = [ 1/Δη² + p[i]/Δη for i=1:N+1]
            B = [-2/Δη² - h[i] for i=1:N+1]
            C = [ 1/Δη² - p[i]/Δη for i=1:N+1]
            D = [ 1 for i=1:N+1]
```

```
            for i=2:N+1
                G[i] = - ( C[i]*G[i-1] + D[i] )/(B[i] + C[i] * H[i-1])
                H[i] = - A[i] /(B[i] + C[i] * H[i-1])
            end

            hold = copy(h)
            for i=N:-1:2
            h[i] = G[i] + H[i] * h[i+1]
            end

            errorBC = abs(h[N+1]-h[N])
            errorProfile = maximum(abs.(hold-h))


            for i = 2:N+1
            p[i] = p[i-1] + (h[i] + h[i-1])*Δη/2
            end
            @printf("%4.4d \t %16.6e \t %16.6e \n", iter, errorProfile, errorBC)


            iter += 1
        end
        println("\n --> time taken for h convergence: $(time()-hstart) \n")
        return η, h

    end


# end of function
```

```
([0.0, 0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.4, 1.6,   more ,10.0], [0.0, 0.242688, 0.44557, 
```

```
Baseline convergence for default inputs:

################################
Beginning h-convergence loop...

h iteration   h profile error   convergence of h(η -> ∞)
---------------------------------------
0000          1.298370e+00        4.258480e-05
0001          5.978384e-01        1.529414e-07
0002          1.261300e-01        1.862502e-10
0003          3.135210e-02        1.063682e-11
0004          6.878020e-03        8.482104e-14
0005          1.408148e-03        0.000000e+00
0006          2.714288e-04        2.220446e-16
0007          4.947427e-05        0.000000e+00
0008          8.618355e-06        0.000000e+00
0009          1.449355e-06        0.000000e+00
0010          2.372800e-07        1.110223e-16

 --> time taken for h convergence: 0.0075190067291259766
```

Tasks given in problem statement

Task 1: Investigate effects of η_{max} changes

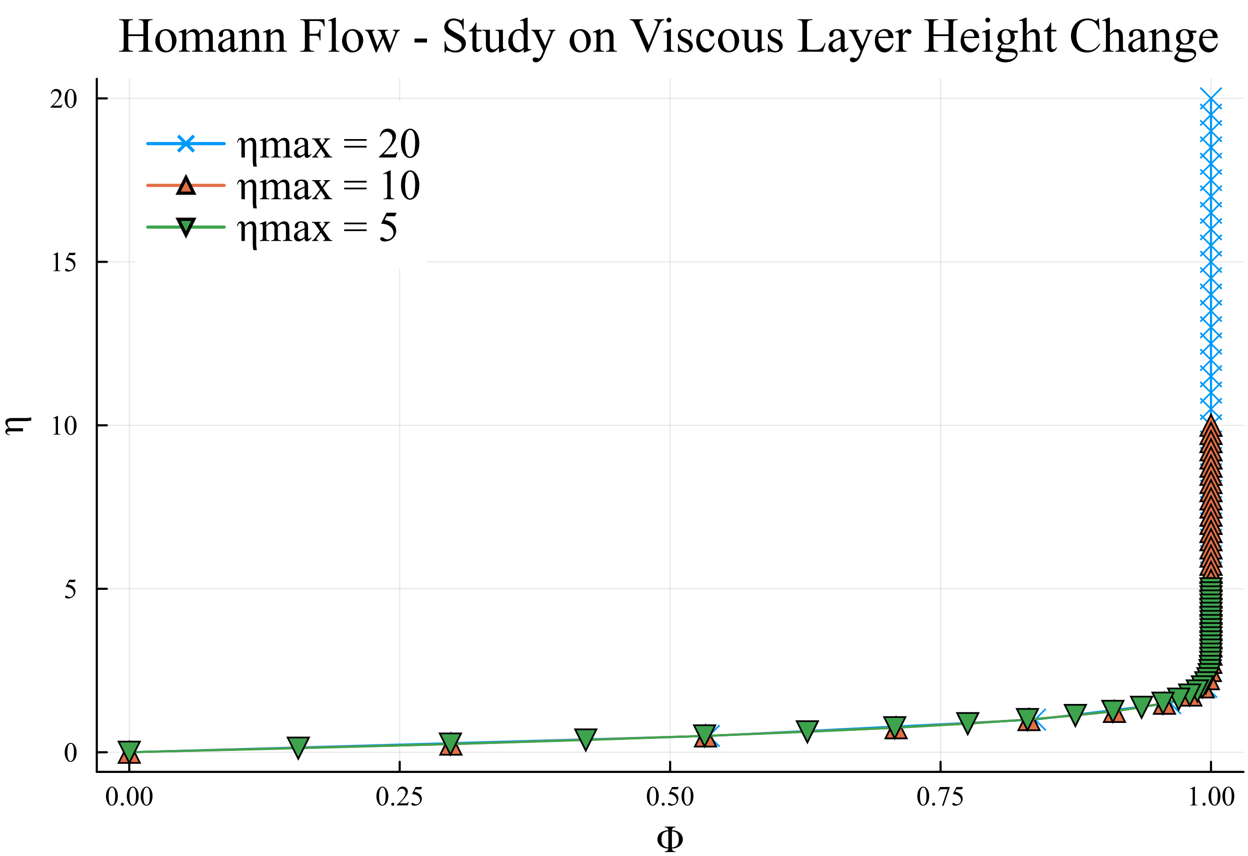
According to the problem statement, the following inputs will be used

N = 40

ηmax = 5 , 10, 20

ϵProfile = 1e-8 (my personal choice)

The results of η vs φ' will be plotted for each ηmax




```julia
begin
    η5, h5 = BL_homann(5, 40, 50, 1e-8)
    η10, h10 = BL_homann(10, 40, 50, 1e-8)
    η20, h20 = BL_homann(20, 40, 50, 1e-8)

    plot(reverse([h5 h10 h20]),reverse([η5 η10 η20]),
        fontfamily="Times",
        title="Homann Flow - Study on Viscous Layer Height Change",
        label = ["ηmax = 20" "ηmax = 10" "ηmax = 5"],
        legend_position = :topleft,
        legendfontsize=12,
        legend_foreground_color=nothing,
        markersize=[5 5 5],
        markershape = [:x :utriangle :dtriangle],
        xlabel="Φ", ylabel="η",
    )


end
```

```
################################
Beginning h-convergence loop...

h iteration   h profile error   convergence of h(η -> ∞)
---------------------------------------
0000         8.994720e-01        5.052074e-05
0001         2.885161e-01        1.485845e-06
0002         2.872255e-02        6.489978e-08
0003         4.562970e-03        2.964637e-09
0004         7.806314e-04        1.305536e-10
0005         1.326575e-04        2.900646e-11
0006         2.205718e-05        1.956491e-11
0007         3.593980e-06        2.023137e-11
0008         5.764474e-07        2.017353e-11
0009         9.141770e-08        2.017964e-11
0010         1.438523e-08        2.017897e-11
0011         2.251834e-09        2.017897e-11

 --> time taken for h convergence: 0.0005841255187988281

################################
Beginning h-convergence loop...

h iteration   h profile error   convergence of h(η -> ∞)
---------------------------------------
0000         1.349629e+00        8.091311e-05
0001         6.547217e-01        2.706832e-07
0002         1.415757e-01        1.835824e-09
0003         3.607937e-02        3.936051e-11
0004         8.043665e-03        1.680878e-13
0005         1.669163e-03        2.664535e-15
0006         3.215307e-04        0.000000e+00
0007         5.842378e-05        0.000000e+00
0008         1.012875e-05        0.000000e+00
0009         1.692803e-06        0.000000e+00
0010         2.750627e-07        0.000000e+00
0011         4.375699e-08        0.000000e+00
0012         6.852644e-09        0.000000e+00

 --> time taken for h convergence: 0.0004980564117431641

################################
Beginning h-convergence loop...

h iteration   h profile error   convergence of h(η -> ∞)
```

Task 1 Discusion

As ηmax increases, regions beyond the boundary layer are better captured. However, at the lowest ηmax value of 5, the entire viscous region is captured. Just how high to set ηmax depends on how much of the inviscous region one would like to capture, but a ηmax value of 20 does a very good job of capturing all of the velocity profile out to the freestream and beyond.

Task 2: Investigate effects of N changes

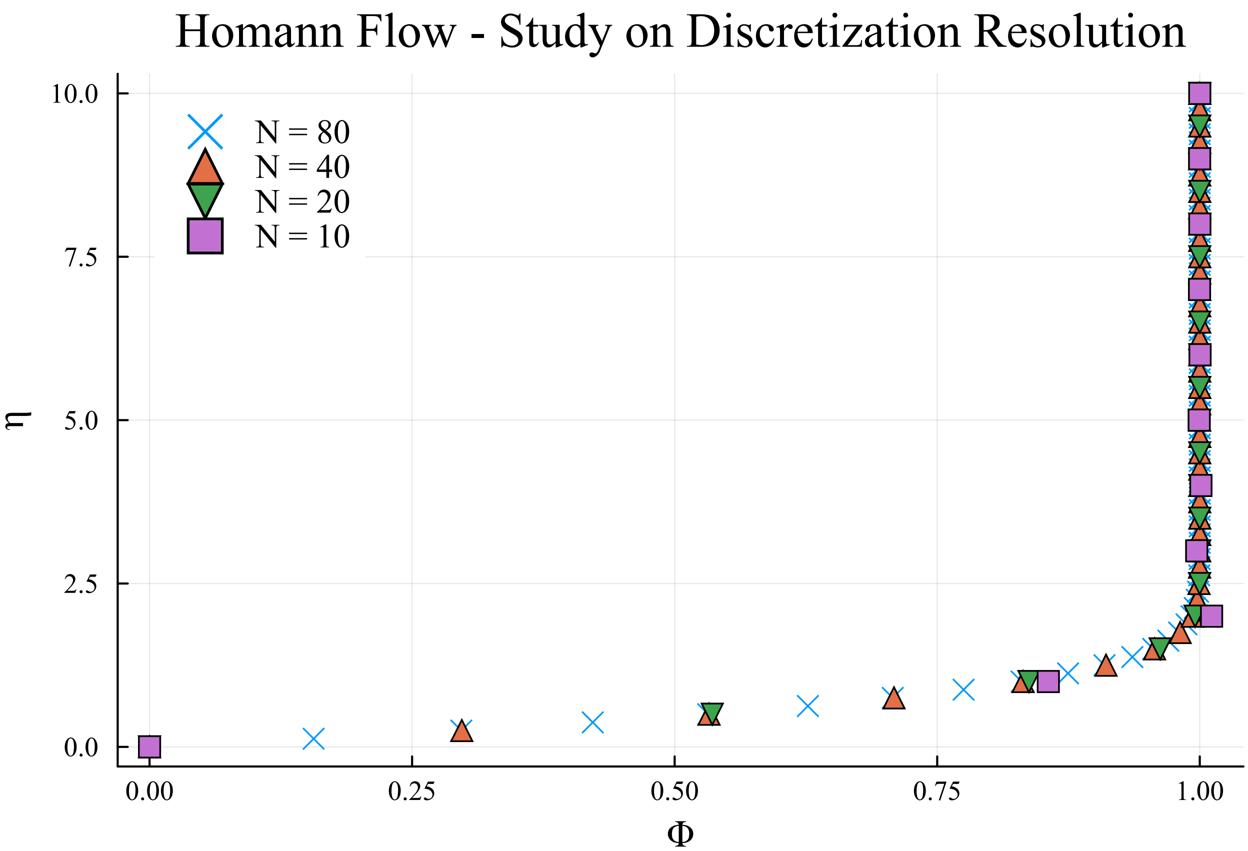
According to the problem statement, the following inputs will be used

N = 10, 20, 40, 80

ηmax = 10

ϵProfile = 1e-8 (my personal choice)

The results of η vs φ' will be plotted for each N



```
begin
    η_N80, h_N80 = BL_homann(10, 80, 50, 1e-6)
    η_N40, h_N40 = BL_homann(10, 40, 50, 1e-6)
    η_N20, h_N20 = BL_homann(10, 20, 50, 1e-6)
    η_N10, h_N10 = BL_homann(10, 10, 50, 1e-6)


    plot([h_N80, h_N40, h_N20, h_N10],[η_N80, η_N40, η_N20, η_N10];
        seriestype=:scatter,
        fontfamily="Times",
        title="Homann Flow - Study on Discretization Resolution",
        label = ["N = 80" "N = 40" "N = 20" "N = 10"],
        legend_position = :topleft,
        legendfontsize=10,
        legend_foreground_color=nothing,
        markersize=[5 5 5 5],
        markershape = [:x :utriangle :dtriangle :square],
        xlabel="Φ", ylabel="η",
    )


end
```

```
##############################
Beginning h-convergence loop...

h iteration   h profile error   convergence of h(η -> ∞)
----------------------------------------
0000         1.209918e+00        1.085382e-05
0001         4.940185e-01        4.645620e-08
0002         9.385258e-02        2.292859e-10
0003         2.214499e-02        8.468781e-13
0004         4.796458e-03        1.776357e-15
0005         9.797532e-04        0.000000e+00
0006         1.879955e-04        0.000000e+00
0007         3.421047e-05        0.000000e+00
0008         5.965864e-06        1.110223e-16
0009         1.006677e-06        0.000000e+00
0010         1.656970e-07        2.220446e-16

 --> time taken for h convergence: 0.0006818771362304688

##############################
Beginning h-convergence loop...

h iteration   h profile error   convergence of h(η -> ∞)
----------------------------------------
0000         1.349629e+00        8.091311e-05
0001         6.547217e-01        2.706832e-07
0002         1.415757e-01        1.835824e-09
0003         3.607937e-02        3.936051e-11
0004         8.043665e-03        1.680878e-13
0005         1.669163e-03        2.664535e-15
0006         3.215307e-04        0.000000e+00
0007         5.842378e-05        0.000000e+00
0008         1.012875e-05        0.000000e+00
0009         1.692803e-06        0.000000e+00
0010         2.750627e-07        0.000000e+00

 --> time taken for h convergence: 0.00040793418884277344

##############################
Beginning h-convergence loop...

h iteration   h profile error   convergence of h(η -> ∞)
----------------------------------------
0000         1.501970e+00        5.708232e-04
0001         8.460316e-01        1.466840e-06
```

Task 2 Discussion

The full "shape" of the transformed velocity profile, φ', is captured in each instance of different values of "N". However, a value of N=10 seems to have two interesting qualities:

- The number of iterations required is much higher than the other values of N
- The curve seems to overshoot at the "turn" which, given the high number of iterations for h, implies that the solution did not converge.

Actually, the N=10 h-iterations hit the maximum (h_itermax=50). Given this info, I'd say a value of N=10 is not appropriate. The other values seem to do fine, but in this case I would use a value close to 100 for confidence.

Task 3: Compare Hiemenz and Homann flow solutions

According to the problem statement, the following inputs will be used

N = 40
ηmax = 10

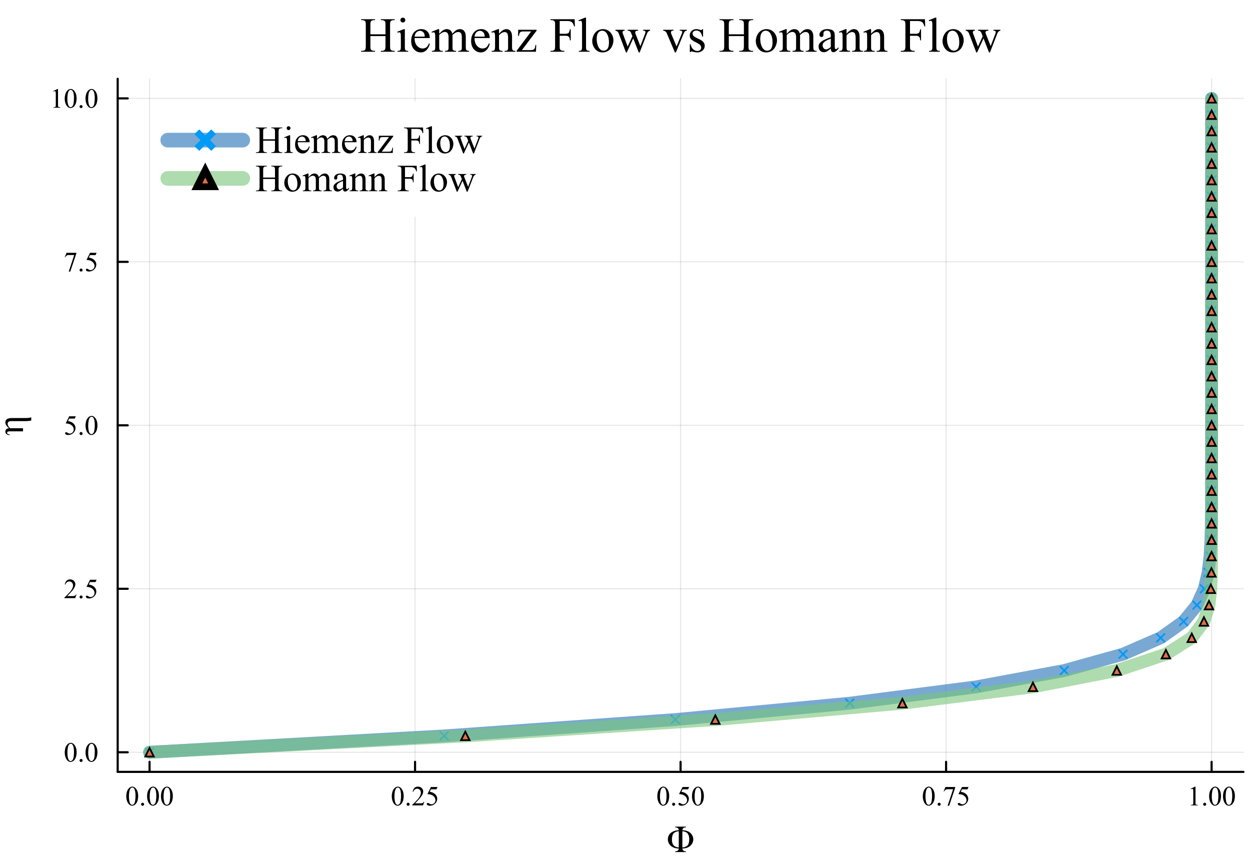
ϵProfile = 1e-8 (my personal choice for Homann function)

The results of η vs φ' will will be plotted for each solution

`BL_hiemenz`

Inserting my Hiemenz flow function from HW7 here. For details, see the .jl script on my Github:

BLHW7HiemenzFlow




```julia
begin
    ηHiemenz, hHiemenz = BL_hiemenz(10, 40, 1e-7)
    ηHomann, hHomann = BL_homann(10, 40, 50, 1e-7, 1e-6)


    plot([hHiemenz, hHomann], [ηHiemenz, ηHomann];
        label=["Hiemenz Flow" "Homann Flow"],
        linecolor = ["#2171b5" "#78c679"],
        lw=6,
        legend_position = :topleft,
        legendfontsize=11,
        legend_foreground_color=nothing,
        markersize=[2 2],
        markershape = [:x :utriangle],
        fontfamily="Times",
        xlabel="Φ", ylabel="η",
        title="Hiemenz Flow vs Homann Flow",
        linealpha=[.6 .6]
        )

end
```

```
################################
Beginning h-convergence loop...

h iteration   h profile error   convergence of h(η -> ∞)
---------------------------------------
0000         1.349629e+00        8.091311e-05
0001         6.547217e-01        2.706832e-07
0002         1.415757e-01        1.835824e-09
0003         3.607937e-02        3.936051e-11
0004         8.043665e-03        1.680878e-13
0005         1.669163e-03        2.664535e-15
0006         3.215307e-04        0.000000e+00
0007         5.842378e-05        0.000000e+00
0008         1.012875e-05        0.000000e+00
0009         1.692803e-06        0.000000e+00
0010         2.750627e-07        0.000000e+00
0011         4.375699e-08        0.000000e+00

 --> time taken for h convergence: 0.0009229183197021484
```

Task 3 Discussion

Both Hiemenz and Homann flow solutions are very close. The area that differs is the gradient at the boundary of viscous and inviscous solutions. Homann flow solution suggests the gradient in this location is slightly higher, but this discrepancy between the two is very small!

END OF ASSIGNMENT :)