1) Now, let's solve with conservation & NS

Conservation: $Q = A_0 u_0 = A(z) u(z)$; assuming incompressible: $\rho(0) = \rho(z) = \rho_{water}$

$$u(z) = \frac{A_0 u_0}{A(z)} = \frac{a^2}{r(z)^2} u_0$$

Finally we have $u(z) = \frac{a^2}{r(z)^2} u_0$

2) Now continue to NS (cylindrical)

$$\hat{z}: \rho\left(\frac{\partial u_z}{\partial t} + u_r \frac{\partial u_z}{\partial r} + \frac{u_\theta}{r}\frac{\partial u_z}{\partial \theta} + u_z \frac{\partial u_z}{\partial z}\right) = -\frac{\partial P}{\partial z} + \mu[\;] + \rho g$$

stable: 0; 0; 0; inviscid: 0

$u = u(z)$ ⇓

$$u_z \frac{\partial u_z}{\partial z} = \frac{-1}{\rho}\frac{\partial P}{\partial z} + g$$ ← Governing equation!

Now, let's find $P(z)$:

$$P(z) = P_{out} + (P_{in} - P_{out}) = P_a + \sigma\left(\frac{1}{r(z)} + \frac{1}{R_2}\right)$$

$\Delta P_{laplace}$

$$\frac{\partial P}{\partial z} = \sigma \frac{(-1)}{r^2}\dot{r} = -\sigma r^{-2}\dot{r}$$

Plug it all in:

$$u\frac{\partial u(z)}{\partial z} = \frac{-1}{\rho}\frac{\partial P}{\partial x} + g$$

$$\frac{\partial}{\partial z}\left[a^2 u_0 r(z)^{-2}\right] = a^2 u_0 (-2) r^{-3}\dot{r}$$

⇓

$$\left(a^2 u_0 r^{-2}\right)\left(a^2 u_0 (-2) r^{-3}\right)\dot{r} = \frac{\sigma}{\rho} r^{-2}\dot{r} + g$$

$$\Rightarrow \dot{r} = g \Big/ \left[-2a^4 u_0^2 r^{-5} - \frac{\sigma}{\rho} r^{-2}\right]$$ ODE for r

5) $R_2 \to \infty$, let's apply Bernoulli's law:

$$\frac{1}{2}\rho U_0^2 + \rho g z + P_A = \frac{1}{2}\rho U(z)^2 + P_{IS} \qquad \begin{cases} R_1 = r \\ R_2 \to \infty \end{cases}$$

Surface pressure due to curvature: $\nabla \cdot n = \frac{1}{R_1} + \frac{1}{R_2} \to \frac{1}{r}$

Now, $P = P_{out} + \Delta P = P_a + \frac{\sigma}{r}$. But P_a cancels out

$$\frac{1}{2}\rho V_0^2 + \rho g \cancel{z}^{\,0} + \underbrace{\frac{\sigma}{a}}_{\text{init radius}} = \frac{1}{2}\rho V(z)^2 + \frac{\sigma}{r}$$

take $z=0$ @ $r=a$

Re-arranging: $$\frac{U(z)}{U_0} = \left[1 + \frac{z}{Fr}\frac{z}{a} + \frac{z}{We}\left(1 - \frac{a}{r}\right)\right]^{1/2}$$

$$\begin{cases} Fr = U_0^2/ga \\ We = \rho U_0^2/\sigma \end{cases}$$

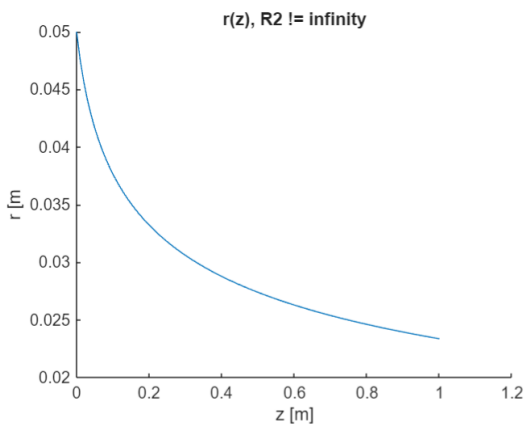
Now, lets use conservation, as we saw before: $r(z)^2 U(z)^2 = a^2 U_0^2$

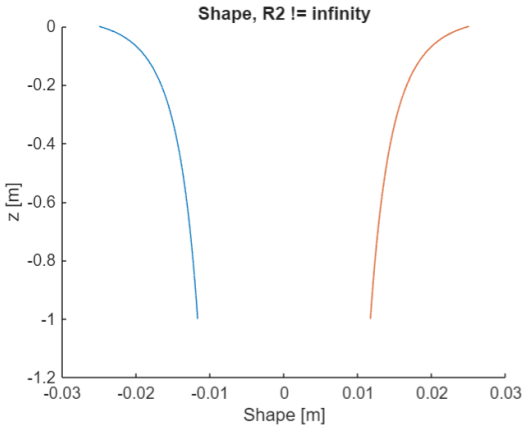
$$\frac{r(z)}{a} = \left[1 + \frac{z}{Fr}\frac{z}{a} + \frac{z}{We}\left(1 - \frac{a}{r}\right)\right]^{-1/4}$$

↳ Solve this numerically in Matlab

3. I used Euler's method for my numerical solver. Euler's method consists of starting at the initial condition and iterating forward according to y_{n+1} = y_n + h f(x_n, y_n) where h is my discretization. I choose h to be 0.01 as it gave me a high resolution of accuracy in my solving. Additionally, my initial condition was r(z=0) = 5 cm.

4. I iterated over the domain of z = 0 [m] to 1 [m] using the givens and the results are shown below:

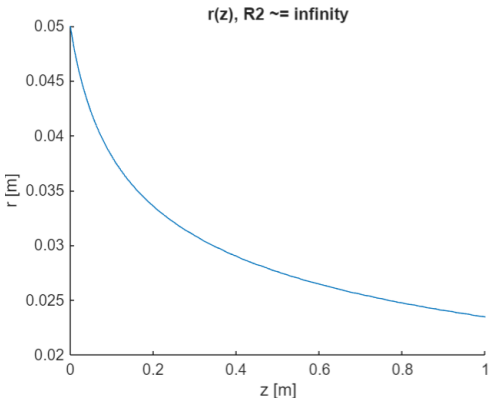


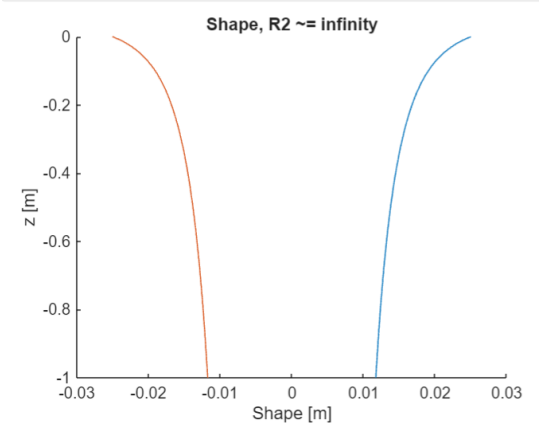


5. We developed the expression from class:

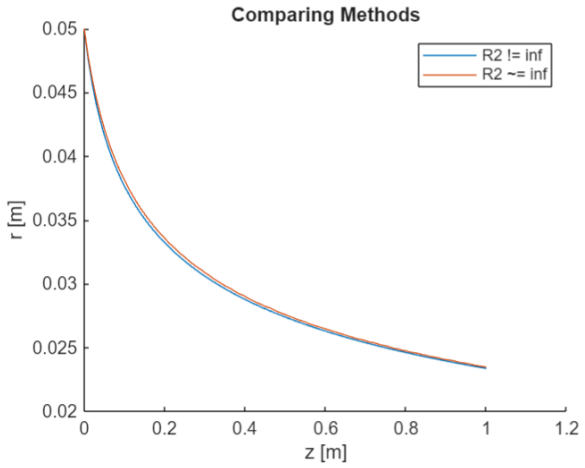
Results from the R2->inf below:

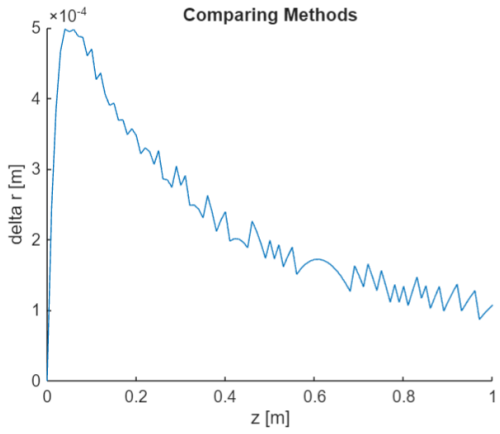
compare the results below:





And we can compare the results directly one with another:





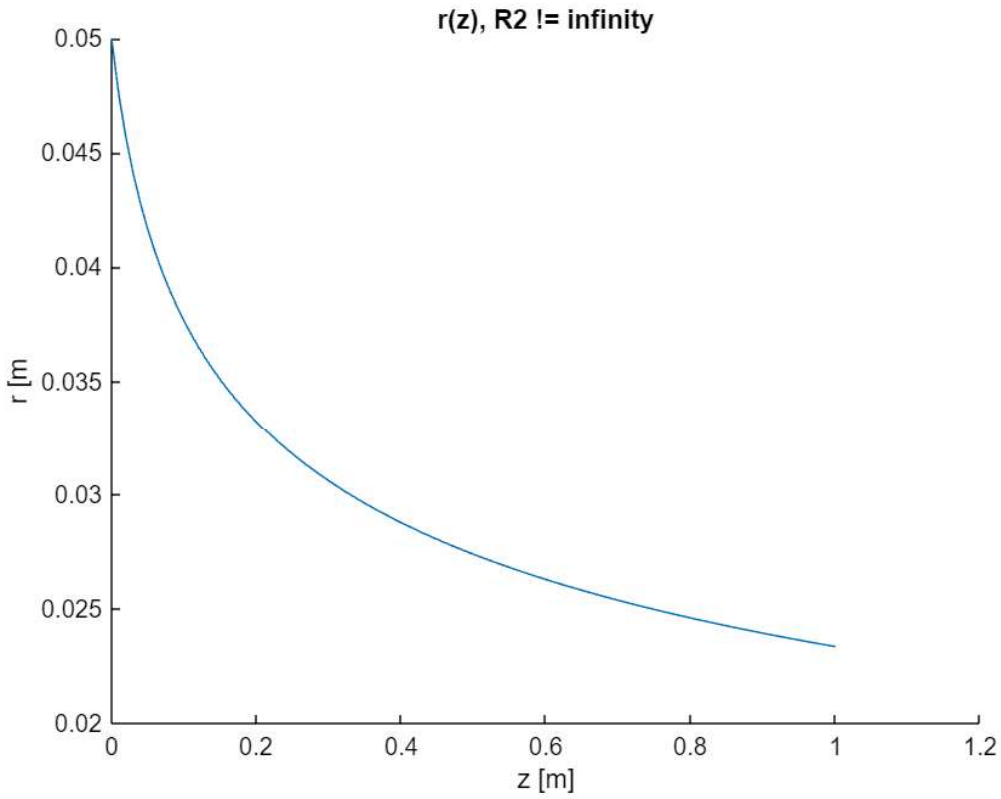
We see that the solutions are very similar and there is not much difference between them. The error is proportional to the radius itself. The error is of order of magnitude 0.1 mm is very small when the size of the radius of order magnitude of cm. 0.1 mm / cm is 0.01 and this is very small!

The error is 0 in the beginning as the boundary condition ensures this and it grows slightly then the error contuse to decay and stabilize near 0.01 mm.

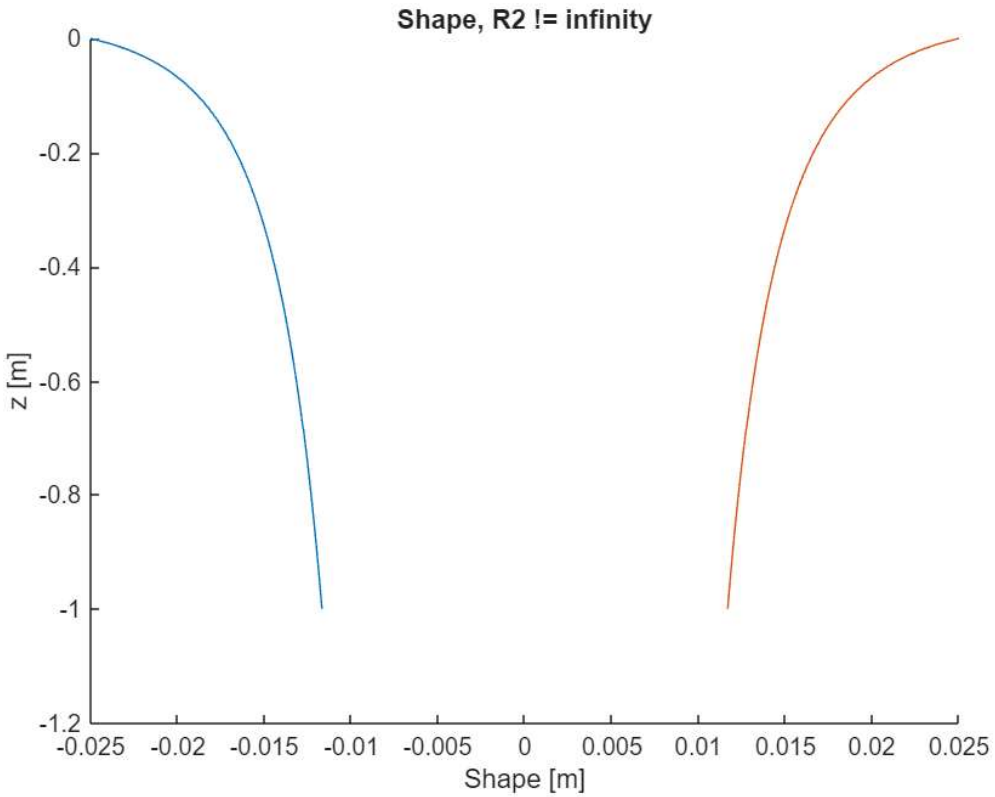
6. We do not see any oscillatory behavior. Oscillatory behaviors, from instabilities, appears when we have t,r dependencies in or Uz and alternately t,z dependencies in r. There is a coupling between time and wake number manifested in the dispersion relation. This is what leads to the oscillatory notion we see in disturbances to the flow. But in our problem we assume steady state as well as U=U(z) and therefore independent of r,t. Therefore, the solutions are without oscillatory disturbances.

However, there is some oscillation in the error magnitude. This is seen in the plot above. We can directly relate this to the discretization size of our solver. For higher discretization of the system we will have more oscillations in the error. This is due to the fact that the steps are very small and the numerical solution changes very very rapidly. Note, this oscillation is very different than the dispersion relation oscillation we saw in class.

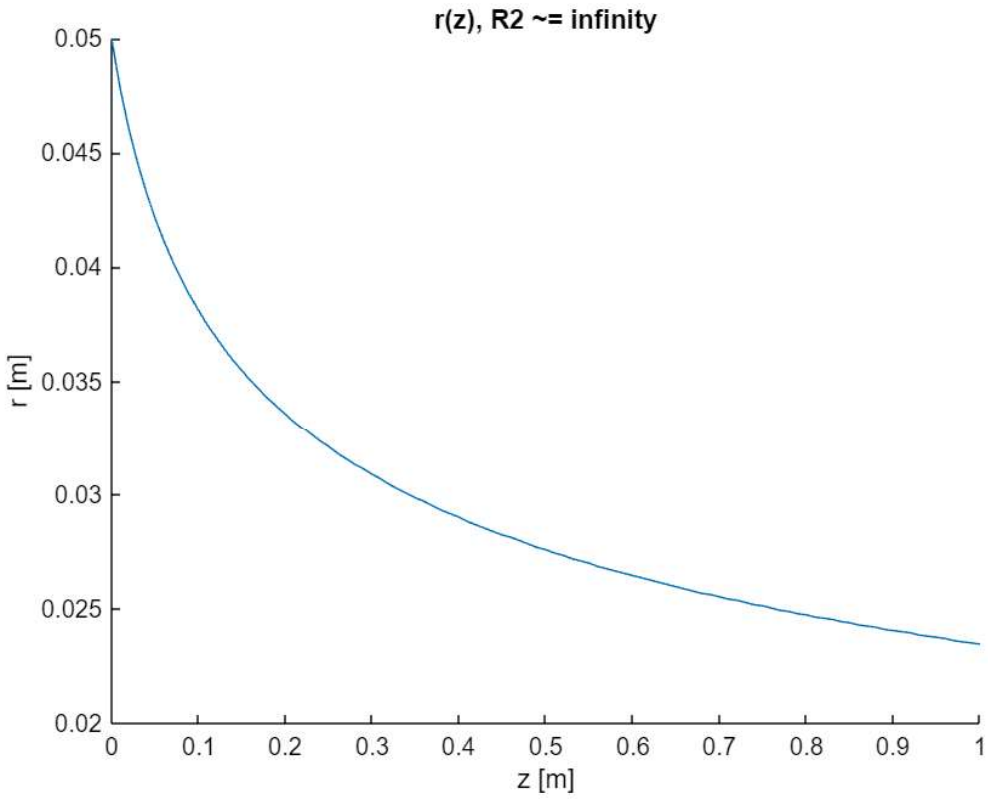
```
g = 9.81;
sigma = 72E-3;
rho = 997;
u0 = 1;
zspan = [0 1];
a = 5/100;
A = -2*a^4*u0^2;
B = -sigma/rho;

drdz = @(z,r) (g) ./ ( (A*r.^-5) + (B*r.^-2) );

z0 = 0;
r0 = a;
h = 0.01;
zf = 1;

n = (zf - z0) / h;

z = zeros(n+1, 1);
r = zeros(n+1, 1);
z(1) = z0;
r(1) = r0;

for i = 1:n
    r(i+1) = r(i) + h * drdz(z(i), r(i));
    z(i+1) = z(i) + h;
end

r_1 = r;
z_1 = z;

figure()
title("r(z), R2 != infinity")
hold on;
plot(z, r)
xlabel("z [m]")
ylabel("r [m")
```



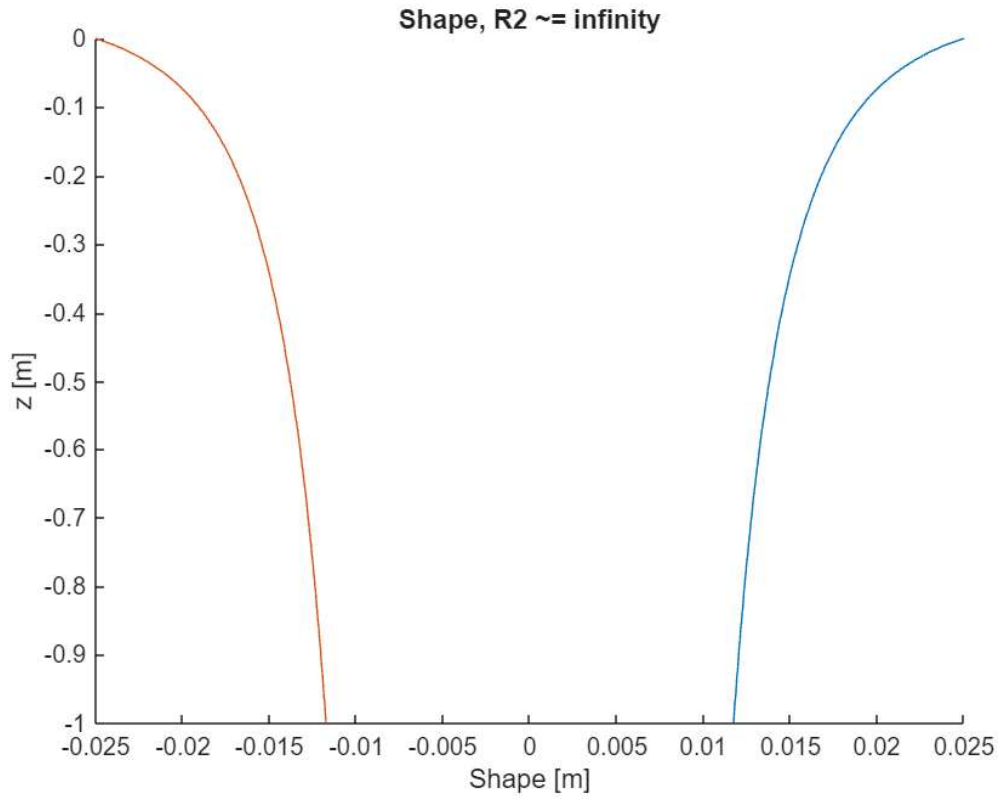

```
figure()
title("Shape, R2 != infinity")
hold on;
plot(-r/2, -z)
plot(+r/2, -z)
ylabel("z [m]")
xlabel("Shape [m]")
hold off;
```



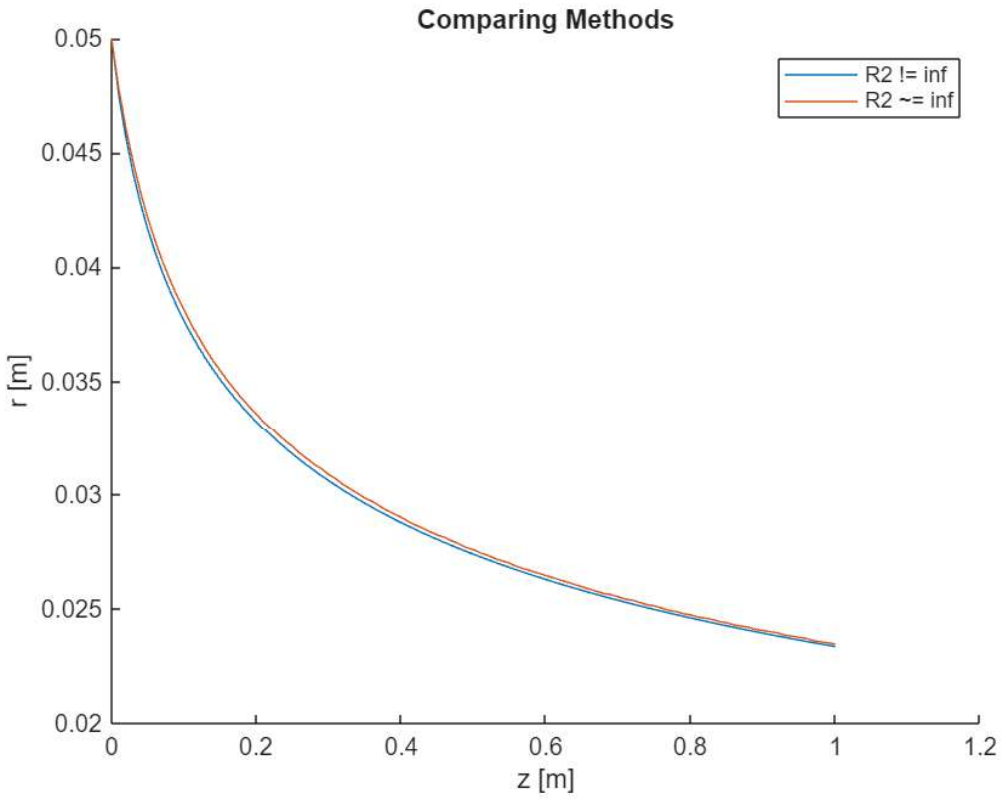

```
niter = length(z_1);
zs = linspace(0,1,niter);
rs = zeros(1,niter);
for i=1:niter
    rs(i) = r_calc(zs(i));
end
r_2 = rs;
z_2 = zs;

figure()
hold on;
title("r(z), R2 ~= infinity")
plot(zs, rs)
xlabel("z [m]")
ylabel("r [m]")
hold off;
```



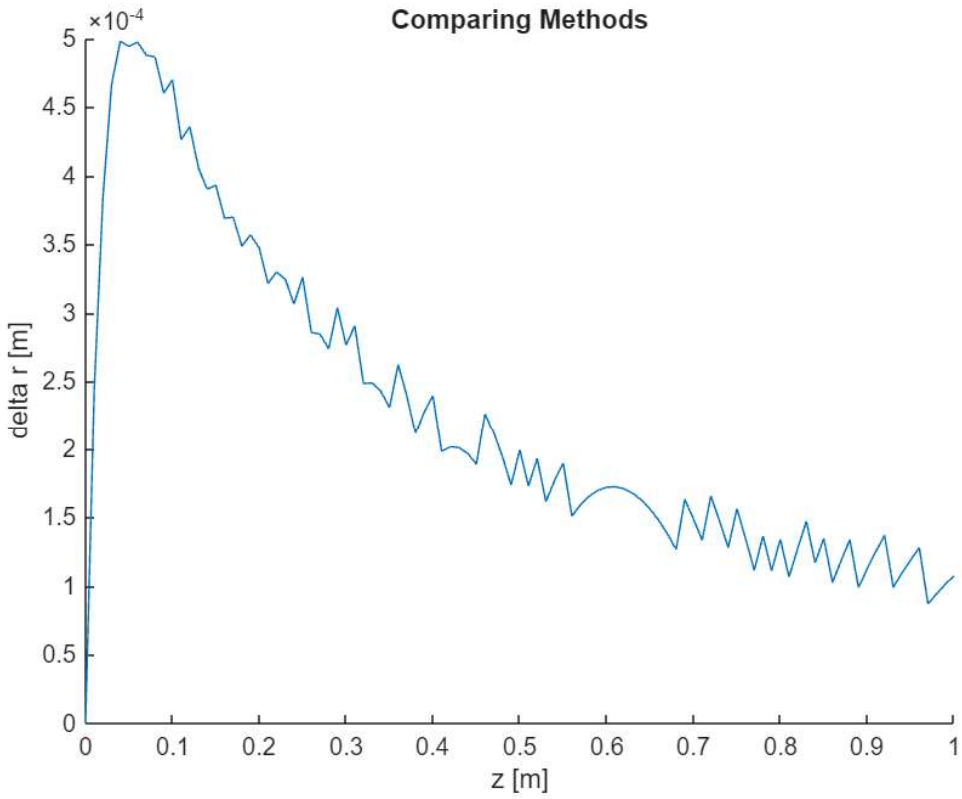

```
figure()
title("Shape, R2 ~= infinity")
hold on;
plot(rs./2, -zs)
plot(-rs./2, -zs)
ylabel("z [m]")
xlabel("Shape [m]")
hold off;
```




```
figure()
hold on;
title("Comparing Methods")
xlabel("z [m]")
ylabel("r [m]")
plot(z_1, r_1, DisplayName="R2 != inf")
plot(z_2, r_2, DisplayName="R2 ~= inf")
legend()
hold off;
```




```
figure()
hold on;
title("Comparing Methods")
xlabel("z [m]")
ylabel("delta r [m]")
plot(z_2, abs(r_2'-r_1))
hold off;
```




```
function r_z=r_calc(z)
    niter = 1000;
    g = 9.81;
    sigma = 72E-3;
    rho = 997;
    u0 = 1;
    a = 0.05;
    Fr = u0^2/(g*a);
    We = rho*u0^2*a/sigma;

    rs = linspace(a, 1/1000, niter);
    eps = 10;
    for i = 1:niter
        r = rs(i);
        LHS = r/a;
        RHS = ( 1 + 2/Fr*z/a + 2/We*(1-a/r) )^(-1/4);
        diff = abs(RHS-LHS);
        if diff < eps
            eps = diff;
            r_final = r;
        end
    end
    r_z = r_final;
end
```