Exercise 1: Womersley velocity profiles in straight pipes for pulsating flow

Main topics: Womersley velocity profiles. Pulsating flow.

Velocity profiles for pulsating flow in straight pipes can be derived from the momentum equation for fully developed flow by using Bessel functions. The momentum equation for fully developed flow on dimensionless form states (see lecture notes for details):

$$\alpha^2 \frac{\partial v}{\partial t} = -\frac{\partial p}{\partial z} + \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial v}{\partial r}\right) \tag{1}$$

We wish to look at velocity profiles in the frequency and time domain. The momentum equation above is linear in v so superposition of harmonics is OK. We introduce the following expressions, representing the pulsatile behaviour of the flow:

$$\begin{aligned} \text{Driving force:} \quad & \frac{\partial p}{\partial z} = \frac{\partial \hat{p}}{\partial z} e^{i\omega t} \\ \text{Velocity:} \quad & v = \hat{v} e^{i\omega t} \end{aligned}$$

By substitution into the momentum equation, as represented in Equation (1), we obtain:

$$i\omega\alpha^2 \hat{v}(r) = -\frac{\partial \hat{p}}{\partial z} + \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial \hat{v}}{\partial r}\right) \tag{2}$$

Furthermore, the Bessel equation for $n = 0$ is obtained if we substitute $s = i^{2/3}\alpha r$ into Equation (2). This is then rearranged as follows:

$$\frac{\partial^2 \hat{v}}{\partial s^2} + \frac{1}{s}\frac{\partial \hat{v}}{\partial s} + \left(1 - \frac{n^2}{s^2}\right) v = \frac{i}{\rho\omega}\frac{\partial \hat{p}}{\partial z} \tag{3}$$

The solution can be expressed with Bessel functions $J_0(r)$:

$$\hat{v}(r) = \frac{i}{\rho\omega}\frac{\partial \hat{p}}{\partial z}\left(1 - \frac{J_0(i^{3/2}\alpha r/a)}{J_0(i^{3/2}\alpha)}\right) \tag{4}$$

From this we find the Womersley profiles for straight pipe flow by reintroducing $v = \hat{v}e^{i\omega t}$:

$$v(r,t) = Re\left[\frac{i}{\rho\omega}\frac{\partial \hat{p}}{\partial z} e^{i\omega t}\left(1 - \frac{J_0(i^{3/2}\alpha r/a)}{J_0(i^{3/2}\alpha)}\right)\right] \tag{5}$$

a) Calculate Womersley velocity profiles from Equation (5) by using the built-in `Matlab` function `besselj(nu,Z)`.

b) Plot Womersley velocity profiles for different time levels and Womersley numbers, to illustrate the pulsatile behaviour of the flow and the influence of the Womersley number. A "skeleton" MATLAB script `Womersley.m` is provided in this exercise and can be used as a starting point.

```
%--------------------- VARIABLES/CONSTANTS -------------------------%
a=0.5e-2;   mu=1.05e-3; T=0.5;  f=1/T;  w=2*pi*f;   rho=1e3;
Ns = 100;                       %-Radial resolution
Nt = 50;                        %-Number of time levels
Np = 10;
Tmin=0.0;   Tmax=Np*T;

r = linspace(0,a,Ns);           %-Make radial vector r
t = linspace(Tmin,Tmax,Nt);     %-Make time vector t

alpha = sqrt(rho*w/mu)*a;       %-Womersley number
p0=1.0;
p=p0*sin(w*t).^2%               %-Make a time varying complex pressure
                                % vector dp/dz(w,t), use a constant
                                % amplitude p0=1.0


%------ CALCULATE WOMERSLEY PROFILES WITH BESSEL FUNCTIONS ----------%
for i=1:Nt
%v(i,:)=...%                    %-Use the built in matlab function
                                % besselj(nu,Z) to calculate the Bessel
                                % functions needed to find an expression
                                % for the velocity vector v(i,:) as a
                                % function of the pressure function p(i).
end


%-------------------- PLOT WOMERSLEY PROFILES -----------------------%
p    = -real(p)/p0;
vmax = max(max(real(v)));
v    = real(v)/vmax;

r = [-r(Ns:-1:1) r];            %-Make r a vector from -a to a
v = [v(:,Ns:-1:1) v(:,:)];      %-Mirror v about r=0

onesArray = ones(size(r));
timeLevel = 1;

subplot(1,2,1);h(1)=plot(r/a,onesArray*p(timeLevel),'r');set(gca,'ylim',[-1 1]);
subplot(1,2,2);h(2)=plot(r/a,v(timeLevel,:));set(gca,'ylim',[-1 1]);
set(h(:),'linewidth',3);
```

Solution. See figures showing velocity profiles at different times

Movie 1: Animation of velocity profiles for $\alpha = 2$. `velocity_0.mp4`

Movie 2: Animation of velocity profiles for $\alpha = 5$. `velocity_1.mp4`

Movie 3: Animation of velocity profiles for $\alpha = 10$. `velocity_2.mp4`

Movie 4: Animation of velocity profiles for $\alpha = 20$. `velocity_3.mp4`

```
## Import
import numpy as np
```

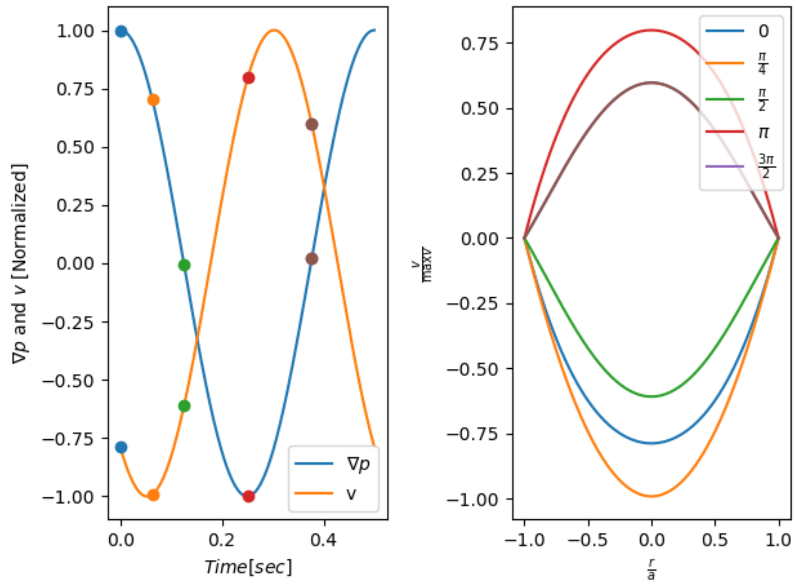

Figure 1: Velocity profiles for $\alpha = 2$.

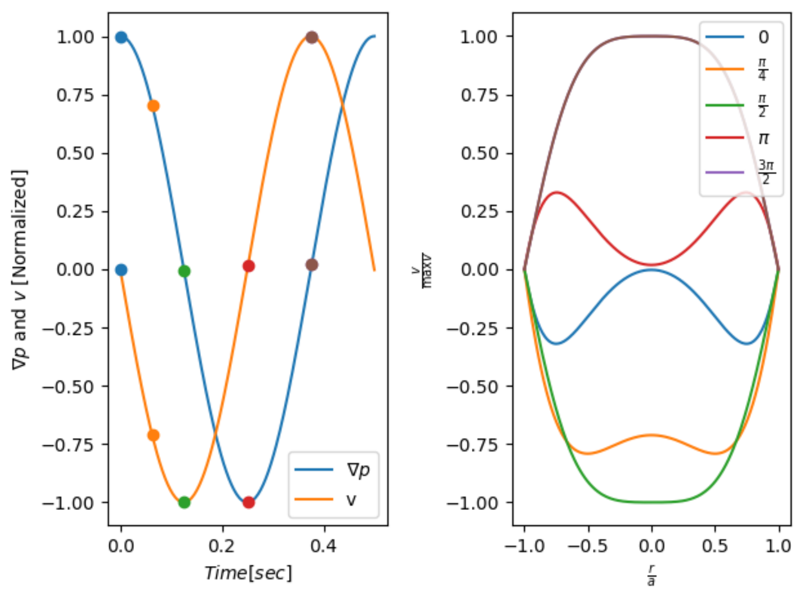

Figure 2: Velocity profiles for $\alpha = 5$.

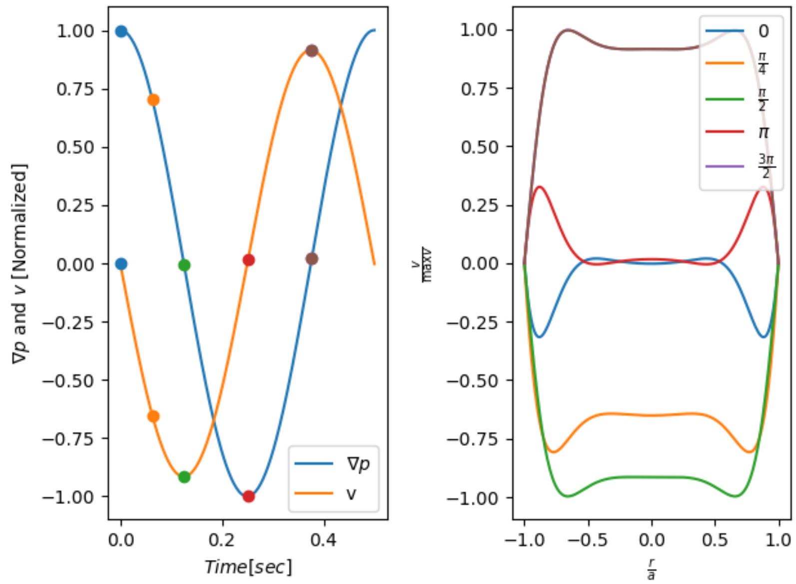

Figure 3: Velocity profiles for $\alpha = 10$.

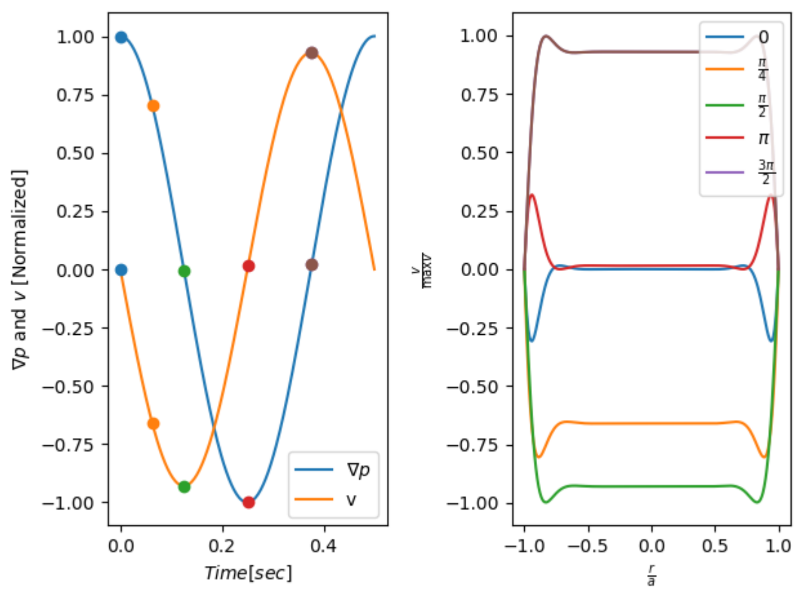

Figure 4: Velocity profiles for $\alpha = 20$.

```
import scipy as sp
import scipy.special
import matplotlib.pyplot as plt
import matplotlib.animation as anim
##-------------------- VARIABLES/CONSTANTS ------------------------#
a=0.5e-2
mu=1.05e-3
T=0.5
f=1/T
w=2*np.pi*f
rho=1e3

Ns = 100                      #-Radial resolution
Nt = 200                      #-Number of time levels
Np = 1                        #-Number of periods
Tmin=0.0
Tmax=Np*T
r = np.linspace(0,a,Ns)          #-Make radial vector r
t = np.linspace(Tmin,Tmax,Nt)    #-Make time vector t
alpha = np.sqrt(rho*w/mu)*a      #-Womersley number
p0=1.0
dp = p0*np.exp(1j*w*t)           #-Make a time varying complex pressure
                                 # vector dp/dz(w,t), use a constant
                                 # amplitude p0=1.0

##
#------ CALCULATE WOMERSLEY PROFILES WITH BESSEL FUNCTIONS ----------#
v2 = np.empty((Nt,Ns))
tau2 = np.empty((Nt,))
v5 = np.empty((Nt,Ns))
tau5 = np.empty((Nt,))
v10 = np.empty((Nt,Ns))
tau10 = np.empty((Nt,))
v20 = np.empty((Nt,Ns))
tau20 = np.empty((Nt,))
for i in range(Nt):
#v(i,:)=...#                     #-Use the built in scipy function
                                 # sp.special.jn(nu,Z) to calculate the Bessel
                                 # functions needed to find an expression
                                 # for the velocity vector v(i,:) as a
                                 # function of the pressure function p(i).
    alpha=2
    v2[i,:] = (1j/w/rho)*dp[ i ]*(1-sp.special.jn(0,1j**(1.5)*alpha*r/a)/
            sp.special.jn(0,1j**(1.5)*alpha))
    tau2[i] = -mu*1j**(5/2)*alpha*dp[i]*sp.special.jn(1,1j**(1.5)*alpha)/
            sp.special.jn(0,1j**(1.5)*alpha)

    alpha=5
    v5[i,:] = (1j/w/rho)*dp[ i ]*(1-sp.special.jn(0,1j**(1.5)*alpha*r/a)/
            sp.special.jn(0,1j**(1.5)*alpha))
    tau5[i] = -mu*1j**(5/2)*alpha*dp[i]*sp.special.jn(1,1j**(1.5)*alpha)/
            sp.special.jn(0,1j**(1.5)*alpha)

    alpha=10
    v10[i,:] = (1j/w/rho)*dp[ i ]*(1-sp.special.jn(0,1j**(1.5)*alpha*r/a)/
            sp.special.jn(0,1j**(1.5)*alpha))
    tau10[i] = -mu*1j**(5/2)*alpha*dp[i]*sp.special.jn(1,1j**(1.5)*alpha)/
            sp.special.jn(0,1j**(1.5)*alpha)

    alpha=20
    v20[i,:] = (1j/w/rho)*dp[ i ]*(1-sp.special.jn(0,1j**(1.5)*alpha*r/a)/
            sp.special.jn(0,1j**(1.5)*alpha))
    tau20[i] = -mu*1j**(5/2)*alpha*dp[i]*sp.special.jn(1,1j**(1.5)*alpha)/
    sp.special.jn(0,1j**(1.5)*alpha)
```

```
#-------------------- PLOT WOMERSLEY PROFILES -----------------------#
p    = np.real(dp)/p0
vmax = np.max(np.real(v2))
v2    = np.real(v2)/vmax
vmax = np.max(np.real(v5))
v5    = np.real(v5)/vmax
vmax = np.max(np.real(v10))
v10    = np.real(v10)/vmax
vmax = np.max(np.real(v20))
v20    = np.real(v20)/vmax
r = np.concatenate((-r[Ns::-1], r))           #-Make r a vector from -a to a
v2 = np.concatenate((v2[:,Ns::-1],v2),axis=1)
v5 = np.concatenate((v5[:,Ns::-1],v5),axis=1)
v10 = np.concatenate((v10[:,Ns::-1],v10),axis=1)
v20 = np.concatenate((v20[:,Ns::-1],v20),axis=1)

# Plot single time point
plt.figure()
plt.plot(r/a, v2[5,:])
plt.xlabel(r"$\frac{r}{a}$")
plt.ylabel(r"$\frac{v}{\max v}$")
plt.show()


# Phase and time plots
v_plot = v2
for alpha_idx, v_plot in enumerate((v2, v5, v10, v20)):
# Choose 0, pi/4, pi/2, pi, and 3pi/2
    fig1, ax = plt.subplots(1,2)
    v_center = v_plot[:,Ns]
    ax[0].plot(t,p,label=r"$\nabla p$")
    ax[0].plot(t,v_center,label="v")
    ax[0].legend()
    ax[0].set_xlabel(r'$Time[sec]$')
    ax[0].set_ylabel(r'$\nabla p$ and $v$ [Normalized]')

    indices = [0, int(np.floor((Nt/Np)/8)), int(np.floor((Nt/Np)/4)),
        int(np.floor((Nt/Np)/2)), int(np.floor(3*(Nt/Np)/4)),
        int(np.floor(3*(Nt/Np)/4))]

    for idx in indices:
      pp, = ax[0].plot(t[idx],p[idx], 'o')
      vp, = ax[0].plot(t[idx],v_center[idx], 'o')
      l, = ax[1].plot(r/a, v_plot[idx,:])
      pp.set_color(l.get_color())
      vp.set_color(l.get_color())

    ax[1].legend((r'$0$', r'$\frac{\pi}{4}$', r'$\frac{\pi}{2}$', r'$\pi$',
        r'$\frac{3\pi}{2}$'), loc=1)
    ax[1].set_xlabel(r'$\frac{r}{a}$')
    ax[1].set_ylabel(r'$\frac{v}{\max v}$')
    plt.tight_layout()
    plt.savefig("velocity_%d.png"%alpha_idx)
    plt.show()
```

c) Describe the phase of the velocity and the pressure gradient for different Womersley numbers. How does this compare for large (20) and small (2) Womersley numbers?

Solution. We see from the plots that for small Womersley number the flow and pressure are 180 degrees out of phase (or forward flow is in phase with decreasing pressure gradient) while at high Wommersley numbers inertial effects cause a the phase difference to tend towards 90 degrees as the flow near the wall remains in phase with the pressure gradient, but the core flow becomes out of phase.

d) Based only on the velocity profiles you have plotted how does the wall shear stress, $\tau_w = -\mu \frac{\partial v}{\partial r}|_a$, change with increasing Womersley number?

Solution. We see that with increasing Womersley number the slope of the velocity profile becomes steeper at the wall, thus the wall shear stress is increased.

e) Using the fact that $\frac{\partial J_0(s)}{\partial s} = -J_1(s)$ derive an expression for τ_w as a function of the Womersley number α. Plot τ_w vs α for a fixed radius a.

Solution. We begin with (5) and note that the only dependence on r is in $\frac{J_0(i^{3/2}\alpha r/a)}{J_0(i^{3/2}\alpha)}$ thus

$$\frac{\partial v}{\partial r} = Re\left[\frac{i}{\rho\omega}\frac{\partial \hat{p}}{\partial z}e^{i\omega t}\frac{\partial J_0(i^{3/2}\alpha r/a)}{\partial r}\frac{1}{J_0(i^{3/2}\alpha)}\right] \tag{6}$$

Now

$$\frac{\partial J_0(i^{3/2}\alpha r/a)}{\partial r} = -J_1(i^{3/2}\alpha r/a)\frac{i^{3/2}\alpha}{a} \tag{7}$$

thus we have

$$\frac{\partial v}{\partial r} = Re\left[\frac{i^{5/2}\alpha}{a\rho\omega}\frac{\partial \hat{p}}{\partial z}e^{i\omega t}\frac{J_1(i^{3/2}\alpha r/a)}{J_0(i^{3/2}\alpha)}\right] \tag{8}$$

So

$$\tau_w = -\mu Re\left[\frac{i^{5/2}\alpha}{a\rho\omega}\frac{\partial \hat{p}}{\partial z}e^{i\omega t}\frac{J_1(i^{3/2}\alpha)}{J_0(i^{3/2}\alpha)}\right] \tag{9}$$

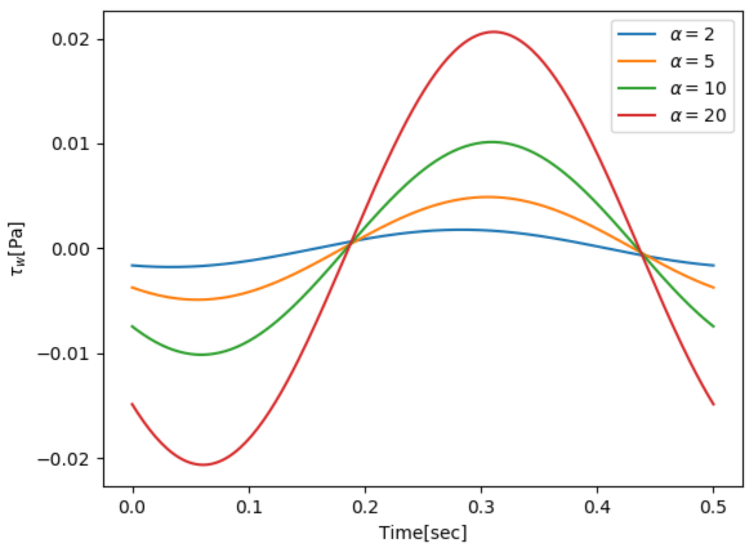

Figure 5: Plot of wall shear stress over time for various Womersley numbers.