EAE 298 Aeroacoustics
Fall Quarter 2016
Homework #4

John Karasinski

December 5, 2016

You are designing a new aircraft engine and analyzing acoustic propagation generated by a non-uniform flow with angle of attack interacting with rotating fans. You obtained flow fields from CFD for a one-sixth small scale of the engine. The radius for the small scale engine is 13 in and the hub radius is 3 in. CFD provides the velocity gust information as a function of its circumferential modes. The circumferential mode for acoustics can be expressed as $m = nB - kV$ where B is the number of blades, V is the number of vanes, n stands for the harmonic of BPF and k is the integer (1, 2, 3...). You consider only positive k at this time (this is related the rotation direction of gust). The number of blades is 18. The number of vanes is considered to be 1 since there is no physical vanes but there is one revolution difference. The Mach number is 0.525 and the fan RPM is 8326.3042, the speed of sound is 13503.937009 in/s and the density is 1.4988E-5 slug/in^3. The dominant noise is generated at the 1st BPF or n=1 in which the angular frequency is given as $RPM \times \frac{2\pi}{60} \times B$. We are interested in the propagation through the inlet of the engine so that sound propagates to -z direction assuming the +z direction is in the flow direction.

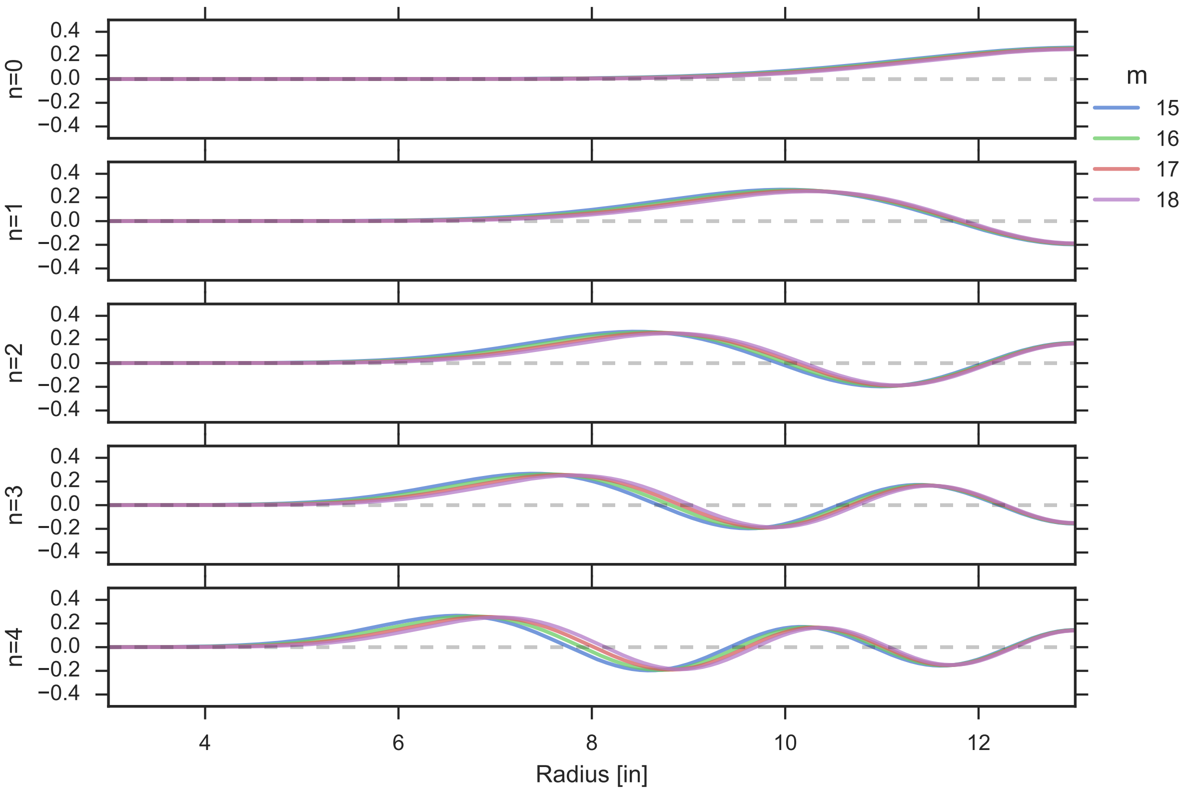
Problem 1. [20 points]

Determine the first five eigenvalues of acoustics for m=18, 17, 16, 15 or (k=1, 2, 3, 4) or (m,n) = (18,0), (18,1), (18,2), (18,3), (18,4), (17,0), (17,1), (17,2), (17,3), (17,4), (16,0), (16,1), (16,2), (16,3), (16,4), (15,0), (15,1), (15,2), (15,3), (15,4)

				m
n	15	16	17	18
0	1.3093	1.3895	1.4696	1.5495
1	1.7032	1.7896	1.8755	1.9612
2	2.0137	2.1036	2.1932	2.2823
3	2.3005	2.3932	2.4855	2.5772
4	2.5753	2.6702	2.7646	2.8585

Problem 2. [20 points]

Plot the five eigenfunctions (radial modes, n=0, 1, 2, 3, 4) for m=18, 17, 16, 15 or (k=1, 2, 3, 4) and verify n describes the number of zero crossings in the radial direction



Problem 3. [20 points]

Determine the wavenumbers in the z direction for (m,n)=(18,0), (18,1), (18,2), (17,0), (17,1), (17,2), (16,0), (16,1), (16,2), (15,0), (15,1), (15,2). Indicate whether the mode is cut-on (propagating) or cut-off (exponentially decaying). Consider only the propagation in the -z direction. Exclude the exponentially growing solution and include only the propagating solutions or exponentially decaying solutions.

m	n	μ	K_z	Cut
15	0	1.3093	-0.386 + 0.000 i	On
15	1	1.7032	-0.842 + 1.195 i	Off
15	2	2.0137	-0.842 + 1.738 i	Off
15	3	2.3005	-0.842 + 2.175 i	Off
15	4	2.5753	-0.842 + 2.565 i	Off
16	0	1.3895	-0.842 + 0.301 i	Off
16	1	1.7896	-0.842 + 1.359 i	Off
16	2	2.1036	-0.842 + 1.880 i	Off
16	3	2.3932	-0.842 + 2.309 i	Off
16	4	2.6702	-0.842 + 2.696 i	Off
17	0	1.4696	-0.842 + 0.638 i	Off
17	1	1.8755	-0.842 + 1.510 i	Off
17	2	2.1932	-0.842 + 2.016 i	Off
17	3	2.4855	-0.842 + 2.440 i	Off
17	4	2.7646	-0.842 + 2.824 i	Off
18	0	1.5495	-0.842 + 0.860 i	Off
18	1	1.9612	-0.842 + 1.653 i	Off
18	2	2.2823	-0.842 + 2.148 i	Off
18	3	2.5772	-0.842 + 2.568 i	Off
18	4	2.8585	-0.842 + 2.950 i	Off

Problem 4. [30 points]

The pressure distribution file at z=0 plane for m=18 or 1 BPF is provided. The first column is the dimensional radius [in], the second column the real part of the pressure [psi], and the third column is the imaginary part of the pressure [psi]. Using this boundary condition, compute the sound power level for (m,n)=(18,0), (18,1), (18,2). This noise is considered for blade self noise that is not associated with the gust response since k=0. Note that the z=0 plane is not the same as the engine inlet. Use the conversion for the unit for the sound power as follows: PWL (dB)=10*log10((Wmn))-10*log10(7.3756E-13)

n	PWL
0	-10.4015
1	-13.3775
2	-22.3880

```
from scipy.special import jv, yv
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import scipy
import seaborn as sns
sns.set(style="ticks", palette="muted", color_codes=True)


##############################################################################
# Problem 1 ##################################################################
##############################################################################


def Jm_(m, val):
    return 0.5 * (jv(m - 1, val) - jv(m + 1, val))


def Ym_(m, val):
    return 0.5 * (yv(m - 1, val) - yv(m + 1, val))


def func(mu, m, Ri, Ro):
    '''Equation taken from "Turbomachinery Noise" notes, page 10.'''
    return Jm_(m, mu * Ro) * Ym_(m, mu * Ri) - Jm_(m, mu * Ri) * Ym_(m, mu * Ro)


Ri, Ro = 3, 13  # inches

eigenvalues = []
for m in np.arange(15, 19):
    for initial_guess in np.linspace(0.1, 3, 300):
        try:
            mu = scipy.optimize.broyden1(lambda guess: func(guess, m, Ri, Ro), initial_guess)
            eigenvalues.append([m, mu])
        except Exception:
            pass
eigenvalues = pd.DataFrame(eigenvalues)
eigenvalues.columns = ['m', 'mu']

eigenvalues.mu = eigenvalues.mu.astype(float).round(8)
eigenvalues.drop_duplicates(inplace=True)
eigenvalues.sort(['m', 'mu'], inplace=True)

eigenvalues = eigenvalues.groupby('m').head()
eigenvalues['n'] = 4 * list(np.arange(0, 5))
print(eigenvalues.pivot('n', 'm'))

##############################################################################
# Problem 2 ##################################################################
##############################################################################


```

```
def eigenfunction(m, mu, r, Ri):
    '''Equation taken from "Turbomachinery Noise" notes, page 10.'''
    return jv(m, mu * r) - (Jm_(m, mu * Ri) / Ym_(m, mu * Ri)) * yv(m, mu * r)

f, ax = plt.subplots(nrows=5, sharex=True, sharey=True, squeeze=True)
r = np.linspace(Ri, Ro, 1000)
for n, group in eigenvalues.groupby('n'):
    for i, g in group.reset_index(drop=True).iterrows():
        ax[n].plot(r, eigenfunction(g.m, g.mu, r, Ri),
                   color=sns.color_palette()[i], alpha=0.75)
        ax[n].set_ylabel("n={}".format(n))

for ax_ in ax:
    ax_.plot([Ri, Ro], [0, 0], '--', alpha=0.25, color='k')

legend = ax[0].legend([15, 16, 17, 18], title='m', loc='center left', shadow=True, bbox_to_anchor=(1, 0
legend.get_frame().set_facecolor('#333333')
plt.xlim(Ri, Ro)
plt.ylim(-.5, .5)
plt.xlabel('Radius [in]')
plt.savefig('tex/figs/problem2.pdf')
plt.close()

###############################################################################
# Problem 3 ###################################################################
###############################################################################


M = 0.525
RPM = 8326.3042  # rotations per minute
B = 18
w = RPM * (2 * np.pi / 60) * B  # angular velocity
c = 13503.937009  # in/s
K = w / c

res = []
for i, (m, mu, n) in eigenvalues.iterrows():
    Kz = K * (- M + np.emath.sqrt(1 - (1 - M**2) * (mu / K)**2)) / (1 - M**2)
    Kzr = float(np.real(Kz))
    Kzi = float(np.imag(Kz))
    res.append([m, n, mu, Kz, Kzr, Kzi])
res = pd.DataFrame(res, columns=['m', 'n', 'mu', 'Kz', 'Kzr', 'Kzi'])

###############################################################################
# Problem 4 ###################################################################
###############################################################################


rho = 1.4988E-5  # slug/in^3
p = pd.DataFrame.from_csv('pressure_input.dat', sep='\t', header=None, index_col=None)
p.columns = ['r', 'Pr', 'Pi']
p['P'] = p.Pr + p.Pi * 1j

PWLs = []
```

```
m = 18
for _, (n, mu, Kz) in res.query('m == @m and n < 3')[['n', 'mu', 'Kz']].iterrows():
    mu = float(mu)
    gamma = (+ 0.5 * (Ro ** 2 - m ** 2 / mu ** 2) * eigenfunction(m, mu, Ro, Ri)**2
             - 0.5 * (Ri ** 2 - m ** 2 / mu ** 2) * eigenfunction(m, mu, Ri, Ri)**2)

    p['psi'] = eigenfunction(m, mu, p.r, Ri)
    A = (1 / gamma) * np.trapz(y=p.P * p.psi * p.r, x=p.r)
    W1 = (np.pi / (rho * c)) * gamma * A * np.conjugate(A)
    W2 = ((1 + M ** 2) * np.real(Kz / (w / c - Kz * M))
          + M * (1 + abs(Kz / (w / c - Kz * M))**2))
    Wmn = W1 * W2
    PWL = 10 * np.log10(abs(Wmn)) - 10 * np.log10(7.3756E-13)
    # print(int(n), PWL)
    PWLs.append([int(n), PWL, Wmn, Kz, mu, gamma, A])
PWLs = pd.DataFrame(PWLs)
PWLs.columns = ['n', 'PWL', 'Wmn', 'Kz', 'mu', 'gamma', 'A']
print(PWLs[['n', 'PWL']])
```