# Amath481 HW2 presentation skills

HAOLOU SUN

October 2022

# 1 Problem 1

## 1.1 Equations

$$\Psi_2(x,t) = \phi_2(x)\cos(\frac{E_2}{2h}t) \tag{1}$$

where $\phi_2$(x) is the second eigenfunction, $E_2$ is the second eigenvalue. Then we set x from -4 to 4 and t from 0 to 5. We get the following plot from a proper angel of view.

## 1.2 Graph

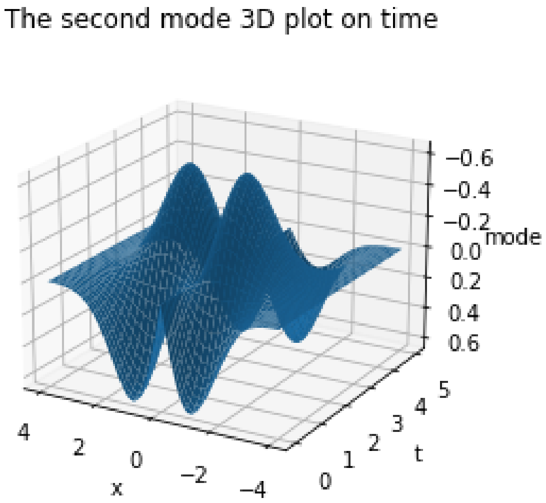

Figure 1: This is the second mode 3D plot view from the angel (-160, 60).

### 1.3 Code

```
import numpy as np
import matplotlib.pyplot as plt
import scipy.integrate
from mpl_toolkits import mplot3d

def rhsfunc(x, y, epsilon):
    f1 = y[1]
    f2 = (np.square(x) - epsilon)*y[0]
    return np.array([f1, f2])

# Define some constants
xp = [-4, 4]
tol = 1e-6

# Define our initial conditions
A = 1

epsilon_start = 0 # This is our initial beta value, we will change it.

eigenfun = np.empty([81, 5])
eigenval = np.empty([1, 5])
phi = np.empty([81,5])

for modes in range(5): # Try to find 5 modes
    epsilon = epsilon_start
    depsilon = 0.05 # This is the amount we will increase epsilon by each time we don’t have
                 # until we get an eigenvalue

    for j in range(1000):
        y0 = np.array([A, np.sqrt((-4)**2 - epsilon)*A])
        x_evals = np.linspace(-4, 4, 20*4+1)
        sol = scipy.integrate.solve_ivp(lambda x,y: rhsfunc(x, y, epsilon), xp, y0, method =
        y_sol = sol.y[0, :]
        yprime = sol.y[1, :]
        error = yprime[-1] + np.sqrt(4**2 - epsilon) * y_sol[-1]

        if np.abs(error)<tol:
            area = np.sqrt(scipy.integrate.trapz(sol.y[0, :]**2, x = x_evals))
            norm = scipy.integrate.trapz((sol.y[0, :]/area)**2, x = x_evals)
            eigenfun[:, modes] = np.abs(sol.y[0, :]/area).T
            eigenval[:, modes] = epsilon
            phi[:, modes] = np.abs(sol.y[0, :]/area).T
            break
        if (-1)**(modes)*error > 0:
            epsilon = epsilon + depsilon
        else:
            epsilon = epsilon - depsilon/2
            depsilon = depsilon/2

    epsilon_start = epsilon + 0.1

A6 = np.reshape(eigenval, (1, 5))
x = np.linspace(-4, 4, 20*4+1)
t = np.linspace(0, 5, 100)
phi_2 = np.reshape(phi[:, 1], (81, 1))
cos = np.reshape(np.cos(A6[0, 1]*t/2), (1,100))
result = np.matmul(phi_2, cos)
fig = plt.figure()
ax = plt.axes(projection = ’3d’)
X, T = np.meshgrid(x, t)
ax.plot_surface(X, T, result.T.real)
ax.set_xlabel(’x’)
ax.set_ylabel(’t’)
ax.set_zlabel(’mode’)
ax.set_title(’The second mode 3D plot on time’)
ax.view_init(-160, 60)
fig.savefig(’3d-160.png’)
plt.show()
```