```python
import matplotlib.pyplot as plt
import scipy.integrate as scii
from scipy.sparse.linalg import eigs
from joblib import Parallel, delayed
import multiprocessing
from functools import partial
import numpy as np


def rhs(f, t):
    return np.array([f[1], f[2], -0.5*f[0]*f[2]])

#blasius profile
def getMesh_U_DuDy(N, y_max):
    initial_data = np.array([0, 0, 1])
    x = np.linspace(0, y_max, N+1)
    y = scii.odeint(rhs, initial_data, x)

    k = y[N][1]
    alpha = 1 / k**(3/2)
    a = alpha**(1/3)
    x = x / a
    y[:, 0] = a * y[:, 0]
    y[:, 1] = a**2 * y[:, 1]
    y[:, 2] = a**3 * y[:, 2]
    return x, y[:, 1], y[:,2]

def boundary_layer_thickness(nu, u_e, x):
    return (nu*x/u_e)**0.5

def getRe_d(nu, u_e, d):
    return u_e*d/nu

def getDimensionalMesh_U_DuDy(x, u_e, nu, y_d, u_d, dudy_d):
    d = boundary_layer_thickness(nu, u_e, x)
    y = y_d*d
    u = u_d*u_e
    dudy = dudy_d*u_e/d
```