Department of Aerospace Engineering
Indian Institute of Technology, Madras

AE21B056 Shreeya Padte

3rd May 2024

AS6320 Acoustic Instabilities in Aerospace Propulsion

Assignment 3

THERMOACOUSTIC INSTABILITY IN A RIJKE TUBE

Contents

1 Introduction **2**

2 Objective **3**

3 Motivation of present problem - Overview of the approach **3**

4 Methodology **4**

5 Results and Discussions **7**
5.1 Evolution of non dimensional acoustic velocity and non-linear evolution of acoustic energy. 7
5.2 Evolution of acoustic velocity projected onto various Galerkin modes. 9
5.3 Bifurcation plot for variation of non-dimensional heater power K. 11

6 Summary **12**

7 Appendix **13**
7.1 Codes to get the plots 13

List of Figures

1 Thermoacoustic instability 2
2 Schematic of horizontal Rijke Tube 4
3 u' with time, $\zeta = 0$, xf = 0.29, K = 0.1 8
4 u' with time, $\zeta = 0$, xf = 0.29, K=1.25 8
5 Energy with time, K = 1.4, $\tau = 0.2$ 9
6 K = 0.2, $\eta_1 = 0.18$ 9
7 K = 1.8, $\eta_1 = 0.18$ 10
8 Parameter values of the system are c1 = 0.1, c2 = 0.06, xf = 0.3 and $\tau = 0.2$. 11
9 3D bifurcation plot of non-dimensional heater power for varying values of time lag 12

1 Introduction

Combustion instabilities are self-sustained large amplitude oscillations of pressure and velocity in combustors with the flame acting as an acoustic actuator and the combustion chamber as an acoustic resonator. The occurence of combustion instability depends on the phase between the heat release fluctuation and the pressure fluctuation at the flame as given by the Rayleigh criteria. If the maximum and minimum of the heat addition occur during the rarefaction and compression phases of pressure oscillation then amplification will take place. [**1**]

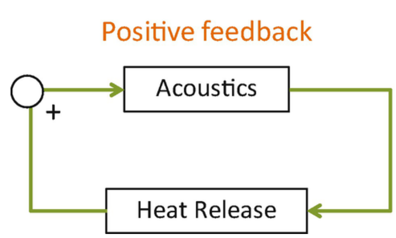

Figure 1: Thermoacoustic instability

Thermoacoustic instability is caused by dynamic coupling between unsteady heat release and acoustic perturbations. It is characterized with self-sustained large-amplitude limit cycle oscillations.

In this type of instabilities the perturbations that grow and alter the features of the flow are of an acoustics nature. Their pressure oscillations can have well defined frequencies with amplitudes high enough to pose a serious hazard to combustion systems. Instabilities are known to destroy gas-turbine-engine components during testing. They represent a hazard to any type of combustion system.

Thermoacoustic combustion instability occurs when a horizontal Rijke tube, with a flat flame, experiences acoustic waves that create a standing wave pattern. This pattern also forms in combustors, but takes a more complex form. The acoustic waves perturb the flame, which in turn affects the acoustics. This feedback between the acoustic waves in the combustor and heat-release fluctuations from the flame is a hallmark of thermoacoustic combustion instabilities.

2 Objective

- The schematic of a horizontal Rijke tube considered. The equations governing the nonlinear stability analysis of evolution of the acoustic field in this are derived. Rijke tube is an acoustic resonator tube, which consists of a heat source positioned at some axial location. A mean flow is maintained at a desired flow rate using a blower. A correlation between the heat release rate fluctuations at the heater location and the acoustic velocity fluctuations at the heater is used to model the fluctuating heat release rate from the heater in the Rijke tube. We use this model problem to understand the nonlinear dynamics in a Rijke tube.
- To explore the evolution of acoustic velocity and energy over time, and to investigate how damping affects oscillations, various concepts such as projected Galerkin modes, triggering instability, and bootstrapping are examined.
- A limit cycle refers to a periodic orbit that a system settles into, where the state variables of the system repeat in a predictable pattern over time. A Hopf bifurcation marks the transition from a stable equilibrium to a stable limit cycle or periodic orbit as the control parameter crosses a critical threshold.
- We will explore how these concepts relate to thermoacoustic systems, where acoustic waves and heat transfer interact within a Rijke tube, giving rise to complex oscillatory behavior. Understanding limit cycles and Hopf bifurcations in such systems is crucial for analyzing stability, predicting instabilities like triggering, and investigating phenomena like bootstrapping, where nonlinear interactions amplify oscillations. [**2**]

3 Motivation of present problem - Overview of the approach

- A linearly stable combustor can be "triggered" by introduction of finite amplitude disturbance. Such a system will be stable with respect to all disturbances whose amplitudes are below a certain threshold value, but transition into pulsating operation will occur when the amplitude of the disturbance exceeds this threshold value.
- There are instances of "bootstrapping" where a mode that decays initially can grow later and ultimately become unstable. A linear stability analysis using normal modes would always indicate stability in the case of bootstrapping and miss the ultimate behavior.
- The pressing need to control this phenomenon in combustion chambers compounded by lack of understanding of combustion instability due to the complex nature of combustion-acoustic interaction in flames led to interest in simpler thermoacoustic systems such as Rijke tubes. A Rijke tube is a relatively simple system: it is a duct with a heat source at quarter length. We use this model problem to understand the nonlinear dynamics in a Rijke tube. [**1**]

4 Methodology

A horizontal Rijke tube with an electric heat source is a system convenient for studying the fundamental principles of thermoacoustic instabilities. The horizontal orientation of the Rijke tube is implemented to exclude the influence of natural convention on the mean flow rate.

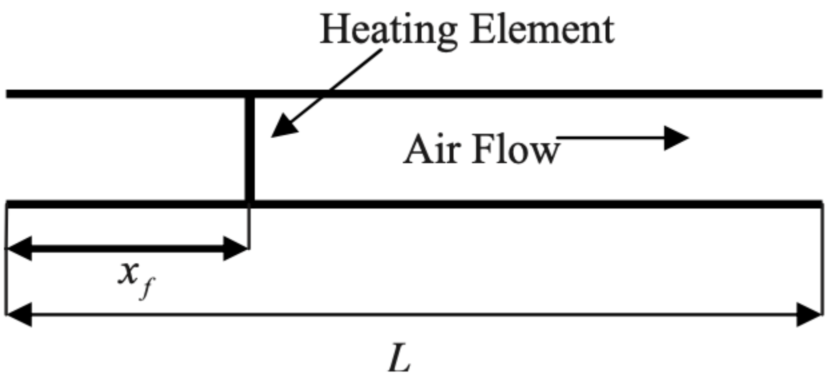

Figure 2: Schematic of horizontal Rijke Tube

We non-dimensionalize the momentum and energy equations. Presence of $\sim$ indicates dimensional quantity and absence of $\sim$ indicates non-dimensional quantities. Neglecting the effect of mean flow and mean temperature gradient in the duct, the governing equations for the one-dimensional acoustic field are:

Linearized momentum equation

$$\rho\frac{\partial\tilde{u'}}{\partial t}+\frac{\partial\tilde{p'}}{\partial\tilde{x}}=0 \tag{1}$$

u_0 is the base flow and L_a is the duct length.

$$\frac{\bar{\rho}}{L_a/c_0}u_0\frac{\partial\tilde{u'}/u_0}{\partial t/(L_a/c_0)}+\frac{\bar{p}}{L_a}\frac{\partial(\tilde{p'}/\bar{(p)})}{\partial\tilde{x}/L_a}=0 \tag{2}$$

$$\frac{\gamma\bar{p}}{\bar{\rho}}=c^2$$

$$\frac{\gamma u_0}{c_0}=\gamma M$$

$$\boxed{\gamma M\frac{\partial u'}{\partial t}+\frac{\partial p'}{\partial x}=0}$$

Linearized energy equation

$$\frac{\partial\tilde{p'}}{\partial t}+\gamma\bar{p}\frac{\partial\tilde{u'}}{\partial x}=(\gamma-1)\dot{\tilde{Q}}' \tag{3}$$

Non dimensionalising this equation gives:

$$\frac{\bar{p}}{L_a/c_0}\frac{\partial\tilde{p'}/\bar{p}}{\partial t/(L_a/c_0)}+\frac{\gamma\bar{p}u_0}{L_a}\frac{\partial\tilde{u'}/u_0}{\partial\tilde{x}/L_a}=(\gamma-1)\dot{\tilde{Q}}' \tag{4}$$

Since King's law exhibits nonlinearity only for velocity perturbations greater than the mean fluctuations, it is used to model the heat release rate:

$$\dot{\tilde{Q}}'=\frac{2L_w}{S\sqrt{3}}(T_w-\bar{T})\sqrt{\pi\lambda C_y\bar{\rho}\frac{dw}{2}}\left[\sqrt{|\frac{u_0}{3}+u_f'(t-\tau)|}-\sqrt{\frac{u_0}{3}}\right]\delta(\tilde{x}-\tilde{x_f})$$

$$\delta(\tilde{x}-\tilde{x_f}) = \delta(L_a(x-x_f)) = \frac{1}{L_a}\delta(x-x_f)$$

$$\frac{\partial p'}{\partial t} + \gamma M\frac{\partial u'}{\partial x} = \frac{\gamma-1}{\bar{p}}\frac{L_a}{c_0}\frac{2L_w}{S\sqrt{3}}(T_w-\bar{T})\sqrt{\pi\lambda C_y\bar{\rho}\frac{dw}{2}u_0}\left[\sqrt{|\frac{1}{3}+u'_f(t-\tau)|}-\sqrt{\frac{1}{3}}\right]\frac{\delta(x-x_f)}{L_a} \tag{5}$$

$$\boxed{\frac{\partial p'}{\partial t} + \gamma M\frac{\partial u'}{\partial x} = K\left[\sqrt{|\frac{1}{3}+u'_f(t-\tau)|}-\sqrt{\frac{1}{3}}\right]\delta(x-x_f)}$$

Thus we have,

$$\gamma M\frac{\partial u'}{\partial t} + \frac{\partial p'}{\partial x} = 0 \tag{6}$$

$$\frac{\partial p'}{\partial t} + \gamma M\frac{\partial u'}{\partial x} = K\left[\sqrt{|\frac{1}{3}+u'_f(t-\tau)|}-\sqrt{\frac{1}{3}}\right]\delta(x-x_f) \tag{7}$$

The Galerkin technique makes use of the fact that any function in a domain can be expressed as a superposition of expansion functions which form a complete basis in that domain. The basis functions are chosen such that they satisfy the boundary conditions. Duct is open at both sides hence p'(x=0) = 0 and p'(x=1) = 0.

$$p'(x,t) = \sum_{j=1}^{N} a_j(t)sin(j\pi x) \tag{8}$$

$$a_j(t) = -\frac{\gamma M}{j\pi}\eta_j\dot{}(t)$$

$$p'(x,t) = \sum_{j=1}^{N} -\frac{\gamma M}{j\pi}\eta_j\dot{}(t)sin(j\pi x) \tag{9}$$

Putting this equation in the linearized momentum equation gives:

$$u'(x,t) = \sum_{j=1}^{N}\eta_j(t)cos(j\pi x)$$

Differentiating acoustic pressure with time and acoustic velocity with x gives:

$$\frac{\partial p'(x,t)}{\partial t} = -\sum_{j=1}^{N}\frac{\gamma M}{j\pi}\frac{d}{dt}(\dot{\eta_j})sin(j\pi x)$$

$$\frac{\partial u'(x,t)}{\partial x} = -\sum_{j=1}^{N} j\pi(\eta_j)sin(j\pi x)$$

Putting these equations in the energy equations give:

$$-\left[\sum_{j=1}^{N}\frac{\gamma M}{j\pi}\frac{d}{dt}(\dot{\eta_j}) + \gamma M j\pi\eta_j\right]sin(j\pi x) = K\left[\sqrt{|\frac{1}{3}+u'_f(t-\tau)|}-\sqrt{\frac{1}{3}}\right]\delta(x-x_f) \tag{10}$$

Computing inner product with basis function: Multiplying both sides by $sin(j\pi x)$ and integrating along the domain [0,1]

$$\int_0^1 \sum_{j=1}^{N} \left[\frac{\gamma M}{j\pi}\frac{d}{dt}(\dot{\eta_j}) + \gamma M j\pi\eta_j\right] sin(j\pi x)sin(n\pi x)dx = \int_0^1 -K\left[\sqrt{|\frac{1}{3} + u'_f(t-\tau)|} - \sqrt{\frac{1}{3}}\right]\delta(x-x_f)sin(n\pi x)dx \tag{11}$$

From the orthogonality relations:

$$\int_0^1 sin(j\pi x)sin(n\pi x)dx = \frac{1}{2}\delta_{jn}$$

$$\int_0^1 f(x)\delta(x-x_f)dx = f(x_f)$$

$$\left[\frac{\gamma M}{j\pi}\frac{d}{dt}(\dot{\eta_j}) + \gamma M j\pi\eta_j\right]\frac{1}{2} = -K\left[\sqrt{|\frac{1}{3} + u'_f(t-\tau)|} - \sqrt{\frac{1}{3}}\right]\delta(x-x_f) \tag{12}$$

We know that, $\omega_j = k_j = j\pi$

$$\boxed{\frac{d\dot{\eta_j}}{dt} + k_j^2\eta_j = -\frac{2kj\pi}{\gamma M}\left[\sqrt{|\frac{1}{3} + u'_f(t-\tau)|} - \sqrt{\frac{1}{3}}\right] sin(j\pi x_f)}$$

$$\boxed{\frac{d\eta_j}{dt} = \dot{\eta_j}}$$

A frequency dependent damping is added according to [3]:

$$\zeta_j = \frac{1}{2\pi}\left[c_1\frac{\omega_j}{\omega_1} + c_2\sqrt{\frac{\omega_1}{\omega_j}}\right]$$

$$\boxed{\frac{d\dot{\eta_j}}{dt} + 2\zeta_j\omega_j\dot{\eta_j} + k_j^2\eta_j = -\frac{2kj\pi}{\gamma M}\left[\sqrt{|\frac{1}{3} + u'_f(t-\tau)|} - \sqrt{\frac{1}{3}}\right] sin(j\pi x_f)}$$

The Runge-Kutta Method is a powerful tool for solving differential equations with high accuracy. It provides an approximate solution for the value of y at a given point x. However, it's important to note that the Runge-Kutta RK4 method is limited to solving only first-order ordinary differential equations (ODEs).

Problem Statement

The RK4 method is employed to approximate the solution to a first-order initial value problem given by:

$$\frac{dy}{dx} = f(x, y)$$

with an initial condition $y(x_0) = y_0$.

Algorithm Steps

1. **Step Size Determination**: Choose an appropriate step size h that defines the spacing between consecutive points at which the solution will be estimated.

2. **Iteration**: At each step i, RK4 calculates four intermediate slopes k_1, k_2, k_3, and k_4 using a weighted combination of function evaluations at different points within the step. These slopes are then used to estimate the solution at the next step.

3. **Update**: Update the solution using a weighted combination of the slopes.

Formulas

$$
\begin{aligned}
k_1 &= hf(x_i, y_i) \\
k_2 &= hf\left(x_i + \frac{h}{2}, y_i + \frac{k_1}{2}\right) \\
k_3 &= hf\left(x_i + \frac{h}{2}, y_i + \frac{k_2}{2}\right) \\
k_4 &= hf(x_i + h, y_i + k_3) \\
y_{i+1} &= y_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)
\end{aligned}
$$

Advantages

- RK4 is relatively simple to implement.
- It provides a good balance between accuracy and computational cost.
- It is more accurate than simpler methods like Euler's method.

5 Results and Discussions

5.1 Evolution of non dimensional acoustic velocity and non-linear evolution of acoustic energy.

- If the system is non-normal as well as non-linear, oscillations can grow even though the eigen values indicate linear stability. For such systems there exist some initial conditions in which the oscillations grow and decay respectively. [**1**]
- In this system the heater is located at xf = 0.29m, u = 0.5 m/s and c = 399.6 m/s and we see trigerring in the absence of damping. "Triggering" in the context of thermoacoustic devices like the Rijke tube and solid rocket motors refers to the sudden onset or initiation of a self-sustained oscillation or combustion process. In the first u' v/s t plot we observe damping for heater value K = 0.1 and in the second u' v/s t plot we observe amplitude of oscillations saturate at heater value K = 1.25. Hence at different initial conditions we see different oscillations. Also saturations can occur in the absence of damping if the phase difference between the acoustic oscillations and the heat release oscillations evolves to 90° as the system approaches limit cycle.
- The non-dimensional acoustic energy initially grows. After sufficient transient growth, the nonlinearity picks up. Hence short-term growth of fluctuations can lead to significant amplitudes where nonlinear effects could cause nonlinear driving.

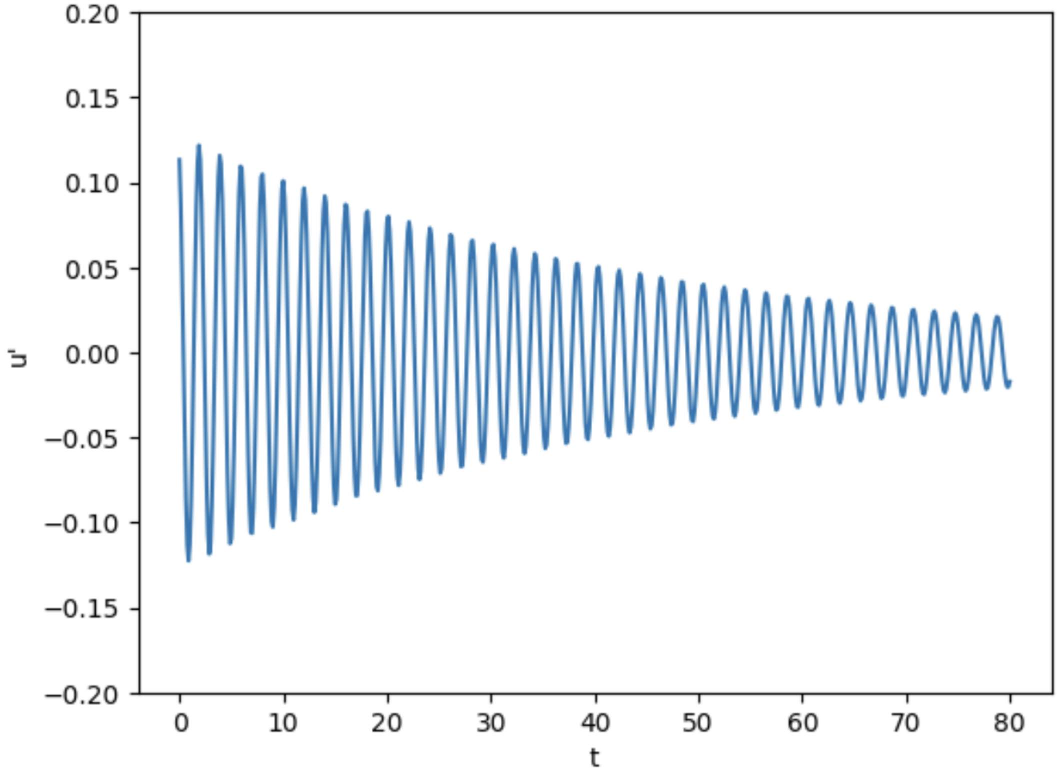

Figure 3: u' with time, $\zeta = 0$, xf = 0.29, K = 0.1

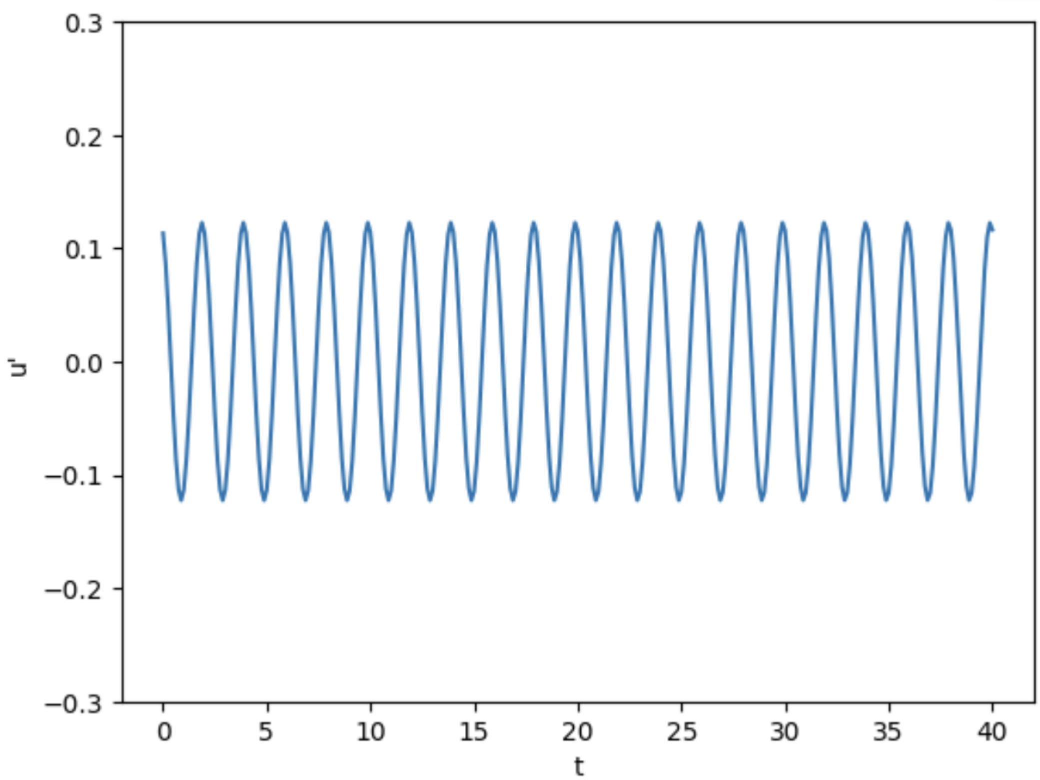

Figure 4: u' with time, $\zeta = 0$, xf = 0.29, K=1.25

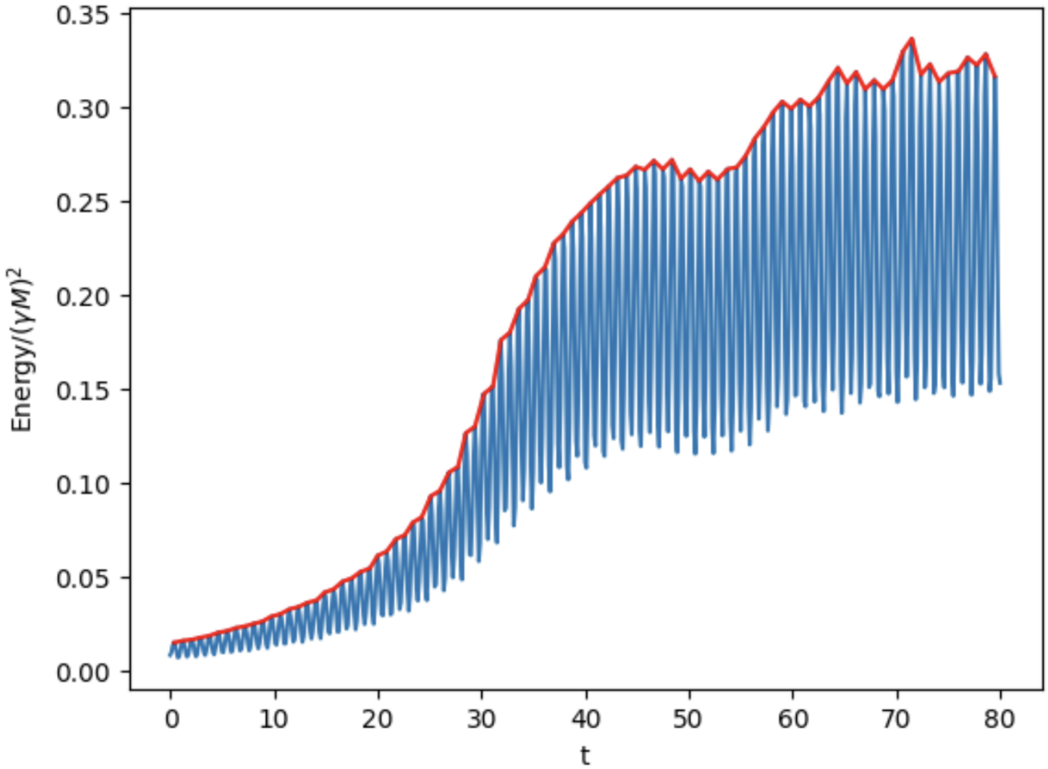

Figure 5: Energy with time, K = 1.4, τ = 0.2

5.2 Evolution of acoustic velocity projected onto various Galerkin modes.

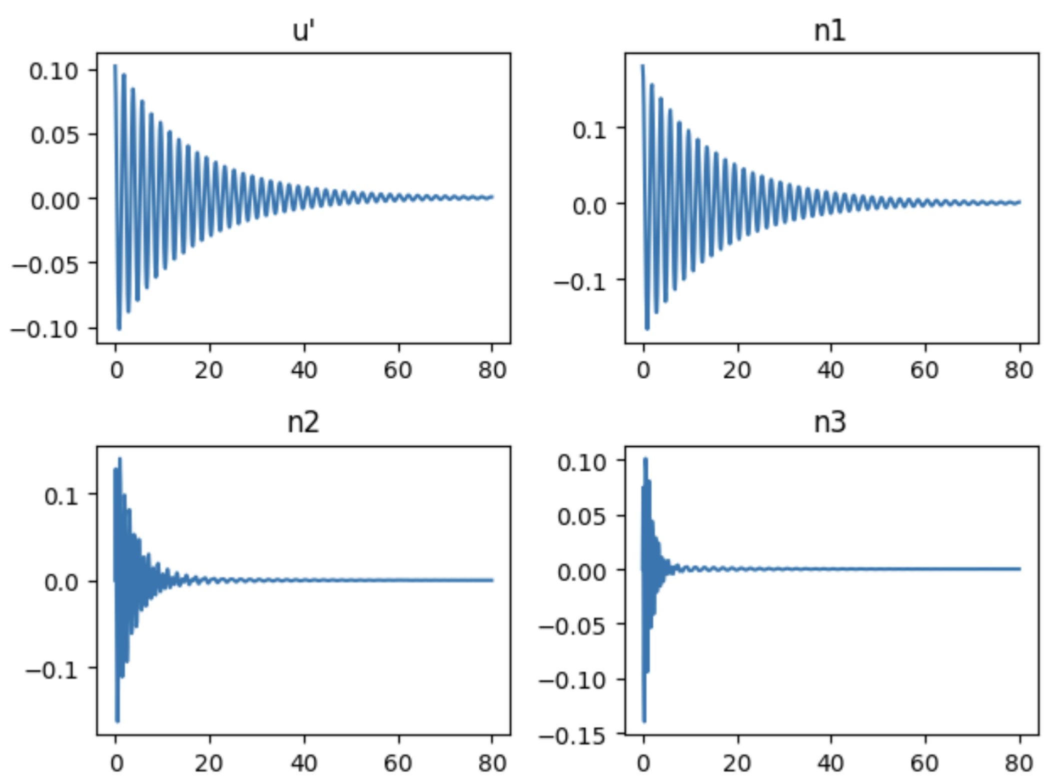

Figure 6: K = 0.2, η_1 = 0.18

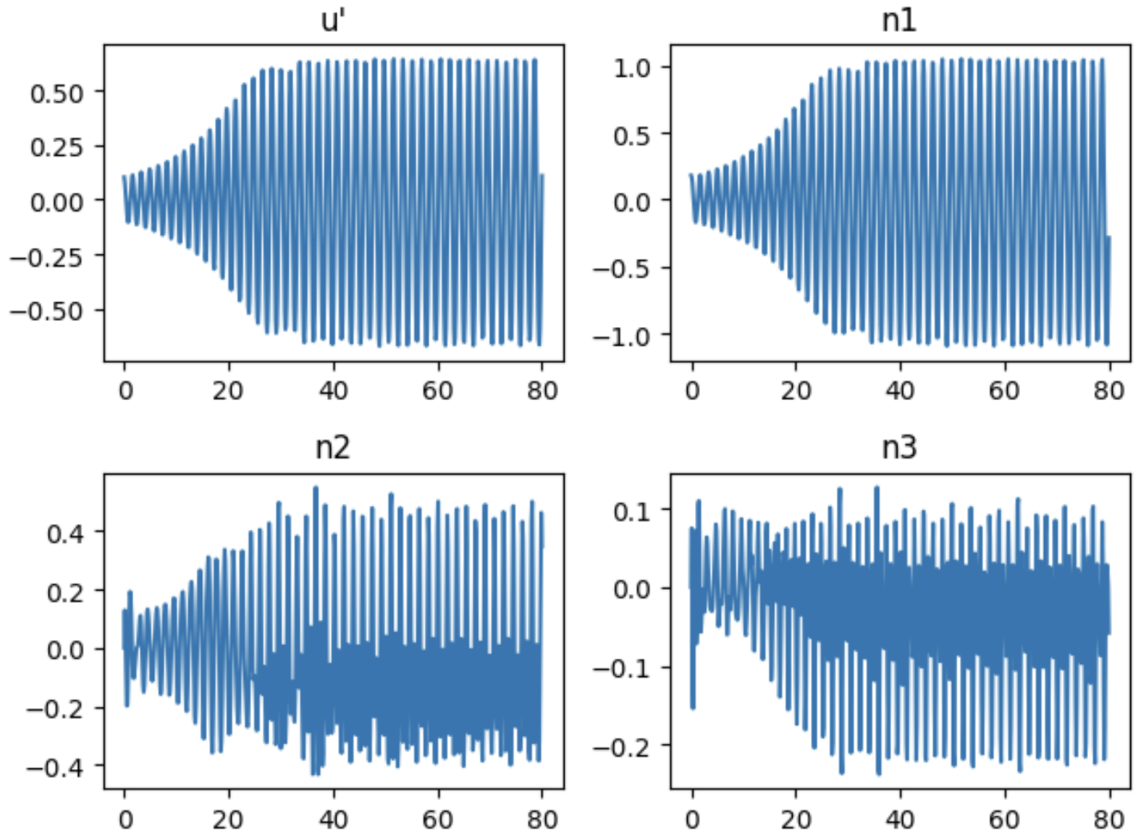

Figure 7: K = 1.8, $\eta_1 = 0.18$

- We observed bootstrapping in a thermoacoustic system which is stable according to classical linear stability analysis based on eigenvalues. Bootstrapping refers to a phenomenon where small perturbations or fluctuations, which are normally insignificant and would decay away in a linearly stable system, are amplified and sustained by nonlinear interactions within the system.
- In this system, heater was located at 1/4 the duct length, the initial conditions were $\eta_1(0) = 0.18$, $\eta_{i\neq1} = 0$ and $\dot{\eta_i}(0) = 0$. Other parameters were maintained to be same as previous example.
- We observe the evolution of acoustic energy at the heater location. It can be seen that low frequency oscillations that are initially present in the system decay and high frequency oscillations set in after some time. Further, it can be seen that the oscillations eventually saturate after nonlinear growth.
- The next figures show the evolution of the acoustic velocity projected on the first three Galerkin expansion functions. After sufficient energy is projected onto the second and third expansion functions, they project the energy back, causing the energy projected on the first expansion function to grow. The net effect of all these energy transfer causes the acoustic velocity to grow and eventually saturate. This feature is known as bootstrapping.

5.3 Bifurcation plot for variation of non-dimensional heater power K.

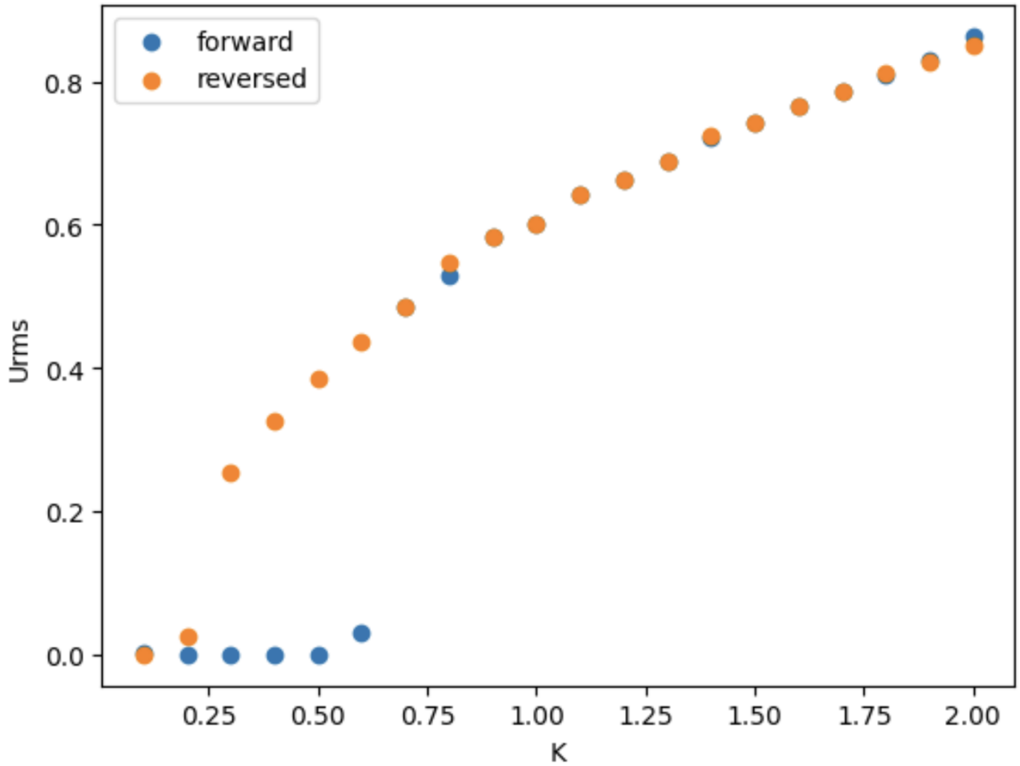

Figure 8: Parameter values of the system are c1 = 0.1, c2 = 0.06, xf = 0.3 and $\tau = 0.2$.

- The effect of varying the non-dimensional heater power (K) on the evolution of the system is analyzed with the bifurcation diagram.
- Increasing the electrical power increases the nondimensional heater power and it represents an increase in the driving force given to the system. Increased driving strives to destabilize the system. Therefore, for small values of K, the equilibrium is stable and all perturbations decay asymptotically to zero. Increasing K decreases the margin of stability of the flow and at a critical value of K, a pair of complex eigenvalues of the system cross over to the right half plane (Hopf bifurcation) and the system becomes linearly unstable.
- The bifurcation is subcritical and the resulting small-amplitude limit cycles close to the Hopf point are unstable. This unstable branch of limit cycles further undergoes a fold or turning point bifurcation and gains stability.
- This bistable region lies between the linear stability boundary given by the Hopf point and the nonlinear stability boundary given by the fold points. Here the system can reach an equilibrium solution or a limit cycle depending on the initial conditions. [**2**]

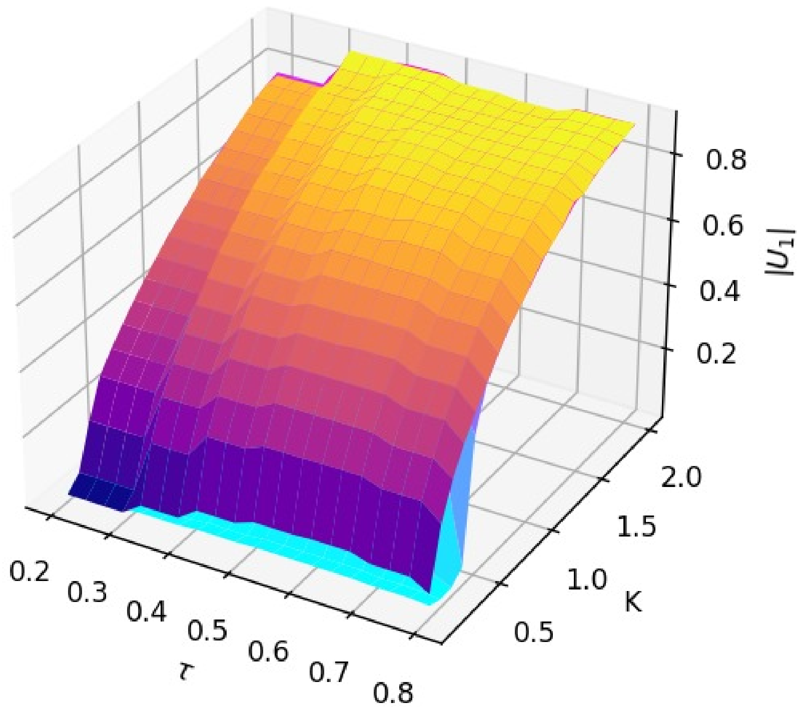

Figure 9: 3D bifurcation plot of non-dimensional heater power for varying values of time lag

6 Summary

Our exploration revolves around the study of thermoacoustic interaction, focusing on a simplified model representing a horizontal Rijke tube—a classic setup for investigating acoustic phenomena.
We start by examining the Runge-Kutta (RK4) method, a widely-used numerical technique for solving differential equations. We explain its application in approximating solutions for first-order ordinary differential equations (ODEs), which form the basis of our modeling approach.
In our modeling efforts, we simplify the system by neglecting mean flow and mean temperature gradient within the duct. Instead, we focus on a tube open at both ends, providing clear boundary conditions for our model. We employ the Galerkin technique—a common method for solving partial differential equations—to derive a solution. Choosing appropriate basis functions ensures adherence to the prescribed boundary conditions.
Throughout our analysis, we track the evolution of acoustic velocity and energy over time. We also explore the concept of damping and its influence on oscillations within the system. Additionally, we investigate phenomena such as projected Galerkin modes, triggering instability, and bootstrapping. Triggering instability refers to transient growth prompting nonlinear driving, initiating oscillation amplitudes. We also discuss subcritical Hopf bifurcations—a crucial aspect offering insights into the behavior of the Rijke tube.

References

[1] K. Balasubramanian and R. I. Sujith, "Thermoacoustic instability in a Rijke tube: Non-normality and nonlinearity," *Phys. Fluids* **20**, 044103 (2008). `https://doi.org/10.1063/1.2895634`

[2] Priya Subramanian, Sathesh Mariappan, R. I. Sujith, and Pankaj Wahi, "Bifurcation analysis of thermoacoustic instability in a horizontal Rijke tube," *International Journal of Spray and Combustion Dynamics*, **2**(4), 325–356 (2010).

[3] K. I. Matveev, Thermoacoustic instabilities in the Rijke tube: Experiments and modeling, Ph.D. thesis, 2003, California Institute of Technology, Pasadena.

7 Appendix

7.1 Codes to get the plots

```python

import numpy as np
import math
import matplotlib.pyplot as plt


# Constants
j = 1
omegaj = j * math.pi
kj = j * math.pi
c1 = 0.1
c2 = 0.06
omega1 = math.pi
r = omega1 / omegaj
zetaj = (1 / (2 * math.pi)) * ((c1 * 1 / r) + (c2 * math.sqrt(r)))
#zetaj = 0
#K = 0.025
tau = 0.2
xf = 0.29
gamma = 1.4
ubar = 0.5
c0 = 399.6
M = ubar/c0
# Define initial conditions and range
nj0 = 0.2
nj_dot0 = 0.0
t0 = 0
tf = 80
h = 0.125  # Step size

num_steps = int((tf - t0)/h)

# Define functions for the first derivatives
def f1(nj, nj_dot):
    return nj_dot


def f2(nj, nj_dot, kj, omegaj, zetaj):
    return ((-2 * zetaj * omegaj * nj_dot) - (kj**2 * nj))


def f3(nj, nj_dot, kj, omegaj, zetaj, ufprime):

    #a = ((2 * j * math.pi * K) / (gamma * M)) * math.sin(j * math.pi * xf)
    #b = math.sqrt(abs(1/3 + ufprime)) - math.sqrt(1/3)

    return -2*zetaj*omegaj*nj_dot - (kj**2)*nj - 2*j*math.pi*K*(math.sqrt(
    abs(1/3 + ufprime)) - math.sqrt(1/3))*math.sin(j*math.pi*xf)



# Define the RK4 solver
def rk4_second_order1(f1, f2, nj0, nj_dot0, h, kj, omegaj, zetaj):
    num_steps1 = int(tau / h)  # Number of steps
    #print(num_steps1)
    #t = t0
```

```python
    nj = nj0
    nj_dot = nj_dot0


    #ts = [t0]
    njs = [nj0]
    nj_dots = [nj_dot0]
    uprime = []
    pprime = []


    for i in range(num_steps1+1):
        k1nj = h * f1(nj, nj_dot)
        k1nj_dot = h * f2(nj, nj_dot, kj, omegaj, zetaj)

        k2nj = h * f1(nj + k1nj/2, nj_dot + k1nj_dot/2)
        k2nj_dot = h * f2(nj + k1nj/2, nj_dot + k1nj_dot/2, kj, omegaj, zetaj)

        k3nj = h * f1(nj + k2nj/2, nj_dot + k2nj_dot/2)
        k3nj_dot = h * f2(nj + k2nj/2, nj_dot + k2nj_dot/2, kj, omegaj, zetaj)

        k4nj = h * f1(nj + k3nj, nj_dot + k3nj_dot)
        k4nj_dot = h * f2(nj + k3nj, nj_dot + k3nj_dot, kj, omegaj, zetaj)

        nj += (k1nj + 2*k2nj + 2*k3nj + k4nj) / 6
        nj_dot += (k1nj_dot + 2*k2nj_dot + 2*k3nj_dot + k4nj_dot) / 6
        #t += h

        #ts.append(t)
        njs.append(nj)
        nj_dots.append(nj_dot)

        uprime_tval = nj * math.cos(j * math.pi * xf)
        uprime.append(uprime_tval)

        pprime_tval = - ((gamma * M)/(j * math.pi)) * nj_dot * math.sin(j * math.pi * xf)
        pprime.append(pprime_tval)

        ''' for different modes
        if j==1:
          uprime_tval = nj * math.cos(j * math.pi * xf)
        elif j==2:
          uprime_tval = nj * math.cos(j * math.pi * xf) + u2[0][i]
        else:
          uprime_tval = nj * math.cos(j * math.pi * xf) + u2[0][i] + u2[1][i]

        uprime.append(uprime_tval)

        '''

    #print(len(uprime))
    return njs, nj_dots, uprime, pprime


# Define the RK4 solver
def rk4_second_order2(f1, f2, f3, nj0, nj_dot0, h, kj, omegaj, zetaj):

```

```python
    nj1, njdot1, u, p = rk4_second_order1(f1, f2, nj0, nj_dot0, h, kj, omegaj, zetaj)



    for i in range(num_steps + 1 - len(u)):
        nj = nj1[-1]
        nj_dot = njdot1[-1]

        k1nj = h * f1(nj, nj_dot)
        k1nj_dot = h * f3(nj, nj_dot, kj, omegaj, zetaj, u[i+1])

        k2nj = h * f1(nj + k1nj/2, nj_dot + k1nj_dot/2)
        k2nj_dot = h * f3(nj + k1nj/2, nj_dot + k1nj_dot/2, kj, omegaj, zetaj, u[i+1])

        k3nj = h * f1(nj + k2nj/2, nj_dot + k2nj_dot/2)
        k3nj_dot = h * f3(nj + k2nj/2, nj_dot + k2nj_dot/2, kj, omegaj, zetaj, u[i+1])

        k4nj = h * f1(nj + k3nj, nj_dot + k3nj_dot)
        k4nj_dot = h * f3(nj + k3nj, nj_dot + k3nj_dot, kj, omegaj, zetaj, u[i+1])

        nj += (k1nj + 2*k2nj + 2*k3nj + k4nj) / 6
        nj_dot += (k1nj_dot + 2*k2nj_dot + 2*k3nj_dot + k4nj_dot) / 6

        #print(nj)
        nj1.append(nj)
        njdot1.append(nj_dot)

        uprime_tval = nj * math.cos(j * math.pi * xf)
        u.append(uprime_tval)

        pprime_tval = - ((gamma * M)/(j * math.pi)) * nj_dot * math.sin(j * math.pi * xf)
        p.append(pprime_tval)

        ''' for different modes
        if j==1:
          uprime_tval = nj * math.cos(j * math.pi * xf)
        elif j==2:
          uprime_tval = nj * math.cos(j * math.pi * xf) + u2[0][i]
        else:
          uprime_tval = nj * math.cos(j * math.pi * xf) + u2[0][i] + u2[1][i]

        u.append(uprime_tval)

        '''

    return nj1, njdot1, u, p


################################# Acoustic Energy and velocity

K = 1.4
t_values = np.arange(t0, tf + h, h)

nj2, njdot2, u2, p2 = rk4_second_order2(f1, f2, f3, nj0, nj_dot0, h, kj, omegaj, zetaj)
```

```python

E = [((0.5 * a**2) + (0.5 * (gamma * M * b)**2))/((gamma * M)**2) for a, b in zip(p2, u2)]

from scipy.signal import find_peaks
E_flat = np.hstack(E)
peaks, _ = find_peaks(E_flat)
#valleys, _ = find_peaks(-E_flat)

plt.plot(t_values, E)
plt.plot(t_values[peaks], E_flat[peaks], "", color='red')
#plt.plot(t_values[valleys], E_flat[valleys], "x", color='red')
#plt.ylim([-0.2, 0.2])
plt.xlabel('t')
plt.ylabel(r'Energy/($\gamma M)^2$')
#plt.ylabel("u'")
plt.show()


##########################################.  Different Galerkin Modes


# Loop through different j values
for j in range(1, 4):
    omegaj = j * np.pi
    kj = j * np.pi
    zetaj = (1 / (2 * np.pi)) * ((c1 * j * np.pi) / (np.pi) + (c2 * np.sqrt(np.pi / (j * np.pi))))

    nj3, njdot3, u3= rk4_second_order2(f1, f2, f3, nj0, nj_dot0, h, kj, omegaj, zetaj)
    nj2.append(nj3)
    njdot2.append(njdot3)
    u2.append(u3)

# Plot nj2 for j=2
nj2[0].pop()
nj2[1].pop()
nj2[2].pop()
# Create subplots
fig, axs = plt.subplots(2, 2)

# Plot data on each subplot
axs[0, 0].plot(t_values, u2[0])
axs[0, 0].set_title("u'")

axs[0, 1].plot(t_values, nj2[0])
axs[0, 1].set_title('n1')

axs[1, 0].plot(t_values, nj2[1])
axs[1, 0].set_title('n2')

axs[1, 1].plot(t_values, nj2[2])
axs[1, 1].set_title('n3')

# Adjust layout to prevent overlapping titles
plt.tight_layout()

# Show the plot
plt.show()

```

```
################################# 2D Bifurcation plot


U2 = []
U3 = []
K2 = np.linspace(0.1, 2, 20)
K3 = np.flip(K2)


t_values = np.arange(t0, tf + h, h)
for K in K2:
  nj2, njdot2, u2= rk4_second_order2(f1, f2, f3, nj0, nj_dot0, h, kj, omegaj
    , zetaj, K)
  subset = u2[len(u2)//2:]
  U = np.sqrt(np.mean(np.square(subset)))
  U2.append(U)
  nj0 = nj2[-1]
  nj_dot0 = njdot2[-1]
  #print(nj0)

nj02 = 0.6
nj_dot02 = 0.1
for K in K3:
  nj3, njdot3, u3= rk4_second_order2(f1, f2, f3, nj02, nj_dot02, h, kj,
    omegaj, zetaj, K)
  subset2 = u3[len(u3)//2:]
  U0 = np.sqrt(np.mean(np.square(subset2)))
  U3.append(U0)
  nj02 = nj3[-1]
  nj_dot02 = njdot3[-1]
  #print(nj2[-1])



plt.scatter(K2, U2)
plt.scatter(K3, U3)
plt.legend(["forward", "reversed"])
plt.xlabel('K')
plt.ylabel("Urms")
plt.show()


#################################### 3D Bifurcation plot



Urms_values = np.zeros((len(tau_values), len(K_values)))

Urms2 = np.zeros((len(tau_values), len(Kflip)))
#Urms_values = []
i = 0
j1 = 0
k = 0
# Calculate Urms for each combination of K and tau
for tau in tau_values:
    nj = 0.2
    nj_dot = 0.0
    for K in K_values:
```

```
        nj2, njdot2, u2 = rk4_second_order2(f1, f2, f3, nj, nj_dot, h, kj, omegaj, zetaj, K, tau)
        subset = u2[len(u2)//2:]
        Urms = np.sqrt(np.mean(np.square(subset)))
        Urms_values[i, j1] = Urms
        nj = nj2[-1]
        nj_dot = njdot2[-1]
        j1 = j1 + 1

    nj = 0.6
    nj_dot = 0.1
    for K in Kflip:
        nj3, njdot3, u3 = rk4_second_order2(f1, f2, f3, nj, nj_dot, h, kj, omegaj, zetaj, K, tau)
        subset = u3[len(u2)//2:]
        U_rms = np.sqrt(np.mean(np.square(subset)))
        Urms2[i, k] = U_rms
        nj = nj3[-1]
        nj_dot = njdot3[-1]
        k = k + 1
    i = i + 1
    j1 = 0
    k = 0

# Create mesh grid for K and tau
tau_mesh, K_mesh = np.meshgrid(tau_values, K_values)

tau_mesh2, K_mesh2 = np.meshgrid(tau_values, Kflip)

# Plot 3D bifurcation plot with z-axis on the left
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.plot_surface(tau_mesh, K_mesh, Urms_values.T, cmap='cool')
ax.plot_surface(tau_mesh2, K_mesh2, Urms2.T, cmap='plasma')
ax.set_xlabel('$\\tau$')
ax.set_ylabel('K')
ax.set_zlabel('U')
plt.show()
```