College of Engineering
School of Aerospace Engineering

ME 6444: Nonlinear Systems

HW3

Professor:
Michael J. Leamy
Gtech ME Professor

Student:
Tomoki Koike
Gtech MS Student

October 7, 2021

Table of Contents

Problem 1

(20 PTS) Hamilton's Principle - Nonlinear String

a. Derive the equations of motion for a spatial string (i.e., string with motions in the x, y, and z directions) using both the full NL strain expression and a NL material law of the form $\sigma = E_1\epsilon + E_2\epsilon^2$. Do not reduce the theory using a quasi-static stretching assumption.

b. Introduce a linear damping (i.e. Kelvin-Voigt) such that the material law takes the form $\sigma = E_1\epsilon + E_2\epsilon^2 + \alpha\dot{\epsilon}$ and re-derive the equations of motion. **Hint:** The problem is now non-conservative and it may be easier to calculate the virtual work done by internal forces rather than using the concept of string energy. Please see Lecture 11 for more information.

Solution:

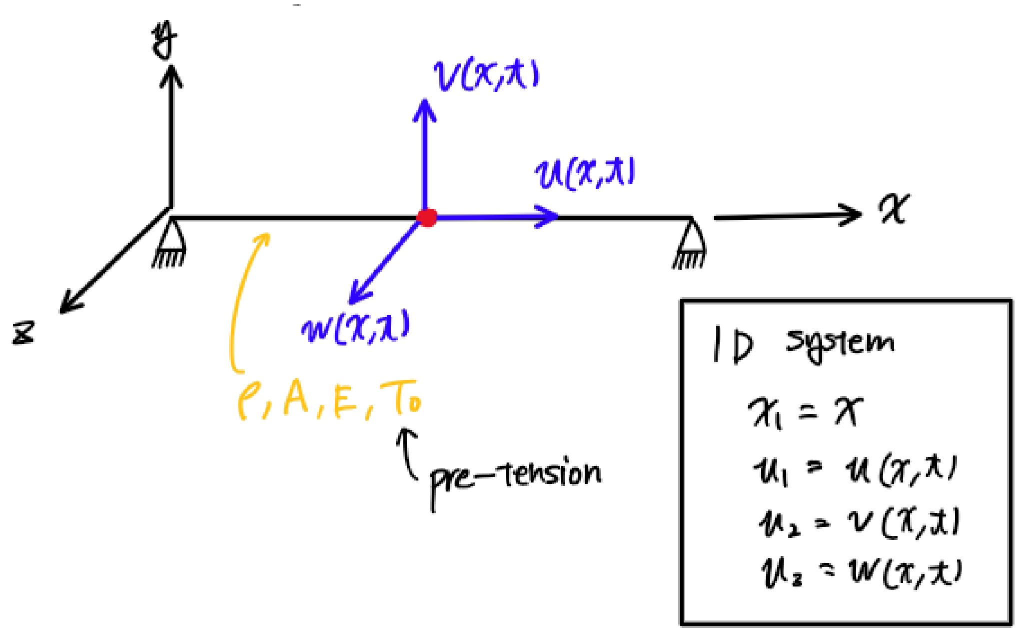

Figure 1: Nonlinear string problem diagram

a) First we must define variables and parameters which characterize the nonlinear string problem shown in Figure 1.

- l: length of the string $[m]$

- ρ: density of string $[kg/m^3]$
- A: cross-sectional area of the string $[m^2]$
- E: Young's modulus of the string $[N/m^2]$
- T_0: pre-tension $[N]$
- u: longitudinal displacement $[m]$
- v: lateral displacement $[m]$
- w: vertical displacement $[m]$

The dynamic exerted on a small portion of the string is shown below.

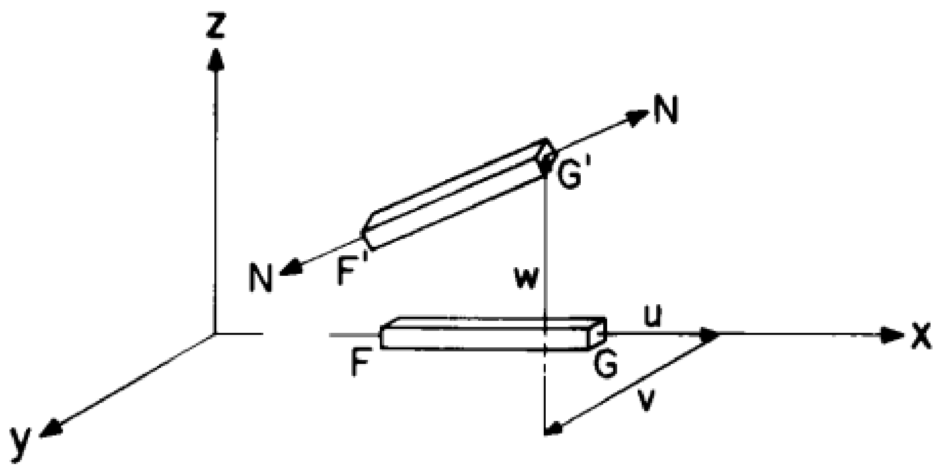

Figure 2: Diagram of virtual displacement [1]

This problem is has geometric nonlinearity and material nonlinearity. For the stress we add an additional pre-stress term as follows.

$$\sigma = \frac{T_0}{A} + E_1\epsilon + E_2\epsilon^2.$$

To find the equation of motion we use the Hamilton's principle. Firstly, the kinetic energy is

$$T = \frac{1}{2}\int_0^l \rho A\left(\dot{u}^2 + \dot{v}^2 + \dot{w}^2\right)dx$$

Recall that for a 1-D string system the only non-zero strain term is ϵ_{11}, and

$$\begin{aligned}
\epsilon_{11} &= \frac{1}{2}\left(u_{i,j} + u_{j,i} + u_{k,i}\cdot u_{k,j}\right) \\
&= \frac{1}{2}\left(u_x + u_x + u_x\cdot u_x + v_x\cdot v_x + w_x\cdot w_x\right) \\
&= u_x + \frac{1}{2}\left(u_x^2 + v_x^2 + w_x^2\right)
\end{aligned}$$

The potential energy can be computed as follows.

$$\begin{aligned}
u &= \int \sigma_x x d\epsilon_{11} \\
&= \int \left(\frac{T_0}{A} + E_1\epsilon_{11} + E_2\epsilon_{11}^2\right) d\epsilon_{11} \\
&= \frac{T_0}{A}\epsilon_{11} + \frac{1}{2}E_1\epsilon_{11}^2 + \frac{1}{3}E_2\epsilon_{11}^3
\end{aligned}$$

Then

$$\begin{aligned}
U &= \int_0^l uAdx \\
&= \int_0^l \left(T_0\epsilon_{11} + \frac{1}{2}E_1A\epsilon_{11}^2 + \frac{1}{3}E_2A\epsilon_{11}^3\right) dx
\end{aligned}$$

The Lagrangian becomes

$$L = T - U = \frac{1}{2}\int_0^l \rho A\left(\dot{u}^2 + \dot{v}^2 + \dot{w}^2\right) dx - \int_0^l \left(T_0\epsilon_{11} + \frac{1}{2}E_1\epsilon_{11}^2 + \frac{1}{3}E_2\epsilon_{11}^3\right) dx.$$

Thus,

$$\delta L = \int_0^l \left[\rho A\left(\dot{u}\delta\dot{u} + \dot{v}\delta\dot{v} + \dot{w}\delta\dot{w}\right) - \left(T_0\epsilon_{11} + E_1A\epsilon_{11} + E_2A\epsilon_{11}^2\right)\delta\epsilon_{11}\right] dx.$$

Now since $\delta W^{NC} = 0$, the extended Hamilton's principle

$$\int_{t_1}^{t_2} \left[\delta L + \delta w^{NC}\right] dt = 0$$

becomes

$$\int_{t_1}^{t_2}\int_0^l \left[\rho A\left(\dot{u}\delta\dot{u} + \dot{v}\delta\dot{v} + \dot{w}\delta\dot{w}\right) - \left(T_0\epsilon_{11} + E_1A\epsilon_{11} + E_2A\epsilon_{11}^2\right)\delta\epsilon_{11}\right] dxdt = 0$$

where

$$\delta\epsilon_{11} = (1 + u_x)\delta u_x + v_x\delta v_x + w_x\delta w_x.$$

Using integration by parts we have

$$\int_0^l \int_{t_1}^{t_2} (\rho A \dot{u} \delta \dot{u}) dt dx = \int_0^l \rho A \Big\{ \big[\dot{u} \delta u \big]_{t_1}^{t_2} - \int_{t_1}^{t_2} \ddot{u} \delta u dt \Big\} dx$$
$$= - \int_0^l \int_{t_1}^{t_2} \Big(\rho A \ddot{u} \delta u \Big) dt dx \qquad \because \delta u(t_1) = \delta u(t_2) = 0$$

Similarly,

$$\int_0^l \int_{t_1}^{t_2} (\rho A \dot{v} \delta \dot{v}) dt dx = - \int_0^l \int_{t_1}^{t_2} \Big(\rho A \ddot{v} \delta v \Big) dt dx$$
$$\int_0^l \int_{t_1}^{t_2} (\rho A \dot{w} \delta \dot{w}) dt dx = - \int_0^l \int_{t_1}^{t_2} \Big(\rho A \ddot{w} \delta w \Big) dt dx$$

We also use integration by part for the following.

$$\int_{t_1}^{t_2} \int_0^l \sigma A \left(1 + u_x\right) \delta u_x dx dt$$
$$= \int_{t_1}^{t_2} \left\{ \Big[\sigma A (1 + u_x) \delta u \Big]_0^l - \int_0^l \Big(\sigma A (1 + u_x) \Big)_x \delta u dx \right\} dt$$

from the boundary condition $u(l, t) = u(0, t) = 0$

$$= - \int_{t_1}^{t_2} \int_0^l \Big(\left(T_0 \epsilon_{11} + E_1 A \epsilon_{11} + E_2 A \epsilon_{11}^2 \right) (1 + u_x) \Big)_x \delta u dx dt$$

Similarly,

$$\int_{t_1}^{t_2} \int_0^l \sigma A v_x \delta v_x dx dt = - \int_{t_1}^{t_2} \int_0^l \left(\left(T_0 \epsilon_{11} + E_1 A \epsilon_{11} + E_2 A \epsilon_{11}^2 \right) v_x \right)_x \delta v dx dt$$
$$\int_{t_1}^{t_2} \int_0^l \sigma A w_x \delta w_x dx dt = - \int_{t_1}^{t_2} \int_0^l \left(\left(T_0 \epsilon_{11} + E_1 A \epsilon_{11} + E_2 A \epsilon_{11}^2 \right) w_x \right)_x \delta w dx dt$$

Finally by combining everything we get the following equation of motion

$$\boxed{\begin{aligned} \rho A \ddot{u} - \left(\left(T_0 \epsilon_{11} + E_1 A \epsilon_{11} + E_2 A \epsilon_{11}^2 \right) (1 + u_x) \right)_x &= 0 \\ \rho A \ddot{v} - \left(\left(T_0 \epsilon_{11} + E_1 A \epsilon_{11} + E_2 A \epsilon_{11}^2 \right) v_x \right)_x &= 0 \\ \rho A \ddot{w} - \left(\left(T_0 \epsilon_{11} + E_1 A \epsilon_{11} + E_2 A \epsilon_{11}^2 \right) w_x \right)_x &= 0 \end{aligned}}$$

(b) For this problem, because it is non-conservative we disregard the potential energy and use only the kinetic energy. Thus,

$$L = T - \cancelto{0}{V} = \frac{1}{2}\int_0^l \rho A\left(\dot{u}^2 + \dot{v}^2 + \dot{w}^2\right)dx$$

and

$$\delta L = \int_0^l \left[\rho A\left(\dot{u}\delta\dot{u} + \dot{v}\delta\dot{v} + \dot{w}\delta\dot{w}\right)\right]dx.$$

However, with the principle of virtual work we have

$$\begin{aligned}
\delta w^{INT} &= -\delta w^{EXT} = -\left[\left(\sigma + \frac{\partial\sigma}{\partial x}dx\right)A\delta\epsilon_{11}dx\right] \\
&= -\sigma A\delta\epsilon_{11}dx \\
&= -\int_0^l \left(T_0 + E_1 A\epsilon_{11} + E_2 A\epsilon_{11}^2 + \alpha A\dot{\epsilon}\right)\delta\epsilon_{11}dx
\end{aligned}$$

and

$$\delta w^{NC} = \delta w^{INT}.$$

The extended Hamilton's principle is

$$\int_{t_1}^{t_2}\left[\delta L + \delta w^{NC}\right]dt = 0$$

and therefore we end up with the following expression

$$\int_{t_1}^{t_2}\int_0^l \left[\rho A\left(\dot{u}\delta\dot{u} + \dot{v}\delta\dot{v} + \dot{w}\delta\dot{w}\right) - \left(T_0 + E_1 A\epsilon_{11} + E_2 A\epsilon_{11}^2 + \alpha A\dot{\epsilon}\right)\delta\epsilon_{11}\right]dxdt = 0$$

which is the exact same that we had from part (a) of this problem. But the only difference is that the stress has a damping term which renders that problem as non-conservative as the following

$$\sigma = \frac{T_0}{A} + E_1\epsilon_{11} + E_2\epsilon_{11} + \alpha\dot{\epsilon_{11}}.$$

Hence, the derivation of the equation of motion is identical to the steps that we took for part (a), and thus, it becomes

$$\boxed{\begin{aligned}
\rho A\ddot{u} - \left(\left(T_0 + E_1 A\epsilon_{11} + E_2 A\epsilon_{11}^2 + \alpha A\dot{\epsilon_{11}}\right)(1 + u_x)\right)_x &= 0 \\
\rho A\ddot{v} - \left(\left(T_0 + E_1 A\epsilon_{11} + E_2 A\epsilon_{11}^2 + \alpha A\dot{\epsilon_{11}}\right)v_x\right)_x &= 0 \\
\rho A\ddot{w} - \left(\left(T_0 + E_1 A\epsilon_{11} + E_2 A\epsilon_{11}^2 + \alpha A\dot{\epsilon_{11}}\right)w_x\right)_x &= 0
\end{aligned}}$$

Problem 2

(30 PTS) Galerkin's Method - Nonlinear String

Continuning from Problem 1.b., study only the in-plane vibration (i.e., set $u(x,t) = w(x,t) = 0$) and only nonlinear terms due to damping (i.e., the only nonlinear terms to keep in the model are those dependent on the damping parameter α).

a. Determine the single PDE governing free motions. Clearly identify the terms arising from Kelvin-Voigt damping. Remark on the characteristic of the damping - would the linear system have a contribution due to damping?

b. Study free motions of a pinned-pinned string using the first two mode shapes of the corresponding linear system. Use Galerkin's method to obtain two nonlinear ODEs.

c. For a parameter set of your choosing, look at pseudo-phase planes in which you plot the free response due to a given set of initial modal displacements and speeds. You should generate two pseudo-phase planes: one corresponding to modal displacement and speed of the first mode, and the second corresponding to the second mode. This can be accomplished using the `scene = [x(t), y(t)]` option in Maple with the `DEPlot` function. Verify that the response for each mode is in fact damped.

Solution:
(a) The given condition of $u(x,t) = w(x,t) = 0$ implies that

$$u_x = u_{xx} = \dot{u} = \ddot{u} = w_x = w_{xx} = \dot{w} = \ddot{w} = 0.$$

Additionally, if we assume that the tension is not effected by the displacement v, we can disregard the first equation and can reduces the equations to

$$\rho A\ddot{v} - \Big(\left(T_0 + E_1 A\epsilon_{11} + E_2 A\epsilon_{11}^2 + \alpha A\dot{\epsilon}_{11}\right) v_x \Big)_x = 0$$
$$\epsilon_{11} = \frac{1}{2}v_x^2$$

Now if we linearize the equation governing $v(x,t)$ while keeping the nonlinear terms associated with the damping term α, we get

$$\rho A\ddot{v} - (T_0 v_x + \alpha A v_x^2 \dot{v}_x)_x = 0$$

which then becomes the single PDE governing the free motions as follows.

$$\boxed{\rho A\ddot{v} - T_0 v_{xx} - 2\alpha A v_x v_{xx}\dot{v}_x - \alpha A v_x^2 \dot{v}_{xx} = 0.}$$

The terms arising from the Kelvin-Voigt damping are colored in blue in the answer. The solution shows that with the nonlinear terms with damping, the linear system will have a contribution due to damping.

(b) Going back to the system of

$$\rho A\ddot{v} - \left(\left(T_0 + E_1 A\epsilon_{11} + E_2 A\epsilon_{11}^2 + \alpha A\dot{\epsilon}_{11}\right) v_x\right)_x = 0$$
$$\epsilon_{11} = \frac{1}{2}v_x^2$$

with the boundary conditions of $v(0,t) = v(l,t) = 0$ with the pinned-pinned string. From the linear system that we have derived in Problem 2.a. and from the boundary condition, we can deduce the basis equations (eigenfunctions) for the Galerikin method to be (with first and second mode)

$$\phi_1 = \sin\left(\frac{\pi x}{l}\right)$$
$$\phi_2 = \sin\left(\frac{2\pi x}{l}\right).$$

And, with the two-term approximation we have

$$\tilde{v}(x,t) = \beta_1(t)\phi_1(x) + \beta_2(t)\phi_2(x).$$

Let,

$$\begin{aligned}\gamma(x,t) &= T_0 + E_1 A\epsilon_{11} + E_2 A\epsilon_{xx}^2 + \alpha A\dot{\epsilon}_{xx} \\ &= T_0 + \frac{1}{2}E_1 A v_x^2 + \frac{1}{4}E_2 A v_x^4 + \alpha A v_x \dot{v}_x\end{aligned}$$

The partial derivatives of $\tilde{v}$, are as follows

$$\frac{\partial}{\partial x}\tilde{v} = \frac{\pi}{l}\beta_1\cos\left(\frac{\pi x}{l}\right) + \frac{2\pi}{l}\beta_2\cos\left(\frac{2\pi x}{l}\right)$$

$$\frac{\partial^2}{\partial x^2}\tilde{v} = -\frac{\pi^2}{l^2}\beta_1\sin\left(\frac{\pi x}{l}\right) - \frac{4\pi^2}{l^2}\beta_2\sin\left(\frac{2\pi x}{l}\right)$$

$$\frac{\partial^2}{\partial x\partial t}\tilde{v} = \frac{\pi}{l}\dot{\beta}_1\cos\left(\frac{\pi x}{l}\right) + \frac{2\pi}{l}\dot{\beta}_2\cos\left(\frac{2\pi x}{l}\right)$$

Going back to our governing equation, we have

$$\begin{aligned}\rho A\ddot{v} - \left(\left(T_0 + E_1 A\epsilon_{11} + E_2 A\epsilon_{11}^2 + \alpha A\dot{\epsilon}_{11}\right) v_x\right)_x &= 0 \\ \rho A\ddot{v} - \gamma_x v_x - \gamma v_{xx} &= 0\end{aligned}$$

If we plug $\tilde{v}$ into $\tilde{\gamma}$ we have

$$\begin{aligned}
\tilde{\gamma} = T_0 &+ \frac{1}{2}E_1 A\left(\frac{\pi}{l}\beta_1\cos\left(\frac{\pi x}{l}\right) + \frac{2\pi}{l}\beta_2\cos\left(\frac{2\pi x}{l}\right)\right)^2 \\
&+ \frac{1}{4}E_2 A\left(\frac{\pi}{l}\beta_1\cos\left(\frac{\pi x}{l}\right) + \frac{2\pi}{l}\beta_2\cos\left(\frac{2\pi x}{l}\right)\right)^4 \\
&+ \alpha A\left(\frac{\pi}{l}\beta_1\cos\left(\frac{\pi x}{l}\right) + \frac{2\pi}{l}\beta_2\cos\left(\frac{2\pi x}{l}\right)\right)\left(\frac{\pi}{l}\dot{\beta}_1\cos\left(\frac{\pi x}{l}\right) + \frac{2\pi}{l}\dot{\beta}_2\cos\left(\frac{2\pi x}{l}\right)\right)
\end{aligned}$$

The residual functions become

$$\tilde{r} = \rho A\ddot{\tilde{v}} - \tilde{\gamma}_x\tilde{v}_x - \tilde{\gamma}\tilde{v}_{xx}$$

and since we know that the inner product between the basis function and the residual function are equal to zero due to their perpendicular traits

$$\begin{aligned}
\langle\tilde{r},\phi_1\rangle_l &= \int_0^l \tilde{r}\phi_1 dx = 0 \\
\langle\tilde{r},\phi_2\rangle_l &= \int_0^l \tilde{r}\phi_2 dx = 0
\end{aligned}$$

This gives us the final output of two nonlinear ODEs

$$\boxed{\begin{aligned}
&\frac{\rho A l}{2}\ddot{\beta}_1 + \frac{\alpha A}{l^3}\left(36.5284\beta_1^2 + 97.4091\beta_2^2\right)\dot{\beta}_1 + \frac{194.8182\alpha A}{l^3}\beta_1\beta_2\dot{\beta}_2 \\
&\qquad \frac{4.9348T_0}{l}\beta_1 + \frac{E_1 A}{l^3}\beta_1\left(18.2642\beta_1^2 + 146.1136\beta_2^2\right) \\
&\qquad \frac{E_2 A}{l^5}\beta_1\left(75.1085\beta_1^4 + 2.1030\times10^3\beta_1^2\beta_2^2 + 3.6052\times10^3\beta_2^4\right) = 0 \\
\\
&\frac{\rho A l}{2}\ddot{\beta}_2 + \frac{\alpha A}{l^3}\left(97.4091\beta_1^2 + 584.4545\beta_2^2\right)\dot{\beta}_2 + \frac{194.8182\alpha A}{l^3}\beta_1\beta_2\dot{\beta}_1 \\
&\qquad \frac{19.7392T_0}{l}\beta_1 + \frac{E_1 A}{l^3}\beta_2\left(146.1136\beta_1^2 + 292.2273\beta_2^2\right) \\
&\qquad \frac{E_2 A}{l^5}\beta_2\left(1.0515\times10^3\beta_1^4 + 7.2104\times10^3\beta_1^2\beta_2^2 + 4.8069\times10^3\beta_2^4\right) = 0
\end{aligned}}$$

The answers were computed using symbolic calculations of MATLAB (refer to code in Problem 2: MATLAB Code).

(c) Now if we plot the pseudo-phase planes they will look as the following

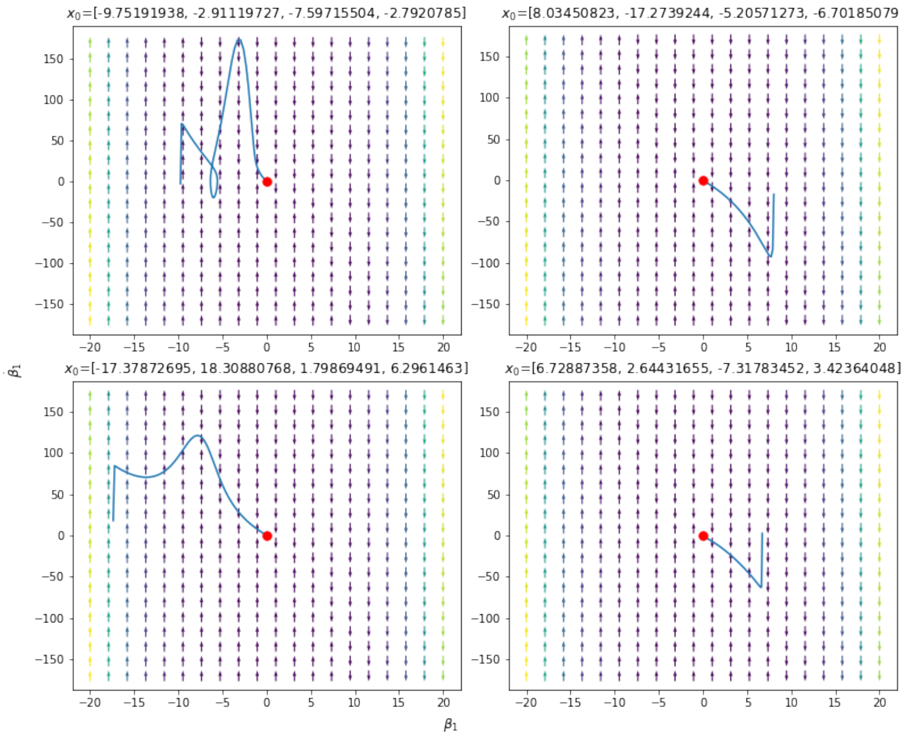

Figure 3: Pseudo-phase plane for mode 1 with 4 different initial conditions

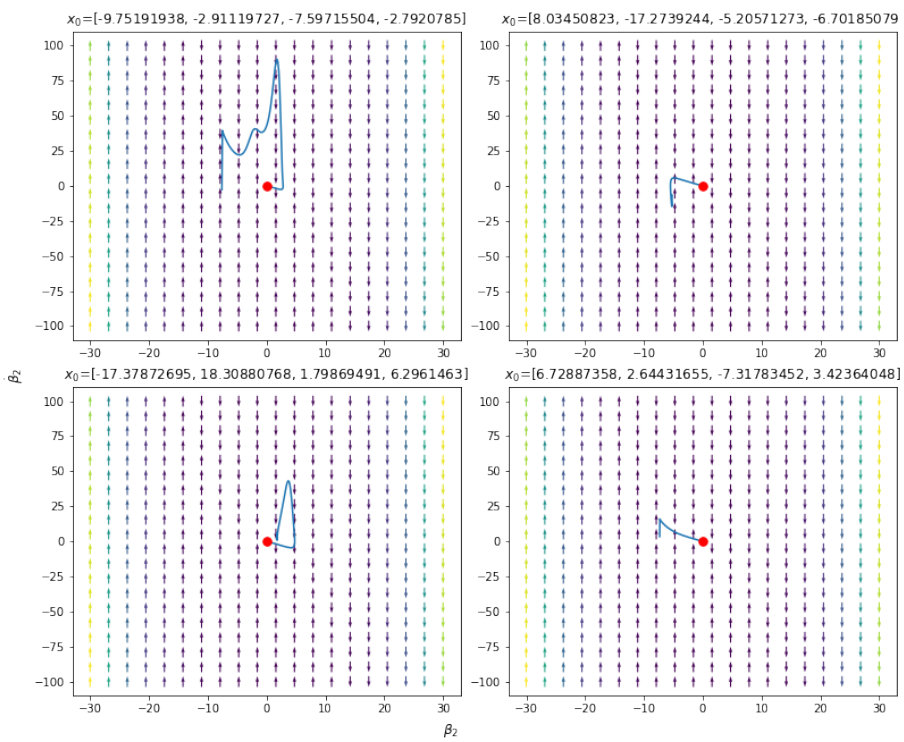

Figure 4: Pseudo-phase plane for mode 2 with 4 different initial conditions

These simulations are done with the following conditions

- ρ: 500
- A: 0.0001
- l: 10
- T_0: 1500
- E_1: 2.5e+6
- E_2: 0.5e+6
- α: 1.2e+6

and with initial conditions of

$$[-9.75191938, -2.91119727, -7.59715504, -2.7920785],$$
$$[8.03450823, -17.2739244, -5.20571273, -6.70185079],$$
$$[-17.37872695, 18.30880768, 1.79869491, 6.2961463],$$
$$[6.72887358, 2.64431655, -7.31783452, 3.42364048]$$

which were generated randomly.

By observing the plots of Figure 3 and 4 we can see that both show the same behavior of terminating itself at the point (0, 0). Hence, we can say that the response of each mode is in fact damped.

The simulations are done using Python and the code is in Problem 2: Python Code.

References

[1] A. H. Nayfeh and D. T. Mook. *Nonlinear Oscillations.* Wiley Classics Library Edition, 1995.

Appendix

4.1 Problem 2: MATLAB Code

```
% ME 6444 HW3 Problem 2 MATLAB code
% Tomoki Koike
clear all; close all; clc;  % housekeeping commands
%%
% Declare and solve necessary conditions
syms gamma(epsilon) epsilon(v) v(x, t)
syms rho A E_1 E_2 alpha T_0 l

v_x = diff(v, x);
v_xx = diff(v_x, x);
v_t = diff(v, t);
v_tt = diff(v_t, t);
%%
epsilon = v_x^2/2
gamma = T_0 + E_1*A*epsilon + E_2*A*epsilon^2 + alpha*A*diff(epsilon, t)
v_eom = rho*A*v_tt - diff(gamma, x) * v_x - gamma * v_xx
%%
syms phi_1(x) phi_2(x) beta_1(t) beta_2(t) v_tilde(x, t)

phi_1(x) = sin(pi*x/l);
phi_2(x) = sin(2*pi*x/l);
v_tilde(x, t) = beta_1(t) * phi_1(x) + beta_2(t) * phi_2(x);
%%
gamma_tilde = subs(gamma, v, v_tilde)
%%
r_tilde(x, t) = subs(v_eom, [v, gamma], [v_tilde, gamma_tilde])
%%
EOM1 = int(r_tilde(x, t) * phi_1(x), x, 0, l) == 0
EOM2 = int(r_tilde(x, t) * phi_2(x), x, 0, l) == 0
```

4.2 Problem 2: Python Code

```
import matplotlib.pyplot as plt
import numpy as np
from scipy.integrate import solve_ivp, DOP853

```

```python
# Defining parameters
params = {
    'rho': 500,
    'A': 0.0001,
    'l': 10,
    'T0': 1500,
    'E1': 2.5e+6,
    'E2': 0.5e+6,
    'alpha': 1.2e+6
}

# System ODE
# x0 : beta1
# x1 : beta1_dot
# x2 : beta2
# x3 : beta2_dot
def nonlinear_string_ode(t, x, rho, A, l, T0, E1, E2, alpha):
    c11 = alpha*A/l**3 * (36.5284*x[0]**2 + 97.4091*x[2]**2)*x[1]
    c12 = 194.8182*alpha*A/l**3 * x[0] * x[2] * x[3]
    c13 = 4.9348*T0/l * x[0]
    c14 = E1*A/l**3 * x[0] * (18.2642 * x[0]**2 + 146.1136 * x[2]**2)
    c15 = E2*A/l**5 * x[0] * (75.1085 * x[0]**4 + 2.1030e+3 * x[0]**2 * x[2]**2 +
    ↪ 3.6052e+3 * x[2]**4)
    beta1ddot = -2/rho/A/l * (c11 + c12 + c13 + c14 + c15)

    c21 = alpha*A/l**3 * (97.4091*x[0]**2 + 584.4545*x[2]**2)*x[3]
    c22 = 194.8182*alpha*A/l**3 * x[0] * x[2] * x[1]
    c23 = 19.7392*T0/l * x[0]
    c24 = E1*A/l**3 * x[2] * (146.1136 * x[0]**2 + 292.2273 * x[2]**2)
    c25 = E2*A/l**5 * x[2] * (1.0515e+3 * x[0]**4 + 7.2104e+3 * x[0]**2 * x[2]**2 +
    ↪ 4.8069e+3 * x[2]**4)
    beta2ddot = -2/rho/A/l * (c21 + c22 + c23 + c24 + c25)

    return [x[1], beta1ddot, x[3], beta2ddot]


def solve_diffeq(func, t, tspan, ic, parameters={}, algorithm='DOP853',
↪ stepsize=np.inf):
    return solve_ivp(fun=func, t_span=tspan, t_eval=t, y0=ic, method=algorithm,
                     args=tuple(parameters.values()), atol=1e-8, rtol=1e-5,
                     ↪ max_step=stepsize)


def phasePlane(x1, x2, x3, x4, func, parameters):
```

```
    X1, X2 = np.meshgrid(x1, x2)  # create grid
    X3, X4 = np.meshgrid(x3, x4)  # create grid

    u1, v1 = np.zeros(X1.shape), np.zeros(X2.shape)
    u2, v2 = np.zeros(X3.shape), np.zeros(X4.shape)

    NI, NJ = X1.shape
    for i in range(NI):
        for j in range(NJ):
            x = X1[i, j]
            y = X2[i, j]
            xx = X3[i, j]
            yy = X4[i, j]
            dx = func(0, (x, y, xx, yy), *parameters.values())  # compute values on grid
            u1[i, j] = dx[0]
            v1[i, j] = dx[1]

            u2[i, j] = dx[2]
            v2[i, j] = dx[3]
    M1 = np.hypot(u1, v1)
    u1 /= M1
    v1 /= M1

    M2 = np.hypot(u2, v2)
    u2 /= M2
    v2 /= M2
    return X1, X2, u1, v1, M1, u2, v2, M2

# x0 = np.random.uniform(-20, 20, (1,4))
x0 = [
    [-9.75191938, -2.91119727, -7.59715504, -2.7920785 ],
    [8.03450823, -17.2739244,   -5.20571273,  -6.70185079],
    [-17.37872695,  18.30880768,   1.79869491,   6.2961463 ],
    [ 6.72887358,  2.64431655, -7.31783452,  3.42364048]
]

x1, x2 = np.meshgrid(np.linspace(-20, 20, 20), np.linspace(-170, 170, 20))
x3, x4 = np.meshgrid(np.linspace(-30, 30, 20), np.linspace(-100, 100, 20))

# Set up the figure the way we want it to look
fig, ax = plt.subplots(2, 2, figsize=(11, 9), constrained_layout=True)

# Phase plane
b1, b1d, b2, b2d = nonlinear_string_ode(2, [x1, x2, x3, x4], *params.values())
```

```
m1 = np.hypot(b1, b1d)
m2 = np.hypot(b2, b2d)

i = 0
t1 = np.linspace(0, 200, 100000)
t_span1 = (np.min(t1), np.max(t1))
for x0i in x0:
    ax[i//2, i%2].quiver(x1, x2, b1/m1, b1d/m1, m1, scale=None, pivot='mid')
    sol1 = solve_diffeq(nonlinear_string_ode, t1, t_span1, x0i, params)
    ax[i//2, i%2].plot(sol1.y[0, :], sol1.y[1, :], '-')
    ax[i//2, i%2].plot(0, 0, '.r', ms=15)
    ax[i//2, i%2].set_title(r'$x_0$='+str(x0i))
    i+=1

fig.supxlabel(r'$\beta_1$')
fig.supylabel(r'$\dot{\beta}_1$')
plt.savefig('../plots/nl_string_mode1_phase.png')
plt.show()

fig, ax = plt.subplots(2, 2, figsize=(11, 9), constrained_layout=True)
i = 0
t1 = np.linspace(0, 200, 100000)
t_span1 = (np.min(t1), np.max(t1))
for x0i in x0:
    ax[i//2, i%2].quiver(x3, x4, b2/m2, b2d/m2, m2, scale=None, pivot='mid')
    sol1 = solve_diffeq(nonlinear_string_ode, t1, t_span1, x0i, params)
    ax[i//2, i%2].plot(sol1.y[2, :], sol1.y[3, :], '-')
    ax[i//2, i%2].plot(0, 0, '.r', ms=15)
    ax[i//2, i%2].set_title(r'$x_0$='+str(x0i))
    i+=1

fig.supxlabel(r'$\beta_2$')
fig.supylabel(r'$\dot{\beta}_2$')
plt.savefig('../plots/nl_string_mode2_phase.png')
plt.show()
```