ABSTRACT

Title of Thesis: ENERGY SINKS WITH NONLINEAR STIFFNESS AND NONLINEAR DAMPING

Matthew Charles Colvin, Master of Science, 2010

Thesis directed by: Professor Amr Baz
Department of Mechanical Engineering

Nonlinear energy sinks (NES) have been shown to be effective for vibration attenuation under certain conditions. The effects of coupling linear systems to NES with nonlinear stiffness and linear damping have been extensively investigated. However, research involving nonlinearly damped NES is minimal. In this thesis, the performance of nonlinearly damped NES are examined and comparisons are made with the performance of linearly damped counterparts. Saddle-node and Hopf bifurcation diagrams are presented to show instabilities of solutions. The strongly modulated response (SMR), which exists only near 1:1 resonance between the forcing frequency and the frequency of the linear system, is investigated as well. Finally, time response comparisons are made between the systems with linearly and nonlinearly damped NES.

ENERGY SINKS WITH NONLINEAR STIFFNESS AND NONLINEAR DAMPING

by

Matthew Charles Colvin

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Master of Science
2010

Advisory Committee:

Professor Amr Baz, Chair
Professor Balakumar Balachandran
Professor Nikhil Chopra

To my parents, William and Linda

TABLE OF CONTENTS

1.0 Introduction

1.1 Overview

Mechanical vibration has undesirable effects in many engineering applications including machine tool operation, earthquake isolation of buildings, electronic packaging, and aerospace structures, to name a few. Linear isolation methods have been developed to help mitigate these problems. However, recent advancements show that, depending on the application, nonlinear energy sinks (NES) can be far more effective in vibration attenuation than linear absorbers. In fact, nonlinear targeted energy transfer (TET), the main motive for attaching a NES, was "first observed by Gendelman (2001) who studied the transient dynamics of a two-DOF system consisting of a damped linear oscillator that was weakly coupled to an essentially (strongly) nonlinear, damped attachment" (Vakakis et al., 2008). Clearly, the study of NES is still in its infancy, but the concept has already been shown effective in vibration mitigation under certain conditions. Fundamentally, a NES is nothing more than a mass attached to a primary system with a nonlinear spring and a linear or nonlinear damper.

This chapter begins by presenting an overview of earlier works related to NES. General comparisons between classical linear vibration absorbers and NES are then discussed, followed by a summary of related experiments. Finally, an overview of the topics discussed in this thesis is presented at the end of the chapter.

1.2 Summary of Related Literature

Targeted energy transfer (TET) can be broadly defined as the case when "energy of some form is directed from a source (donor) to a receiver (recipient) in a one-way irreversible

fashion" (Vakakis et al., 2008). In the case of a nonlinear energy sink (NES) attached to a linear system, TET refers to the irreversible transfer of energy from the linear system to the NES, hence the name "sink". This process of TET is also referred to as energy pumping.

Energy pumping from an impulsively loaded linear system to a strongly nonlinear (incapable of being linearized) attachment was investigated by Vakakis (2001) by assuming the linear system to be a chain of elastically interconnected particles. Initially, Vakakis introduced energy pumping concepts by application to an impulsively loaded two-degree of freedom system given by

$$\begin{aligned} &\ddot{y}_1 + \varepsilon\lambda\dot{y}_1 + C{y_1}^3 + \varepsilon(y_1 - y_2) = 0, \qquad y_1(0) = \dot{y}_1(0) = 0 \\ &\ddot{y}_2 + \varepsilon\lambda\dot{y}_2 + {\omega_2}^2 y_2 - \varepsilon(y_1 - y_2) = F\delta(t), \; y_2(0) = \dot{y}_2(0) = 0. \end{aligned} \tag{1.1}$$

Note that y_1 and y_2 correspond to displacements of the nonlinear attachment and primary system, respectively. The system was simulated numerically for illustration of energy pumping by assigning $\lambda = 0.5$, ${\omega_2}^2 = 0.9$, $C = 5.0$, and $\varepsilon = 0.1$. The different responses for varying impulse magnitudes of $F = 1.0$, $F = 1.26$, and $F = 1.50$ are presented in Figure 1-1. From inspection of each of the plots in Figure 1-1, it is evident that increasing values of the impulse magnitude results in reduced displacements of the primary system. From Figure 1-1b and Figure 1-1c, it is evident that energy pumping has occurred since the displacement of the nonlinear attachment has increased relative to the displacement of the primary system. Additionally, the displacement of the primary system is shown to decay faster that the displacement of the nonlinear attachment in the presence of energy pumping. In other words, energy has been transferred irreversibly from the primary system to the nonlinear attachment.

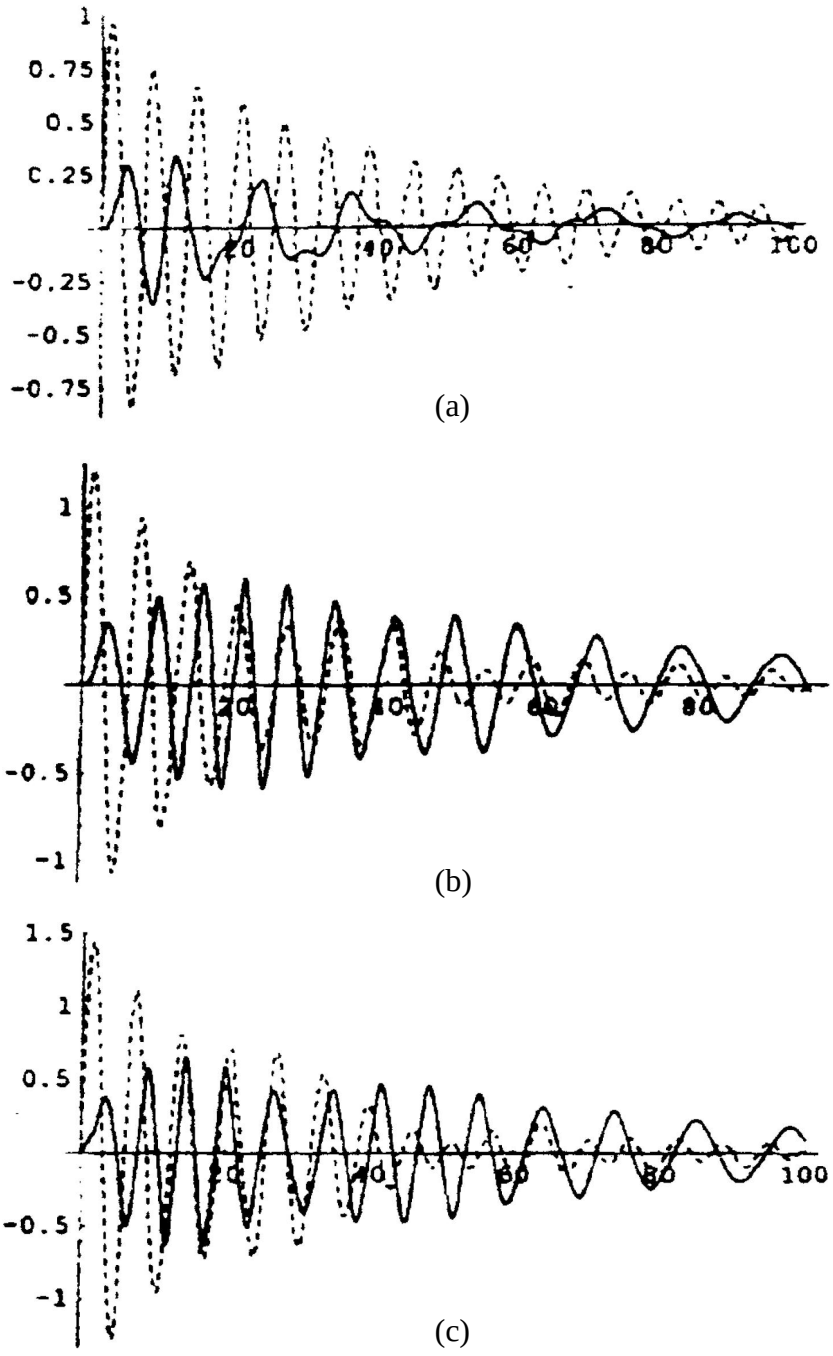


Figure 1-1. Time response plots for (a) $F = 1.0$, (b) $F = 1.26$, and (c) $F = 1.5$. Primary System: - - - - - -, Nonlinear Attachment: ————— (Vakakis, 2001 and Gendelman et al., 2001)

The concepts from Vakakis (2001) are then extended to a more complex system, consisting of a linear chain with 101 particles and a strongly nonlinear attachment at one end, as shown in Figure 1-2. The system is impulsively loaded on the fourth particle of the linear chain, thus the equations of motion with initial conditions are

$$\begin{gathered}
\ddot{y}_1 + \varepsilon\lambda\dot{y}_1 + C{y_1}^3 + \varepsilon(y_1 - y_2) = 0 \\
\ddot{y}_2 + \varepsilon\lambda\dot{y}_2 + {\omega_2}^2 y_2 - \varepsilon(y_1 - y_2) + d(y_2 - y_3) = 0 \\
\ddot{y}_i + \varepsilon\lambda\dot{y}_i + {\omega_2}^2 y_i + d(2y_i - y_{i-1} - y_{i+1}) = 0, \;\; i \in [3, +\infty) \\
y_i(0) = \dot{y}_i(0) = 0, \qquad y_p(0) = 0, \;\; \dot{y}_p(0) = F_p, \qquad i \neq p.
\end{gathered} \tag{1.2}$$

Vakakis (2001) graphically presented the numerical results for varying values of the linear coupling stiffness between particles, *d*, and for different values of the grounding stiffness parameter, ${\omega_2}^2$. The case without energy pumping is shown in Figure 1-3, in which the nonlinear attachment displacement decays at approximately the same time as the linear system. In contrast, energy pumping does occur in Figure 1-4. By decreasing the linear oscillator grounding stiffness to 0.4, the response decays faster than in the previous case with ${\omega_2}^2$ = 0.9. In addition, the response of the nonlinear attachment decays after the response of the linear system, indicating that there has been an irreversible transfer of energy to the attachment. Keeping ${\omega_2}^2$ = 0.4, the coupling stiffness between particles is increased to 3.5 for the responses shown in Figure 1-5. The results shown in Figure 1-5 are very close to those shown in Figure 1-4, again indicating that energy pumping has occurred. Vakakis (2001) then showed that the analysis of energy pumping in the multi-degree of freedom system could be reduced to an analysis of a two-degree of freedom system due to in-phase vibration of the chain of particles.

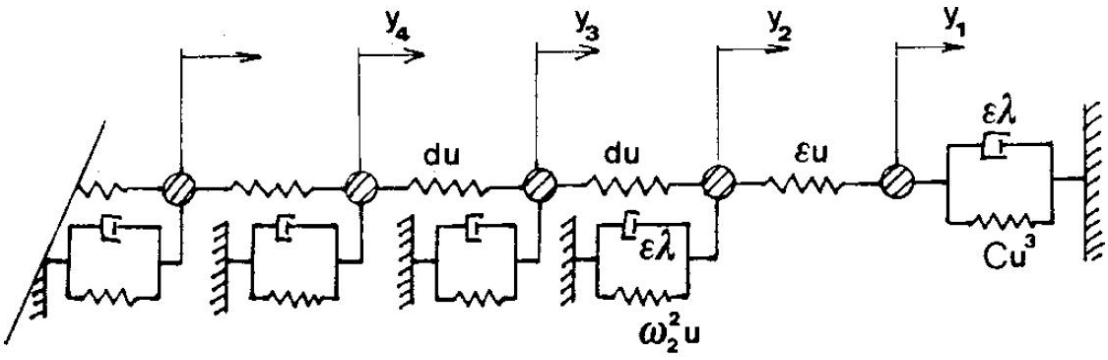
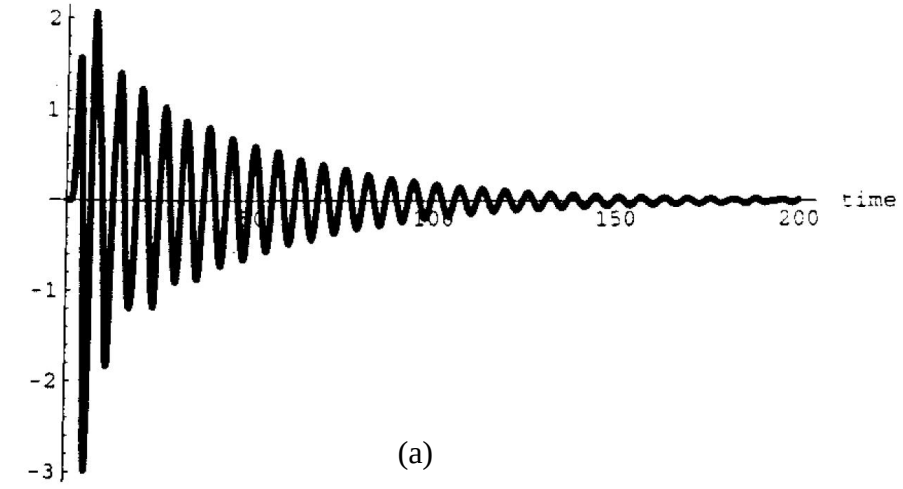
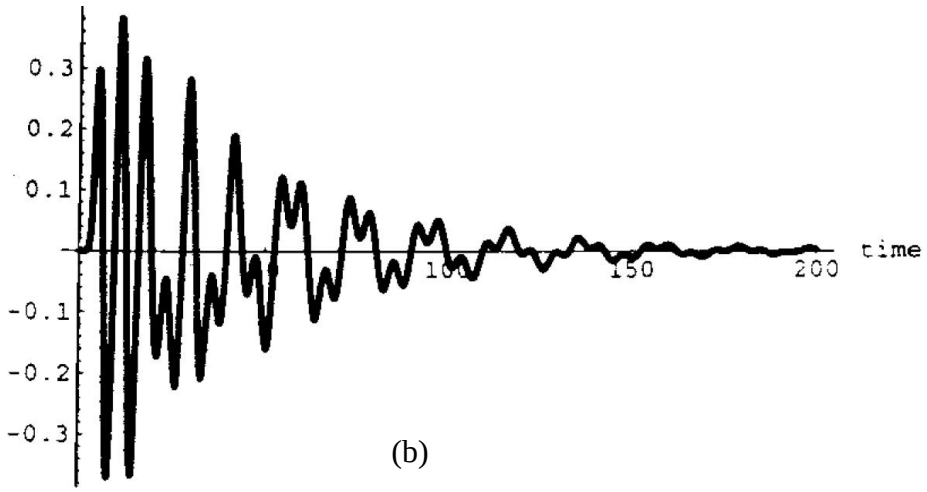


Figure 1-2. Depiction of system in equations (1.2); (Vakakis, 2001)



(a)



(b)

Figure 1-3. Time response plots for (a) the particle of the linear system adjacent to the nonlinear attachment and (b) the nonlinear attachment. Parameters are $\varepsilon = 0.1$, $\lambda = 0.5$, $C = 5.0$, $d = 3.5$, and $\omega_2^2 = 0.9$ (Vakakis, 2001).

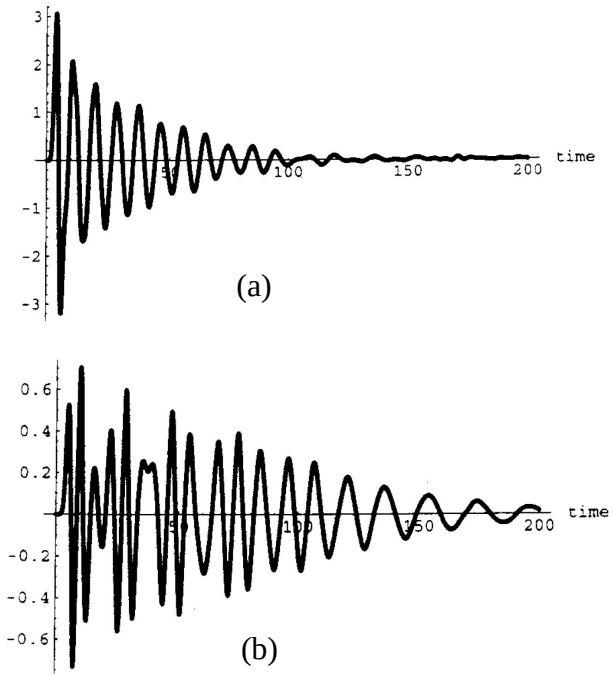


Figure 1-4. Time response plots for (a) the particle of the linear system adjacent to the nonlinear attachment and (b) the nonlinear attachment. Parameters are $\varepsilon = 0.1$, $\lambda = 0.5$, $C = 5.0$, $d = 1.5$, and $\omega_2^2 = 0.4$ (Vakakis, 2001).

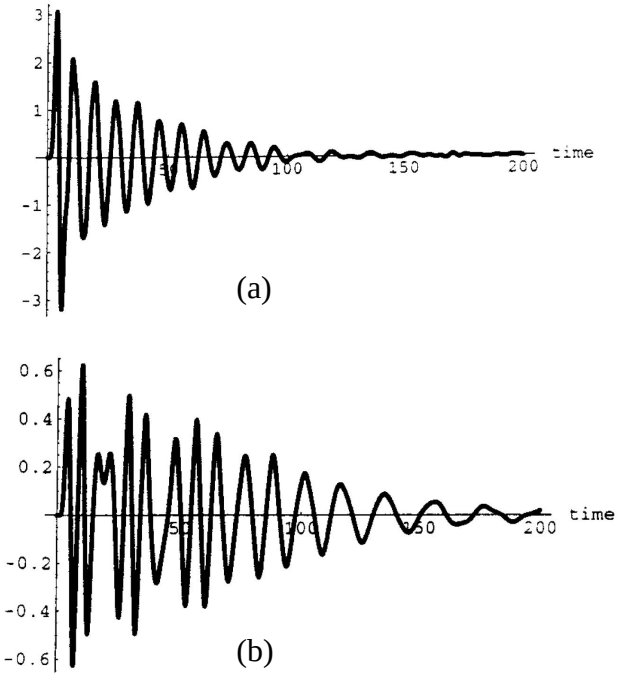


Figure 1-5. Time response plots for (a) the particle of the linear system adjacent to the nonlinear attachment and (b) the nonlinear attachment. Parameters are $\varepsilon = 0.1$, $\lambda = 0.5$, $C = 5.0$, $d = 3.5$, and $\omega_2^2 = 0.4$ (Vakakis, 2001).

The effects of attaching a NES to a linear system with (*N+1*) degrees of freedom by means of a spring with low stiffness are studied by Vakakis et al. (2003). Refer to Figure 1-6 for a schematic of the system. In their study, Vakakis et al. (2003) focus on the nonlinear normal modes (NNMs) of a conservative system in order to explain the dynamics of the same system with damping. The NNMs are defined as "the free periodic and synchronous oscillations of the undamped, unforced system, that are, in essence, the non-linear analogs of the linear modes of classical vibration theory" (Vakakis et al., 2003). In other words, the NNMs define the shape of the vibrating nonlinear system at specific frequencies. In addition, it was shown that if damping and impulse forcing are included in the system, the NES can vibrate at a different frequency than the linear substructure at any given instant prior to reaching steady-state oscillations. During this transient period, provided the external forcing is high enough, energy pumping can occur. Energy pumping in multi-degree of freedom systems is due to resonance capture cascades, defined as "a sequence of multiple resonance interactions of the non-linear attachment with more than one modal oscillators of the linear substructure" (Vakakis et al., 2003). When multiple modes of the system are excited during transient vibrations, the attachment is able to resonate with these different modes due to its nonlinearity. In contrast, a linear vibration absorber only has one frequency with which it can resonate, greatly limiting the design options compared to those available when designing a NES.

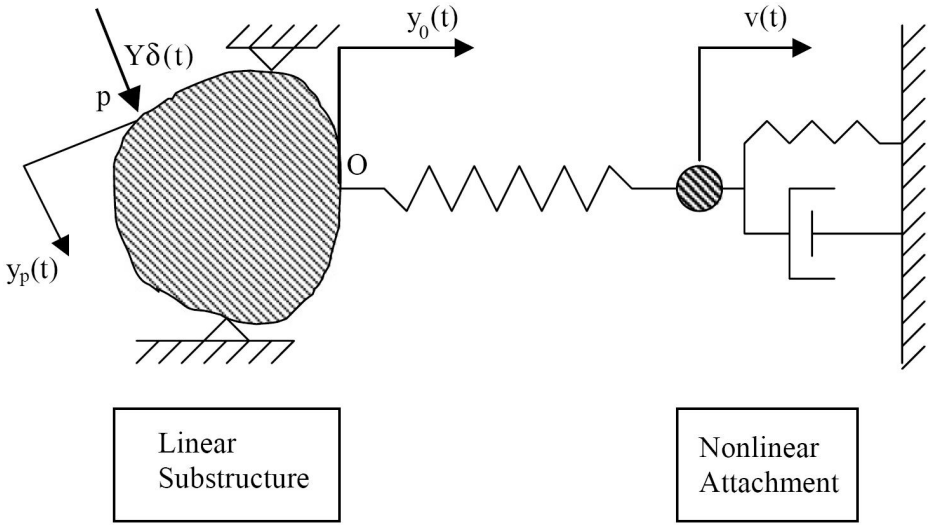


Figure 1-6. Depiction of system studied by Vakakis et al. (2003)

Although similar to the resonance capture cascades of multi-degree of freedom linear systems with NES, resonance capture in single-degree of freedom systems omits the word "cascades" because there is only one linear natural frequency with which to resonate. In order to understand resonance capture, the concept of *resonance manifold* must first be grasped. Vakakis et al. (2008) reference Sanders and Verhulst (1985) by introducing the following damped nonlinear "system in polar form with multiple phase angles":

$$I' = \varepsilon R(\phi, I),\ I \in R^{+p}$$
$$\phi' = \omega(I),\ \phi \in T^q, \tag{1.3}$$

where I "represents energy-like amplitudes" (Vakakis et al., 2008), and ϕ is a "vector of angles" (Vakakis et al., 2008). The vector I has length p, and the vector ϕ has length q. The set of p positive real numbers is represented by R^{+p}, and the q-torus, or the torus requiring q "angular coordinates to describe the motion" (Nayfeh and Balachandran,

1995), is represented by T^q. The frequency vector is given by

$$\omega(I) = \left[\omega_1(I), \omega_2(I), \ldots, \omega_q(I)\right]^T. \tag{1.4}$$

Vakakis et al. (2008) define a *resonance manifold* as "the set of points in $D \subset R^p$ where $\omega_i(I) \equiv 0, i = 1,2, \ldots q$". In other words, a resonance manifold is the set of points in D for which all frequencies are identical to zero. Resonance capture can be divided into two types: *Transient Resonance Capture* (*TRC*) and *Sustained Resonance Capture* (*SRC*). Vakakis et al. (2008) define TRC "as capture into a resonance manifold which occurs and continues for a certain period of time, followed by a transition to escape from capture". On the other hand, SRC is "defined as resonance capture that will never escape with increasing time" (Vakakis et al., 2008). When observing a phase portrait, resonance capture occurs when the trajectory of the system becomes attracted and locked to the resonance manifold. In the case of a single-degree of freedom linear system coupled to a NES, significant amounts of energy can be exchanged between the two oscillators during resonance capture as illustrated by the concept of targeted energy transfer.

The effect of resonance capture on targeted energy transfer (TET) is illustrated in an example by Lee et al. (2008) of an impulsively loaded system. The system,

$$\begin{aligned} \ddot{x} + {\omega_0}^2 x + \lambda_1 \dot{x} + \lambda_2(\dot{x} - \dot{v}) + C(x - v)^3 = 0 \\ \epsilon \ddot{v} + \lambda_2(\dot{v} - \dot{x}) + C(v - x)^3 = 0, \end{aligned} \tag{1.5}$$

is given parameters $\omega_0^2 = k_1/m_1 = 1$, $C = k_2/m_2 = 1$, $\epsilon = m_2/m_1 = 0.05$, $\lambda_1 = c_1/m_1 = 0.0015$, $\lambda_2 = c_2/m_2 = 0.0015$. Lee et al. (2008) discuss three different ways in which targeted energy transfer (TET) can occur: *fundamental TET*, *subharmonic TET*, and *TET initiated by non-linear beating*. In the case of fundamental TET, "the linear oscillator and the non-linear attachment oscillate with identical frequencies in the neighbourhood of the

fundamental frequency ω_0" (Lee et al., 2008). Figure 1-7 presents percentages of energy transfer to and energy dissipation by the NES. As seen from Figure 1-7, almost all of the energy is eventually transferred to the NES. Corresponding to the trend of energy transfer to the NES is the trend of energy dissipated by the NES. From these two plots, it can be seen that after some initial transients, the NES effectively dissipates over 70% of the total energy in the system. For illustration of the conditions at which this energy pumping occurs, Figure 1-8 presents frequency-energy plots versus the total system energy in the form of wavelet transforms (WTs). The darker regions represent high amplitudes of the WT, and the lighter regions depict the opposite. Refer to Vakakis et al. (2008) for detailed discussions on the use of WTs. As seen from the contour plots, as the energy in the system increases, the frequency of the linear system and NES tend to unity, indicating that 1:1 resonance capture is responsible for this type of TET.

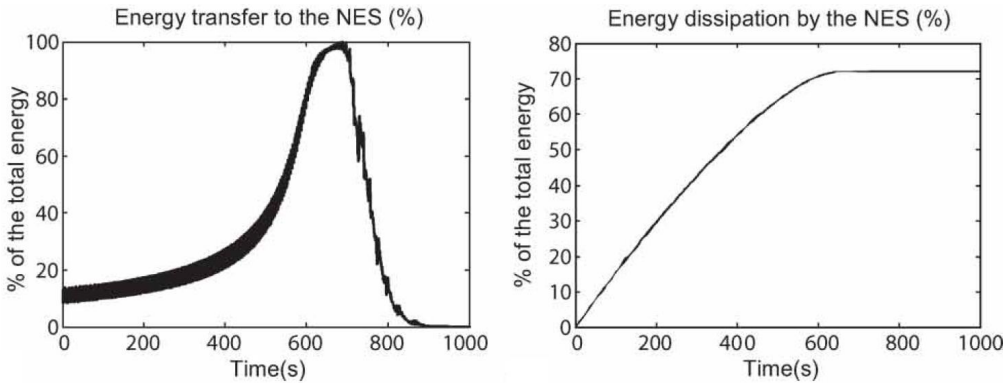


Figure 1-7. *Fundamental TET*. Percentage of the total energy versus time for energy transfer to the NES and energy dissipation by the NES (Lee et al., 2008).

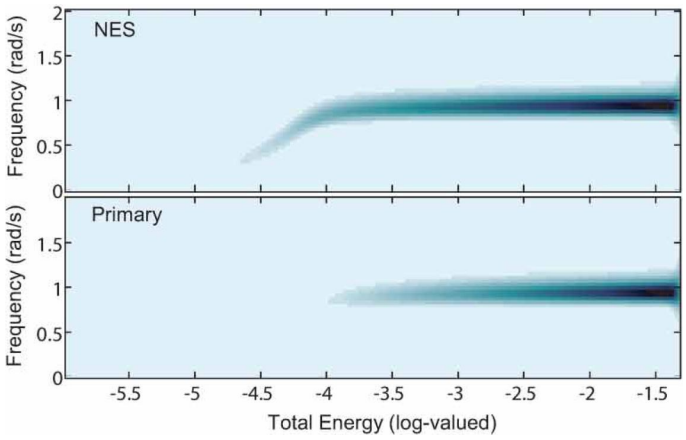


Figure 1-8. *Fundamental TET*. Contour plots of WTs depicting the frequency-energy dependence of the primary system and the NES (Lee et al., 2008).

The second way in which TET can occur is through subharmonic TET. Subharmonic TET refers to exciting "families of NNMs of the underlying Hamiltonian system with the nonlinear attachment engaging in *m*:*n* internal resonance with the linear oscillator (LO) (where *m*, *n* are integers with $m < n$)" (Vakakis et al., 2008). A Hamiltonian system is defined by the following relations (Nayfeh and Balachandran, 1995):

$$\dot{q}_i = \frac{\partial H}{\partial p_i}, \quad \dot{p}_i = -\frac{\partial H}{\partial q_i}, \quad i = 1,2,\dots,n, \quad H = H(q_1, q_2, \dots q_n, p_1, p_2, \dots p_n, t)\,. \qquad (1.6)$$

Internal resonance refers to the condition where the linear natural frequencies, ω_i, of a system are integer multiples of each other. In other words, "there exist positive or negative integers m_1, m_2, m_3, …, m_n such that $m_1\omega_1 + m_2\omega_2 + m_3\omega_3 + \dots + m_n\omega_n \approx 0$" (Nayfeh and Mook, 1995). Lee et al. (2008) continue the numerical simulation by showing results for the case of subharmonic TET with a 1:3 resonance capture, that is, the primary system "oscillates with a frequency approximately three times that of the NES" (Vakakis et al., 2008). Figure 1-9 presents percentages of energy transfer to and energy dissipation by the NES. From inspection of the plot of energy transfer to the

NES, approximately 70% of the total system energy is transferred to the NES. As opposed to the case of fundamental TET, Figure 1-10 shows the frequency localized to approximately 1/3 for the NES and unity for the primary system.

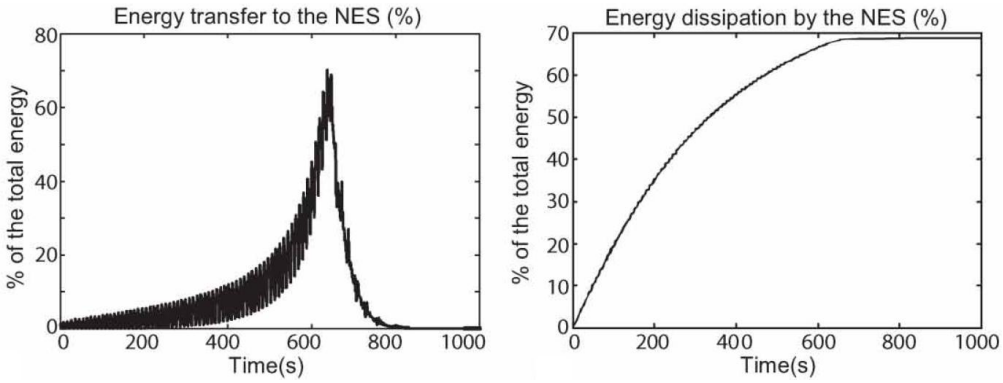


Figure 1-9. *Subharmonic TET*. Percentage of the total energy versus time for energy transfer to the NES and energy dissipation by the NES (Lee et al., 2008).

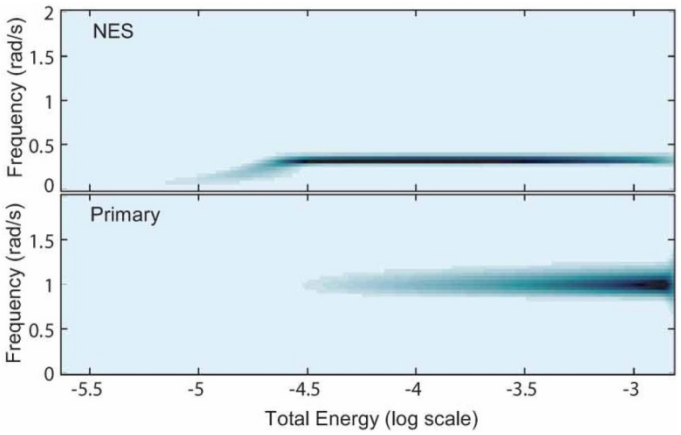


Figure 1-10. *Subharmonic TET*. Contour plots of WTs depicting the frequency-energy dependence of the primary system and the NES (Lee et al., 2008).

In order for fundamental TET and subharmonic TET to occur in an impulsively loaded system, TET must be initiated by nonlinear beating. As reasoned by Lee et al. (2008), fundamental TET and subharmonic TET "cannot be activated with the NES at rest, since in both cases the motion is initialized from a non-localized state of the

system". Figure 1-11 illustrates one case of the energy exchanges during nonlinear beat phenomena in an undamped system. As seen from the plots, the energy in the primary system is a mirror image (about a horizontal axis at 50%) of the energy in the NES, indicating a direct energy exchange between the two. As this energy transfer is clearly reversible, introduction of damping in the system is required for the exchange to be final. With damping in the system and after establishing in the response "an initial non-linear beat phenomenon, either one of the main (fundamental or subharmonic) TET mechanisms can be activated by a non-linear transition (jump) in the dynamics" (Lee et al., 2008). By inspection of the percentage of total energy in the NES as shown in Figure 1-12, the nonlinear beat phenomena is seen early in the motion as the energy percentage fluctuates greatly before converging. Once the nonlinear beating has subsided, fundamental TET increases energy in the NES to nearly 100% of the total system energy. As evident from Figure 1-13, the 1:1 resonance (i.e. fundamental TET) is the means of energy pumping in this case.

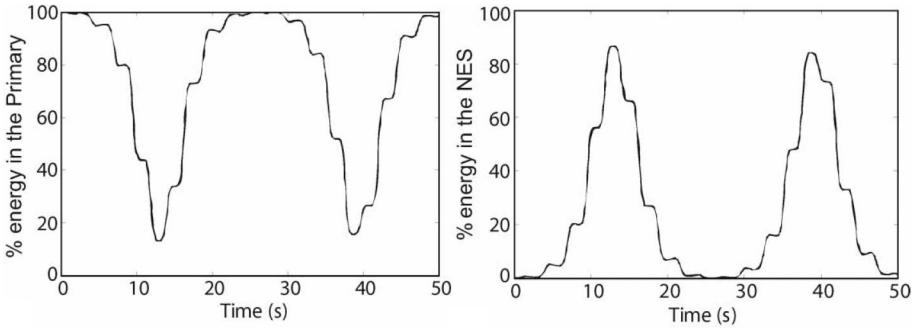


Figure 1-11. Example of energy exchanges due to nonlinear beat phenomena (Lee et al., 2008).

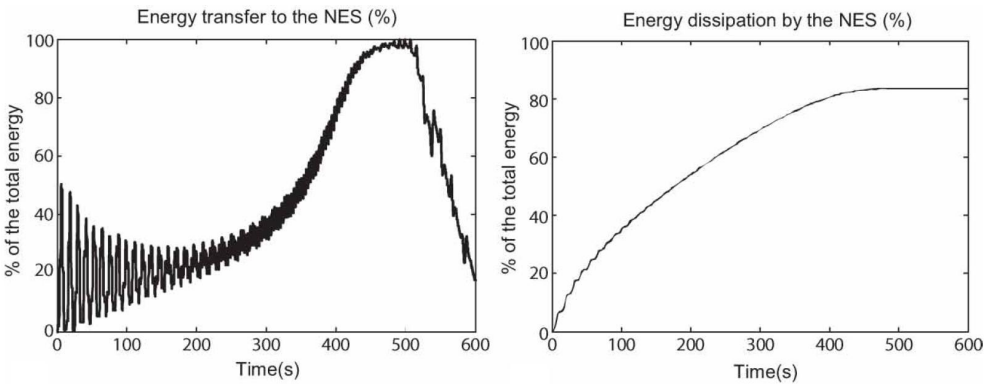


Figure 1-12. *Initiating TET*. Percentage of the total energy versus time for energy transfer to the NES and energy dissipation by the NES (Lee et al., 2008).

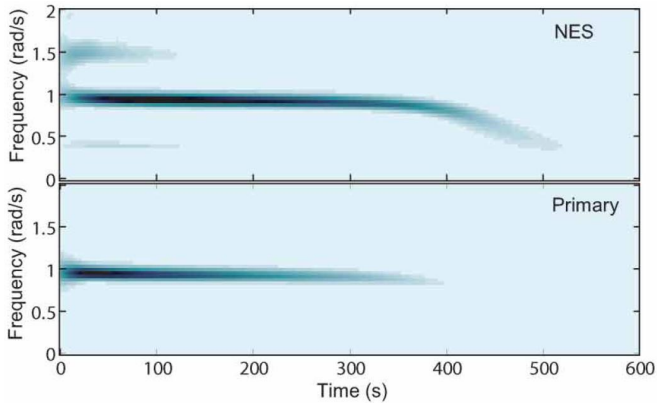


Figure 1-13. *Initiating TET*. Contour plots of WTs depicting the frequency-energy dependence of the primary system and the NES (Lee et al., 2008).

Of particular interest to this thesis is the work presented in Starosvetsky and Gendelman (2008a) that involved studying the system

$$\begin{aligned} \ddot{y}_1 + \varepsilon\lambda(\dot{y}_1 - \dot{y}_2) + (1+\varepsilon\sigma)y_1 + \tfrac{4}{3}\varepsilon(y_1 - y_2)^3 &= \varepsilon A\cos t \\ \varepsilon\ddot{y}_2 + \varepsilon\lambda(\dot{y}_2 - \dot{y}_1) + \frac{4}{3}\varepsilon(y_2 - y_1)^3 &= 0 \end{aligned} \tag{1.7}$$

which is represented schematically in Figure 1-14. From the figure, it is evident that $\varepsilon\lambda$ is the damping coefficient, $(1+\varepsilon\sigma)$ is the linear spring stiffness, and $4\varepsilon/3$ is the nonlinear spring stiffness. In addition, the force F shown in the diagram represents the harmonic force with amplitude εA. As shown in Figure 1-15, saddle-node bifurcation diagrams are

presented in the λ-A plane to show regions of one versus three periodic solutions for different values of the detuning parameter, σ, while keeping the other parameters fixed. For each chosen value of σ, regions of one periodic solution are located outside the curves, while regions of three periodic solutions are bounded inside the curves. Figure 1-16 shows bifurcation diagrams used again, now in the case of Hopf bifurcations to show regions of stability and instability. Starosvetsky and Gendelman (2008a) showed also a method of predicting the occurrence of the Strongly Modulated Response (SMR), which differs from other methods shown in earlier works. The SMR refers to quasiperiodic response, that "is characterized by very deep oscillations of the modulated amplitude comparable to the amplitude of the response itself" (Starosvetsky and Gendelman, 2008a). The response is considered modulated because the "response may be also phase locked or chaotic". Finally, numerical simulations are used to validate the results of the analysis. Notably, Starosvetsky and Gendelman (2008a) showed that two to three different responses can coexist for the studied system.

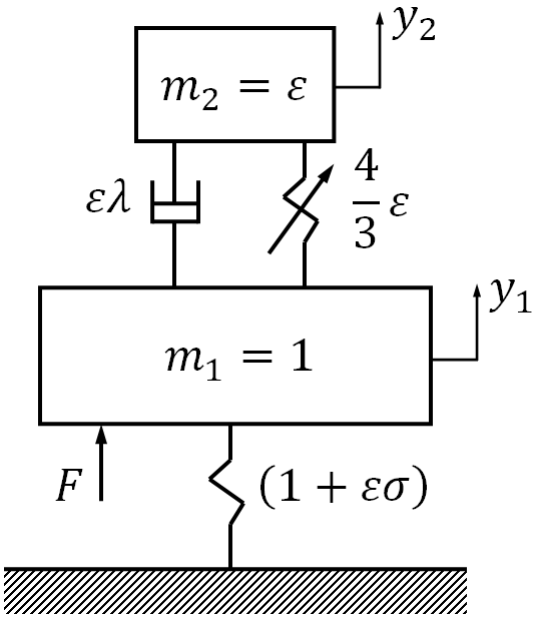


Figure 1-14. System considered by Starosvetsky and Gendelman (2008a)

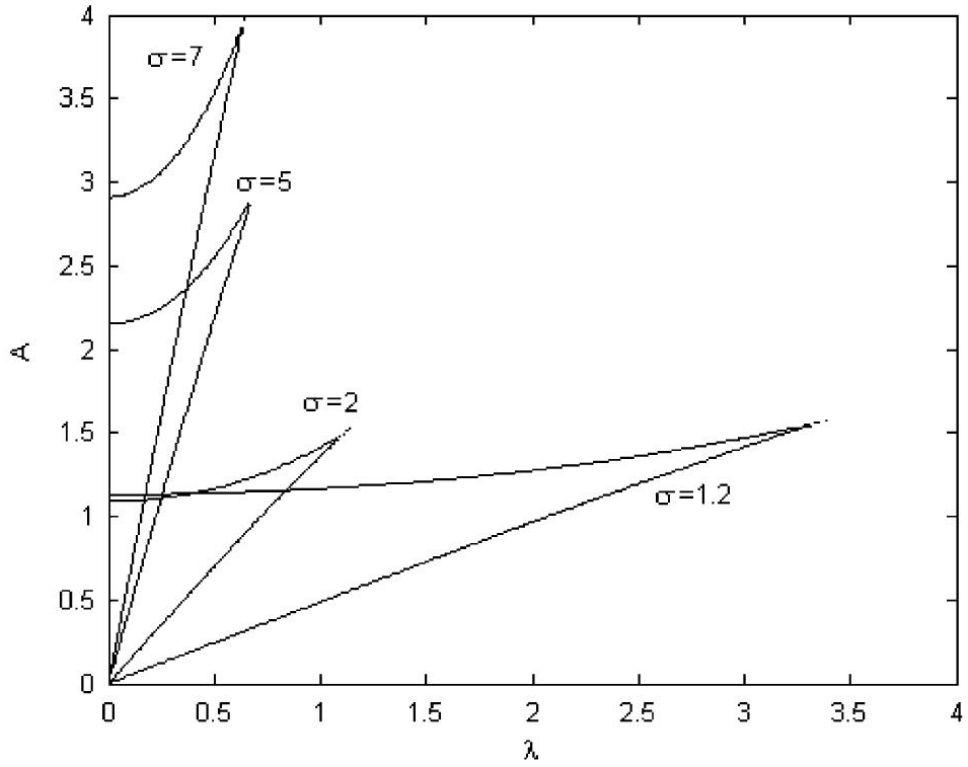


Figure 1-15. Saddle-node bifurcation diagram (Starosvetsky and Gendelman, 2008a)

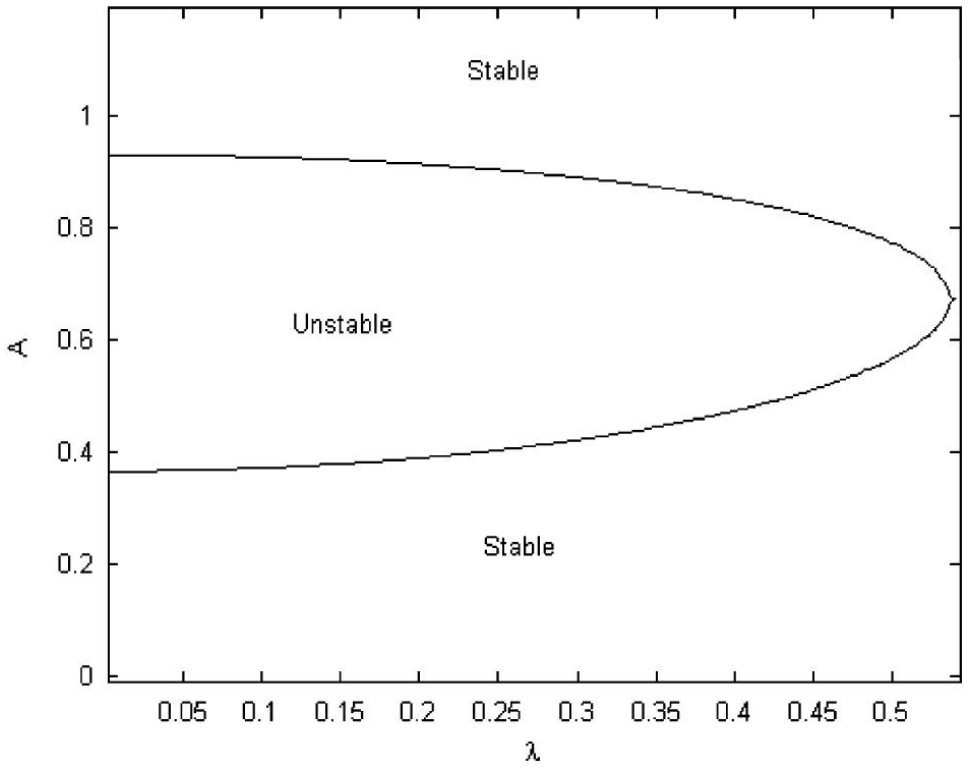


Figure 1-16. Hopf bifurcation diagram (Starosvetsky and Gendelman, 2008a)

Starosvetsky and Gendelman (2008b) have presented detailed discussions on the analysis of the Strongly Modulated Response (SMR) by again considering the system given by equation (1.7). The SMR "exists in a vicinity of exact 1:1 resonance and is characterized by relaxation oscillations between stable branches of the slow invariant manifold" Starosvetsky and Gendelman (2008b). The slow invariant manifold (SIM) is defined by an equation found through the analysis relating the fixed points of the system, Φ, to the slow time scale, τ_1; in this case,

$$\frac{i}{2}\Phi + \frac{\lambda}{2}\Phi - \frac{i}{2}|\Phi^2|\Phi = C(\tau_1). \tag{1.8}$$

Note that $C(\tau_1)$ is simply a consequence of an integration. The relaxation oscillations refer to jumps from one stable branch to another on the SIM. In order for the SMR to occur, the system must be essentially nonlinear and possess mass asymmetry. In other words, the system cannot be reduced to a linear form, and the mass of the attachment must be much less than the mass of the primary system. Letting the ratio of the mass of the attachment to the mass of the primary system be denoted by ε, the condition for this analysis assumes that $\varepsilon << 1$. Beginning with the system given by equation (1.7), the system equations are manipulated such that they are placed in a form suitable for studying the SMR, namely, separation into the slow and fast time scales. Phase portraits are presented for varying amplitudes of the external force, showing how the trajectory can transition from one stable branch to another. From these phase portraits, a one-dimensional mapping procedure was discussed. The one-dimensional maps provide a method of determining regions of existence of the SMR in terms of the detuning parameter by observation of the existence of attractors on the maps. Since the detuning parameter is directly related to the natural frequency of the linear system, the conclusion

of the SMR existing only near 1:1 resonance was determined in this manner. Further details on the SMR will be discussed in Chapter 4 of this thesis.

Inclusion of nonlinear damping in a NES attached to a single-degree of freedom linear system is discussed by Starosvetsky and Gendelman (2009). The system studied,

$$\begin{aligned} \ddot{y_1} + \varepsilon f(y_1 - y_2, \dot{y_1} - \dot{y_2}) + y_1 + \tfrac{4}{3}\varepsilon(y_1 - y_2)^3 &= \varepsilon A\cos\big((1+\varepsilon\sigma)t\big) \\ \varepsilon\ddot{y_2} - \varepsilon f(y_1 - y_2, \dot{y_1} - \dot{y_2}) - \tfrac{4}{3}\varepsilon(y_2 - y_1)^3 &= 0, \end{aligned} \tag{1.9}$$

regards f as the component possessing the nonlinear damping characteristics. The nonlinear damping in the system is due to the drag of a fluid through a variable orifice. See Figure 1-17 for a schematic of the type of device considered. The nonlinearity in the system is "piecewise-quadratic" since the damping coefficient can be one of two values depending on the configuration of the flow allowed through the device. The damping coefficients are denoted by λ_1 and λ_2, in which λ_2 is greater than λ_1. Thus, for the system considered, the damping function is given by

$$f = \begin{cases} \lambda_1(\dot{y_1} - \dot{y_2})|\dot{y_1} - \dot{y_2}|, & |y_1 - y_2| < a_{cr} \\ \lambda_2(\dot{y_1} - \dot{y_2})|\dot{y_1} - \dot{y_2}|, & |y_1 - y_2| > a_{cr} \end{cases}, \tag{1.10}$$

where a_{cr} is a predetermined value inherent in the design of the nonlinear damping device.

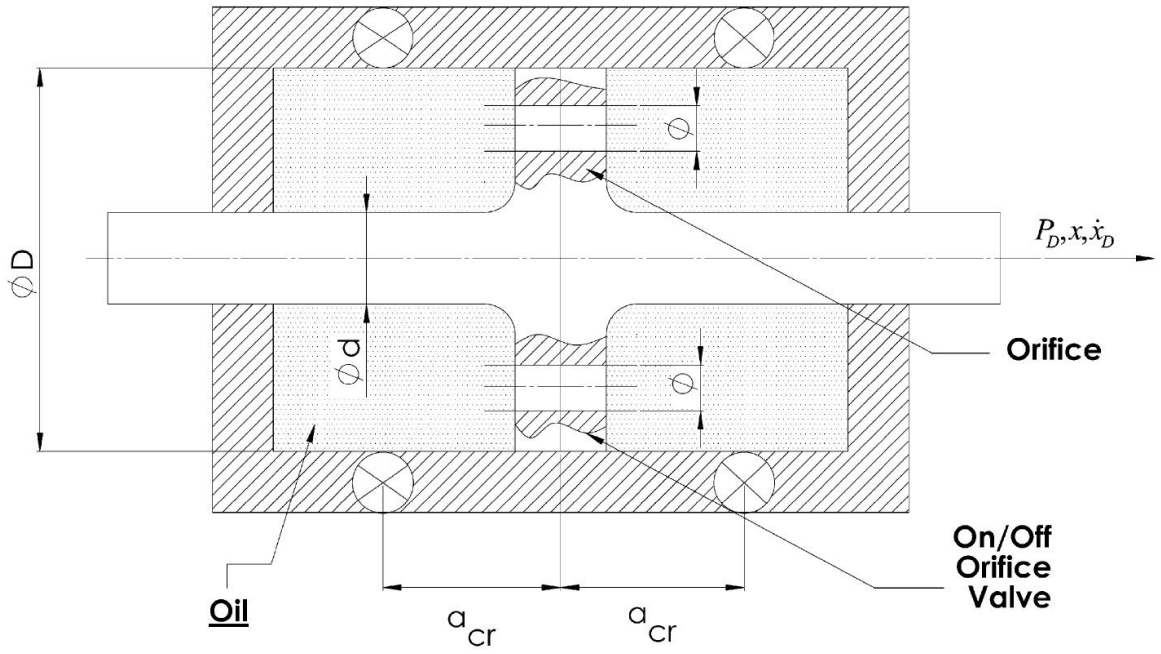


Figure 1-17. Schematic of the nonlinear damper
(Starosvetsky and Gendelman, 2009)

The main goal of Starosvetsky and Gendelman (2009) was to show that the inclusion of nonlinear damping in the NES can remove unwanted responses inherent with a linearly damped NES coupled to a harmonically-forced linear oscillator. Figure 1-18 depicts a linear system coupled to a NES with linear damping. As seen from Figure 1-18a, an undesired response exists while the SMR (desired) also exists. By increasing the linear damping coefficient as shown in Figure 1-18b, the undesired response is eliminated, but the SMR disappears as well. Starosvetsky and Gendelman (2009) demonstrate in Figure 1-19 how nonlinear damping can resolve this problem. By choosing certain values for λ_1 and λ_2, the SMR can exist without the undesired response.

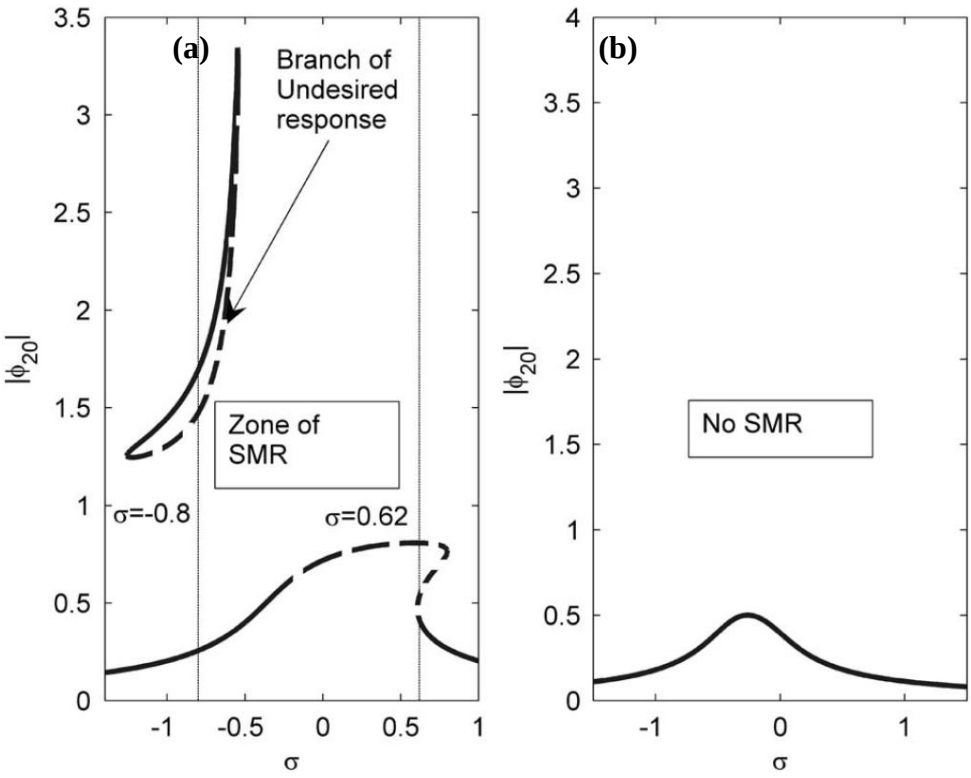


Figure 1-18. *Linear Damping*. (a) $\lambda = 0.2$, (b) $\lambda = 1$ (Starosvetsky and Gendelman, 2009).

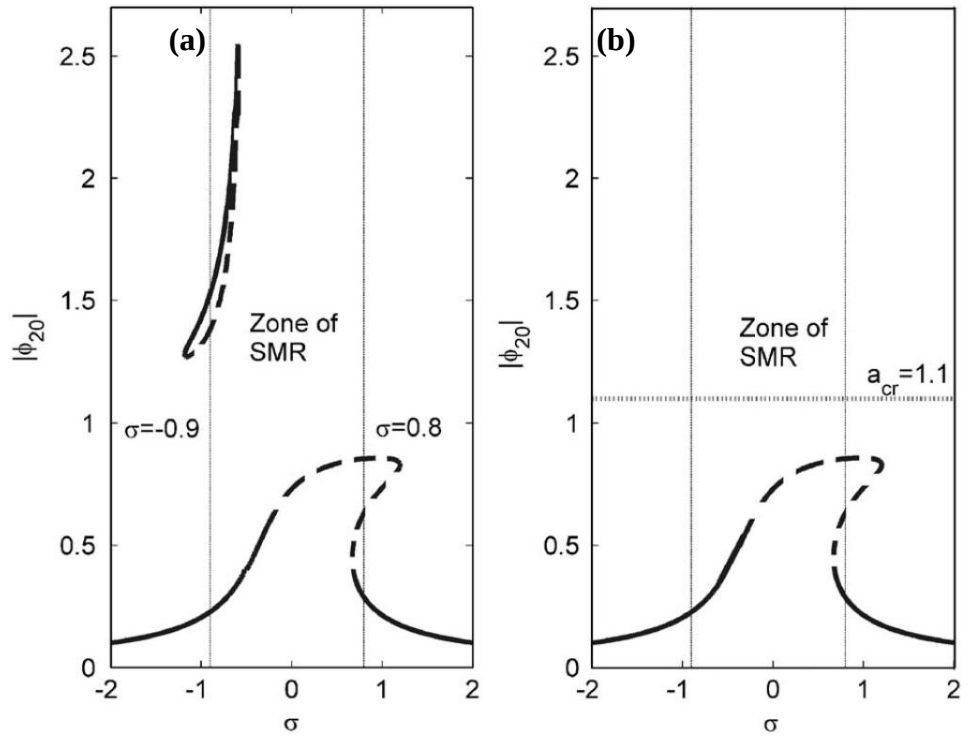


Figure 1-19. *Linear versus nonlinear (piecewise-quadratic) damping*. (a) $\lambda_1 = \lambda_1 = 0.2$, (b) $\lambda_1 = 0.2$ and $\lambda_2 = 6$ (Starosvetsky and Gendelman, 2009).

1.3 Comparison of Nonlinear Energy Sinks with Linear Vibration Absorbers

A classical method of reducing vibration in a system is by attaching a linear vibration absorber, which effectively transfers energy away from the primary system. The simplest absorber consists of a mass attached to a linear spring, with the spring also attached to the main system. If designed properly, the vibration energy of the main system is transferred to the absorber, reducing the vibration in the main system. Ideally, the absorber takes all of the vibration energy away from the primary system at the fundamental frequency of the main system, resulting in zero displacement of the main system at this frequency and very low displacements close to this frequency.

Extending the concept of the vibration absorber to the nonlinear case, a very powerful method of reducing unwanted vibration is achieved by attaching a nonlinear energy sink (NES) to the system. A NES is similar to the classical vibration absorber in that energy is transferred from the primary system to the attachment. However, the NES can reduce vibrations in the main system at multiple resonant frequencies, as opposed to the linear vibration absorber which is tuned to operate near a single resonant frequency. Consequently, the performance of the linear vibration absorber could degrade "over time due to aging of the system, temperature or humidity variations and so forth, thus requiring additional adjustment or tuning of parameters" (Lee et al., 2008). Hence, the degraded linear vibration absorber may become detuned and would operate effectively at a frequency other than the original if, for example, the spring constant was affected. On the other hand, if a NES degraded over time in a similar fashion, since there is no preferred frequency of operation for the NES, the compromise in performance of the NES would be minimal compared to that of the linear vibration absorber.

Vakakis et al. (2008) give an example to illustrate some of the benefits of the NES over classical linear vibration absorbers; in this case, a tuned mass damper (TMD) is considered. Figure 1-20 shows the two systems considered. In the system with the linear oscillator (LO) coupled to the TMD, it is clear that the natural frequency of the TMD alone is $\sqrt{20}$ rad/s $\approx$ 4.5 rad/s. Figure 1-21 was generated by varying the stiffness, k_1, of the linear oscillator in each system. In Figure 1-21a, it is evident that the highest percentage of energy dissipation occurs when the LO natural frequency is approximately equal to that of the TMD (4.5 rad/s). The energy dissipation drops sharply as the LO natural frequency deviates from the TMD natural frequency. On the other hand, Figure 1-21b shows that the NES is effective at energy dissipation with a much wider range of LO natural frequencies compared with the TMD performance. Also notable from Figure 1-21 is that the TMD is independent of, but the NES is heavily reliant on, the impulse magnitude. Hence, the advantage of having a wide range of frequencies at which the NES is effective is at the cost of being dependent on the impulse. Chapter 5 of this thesis presents numerical evidence of the NES dependence on impulse magnitude for a specific system with nonlinear damping.

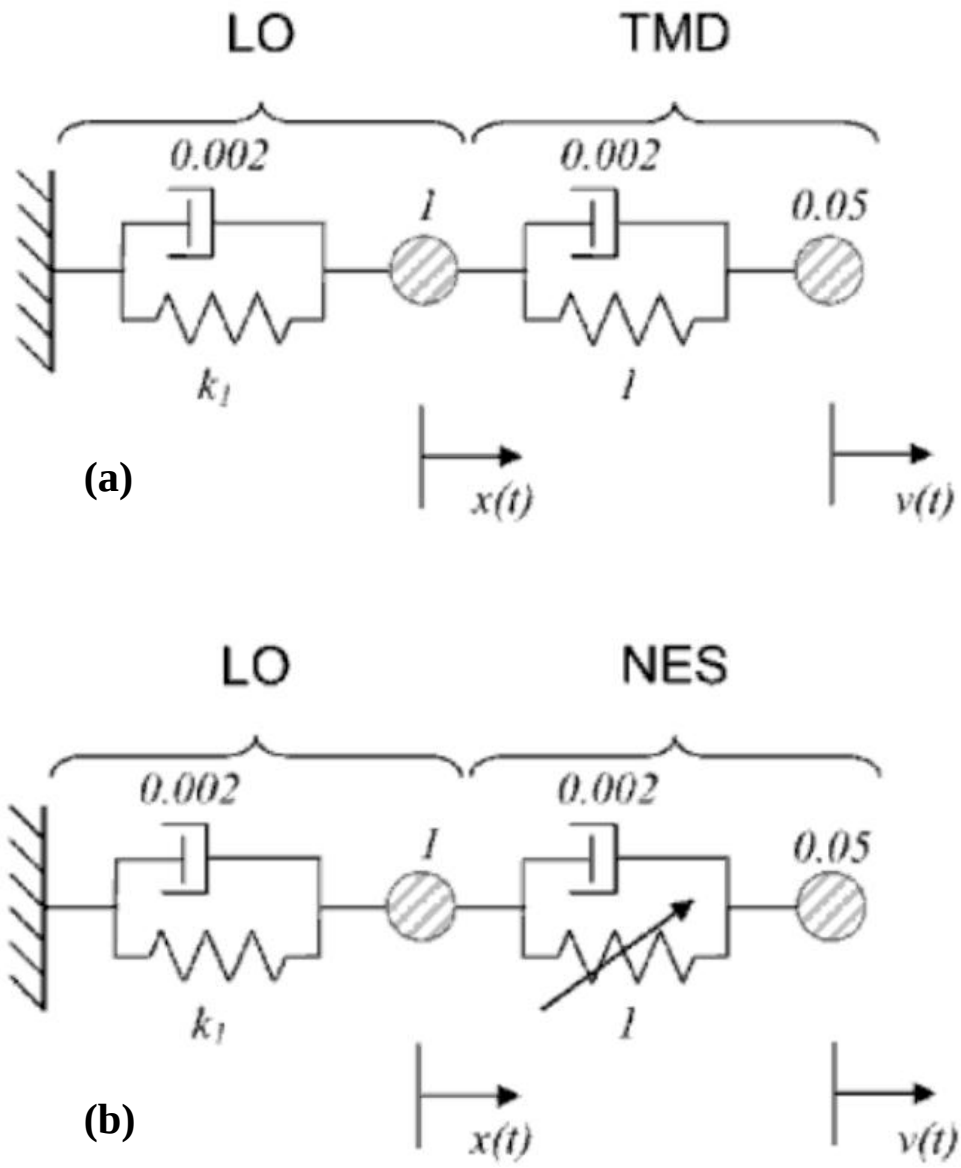


Figure 1-20. Schematic of the linear system attached to a (a) TMD and (b) NES. Vakakis et al (2008); courtesy of *Google Books*.

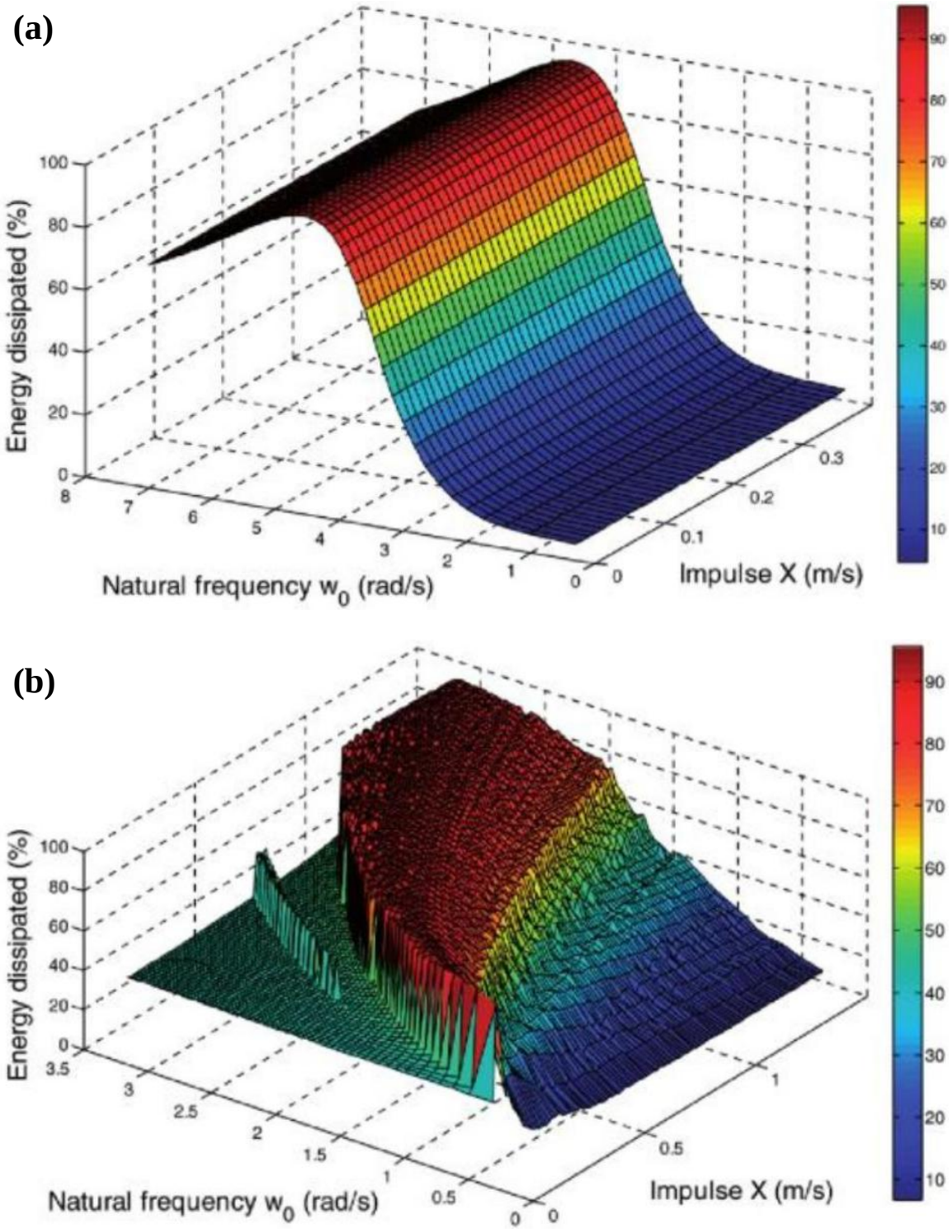


Figure 1-21. Depiction of energy dissipated as a function of linear natural frequency and impulse magnitude for (a) TMD and (b) NES.
Vakakis et al (2008); courtesy of *Google Books*.

1.4 Experiments Related to NES

Experiments have been performed to verify the theories of NES. Two requirements for an NES are that it has "*essential (nonlinearizable) stiffness nonlinearities*, and that there exists *weak damping dissipation* in the integrated linear system-nonlinear attachment configuration" (Vakakis et al., 2008). As described by (Vakakis et al., 2008), a common method of achieving nonlinear stiffness in an experimental setup is by using the inherent nonlinearity in a fixed wire with an applied force at the center corresponding to a displacement of x. Refer to Figure 1-22 for a depiction of this scenario (Vakakis et al., 2008). The force,

$$F = kx\left[1 - L(L^2 + x^2)^{-1/2}\right], \tag{1.11}$$

is found from the geometry of the wire and location of force application (Vakakis et al., 2008). Performing a Taylor series expansion of $(L^2 + x^2)^{-1/2}$ about $x = 0$ gives

$$(L^2 + x^2)^{-1/2} = \frac{1}{L} - \frac{x^2}{2L^3} + \frac{3x^4}{8L^5} + O(x^6). \tag{1.12}$$

Substituting equation (1.12) into equation (1.11) and omitting higher order terms, the force is expressed with a cubic stiffness relation,

$$F = \frac{k}{2L^2}x^3 + O(x^5), \tag{1.13}$$

in which the terms of $O(x^5)$ are considered negligible. Caution must be taken in an experimental setup to ensure very little preload is in the wire while the system is at rest, as this preload adds a linear stiffness term that makes the stiffness no longer essential (i.e. the stiffness can be linearized).

Examples of setups for performing experiments relating to NES are discussed by (Lee et al., 2008) and (Vakakis et al., 2008). One experimental setup, as shown in Figure

1-23, involves having the primary system attached to a fixed location with a linear spring, and the NES is attached to a fixed location with the wire which acts as a nonlinear spring. Both the linear oscillator and NES are considered grounded in this configuration, and both masses are connected with a linear spring. The primary system and the NES are placed on an air track to reduce frictional losses during motion. Another experimental setup involves taking nearly the same system, but now having the location at which the ends of the wire are attached move with the NES, thus having an ungrounded NES. This configuration can simulate a system that is not necessarily confined to one location. Refer to Figure 1-24 for a depiction of this setup. Figures 1-25 and 1-26 show these two experimental setups in a schematic form. Since in reality a small amount of damping is always present, dampers are shown with the linear springs in the diagrams.

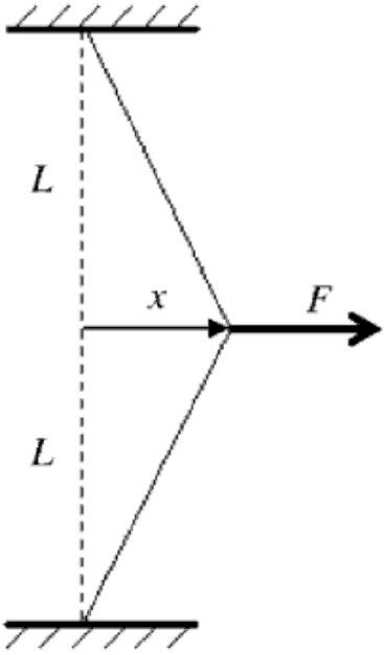
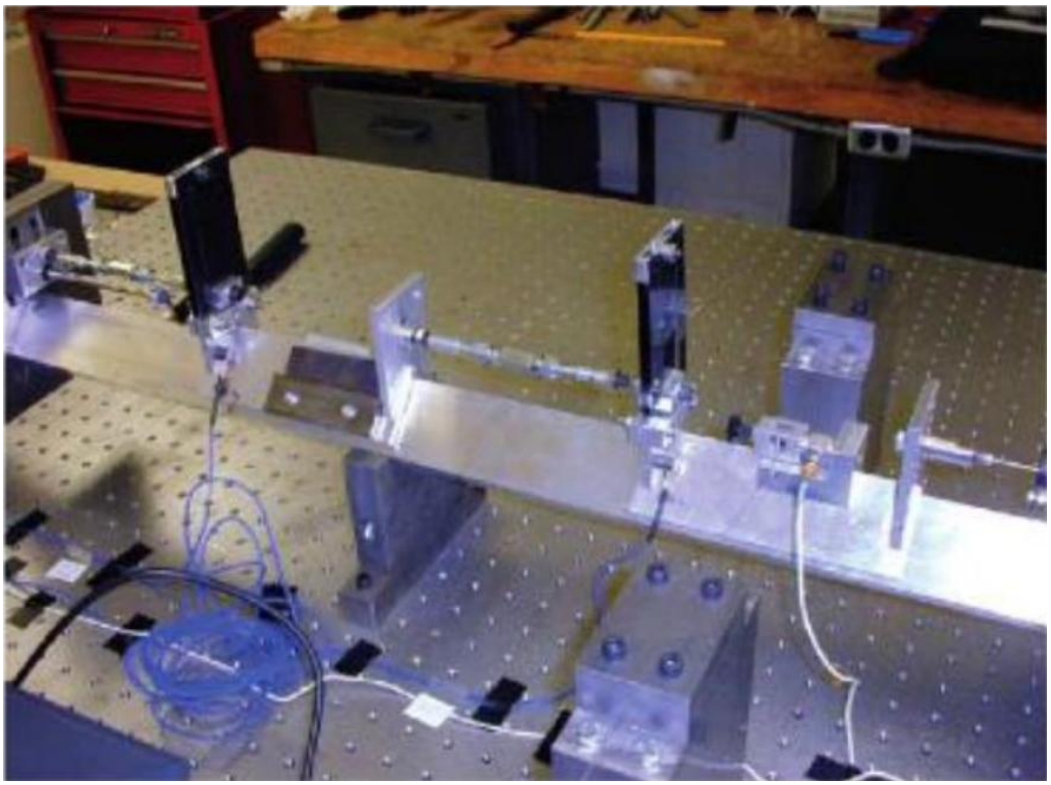
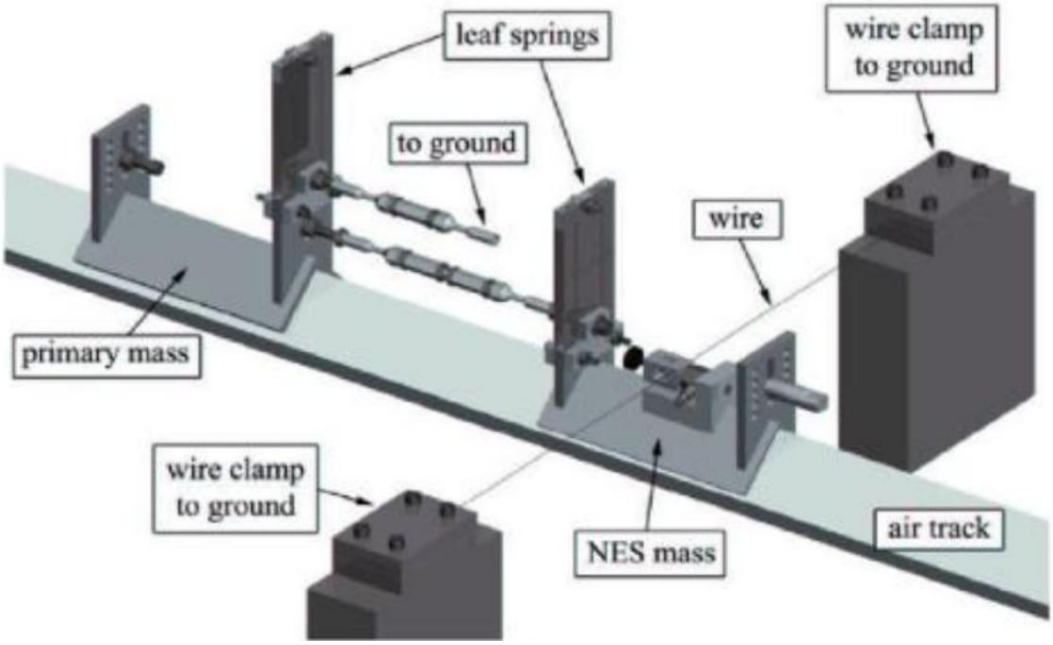


Figure 1-22. Depiction of geometry used for stiffness nonlinearity
Vakakis et al (2008); courtesy of *Google Books*.



(a)



(b)

Figure 1-23. *Grounded NES in an experimental setup*. (a) photograph of setup, (b) schematic of the setup.
Vakakis et al (2008); courtesy of *Google Books*.

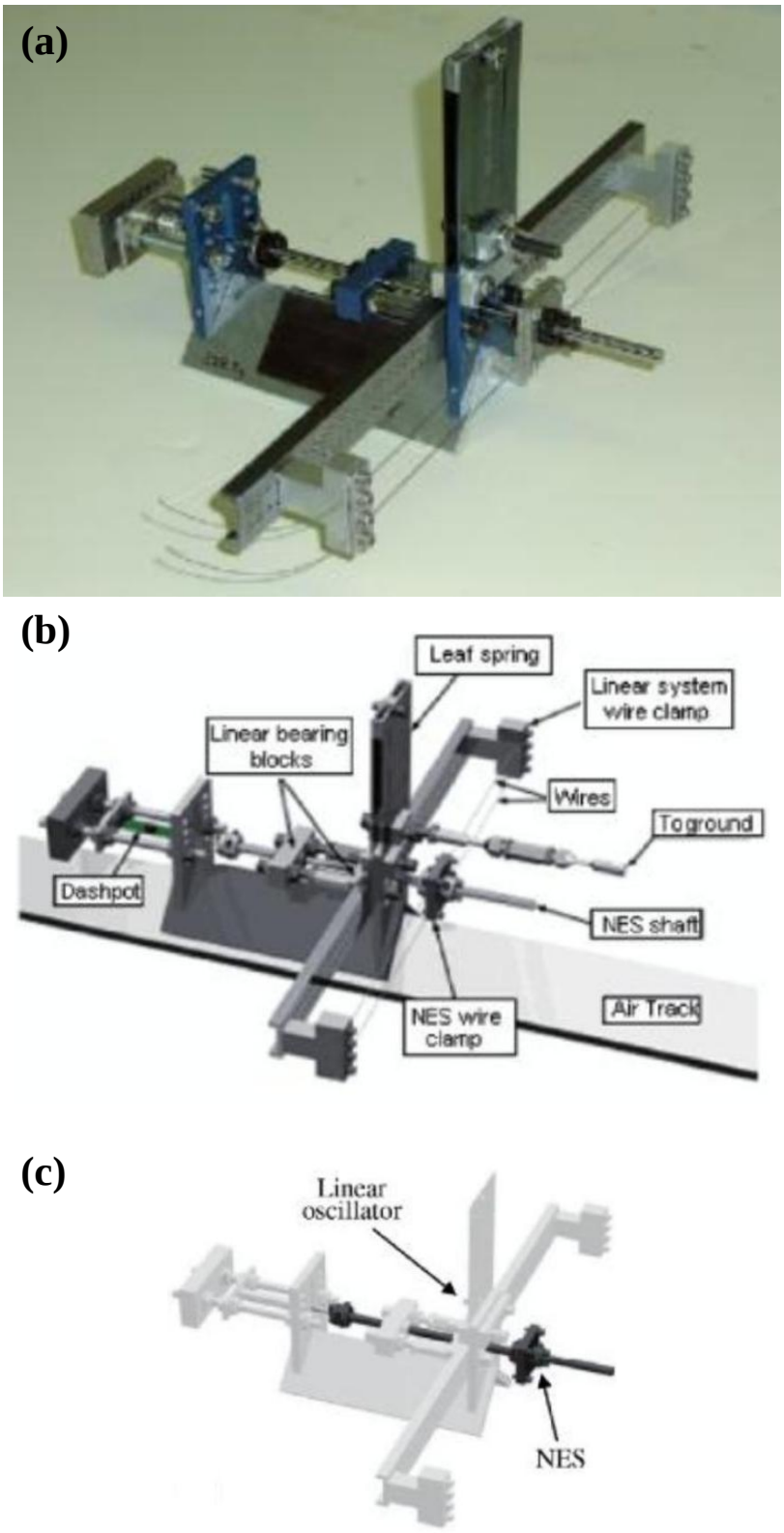


Figure 1-24. *Ungrounded NES in an experimental setup*. (a) photograph of setup (b) schematic of the setup, (c) location of NES and LO. Vakakis et al (2008); courtesy of *Google Books*.

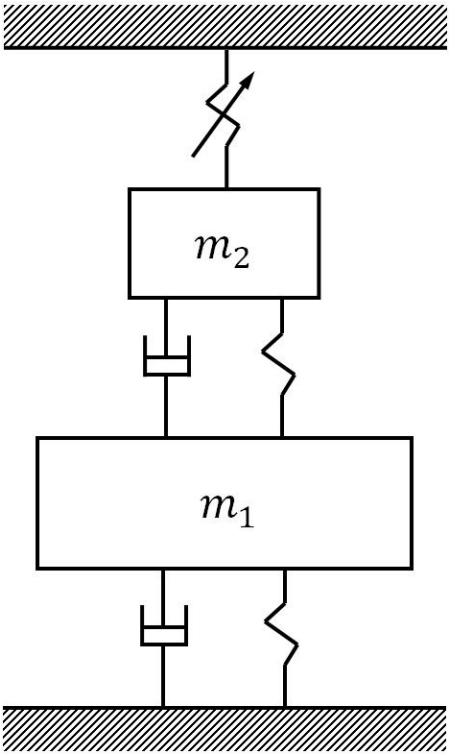


Figure 1-25. Primary system and NES attached to fixed locations (grounded system).

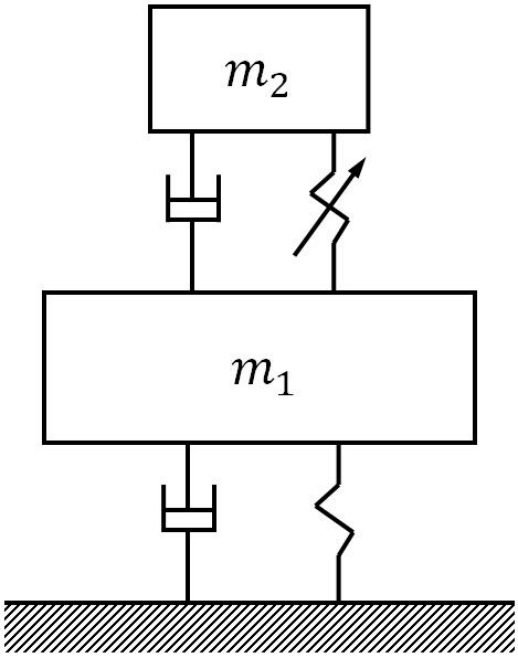


Figure 1-26. Primary system attached to fixed location with NES free (ungrounded system).

1.5 Outline of this Thesis

The goal of this thesis is to investigate the use of nonlinear damping in a NES attached to a single degree of freedom linear oscillator. An introduction to the benefits of nonlinear damping is presented in Chapter 2 with comparisons between the performances of linear, nonlinear with linear damping (NES), and nonlinear with nonlinear damping one degree of freedom systems without attachments. Chapter 3 begins the discussions related to the specific system

$$\begin{aligned} \ddot{y}_1 + \varepsilon\lambda(\dot{y}_1 - \dot{y}_2)^3 + (1 + \varepsilon\sigma)y_1 + \tfrac{4}{3}\varepsilon(y_1 - y_2)^3 &= F \\ \varepsilon\ddot{y}_2 + \varepsilon\lambda(\dot{y}_2 - \dot{y}_1)^3 + \tfrac{4}{3}\varepsilon(y_2 - y_1)^3 &= 0, \end{aligned} \tag{1.14}$$

quantifying the performance when subjected to harmonic forcing ($F = \varepsilon A \cos t$). The strongly modulated response (SMR) is investigated in Chapter 4 through seeking conditions for the existence of this response. Chapter 5 compares the performance of the same system subjected now to impulsive loading of different amplitudes with a similar system having linear damping. Additionally, phase portraits and Poincaré maps are shown in Chapter 5 to further examine the performance of the linearly and nonlinearly damped systems subjected to harmonic excitations. Conclusions and recommendations are presented in Chapter 6.

1.6 Summary

This chapter has presented a brief overview of the concepts of nonlinear energy sinks (NES), targeted energy transfer (TET), and resonance capture (RC), which are essential to the understanding of the mitigation of vibration using a special class of nonlinear absorbers. Pertinent contributions to the theoretical and experimental behavior of this

class of nonlinear absorbers are presented. Also, the relationship of such contributions to the scope of this thesis is outlined.

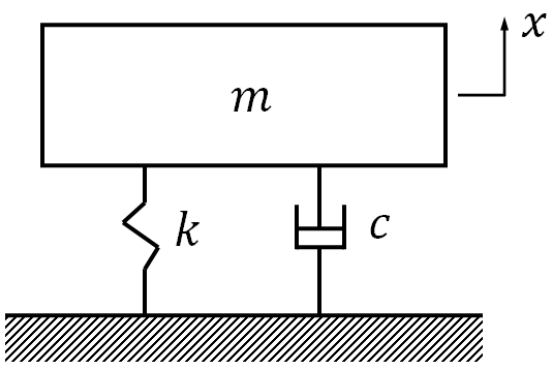
2.0 Concept Energy Sinks with Nonlinear Stiffness and Damping

2.1 Introduction

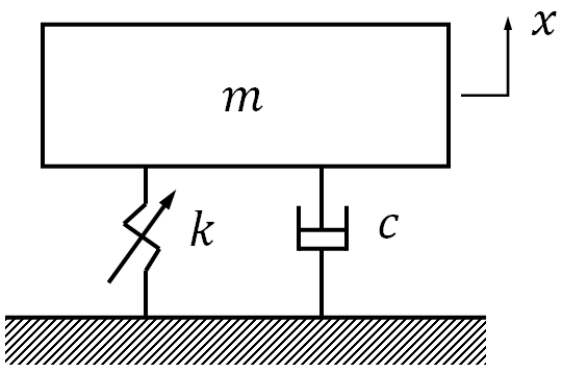
Substantial benefits regarding vibration mitigation can be achieved by including nonlinear components in a system. Since nonlinear springs and nonlinear viscous dampers can have forces proportional to any exponential power of displacement and velocity, respectively, greater flexibility in system design is allowed with their inclusion. This section presents simple examples to illustrate the advantages of having nonlinear components in systems.

2.2 Description of Considered Systems

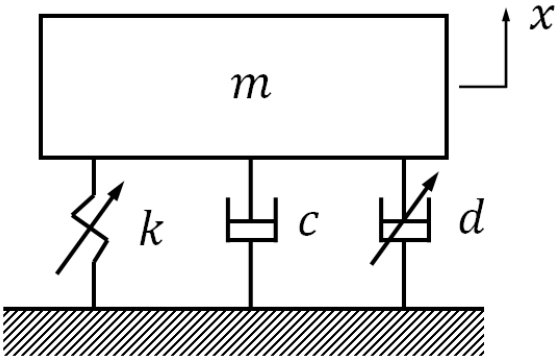
Three systems are compared in this section in order to illustrate the advantages of nonlinear stiffness and nonlinear damping characteristics in terms of vibration mitigation. The three systems under consideration are: a linear system, a NES system, and a nonlinear system with nonlinear damping. For simplicity, only one degree of freedom systems are considered. The linear system consists of a mass attached to a linear spring and a linear damper as shown in Figure 2-1a. The NES system replaces the linear spring with a nonlinear spring having restoring force proportional to the cube of displacement as shown in Figure 2-1b. Finally, the nonlinear system with nonlinear damping expands on the NES system by including a nonlinear damper having a damping force proportional to the velocity cubed as shown in Figure 2-1c.



(a). Linear system



(b). NES system



(c). Nonlinear system with nonlinear damping

Figure 2-1. Different types of considered systems

2.3 Analysis of the Different Systems

2.3.1 Linear System

The equation of motion is written as

$$m\ddot{x} + c\dot{x} + kx = 0. \tag{2.1}$$

In order to perform the numerical integration, the equation of motion is cast into the following state space form:

$$\begin{Bmatrix} \dot{x_1} \\ \dot{x_2} \end{Bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{c}{m} \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix}, \tag{2.2}$$

where

$$x_1 = x \text{ and } x_2 = \dot{x}. \tag{2.3}$$

The system total energy is given by

$$E = \frac{1}{2}m\dot{x}^2 + \frac{1}{2}kx^2. \tag{2.4}$$

Taking the derivative of the energy equation with respect to time gives the rate of energy decay,

$$\dot{E} = (m\ddot{x} + kx)\dot{x} = -c\dot{x}^2 < 0. \tag{2.5}$$

2.3.2 NES System

The equation of motion is written as

$$m\ddot{x} + c\dot{x} + kx^3 = 0. \tag{2.6}$$

In order to perform the numerical integration, the equation of motion is cast into the following state space form:

$$\begin{Bmatrix} \dot{x_1} \\ \dot{x_2} \end{Bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{kx_1{}^2}{m} & -\frac{c}{m} \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix}, \tag{2.7}$$

where

$$x_1 = x \text{ and } x_2 = \dot{x}. \tag{2.8}$$

The system total energy is given by

$$E = \frac{1}{2}m\dot{x}^2 + \frac{1}{4}kx^4. \tag{2.9}$$

Taking the derivative of the energy equation with respect to time gives the rate of energy decay,

$$\dot{E} = (m\ddot{x} + kx^3)\dot{x} = -c\dot{x}^2 < 0. \tag{2.10}$$

2.3.3 Nonlinear System with Nonlinear Damping

The equation of motion is written as

$$m\ddot{x} + c\dot{x} + kx^3 + d\dot{x}^3 = 0. \tag{2.11}$$

In order to perform the numerical integration, the equation of motion is cast into the following state space form:

$$\begin{Bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{Bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{kx_1{}^2}{m} & -\frac{c}{m} - \frac{dx_2{}^2}{m} \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix}, \tag{2.12}$$

where

$$x_1 = x \text{ and } x_2 = \dot{x}. \tag{2.13}$$

The system total energy is given by

$$E = \frac{1}{2}m\dot{x}^2 + \frac{1}{4}kx^4. \tag{2.14}$$

Taking the derivative of the energy equation with respect to time gives the rate of energy decay,

$$\dot{E} = (m\ddot{x} + kx^3)\dot{x} = -c\dot{x}^2 - d\dot{x}^4 < 0. \tag{2.15}$$

2.4 Performance Comparisons

For comparison between the performances of each system, the equations of motion, equations of the total system energy, and equations of system energy decay were generated for all three cases. The equations of motion,

$$Linear \qquad m\ddot{x} + c\dot{x} + kx = 0 \qquad (2.16a)$$

$$NES \qquad m\ddot{x} + c\dot{x} + kx^3 = 0 \qquad (2.16b)$$

$$Nonlinear\ with\ nonlinear\ damping \qquad m\ddot{x} + c\dot{x} + kx^3 + d\dot{x}^3 = 0, \qquad (2.16c)$$

were used to determine displacement as a function of time. The Runge-Kutta 4 numerical method was used to solve these equations. Parameters were chosen for a lightly damped ($\zeta = 0.075$) linear system as shown in Table 2-1.

Table 2-1. System Parameters for Numerical Solutions

System	*m*	*k*	*c*	*d*
Linear	1	4	0.3	0
NES	1	4	0.3	0
Nonlinear with nonlinear damping	1	4	0.3	2

In performing the analysis, each system was initially at rest and given an initial displacement of 1. As shown in Figure 2-2, the frequency of oscillation of each system is different. The linear system has a faster oscillation frequency than the NES system, and the nonlinear system with nonlinear damping has the slowest oscillation frequency. The

displacement of the nonlinear system with nonlinear damping appears to have smoother transitions from peak to peak, and the amplitude of the displacement is reduced faster than that for the linear and NES systems.

Further comparisons between the performances of each system can be made by observing the total system energy as a function of time. The total system energy is governed by

$$Linear \qquad E = \frac{1}{2}m\dot{x}^2 + \frac{1}{2}kx^2 \tag{2.17a}$$

$$NES \qquad E = \frac{1}{2}m\dot{x}^2 + \frac{1}{4}kx^4 \tag{2.17b}$$

$$Nonlinear\ with\ nonlinear\ damping \qquad E = \frac{1}{2}m\dot{x}^2 + \frac{1}{4}kx^4. \tag{2.17c}$$

By plotting the total energy for each system versus time, Figure 2-3 was obtained. Initially, the linear system has twice as much energy as the other two systems. Since the initial velocity is zero, the greater initial system energy is a consequence of the potential energy term in the equations. Observing the graphs, it is clear that the nonlinear system with nonlinear damping decays faster than both the NES and linear systems.

By taking the derivative with respect to time of the energy equations, the equations representing energy decay,

$$Linear \qquad \dot{E} = (m\ddot{x} + kx)\dot{x} = -c\dot{x}^2 \tag{2.18a}$$

$$NES \qquad \dot{E} = (m\ddot{x} + kx^3)\dot{x} = -c\dot{x}^2 \tag{2.18b}$$

$$Nonlinear\ with\ nonlinear\ damping \qquad \dot{E} = (m\ddot{x} + kx^3)\dot{x} = -c\dot{x}^2 - d\dot{x}^4, \tag{2.18c}$$

are determined. Figure 2-4 shows the magnitude of energy decay for each system as a function of time. The linear system evidently takes the longest time to finish dissipating the system energy. The NES system energy decay is faster than that of the linear system.

Although the first peak of energy decay on the NES system plot is lower than that of the linear system, the slower period of energy decay in the NES system allows energy to be dissipated more rapidly. In contrast to the linear and NES systems, the plot of energy decay for the nonlinear system with nonlinear damping shows a relatively high peak followed by a rapid decrease in energy decay. Due to the large initial energy decay, the system energy is dissipated much more quickly than the other two systems. Thus, the nonlinear system with nonlinear damping is more effective at energy dissipation than the linear or NES systems.

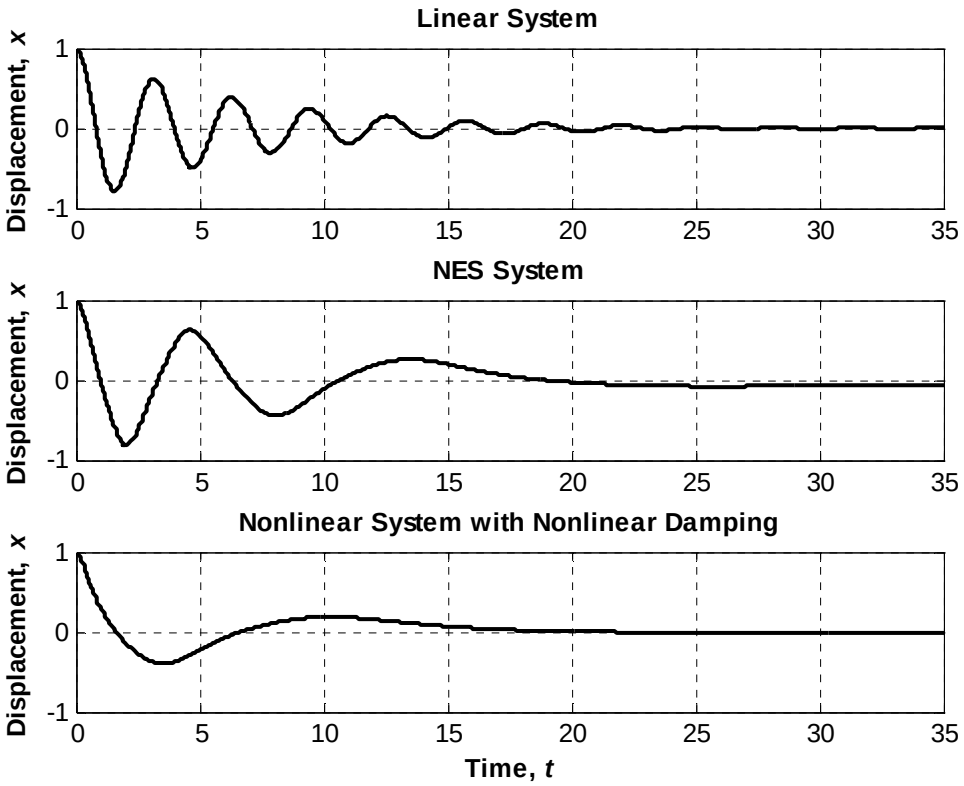


Figure 2-2. Time response of each system

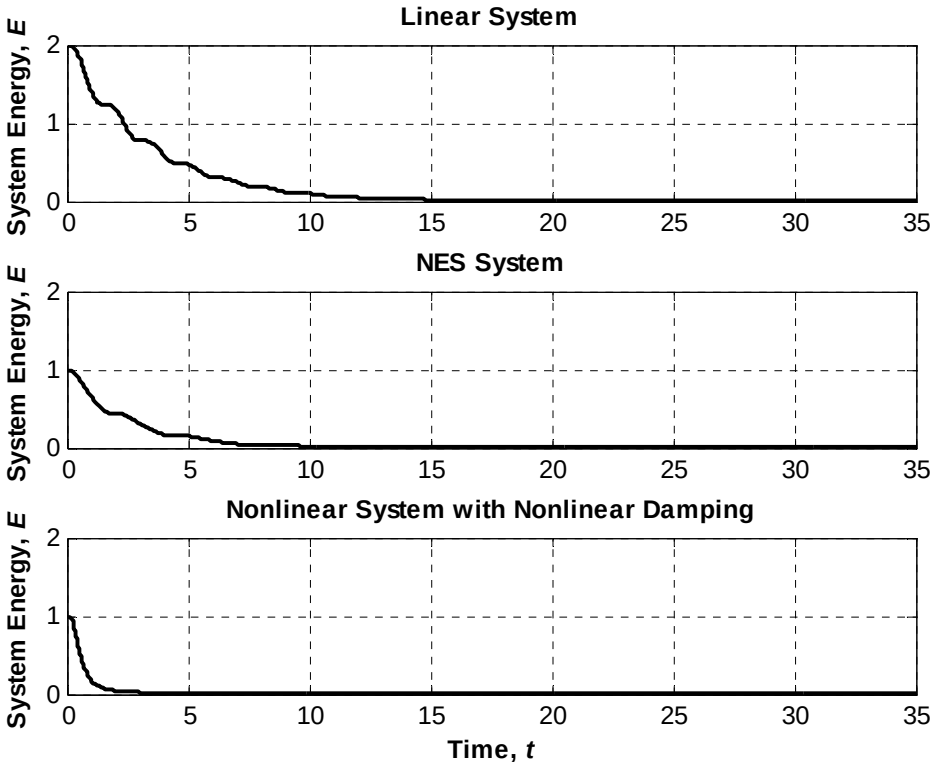


Figure 2-3. System Energy versus time for each system

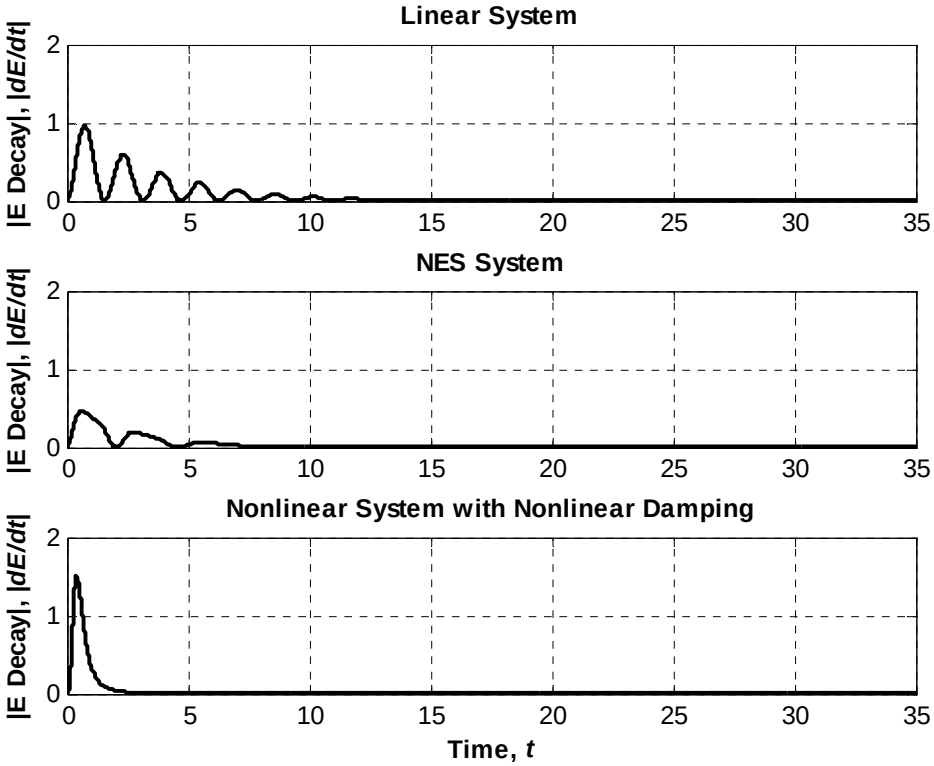


Figure 2-4. Magnitude of energy decay versus time for each system

The results presented in this section have shown that the nonlinear system with nonlinear damping can be far more effective at mitigating vibration than the linear or NES counterparts. The motion of the nonlinear system with nonlinear damping comes to rest more quickly than the other two systems. The rapid reduction in system displacement is due to the system energy being dissipated faster in the nonlinear system with nonlinear damping than in the NES or linear systems. Including nonlinear damping in the system clearly can have a dramatic influence on the system performance. Since the nonlinear damping term is a function of velocity cubed, this term is much more effective at dissipating system energy than a linear damping term.

2.5 Summary

As evident from the examples presented in this section, systems possessing nonlinear damping characteristics can vastly outperform their linear counterparts. In designing a system, the coefficient of the nonlinear damping term could be chosen to reduce system energy much more quickly than a system with only linear damping. Practically, including nonlinear terms allows the designer more options when designing a system.

2.A Appendix

In MATLAB, the Runge-Kutta 4 method was used to numerically integrate the equations of motion, equations of system energy, and equations of energy decay. For ease of explanation, the magnitude of energy decay was determined. Note that the energy decay is always negative; thus the magnitude of energy decay is equal to the energy decay multiplied by -1. The following code was used to generate the plots shown in Figure 2-2, Figure 2-3, and Figure 2-4.

```
%Numerical Comparison of NES, Linear, and Nonlinear with Nonlinear
%Damping Systems

close all
clear all

T = 0.001; h = 1; c = .3; k = 4; m = 1; d = 2;

t_end = 35; %end of time interval
x10 = 1; %initial displacement
x20 = 0; %initial velocity

t_vec = zeros(1,t_end/T + 1);
x1_NES_vec = t_vec; x2_NES_vec = t_vec;
x1_Lin_vec = t_vec; x2_Lin_vec = t_vec;
x1_NonL_vec = t_vec; x2_NonL_vec = t_vec;
E_NES_vec = t_vec; E_dot_NES_vec = t_vec;
E_Lin_vec = t_vec; E_dot_Lin_vec = t_vec;
E_NonL_vec = t_vec; E_dot_NonL_vec = t_vec;

x1_NES = x10; x1_NES_vec(1) = x10; x2_NES = x20; x2_NES_vec(1) = x20;
x1_Lin = x10; x1_Lin_vec(1) = x10; x2_Lin = x20; x2_Lin_vec(1) = x20;
x1_NonL = x10; x1_NonL_vec(1) = x10; x2_NonL = x20; x2_NonL_vec(1) =
x20;
E_NES_vec(1) = 1/2*m*x2_NES^2 + 1/4*k*x1_NES^4;
E_dot_NES_vec(1) = c*x2_NES^2;
E_Lin_vec(1) = 1/2*m*x2_Lin^2 + 1/2*k*x1_Lin^2;
E_dot_Lin_vec(1) = c*x2_Lin^2;
E_NonL_vec(1) = 1/2*m*x2_NonL^2 + 1/4*k*x1_NonL^4;
E_dot_NonL_vec(1) = c*x2_NonL^2;

%Runge Kutta 4
%From Fundamentals of Vibrations by Leonard Meirovitch, p. 677-679:

for t = T:T:t_end
    h = h+1; t_vec(h) = t;
```

```
%NES System:

f11 = x2_NES; f12 = -(c*x2_NES + k*x1_NES^3)/m;
g11 = T*f11; g12 = T*f12;

f21 = x2_NES + 1/2*g12; f22 = -(c*(x2_NES+1/2*g12) + ...
    k*(x1_NES+1/2*g11)^3)/m;
g21 = T*f21; g22 = T*f22;

f31 = x2_NES + 1/2*g22; f32 = -(c*(x2_NES+1/2*g22) + ...
    k*(x1_NES+1/2*g21)^3)/m;
g31 = T*f31; g32 = T*f32;

f41 = x2_NES + g32; f42 = -(c*(x2_NES+g32) + k*(x1_NES+g31)^3)/m;
g41 = T*f41; g42 = T*f42;

x1_NES = x1_NES + 1/6*(g11 + 2*g21 + 2*g31 + g41);
x2_NES = x2_NES + 1/6*(g12 + 2*g22 + 2*g32 + g42);
E_NES = 1/2*m*x2_NES^2 + 1/4*k*x1_NES^4;
E_dot_NES = c*x2_NES^2;

x1_NES_vec(h) = x1_NES; x2_NES_vec(h) = x2_NES;
E_NES_vec(h) = E_NES; E_dot_NES_vec(h) = E_dot_NES;

%Linear System:

f11 = x2_Lin; f12 = -(c*x2_Lin + k*x1_Lin)/m;
g11 = T*f11; g12 = T*f12;

f21 = x2_Lin + 1/2*g12; f22 = -(c*(x2_Lin+1/2*g12) + ...
    k*(x1_Lin+1/2*g11))/m;
g21 = T*f21; g22 = T*f22;

f31 = x2_Lin + 1/2*g22; f32 = -(c*(x2_Lin+1/2*g22) + ...
    k*(x1_Lin+1/2*g21))/m;
g31 = T*f31; g32 = T*f32;

f41 = x2_Lin + g32; f42 = -(c*(x2_Lin+g32) + k*(x1_Lin+g31))/m;
g41 = T*f41; g42 = T*f42;

x1_Lin = x1_Lin + 1/6*(g11 + 2*g21 + 2*g31 + g41);
x2_Lin = x2_Lin + 1/6*(g12 + 2*g22 + 2*g32 + g42);
E_Lin = 1/2*m*x2_Lin^2 + 1/2*k*x1_Lin^2;
E_dot_Lin = c*x2_Lin^2;

x1_Lin_vec(h) = x1_Lin; x2_Lin_vec(h) = x2_Lin;
E_Lin_vec(h) = E_Lin; E_dot_Lin_vec(h) = E_dot_Lin;

%Nonlinear System:

f11 = x2_NonL; f12 = -(c*x2_NonL + k*x1_NonL^3 + d*x2_NonL^3)/m;
g11 = T*f11; g12 = T*f12;

f21 = x2_NonL + 1/2*g12; f22 = -(c*(x2_NonL+1/2*g12) + ...
```

```
        k*(x1_NonL+1/2*g11)^3 + d*(x2_NonL+1/2*g12)^3)/m;
    g21 = T*f21; g22 = T*f22;

    f31 = x2_NonL + 1/2*g22; f32 = -(c*(x2_NonL+1/2*g22) + ...
        k*(x1_NonL+1/2*g21)^3 + d*(x2_NonL+1/2*g22)^3)/m;
    g31 = T*f31; g32 = T*f32;

    f41 = x2_NonL + g32; f42 = -(c*(x2_NonL+g32) + ...
        k*(x1_NonL+g31)^3 + d*(x2_NonL+g32)^3)/m;
    g41 = T*f41; g42 = T*f42;

    x1_NonL = x1_NonL + 1/6*(g11 + 2*g21 + 2*g31 + g41);
    x2_NonL = x2_NonL + 1/6*(g12 + 2*g22 + 2*g32 + g42);
    E_NonL = 1/2*m*x2_NonL^2 + 1/4*k*x1_NonL^4;
    E_dot_NonL = c*x2_NonL^2 + d*x2_NonL^4;

    x1_NonL_vec(h) = x1_NonL; x2_NonL_vec(h) = x2_NonL;
    E_NonL_vec(h) = E_NonL; E_dot_NonL_vec(h) = E_dot_NonL;
end

figure
subplot(3,1,1); plot(t_vec, x1_Lin_vec, '-k','LineWidth', 2); grid on;
title('\bfLinear System'); ylabel('\bfDisplacement, \itx');
subplot(3,1,2); plot(t_vec, x1_NES_vec, '-k','LineWidth', 2); grid on;
title('\bfNES System'); ylabel('\bfDisplacement, \itx');
subplot(3,1,3); plot(t_vec, x1_NonL_vec, '-k','LineWidth', 2); grid on;
title('\bfNonlinear System with Nonlinear Damping')
xlabel('\bfTime, \itt'); ylabel('\bfDisplacement, \itx');

figure
subplot(3,1,1); plot(t_vec, E_Lin_vec, '-k','LineWidth', 2); grid on;
title('\bfLinear System'); ylabel('\bfSystem Energy, \itE');
subplot(3,1,2); plot(t_vec, E_NES_vec, '-k','LineWidth', 2); grid on;
title('\bfNES System'); ylabel('\bfSystem Energy, \itE');
axis([0 t_end 0 2]);
subplot(3,1,3); plot(t_vec, E_NonL_vec, '-k','LineWidth', 2); grid on;
title('\bfNonlinear System with Nonlinear Damping')
xlabel('\bfTime, \itt'); ylabel('\bfSystem Energy, \itE');
axis([0 t_end 0 2]);

figure
subplot(3,1,1); plot(t_vec, E_dot_Lin_vec, '-k','LineWidth', 2); grid
on;
title('\bfLinear System'); ylabel('\bf|E Decay|, \it|dE/dt|');
axis([0 t_end 0 2]);
subplot(3,1,2); plot(t_vec, E_dot_NES_vec, '-k','LineWidth', 2); grid
on;
title('\bfNES System'); ylabel('\bf|E Decay|, \it|dE/dt|');
axis([0 t_end 0 2]);
subplot(3,1,3); plot(t_vec, E_dot_NonL_vec, '-k','LineWidth', 2); grid
on;
title('\bfNonlinear System with Nonlinear Damping')
xlabel('\bfTime, \itt'); ylabel('\bf|E Decay|, \it|dE/dt|');
```

3.0 Nonlinear Analysis of Nonlinear Energy Sinks

3.1 Introduction

This section analyzes the performance of a selected system, which is a modification of the system studied in "Response regimes of linear oscillator coupled to nonlinear energy sink with harmonic forcing and frequency detuning" by Starosvetsky and Gendelman (2008). Starosvetsky and Gendelman (2008) considered a linear system with an attachment consisting of linear damping and cubic nonlinear stiffness. This thesis extends the work to the case of the linear system with the attachment now having cubic nonlinear damping. Physically, the system under consideration in both cases corresponds to a harmonically forced linear structure with a strongly nonlinear attachment acting as a nonlinear energy sink (NES). The equations of motion of the system under consideration are given by

$$\begin{aligned} \ddot{y_1} + \varepsilon\lambda(\dot{y_1} - \dot{y_2})^3 + (1 + \varepsilon\sigma)y_1 + \tfrac{4}{3}\varepsilon(y_1 - y_2)^3 &= \varepsilon A\cos t \\ \varepsilon\ddot{y_2} + \varepsilon\lambda(\dot{y_2} - \dot{y_1})^3 + \tfrac{4}{3}\varepsilon(y_2 - y_1)^3 &= 0, \end{aligned} \tag{3.1}$$

where y_1 is the displacement of the linear oscillator, y_2 is the displacement of the attachment, $\varepsilon\lambda$ is the damping coefficient, εA is the amplitude of external force, and $\varepsilon\sigma$ is the frequency detuning parameter. From equations (3.1), it is evident that the nonlinearities appear in both equations as the cube of difference in velocity and difference in displacement between the linear system and the NES. For simplicity in this chapter of the thesis, the terms "damping coefficient", "amplitude of external force", and "frequency detuning parameter" refer to the same parameters as Starosvetsky and Gendelman (2008), without the coefficient, ε.

In this thesis, periodic solutions are described in Section 3.2 and Section 3.3 with the use of saddle-node and Hopf bifurcations, respectively. Finally, conclusions are discussed at the end of this chapter.

3.2 Saddle-Node Bifurcation

3.2.1 Saddle-Node Bifurcation Background

The goal of this section is to describe the periodic solutions of the system through examining the saddle-node bifurcations. A bifurcation represents "a qualitative change in the features of a system, such as the number and type of solutions," due to "the variation of one or more parameters on which the considered system depends" (Nayfeh and Balachandran, 1995). By looking at a plot of system parameters (displacement versus a scalar parameter, for example), one can often decipher the type of bifurcation by visual inspection. In order to study bifurcations, the system is frequently put in the form

$$\dot{\boldsymbol{x}} = \boldsymbol{F}(\boldsymbol{x}; \mu), \tag{3.2}$$

in which $\boldsymbol{x}$ is the state vector and $\dot{\boldsymbol{x}}$ is the derivative of $\boldsymbol{x}$ with respect to time. On the right hand side of (3.2), $\boldsymbol{F}$ is called the vector field and is a function of $\boldsymbol{x}$ and μ, a scalar parameter (Nayfeh and Balachandran, 1995).

For a saddle-node bifurcation to exist, $\boldsymbol{F}$ must equal the zero vector, and the Jacobian of $\boldsymbol{F}$ (denoted by $D_x\boldsymbol{F}$) must have at least one zero eigenvalue and the remaining eigenvalues having nonzero real parts (Nayfeh and Balachandran, 1995). The first of these prerequisites is the condition for a fixed point to exist. At the fixed point, $\dot{\boldsymbol{x}}$ equals the zero vector, and $\boldsymbol{x}$ can be superficially replaced with $\boldsymbol{x_0}$ in order to represent the fixed

point. In addition to the aforementioned two criteria for a saddle-node bifurcation, one final condition must be met. Let

$$\boldsymbol{F}_{\boldsymbol{\mu}} = \frac{\partial \boldsymbol{F}}{\partial \mu}, \tag{3.3}$$

with $\boldsymbol{F}$ being an n x 1 vector, thus $D_x\boldsymbol{F}$ is an n x n vector. For a saddle-node bifurcation to occur, the n x (n + 1) matrix $[D_x\boldsymbol{F} \mid \boldsymbol{F}_{\boldsymbol{\mu}}]$ must be of rank n (Nayfeh and Balachandran, 1995).

A simple example of a saddle-node bifurcation (given by Nayfeh and Balachandran, 1995) involves the system

$$\dot{x} = \mu - x^2, \tag{3.4}$$

where μ is a scalar control parameter. By setting $\dot{x}$ equal to zero and solving for x, the fixed points of the system,

$$x = \pm\sqrt{\mu}, \tag{3.5}$$

are obtained. Figure 3-1 shows a plot of x versus μ, with a saddle-node bifurcation occurring at the origin. This bifurcation is a saddle-node because the three conditions are satisfied for this system. The first condition ($\boldsymbol{F}$ must equal the zero vector) is satisfied by setting $\dot{x}$ equal to zero since $\dot{x} = F$. Taking the Jacobian of $\boldsymbol{F}$ gives

$$D_x F = \frac{\partial F}{\partial x} = -2x, \tag{3.6}$$

and at $x = 0$, $D_x\boldsymbol{F} = 0$. Setting

$$D_x F - \lambda = 0 \tag{3.7}$$

gives the eigenvalue $\lambda = 0$. Thus, this system has a zero eigenvalue at the bifurcation point. Finally, solving for

$$\boldsymbol{F}_{\boldsymbol{\mu}} = \frac{\partial \boldsymbol{F}}{\partial \mu} = 1, \tag{3.8}$$

the matrix $[D_x\boldsymbol{F} \mid \boldsymbol{F}_\mu]$ becomes [0 1] at the point $x = 0$, $\mu = 0$. Since this matrix has only one linearly independent column, the rank of the matrix is one. The bifurcation is a saddle node bifurcation because all three conditions have been satisfied.

The plot in Figure 3-1 consists of two branches, each corresponding to one of the fixed points. In general, saddle-node bifurcations are characterized by two branches terminating at a single point, the bifurcation point. Note that the qualitative change in Figure 3-1 occurs at the location where the stable branch meets the unstable branch. At the origin, there is a single fixed point solution. However, at values of $\mu > 0$, there are two solutions; thus we have a qualitative change in the system. Stability will be discussed in Section 3.3 with the study of Hopf bifurcations, as this knowledge is not necessary for understanding the saddle-node bifurcation impact on number of solutions.

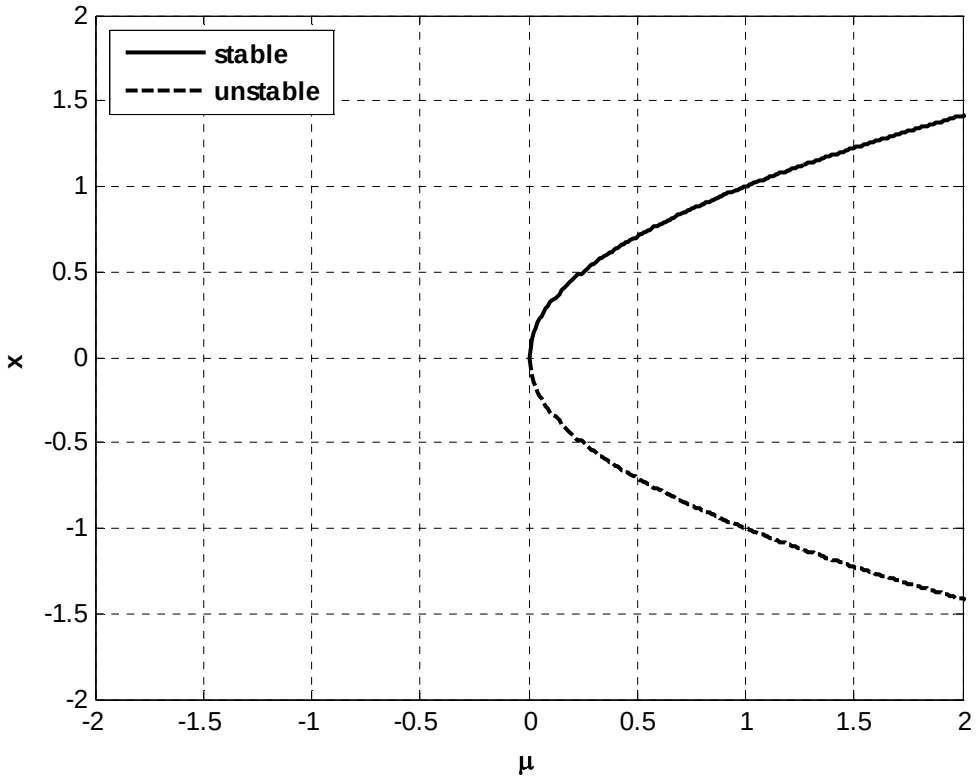


Figure 3-1. Saddle-node bifurcation given by equation (3.4)

3.2.2 Saddle-Node Bifurcation Analysis for the Linearly Damped System

Armed with a general understanding of saddle-node bifurcations, the focus of this section is shifted to the system with linear damping given by

$$\begin{aligned} \ddot{y}_1 + \varepsilon\lambda(\dot{y}_1 - \dot{y}_2) + (1+\varepsilon\sigma)y_1 + \tfrac{4}{3}\varepsilon(y_1 - y_2)^3 &= \varepsilon A\cos t \\ \varepsilon\ddot{y}_2 + \varepsilon\lambda(\dot{y}_2 - \dot{y}_1) + \tfrac{4}{3}\varepsilon(y_2 - y_1)^3 &= 0. \end{aligned} \tag{3.9}$$

The following derivations are based on the work done by Starosvetsky and Gendelman (2008). By making a change of variables according to

$$\begin{aligned} v = y_1 + \varepsilon y_2, \quad w = y_1 - y_2, \\ \varphi_1 e^{it} = \dot{v} + iv, \quad \varphi_2 e^{it} = \dot{w} + iw, \end{aligned} \tag{3.10}$$

and omitting exponential terms from the resulting set of equations, we can rewrite the system as

$$\dot{\varphi_1} + \tfrac{i\varepsilon}{2(1+\varepsilon)}(\varphi_1 - \varphi_2) - \tfrac{i\varepsilon\sigma}{2(1+\varepsilon)}(\varphi_1 + \varepsilon\varphi_2) = \tfrac{\varepsilon A}{2}$$

and (3.11)

$$\dot{\varphi_2} + \tfrac{\lambda(1+\varepsilon)}{2}\varphi_2 + \tfrac{i}{2(1+\varepsilon)}(\varphi_2 - \varphi_1) - \tfrac{i\varepsilon\sigma}{2(1+\varepsilon)}(\varphi_1 + \varepsilon\varphi_2) - \tfrac{i}{2}(1+\varepsilon)|\varphi_2|^2\varphi_2 = \tfrac{\varepsilon A}{2}.$$

Setting the time derivatives of equation (3.11) to zero gives

$$\tfrac{i\varepsilon}{2(1+\varepsilon)}(\varphi_{10} - \varphi_{20}) - \tfrac{i\varepsilon\sigma}{2(1+\varepsilon)}(\varphi_{10} + \varepsilon\varphi_{20}) = \tfrac{\varepsilon A}{2},$$

and (3.12)

$$\begin{aligned} \tfrac{\lambda(1+\varepsilon)}{2}\varphi_{20} + \tfrac{i}{2(1+\varepsilon)}(\varphi_{20} - \varphi_{10}) - \tfrac{i\varepsilon\sigma}{2(1+\varepsilon)}(\varphi_{10} + \varepsilon\varphi_{20}) \\ -\tfrac{i}{2}(1+\varepsilon)|\varphi_{20}|^2\varphi_{20} = \tfrac{\varepsilon A}{2}, \end{aligned}$$

where φ_{10} and φ_{20} are the fixed points of the system. The equations (3.12) lead to

$$\varphi_{10} = \tfrac{(1+\varepsilon\sigma)\varphi_{20} - i(1+\varepsilon)A}{1-\sigma} \tag{3.13}$$

and

$$\left[\lambda^2 + \frac{\sigma^2}{(1-\sigma)^2}\right]|\varphi_{20}|^2 + \frac{2\sigma}{(1-\sigma)}|\varphi_{20}|^4 + |\varphi_{20}|^6 = \frac{A^2}{(1-\sigma)^2}. \tag{3.14}$$

As a simplification, equation (3.14) can be rewritten as

$$\alpha_1 Z + \alpha_2 Z^2 + \alpha_3 Z^3 + \alpha_4 = 0, \tag{3.15}$$

where

$$\begin{gathered} |\varphi_{20}|^2 = Z, \quad \alpha_1 = \lambda^2 + \frac{\sigma^2}{(1-\sigma)^2} \\ \alpha_2 = \frac{2\sigma}{1-\sigma}, \quad \alpha_3 = 1, \quad \alpha_4 = \frac{-A^2}{(1-\sigma)^2}. \end{gathered} \tag{3.16}$$

Taking the derivative of equation (3.15) with respect to *Z* gives

$$3\alpha_3 Z^2 + 2\alpha_2 Z + \alpha_1 = 0. \tag{3.17}$$

Eliminating *Z* from equations (3.15) and (3.17) as shown in Section 3.A.1 gives

$$\alpha_4 = -\alpha_1\left(\frac{-\alpha_2}{3\alpha_3} \pm \frac{\sqrt{{\alpha_2}^2 - 3\alpha_1\alpha_3}}{3\alpha_3}\right) - \alpha_2\left(\frac{-\alpha_2}{3\alpha_3} \pm \frac{\sqrt{{\alpha_2}^2 - 3\alpha_1\alpha_3}}{3\alpha_3}\right)^2 - \alpha_3\left(\frac{-\alpha_2}{3\alpha_3} \pm \frac{\sqrt{{\alpha_2}^2 - 3\alpha_1\alpha_3}}{3\alpha_3}\right)^3. \tag{3.18}$$

The expression given in equation (3.18) represents the boundary of the saddle-node bifurcation, which separates regions of one periodic solution from regions of three periodic solutions. From equation (3.18), Figure 3-2, Figure 3-3, and Figure 3-4 were generated. The boundary can be checked by substituting values for λ and *A* into equations (3.16) and (3.18), then determining the number of real periodic solutions. For example, from Figure 3-2, choosing $\lambda = 0.3$ and $A = 1$ falls within the region of three real periodic solutions. However, for $\lambda = 0.3$ and $A = 0.2$, there is only one real periodic solution.

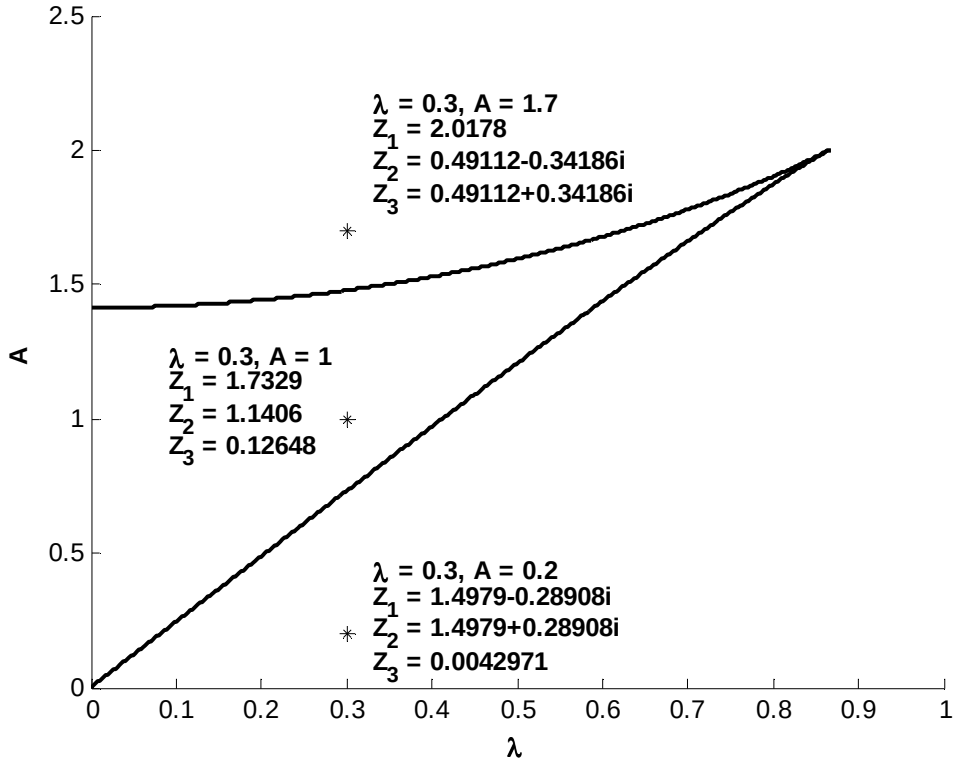


Figure 3-2. Saddle-node bifurcation for $\sigma = 3$ (linear damping)

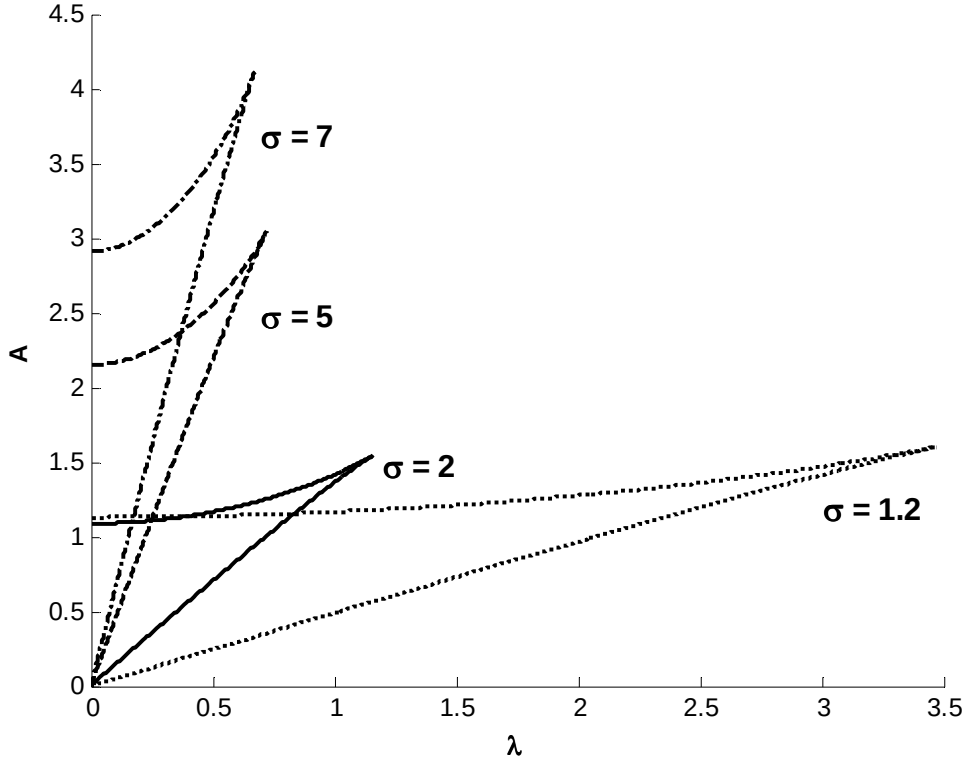


Figure 3-3. Saddle-node bifurcations for positive values of σ (linear damping)

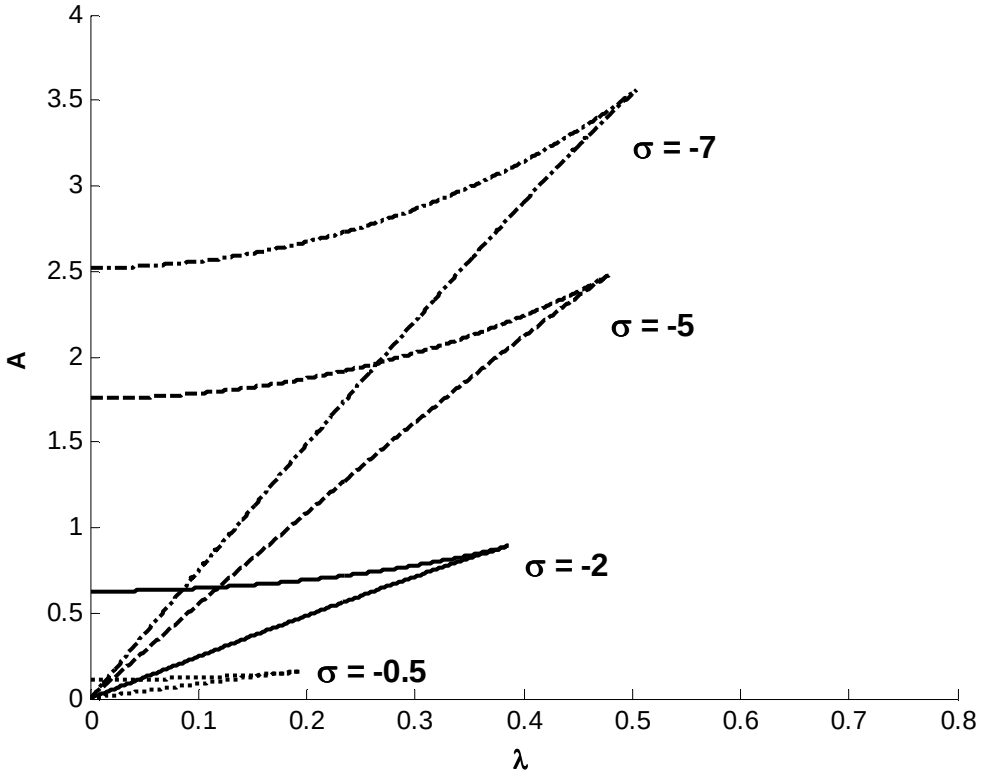


Figure 3-4. Saddle-node bifurcations for negative values of σ (linear damping)

3.2.3 Saddle-Node Bifurcation Analysis for the Nonlinearly Damped System

Continuing the analysis of saddle-node bifurcations, this section focuses on the system with nonlinear damping, given by equation (3.1). The following derivations are based on the work done by Starosvetsky and Gendelman (2008). By making a change of variables according to

$$v = y_1 + \varepsilon y_2, \quad w = y_1 - y_2,$$
$$\varphi_1 e^{it} = \dot{v} + iv, \quad \varphi_2 e^{it} = \dot{w} + iw, \tag{3.19}$$

and omitting exponential terms from the resulting set of equations, we can rewrite the system as

$$\dot{\varphi_1} + \frac{i\varepsilon}{2(1+\varepsilon)}(\varphi_1 - \varphi_2) - \frac{i\varepsilon\sigma}{2(1+\varepsilon)}(\varphi_1 + \varepsilon\varphi_2) = \frac{\varepsilon A}{2}$$

and (3.20)

$$\dot{\varphi_2} + \frac{3}{8}\lambda(1+\varepsilon)|\varphi_2|^2\varphi_2 + \frac{i}{2(1+\varepsilon)}(\varphi_2 - \varphi_1) - \frac{i\varepsilon\sigma}{2(1+\varepsilon)}(\varphi_1 + \varepsilon\varphi_2) - \frac{i}{2}(1+\varepsilon)|\varphi_2|^2\varphi_2 = \frac{\varepsilon A}{2}.$$

Setting the time derivatives of equation (3.20) to zero gives

$$\frac{i\varepsilon}{2(1+\varepsilon)}(\varphi_{10} - \varphi_{20}) - \frac{i\varepsilon\sigma}{2(1+\varepsilon)}(\varphi_{10} + \varepsilon\varphi_{20}) = \frac{\varepsilon A}{2},$$

and (3.21)

$$\frac{3}{8}\lambda(1+\varepsilon)|\varphi_{20}|^2\varphi_{20} + \frac{i}{2(1+\varepsilon)}(\varphi_{20} - \varphi_{10}) - \frac{i\varepsilon\sigma}{2(1+\varepsilon)}(\varphi_{10} + \varepsilon\varphi_{20})$$

$$-\frac{i}{2}(1+\varepsilon)|\varphi_{20}|^2\varphi_{20} = \frac{\varepsilon A}{2},$$

where φ_{10} and φ_{20} are the fixed points of the system. The equations (3.21) lead to

$$\varphi_{10} = \frac{(1+\varepsilon\sigma)\varphi_{20} - i(1+\varepsilon)A}{1-\sigma} \tag{3.22}$$

and

$$\frac{\sigma^2}{(1-\sigma)^2}|\varphi_{20}|^2 + \frac{2\sigma}{(1-\sigma)}|\varphi_{20}|^4 + \left[\frac{9}{16}\lambda^2 + 1\right]|\varphi_{20}|^6 = \frac{A^2}{(1-\sigma)^2}. \tag{3.23}$$

As a simplification, equation (3.23) can be rewritten as

$$\alpha_1 Z + \alpha_2 Z^2 + \alpha_3 Z^3 + \alpha_4 = 0, \tag{3.24}$$

where

$$|\varphi_{20}|^2 = Z, \quad \alpha_1 = \frac{\sigma^2}{(1-\sigma)^2}$$

$$\alpha_2 = \frac{2\sigma}{1-\sigma}, \quad \alpha_3 = \frac{9}{16}\lambda^2 + 1, \quad \alpha_4 = \frac{-A^2}{(1-\sigma)^2}. \tag{3.25}$$

Taking the derivative of equation (3.24) with respect to Z gives

$$3\alpha_3 Z^2 + 2\alpha_2 Z + \alpha_1 = 0. \tag{3.26}$$

Eliminating Z from equations (3.24) and (3.26) as shown in Section 3.A.2 gives

$$\alpha_4 = -\alpha_1\left(\frac{-\alpha_2}{3\alpha_3} \pm \frac{\sqrt{{\alpha_2}^2-3\alpha_1\alpha_3}}{3\alpha_3}\right) - \alpha_2\left(\frac{-\alpha_2}{3\alpha_3} \pm \frac{\sqrt{{\alpha_2}^2-3\alpha_1\alpha_3}}{3\alpha_3}\right)^2 - \alpha_3\left(\frac{-\alpha_2}{3\alpha_3} \pm \frac{\sqrt{{\alpha_2}^2-3\alpha_1\alpha_3}}{3\alpha_3}\right)^3. \quad (3.27)$$

The expression given in equation (3.27) represents the boundary of the saddle-node bifurcation, which separates regions of one periodic solution from regions of three periodic solutions. From equation (3.27), Figure 3-5, Figure 3-6, and Figure 3-7 were generated. The boundary can be checked by substituting values for λ and A into equations (3.15) and (3.27), then determining the number of real periodic solutions. For example, from Figure 3-5, choosing $\lambda = 0.3$ and $A = 1$ falls within the region of three real periodic solutions. However, for $\lambda = 0.3$ and $A = 0.2$, there is only one real periodic solution.

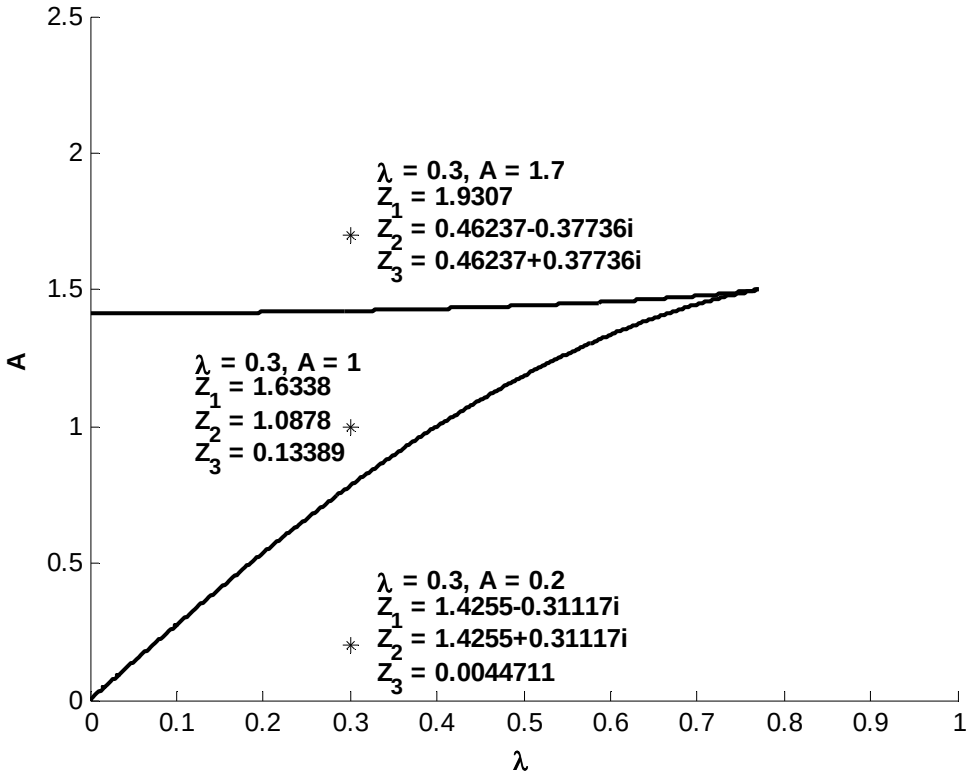


Figure 3-5. Saddle-node bifurcation for $\sigma = 3$ (nonlinear damping)

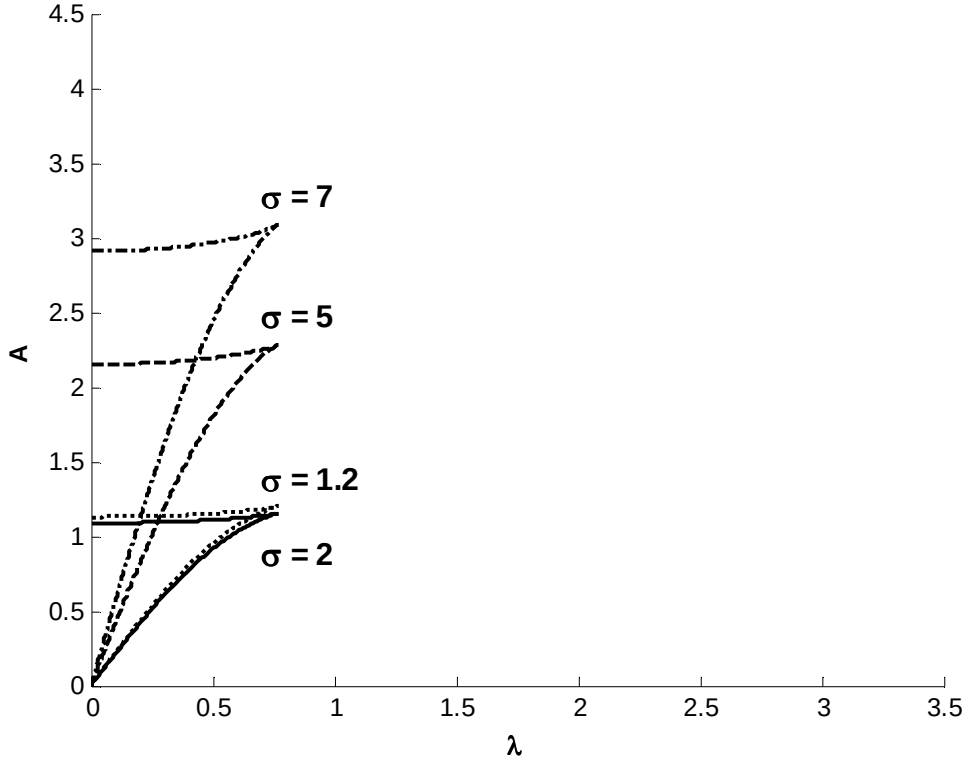


Figure 3-6. Saddle-node bifurcations for positive values of σ (nonlinear damping)

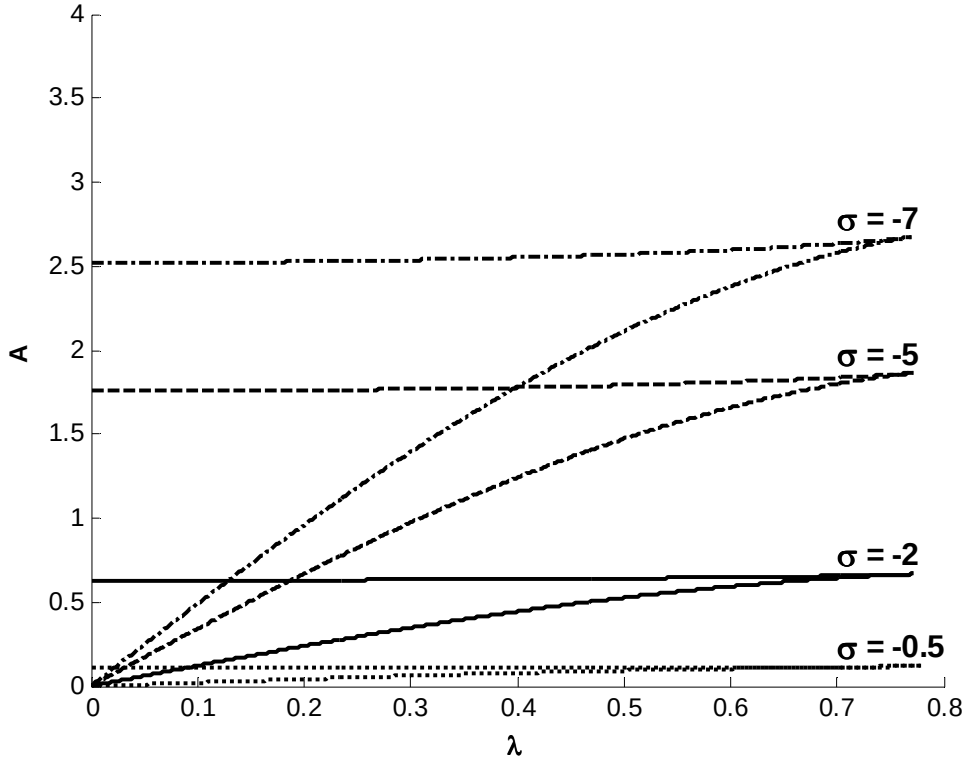


Figure 3-7. Saddle-node bifurcations for negative values of σ (nonlinear damping)

3.2.4 Discussion of Results for Saddle-Node Bifurcations

From inspection of Figures 3-2 through 3-7, it is evident that the saddle-node bifurcations all have the same shape for the various chosen values for σ. Additionally, by observation of the saddle-node bifurcation diagrams, the value of A for the fixed points increases as the magnitude of σ increases.

In Figures 3-2 and 3-5, the solutions are shown at three different points as a spot-check of the different regions. For any λ and A within the region bounded by the two curves, all three periodic solutions are real. However, for all λ and A outside of this region, only one of the three periodic solutions is real, and the other two have imaginary components. Figures 3-3, 3-4, 3-6, and 3-7 represent the same scenario of real periodic solutions for varying values of σ.

Comparing the boundary of the saddle-node bifurcation for the linearly damped system (Figure 3-2) with that for the nonlinearly damped system (Figure 3-5), there has been a clearly visible qualitative change. For both systems, each branch starts off at approximately the same initial value for $\lambda = 0$. The upper branch of the linearly damped system shows a more well-defined concave shape as opposed to the upper branch of the nonlinearly damped system. Conversely, the lower branch of the linearly damped system appears more linear than the curved shape of the lower branch of the nonlinearly damped system. Lastly, it can be seen that the two branches converge for a lower value of λ for the nonlinearly damped system than for the linearly damped system. The same qualitative trend is apparent for different values of sigma, as depicted in Figures 3-3, 3-4, 3-6, and 3-7. These plots show that with an increasing magnitude of σ, there is an

increasing trend in the amplitude, A. However, there is one exception to this trend, since in the nonlinearly damped system the amplitude for $\sigma = 1.2$ is greater than that for $\sigma = 2$.

3.3 Hopf Bifurcation

3.3.1 Hopf Bifurcation Background

The purpose of studying Hopf bifurcations in the context of this thesis is to determine regions of stability of the periodic solutions. Similar to saddle-node bifurcations, Hopf bifurcations represent a qualitative change in the system. For a Hopf bifurcation to exist, the vector field $\boldsymbol{F}$, given in equation (3.2), must equal the zero vector. When the vector field $\boldsymbol{F}$ is equal to the zero vector, let $\boldsymbol{x} = \boldsymbol{x_0}$ and $\mu = \mu_c$. Hence, another necessary criterion for a Hopf bifurcation is that the Jacobian of $\boldsymbol{F}$ (denoted by $D_x\boldsymbol{F}$) must have at least one "pair of purely imaginary eigenvalues, while all of its other eigenvalues have nonzero real parts at" ($\boldsymbol{x_0}; \mu_c$) (Nayfeh and Balachandran, 1995). The final condition for a Hopf bifurcation is that the derivative of the real part of the eigenvalues with respect to μ does not equal zero at μ_c.

An example of a Hopf bifurcation was presented by Nayfeh and Balachandran (1995). With the scalar control parameter μ, the following system is considered:

$$\begin{aligned} \dot{x} &= \mu x - \omega y + (\alpha x - \beta y)(x^2 + y^2) \\ \dot{y} &= \omega x + \mu y + (\beta x + \alpha y)(x^2 + y^2). \end{aligned} \tag{3.28}$$

The system given in equation (3.28) can be written in the form

$$\begin{Bmatrix} \dot{x} \\ \dot{y} \end{Bmatrix} = \boldsymbol{F}(x, y; \mu), \tag{3.29}$$

where the vector $\boldsymbol{F}$ represents the right hand side of equation (3.28). Taking the Jacobian of $\boldsymbol{F}$ from equation (3.29) gives

$$D_x\boldsymbol{F} = \begin{bmatrix} \frac{\partial F(1)}{\partial x} & \frac{\partial F(1)}{\partial x} \\ \frac{\partial F(2)}{\partial x} & \frac{\partial F(2)}{\partial x} \end{bmatrix} = \begin{bmatrix} \frac{\partial \dot{x}}{\partial x} & \frac{\partial \dot{x}}{\partial y} \\ \frac{\partial \dot{y}}{\partial x} & \frac{\partial \dot{y}}{\partial y} \end{bmatrix}. \tag{3.30}$$

Thus,

$$D_x\boldsymbol{F} = \begin{bmatrix} \mu + 3\alpha x^2 + 2\beta xy + \alpha y^2 & -\omega - 3\beta y^2 + 2\alpha xy - \beta x^2 \\ \omega + 3\beta x^2 + 2\alpha xy + \beta y^2 & \mu + 3\alpha y^2 + 2\beta xy + \alpha x^2 \end{bmatrix}. \tag{3.31}$$

From inspection of equation (3.28), it is obvious that (0,0) is a fixed point because at this condition $\dot{x} = 0$ and $\dot{y} = 0$. For this fixed point, the Jacobian reduces to

$$D_x\boldsymbol{F} = \begin{bmatrix} \mu & -\omega \\ \omega & \mu \end{bmatrix}. \tag{3.32}$$

The eigenvalues are found by taking the determinant of $D_x\boldsymbol{F} - \lambda[\mathrm{I}]$ and setting the expression equal to zero. Thus,

$$\begin{vmatrix} \mu - \lambda & -\omega \\ \omega & \mu - \lambda \end{vmatrix} = 0. \tag{3.33}$$

Taking the determinant in equation (3.33) and solving the characteristic equation for λ gives the eigenvalues

$$\lambda_{1,2} = \mu \pm i\omega. \tag{3.34}$$

As shown by Nayfeh and Balachandran (1995), the condition for the derivative of the eigenvalues with respect to μ not being equal to zero is satisfied since

$$\frac{d\lambda_1}{d\mu} = 1 \text{ and } \frac{d\lambda_2}{d\mu} = 1. \tag{3.35}$$

For a Hopf bifurcation to occur, the eigenvalues must be purely imaginary. Thus, letting $\mu = 0$, the eigenvalues become

$$\lambda_{Hopf\ 1,2} = \pm i\omega. \tag{3.36}$$

In order to create the bifurcation diagrams, first equations (3.28) are converted to polar form using the relations

$$x = r\cos\theta \text{ and } y = r\sin\theta. \tag{3.37}$$

The system equations then become

$$\begin{aligned} \dot{r} &= \mu r + \alpha r^3 \\ \dot{\theta} &= \omega + \beta r^2. \end{aligned} \tag{3.38}$$

Fixed points are determined by setting the time derivatives in equation (3.38) equal to zero. Solutions for the trivial and nontrivial fixed points are

$$r = 0 \text{ and } r = \pm i\sqrt{\frac{\mu}{\alpha}}, \tag{3.39}$$

respectively. The Jacobian is found from

$$D_x\boldsymbol{F} = \frac{\partial F}{\partial r} = \frac{\partial \dot{r}}{\partial r} = \mu + 3\alpha r^2. \tag{3.40}$$

The eigenvalues are determined by setting $D_x\boldsymbol{F} - \lambda[\mathrm{I}]$ equal to zero and solving for λ. Thus,

$$\lambda = \mu + 3\alpha r^2. \tag{3.41}$$

For the trivial fixed point, equation (3.41) reduces to

$$\lambda_{r=0} = \mu, \tag{3.42}$$

and for the nontrivial fixed points, equation (3.41) becomes

$$\lambda_{nontrivial} = -2\mu. \tag{3.43}$$

The corresponding bifurcation diagrams for the fixed points from equation (3.39) are presented in Figure 3-8 for α = -1 and Figure 3-9 for α = 1.

From inspection of Figures 3-8 and 3-9, the Hopf bifurcation occurs as predicted at μ = 0. At this point, there is a qualitative change in the system, namely, the number of

solutions and stability. For α = -1 as shown in Figure 3-8, any given negative value of μ results in one stable solution, while any given positive value of μ results in two stable solutions and one unstable solution. Note that stability was assessed by the sign of the real part of the eigenvalues, based on equation (3.42) for the trivial fixed point and equation (3.43) for the nontrivial fixed points. Negative real parts result in stable solutions, and positive real parts result in unstable solutions. A similar, but different, scenario from Figure 3-8 is depicted in Figure 3-9 for α = 1. In this case, a positive value of μ results in a single unstable solution, but negative values of μ result in two unstable solutions and one stable solution.

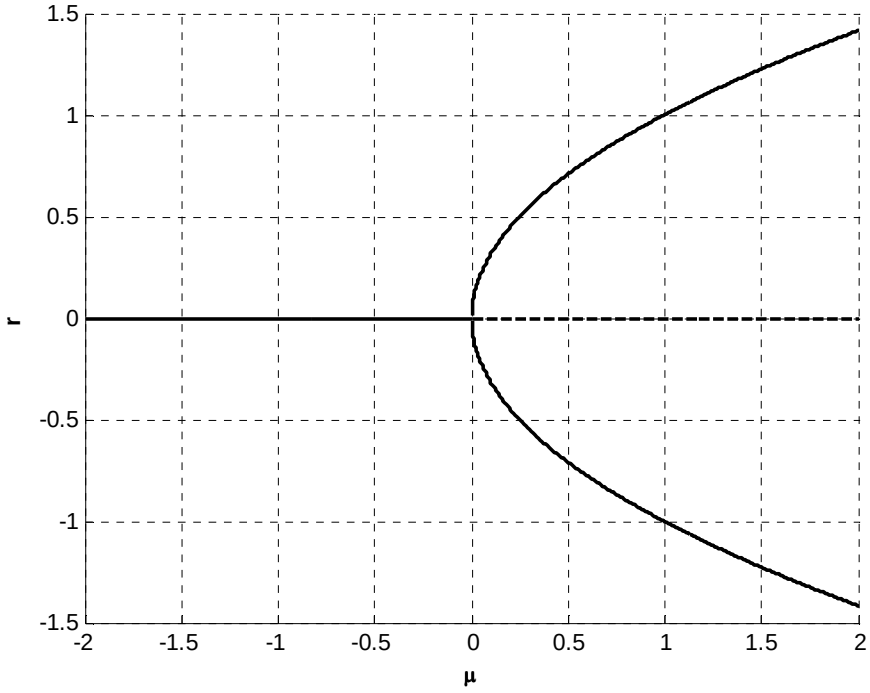


Figure 3-8. Bifurcation diagram of system (3.28) in polar form for α = -1.
——— = stable; – – – – = unstable

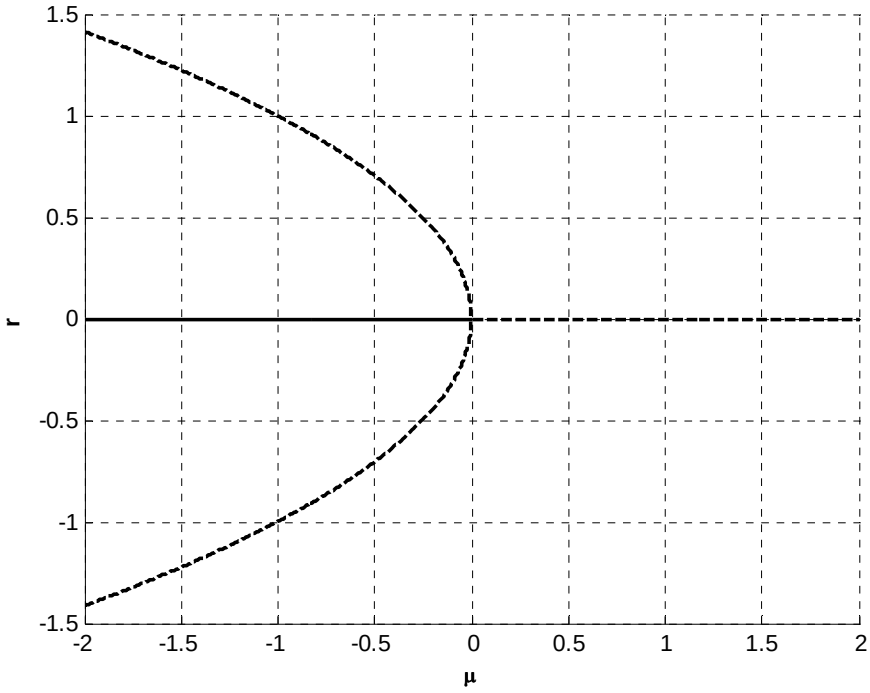


Figure 3-9. Bifurcation diagram of system (3.28) in polar form for $\alpha = 1$. ——— = stable; – – – – = unstable

3.3.2 Hopf Bifurcation Analysis for the System with Linear Damping

For the dynamical system given by equation (3.9), let

$$\varphi_1 = \varphi_{10} + \delta_1, \quad \varphi_2 = \varphi_{20} + \delta_2, \tag{3.44}$$

and substituting into equations (3.11) while omitting the nonlinear terms gives

$$\begin{aligned}
\dot{\delta_1} &= -\frac{i\varepsilon}{2(1+\varepsilon)}(\delta_1 - \delta_2) + \frac{i\varepsilon\sigma}{2(1+\varepsilon)}(\delta_1 + \varepsilon\delta_2), \\
\dot{\delta_1^*} &= \frac{i\varepsilon}{2(1+\varepsilon)}(\delta_1^* - \delta_2^*) - \frac{i\varepsilon\sigma}{2(1+\varepsilon)}(\delta_1^* + \varepsilon\delta_2^*), \\
\dot{\delta_2} &= -\frac{\lambda(1+\varepsilon)}{2}\delta_2 - \frac{i}{2(1+\varepsilon)}(\delta_2 - \delta_1) + \frac{i\varepsilon\sigma}{2(1+\varepsilon)}(\delta_1 + \varepsilon\delta_2) \\
&\quad + i(1+\varepsilon)|\varphi_{20}|^2\delta_2 + \frac{i(1+\varepsilon)}{2}{\varphi_{20}}^2\delta_2^*, \\
\dot{\delta_2^*} &= -\frac{\lambda(1+\varepsilon)}{2}\delta_2^* + \frac{i}{2(1+\varepsilon)}(\delta_2^* - \delta_1^*) - \frac{i\varepsilon\sigma}{2(1+\varepsilon)}(\delta_1^* + \varepsilon\delta_2^*) \\
&\quad - i(1+\varepsilon)|\varphi_{20}|^2\delta_2^* - \frac{i(1+\varepsilon)}{2}{\varphi_{20}^*}^2\delta_2.
\end{aligned} \tag{3.45}$$

The characteristic polynomial can be written as

$$\mu^4 + \gamma_1\mu^3 + \gamma_2\mu^2 + \gamma_3\mu + \gamma_4 = 0, \tag{3.46}$$

where μ are the eigenvalues, and

$$|\varphi_{20}| = N_{20}, \quad {N_{20}}^4 = {\varphi_{20}}^2{\varphi_{20}^*}^2, \tag{3.47}$$

$$\gamma_1 = \lambda(1+\varepsilon),$$

$$\gamma_2 = \left(\frac{3}{2}\varepsilon + \frac{3}{4} + \frac{3}{4}\varepsilon^2\right){N_{20}}^4 + (\varepsilon^2\sigma - 1){N_{20}}^2 + \frac{1}{4}\lambda^2(\varepsilon+1)^2 + \frac{1}{4}(\varepsilon^2\sigma^2 + 1),$$

$$\gamma_3 = \tfrac{1}{4}\lambda\varepsilon(\varepsilon\sigma^2 + 1),$$

$$\gamma_4 = \tfrac{3}{16}\varepsilon^2(1-\sigma)^2{N_{20}}^4 + \tfrac{1}{4}\varepsilon^2\sigma(1-\sigma){N_{20}}^2 + \tfrac{1}{16}\varepsilon^2[(1-\sigma)^2\lambda^2 + \sigma^2].$$

For the Hopf bifurcation to occur, we must have

$$\mu = \pm i\Omega, \tag{3.48}$$

in which Ω is a real number. Substituting equation (3.48) into equation (3.46) and separating into real and imaginary parts gives

$$\Omega^4 - \gamma_2\Omega^2 + \gamma_4 = 0, \quad \Omega(\gamma_1\Omega^2 - \gamma_3) = 0 \Rightarrow \Omega^2 = \frac{\gamma_3}{\gamma_1}. \tag{3.49}$$

Substituting the relation for Ω^2 into the first equation of equation (3.49) gives

$${\gamma_3}^2 - \gamma_2\gamma_3\gamma_1 + \gamma_4{\gamma_1}^2 = 0. \tag{3.50}$$

MATLAB was used to determine the coefficients, v_i, for

$$v_1z^2 + v_2z + v_3 = 0, \tag{3.51}$$

based on equation (3.50). Solving for z in equation (3.51) gives

$$z_{1,2} = \frac{-v_2 \mp \sqrt{{v_2}^2 - 4v_3v_1}}{2v_1}, \tag{3.52}$$

and from equation (3.15), the boundaries of stability are given by

$$\alpha_1z_i + \alpha_2{z_i}^2 + \alpha_3{z_i}^3 + \alpha_4 = 0; \quad z_i = Z = {N_{20}}^2; \quad i = 1,2. \tag{3.53}$$

Refer to Section 3.A.4 for details on the remainder of the analysis. Figures 3-10 through 3-14 depict bifurcation diagrams, amplitude-response, and frequency-response for the system.

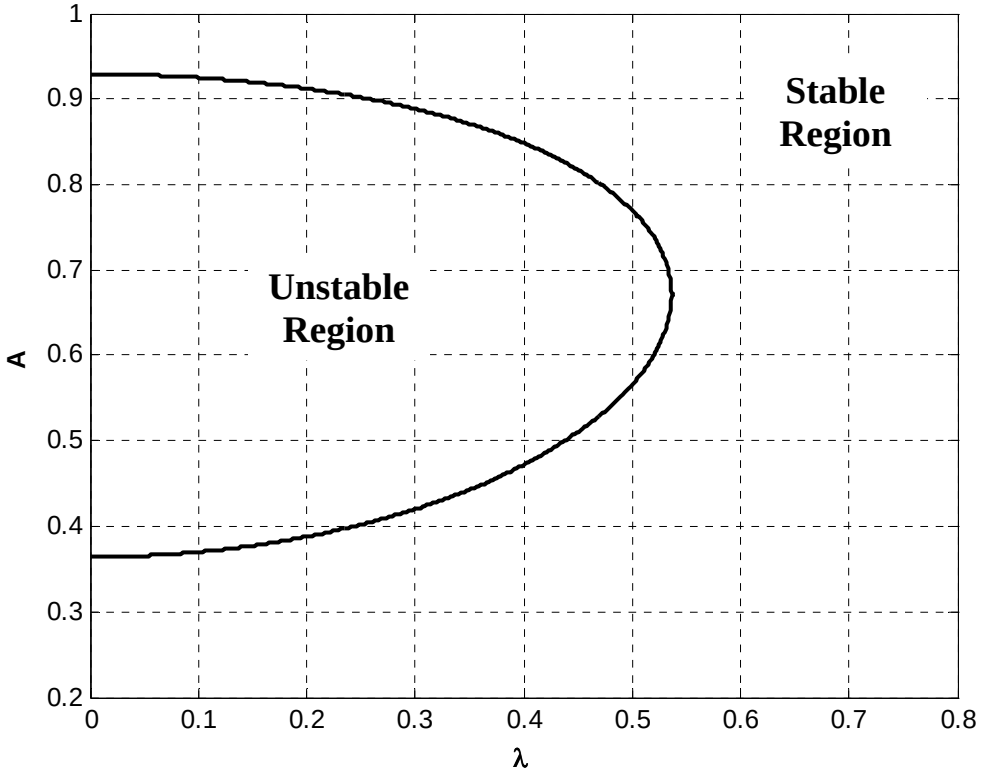


Figure 3-10. Hopf bifurcation for $\sigma = 0.5$, $\varepsilon = 0.05$ (linear damping)

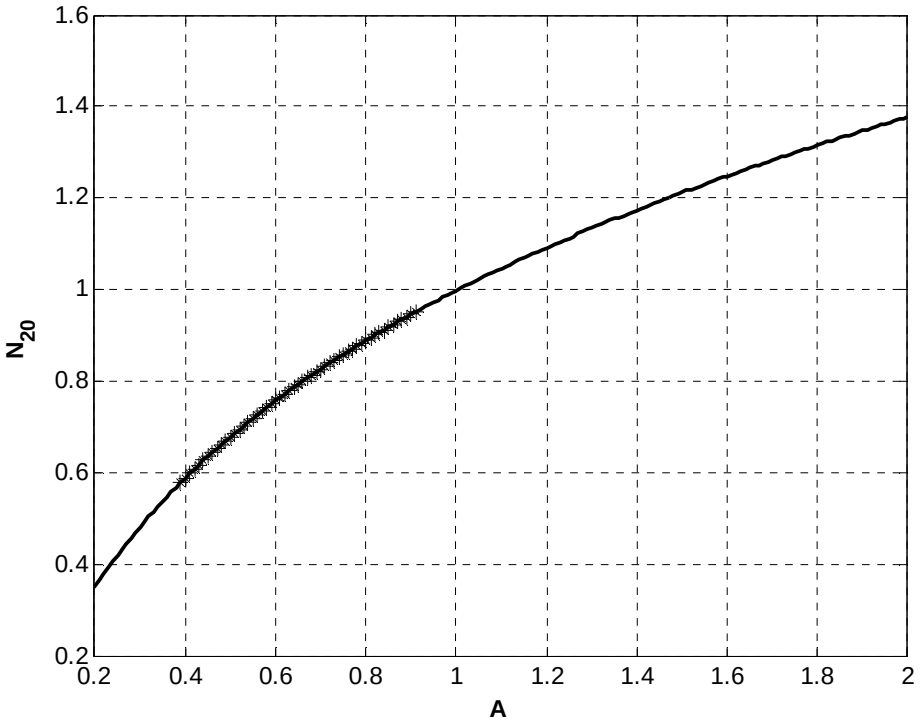


Figure 3-11. Amplitude-response for $\sigma = 0.5$, $\varepsilon = 0.05$, $\lambda = 0.2$ (linear damping)
——— = stable; ********* = unstable

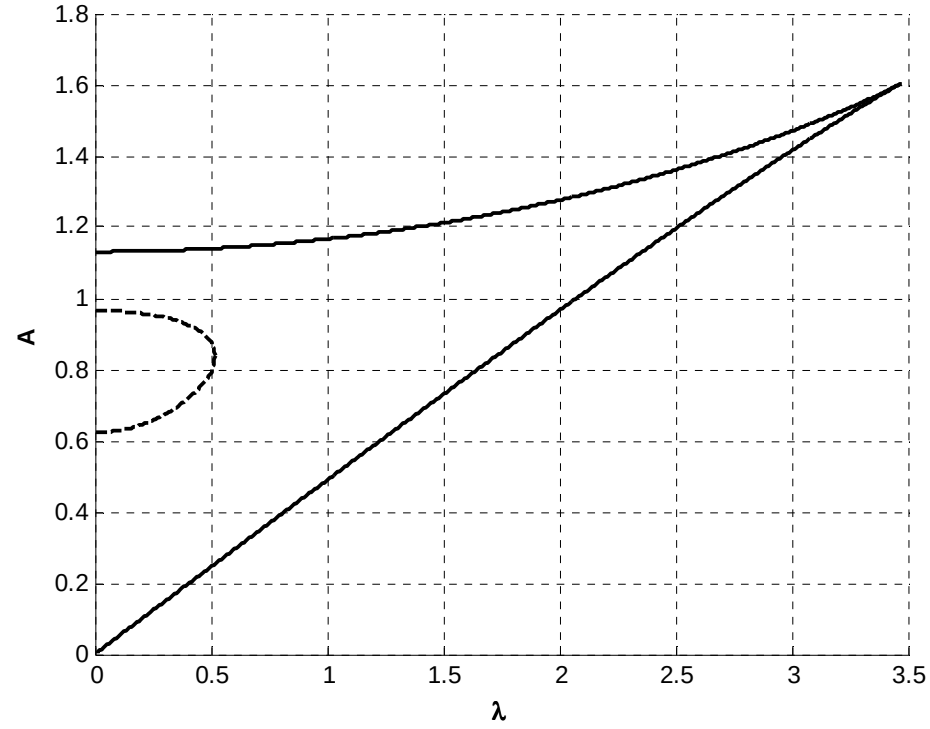


Figure 3-12. Hopf and saddle-node bifurcations for $\sigma = 1.2$, $\varepsilon = 0.05$ (linear damping)
——— = saddle-node; – – – – = Hopf

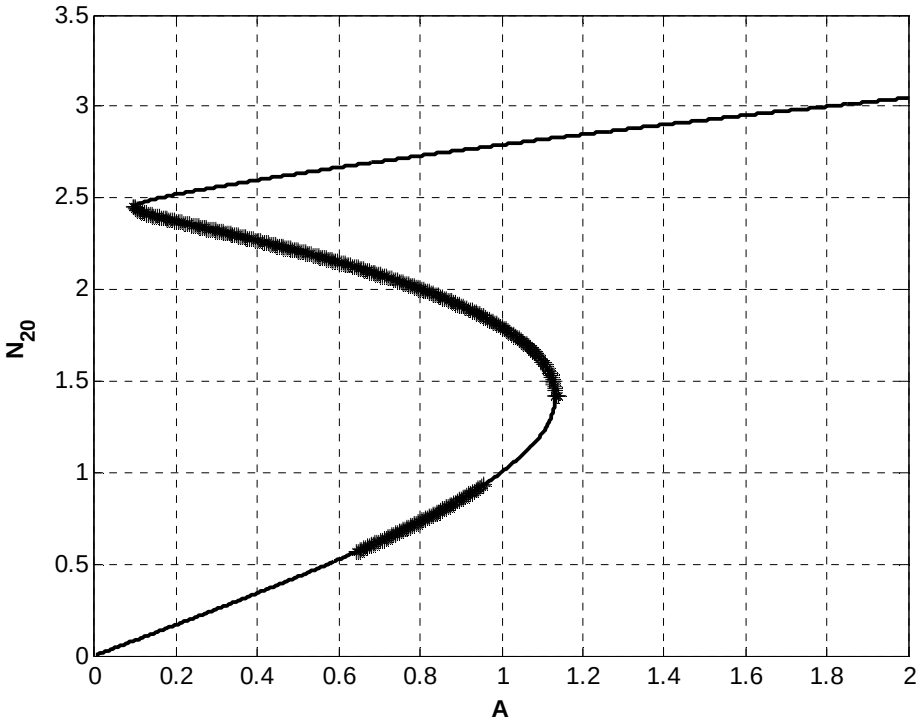


Figure 3-13. Amplitude-response for $\sigma = 1.2$, $\varepsilon = 0.05$, $\lambda = 0.2$ (linear damping)
——— = stable; ********* = unstable

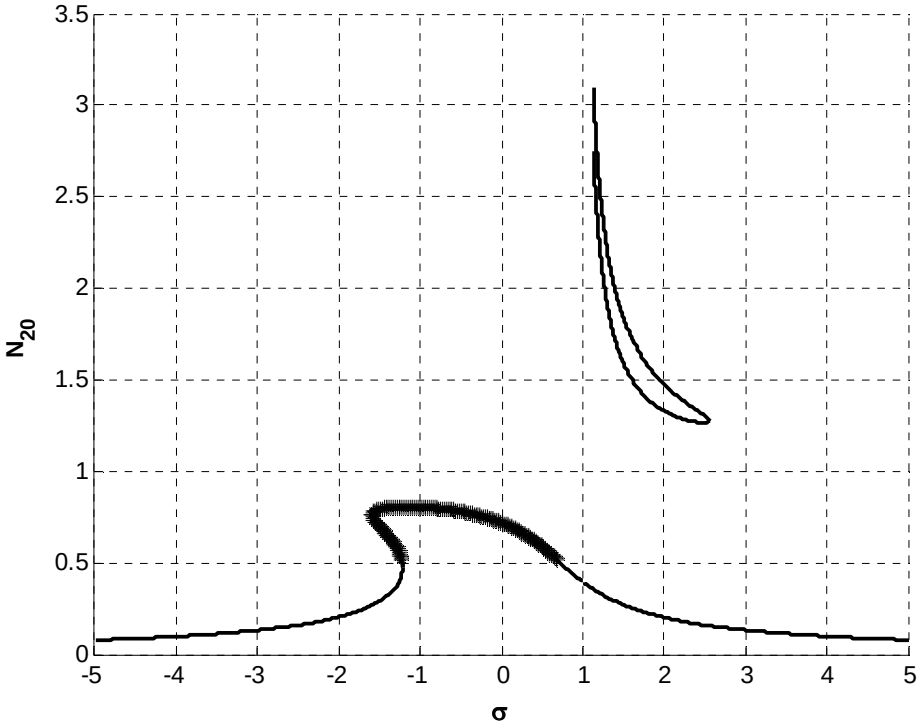


Figure 3-14. Frequency-response for $A = 0.4$, $\varepsilon = 0.05$, $\lambda = 0.2$ (linear damping)
——— = stable; ********* = unstable

3.3.3 Hopf Bifurcation Analysis for the System with Nonlinear Damping

For the dynamical system given by equation (3.1), let

$$\varphi_1 = \varphi_{10} + \delta_1, \quad \varphi_2 = \varphi_{20} + \delta_2, \tag{3.54}$$

and substituting into equation (3.20) while omitting the nonlinear terms gives

$$\begin{aligned} \dot{\delta_1} &= -\tfrac{i\varepsilon}{2(1+\varepsilon)}(\delta_1 - \delta_2) + \tfrac{i\varepsilon\sigma}{2(1+\varepsilon)}(\delta_1 + \varepsilon\delta_2), \\ \dot{\delta_1^*} &= \tfrac{i\varepsilon}{2(1+\varepsilon)}(\delta_1^* - \delta_2^*) - \tfrac{i\varepsilon\sigma}{2(1+\varepsilon)}(\delta_1^* + \varepsilon\delta_2^*), \end{aligned} \tag{3.55}$$

$$\dot{\delta_2} = -\tfrac{i}{2(1+\varepsilon)}(\delta_2 - \delta_1) + \tfrac{i\varepsilon\sigma}{2(1+\varepsilon)}(\delta_1 + \varepsilon\delta_2) - \tfrac{3\lambda(1+\varepsilon)}{4}|\varphi_{20}|^2\delta_2 + i(1+\varepsilon)|\varphi_{20}|^2\delta_2 -$$

$$\tfrac{3\lambda(1+\varepsilon)}{8}{\varphi_{20}}^2\delta_2^* + \tfrac{i(1+\varepsilon)}{2}{\varphi_{20}}^2\delta_2^*,$$

$$\dot{\delta_2^*} = \tfrac{i}{2(1+\varepsilon)}(\delta_2^* - \delta_1^*) - \tfrac{i\varepsilon\sigma}{2(1+\varepsilon)}(\delta_1^* + \varepsilon\delta_2^*) - \tfrac{3\lambda(1+\varepsilon)}{4}|\varphi_{20}|^2\delta_2^* - i(1+\varepsilon)|\varphi_{20}|^2\delta_2^* -$$

$$\tfrac{3\lambda(1+\varepsilon)}{8}{\varphi_{20}^*}^2\delta_2 - \tfrac{i(1+\varepsilon)}{2}{\varphi_{20}^*}^2\delta_2.$$

The characteristic polynomial can be written as

$$\mu^4 + \gamma_1\mu^3 + \gamma_2\mu^2 + \gamma_3\mu + \gamma_4 = 0, \tag{3.56}$$

where μ are the eigenvalues, and

$$|\varphi_{20}| = N_{20}, \quad {N_{20}}^4 = {\varphi_{20}}^2{\varphi_{20}^*}^2, \tag{3.57}$$

$$\gamma_1 = \lambda(1+\varepsilon),$$

$$\gamma_2 = \left(\frac{3}{2}\varepsilon + \frac{3}{4} + \frac{3}{4}\varepsilon^3\right){N_{20}}^4 + (\varepsilon^2\sigma - 1){N_{20}}^2 + \frac{1}{4}\lambda^2(\varepsilon+1)^2 + \frac{1}{4}(\varepsilon^2\sigma^2 + 1),$$

$$\gamma_3 = \tfrac{1}{4}\lambda\varepsilon(\varepsilon\sigma^2 + 1),$$

$$\gamma_4 = \tfrac{3}{16}\varepsilon^2(1-\sigma)^2{N_{20}}^4 + \tfrac{1}{4}\varepsilon^2\sigma(1-\sigma){N_{20}}^2 + \tfrac{1}{16}\varepsilon^2[(1-\sigma)^2\lambda^2 + \sigma^2].$$

For the Hopf bifurcation to occur, we must have

$$\mu = \pm i\Omega, \tag{3.58}$$

in which Ω is a real number. Substituting equation (3.58) into equation (3.56) and separating into real and imaginary parts gives

$$\Omega^4 - \gamma_2\Omega^2 + \gamma_4 = 0, \quad \Omega(\gamma_1\Omega^2 - \gamma_3) = 0 \Rightarrow \Omega^2 = \frac{\gamma_3}{\gamma_1}. \tag{3.59}$$

Substituting the relation for Ω^2 into the first equation of equation (3.59) gives

$$\gamma_3{}^2 - \gamma_2\gamma_3\gamma_1 + \gamma_4\gamma_1{}^2 = 0. \tag{3.60}$$

MATLAB was used to determine the coefficients, v_i, for

$$v_1 z^2 + v_2 z + v_3 = 0, \tag{3.61}$$

based on equation (3.60). Solving for z in equation (3.61) gives

$$z_{1,2} = \frac{-v_2 \mp \sqrt{v_2{}^2 - 4v_3v_1}}{2v_1}, \tag{3.62}$$

and from equation (3.24), the boundaries of stability are given by

$$\alpha_1 z_i + \alpha_2 z_i{}^2 + \alpha_3 z_i{}^3 + \alpha_4 = 0; \quad z_i = Z = N_{20}{}^2; \quad i = 1,2. \tag{3.63}$$

Refer to Section 3.A.5 for details on the remainder of the analysis. Figures 3-15 through 3-19 depict bifurcation diagrams, amplitude-response, and frequency-response for the system.

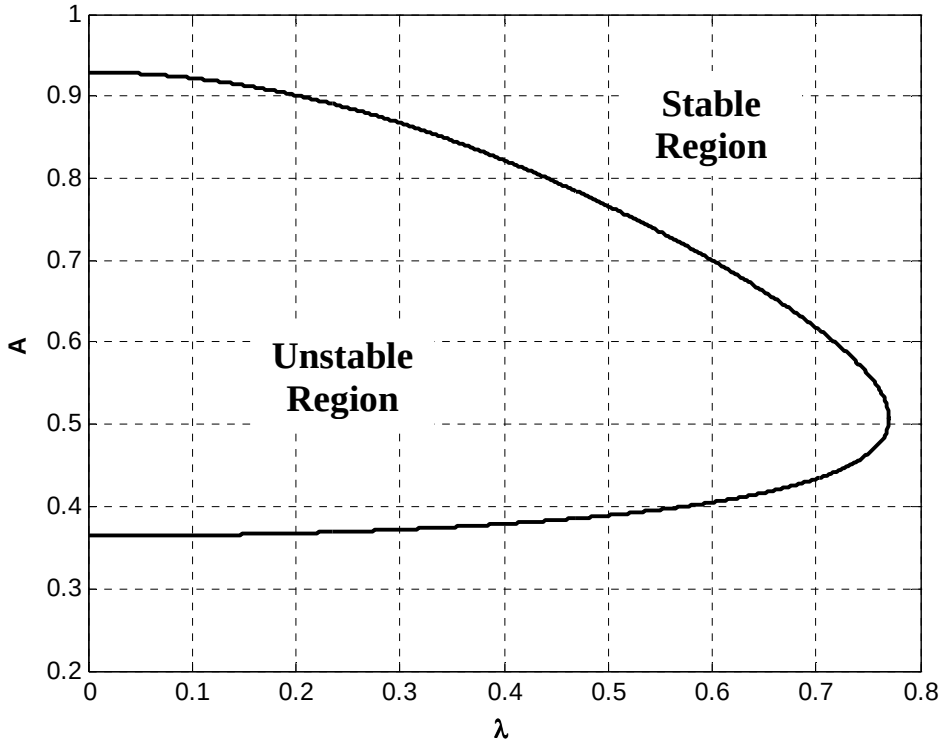


Figure 3-15. Hopf bifurcation for $\sigma = 0.5$, $\varepsilon = 0.05$ (nonlinear damping)

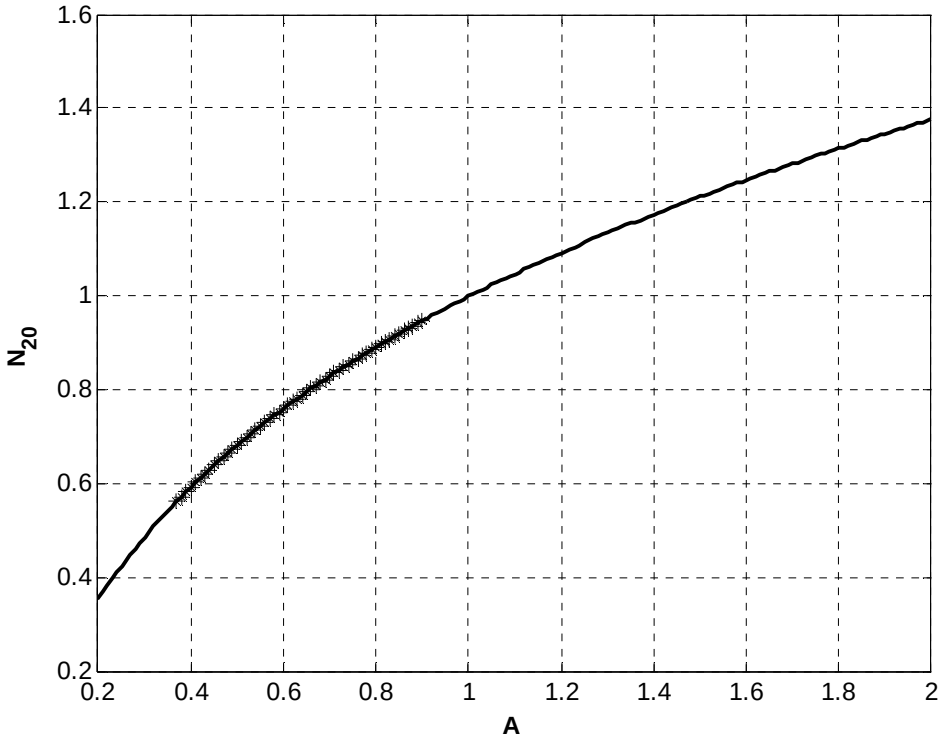


Figure 3-16. Amplitude-response for $\sigma = 0.5$, $\varepsilon = 0.05$, $\lambda = 0.2$ (nonlinear damping)
——— = stable; ********* = unstable

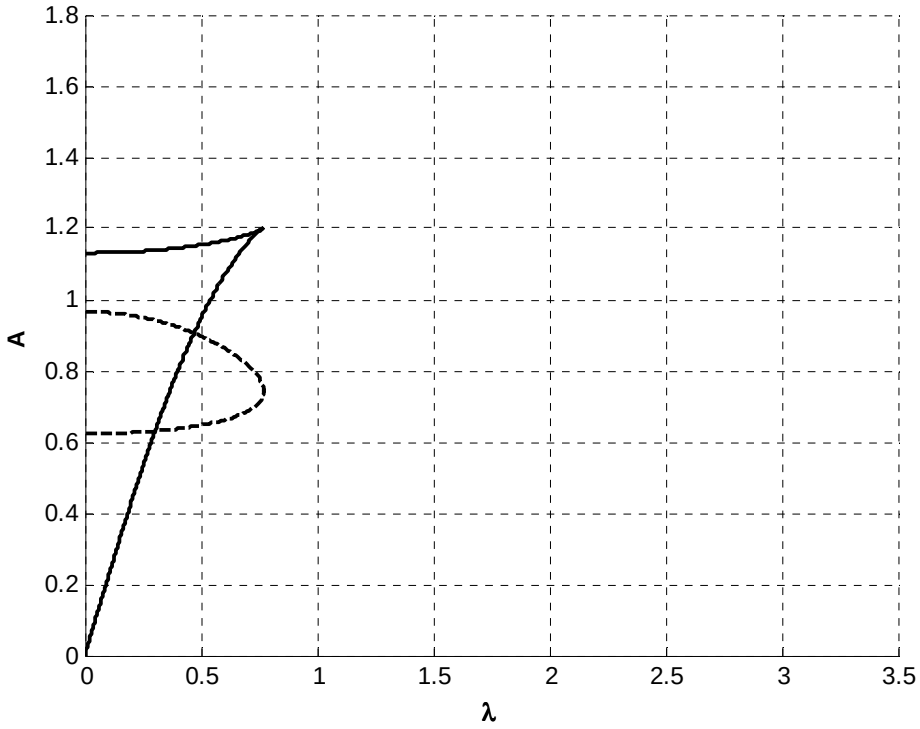


Figure 3-17. Hopf and saddle-node bifurcations for $\sigma = 1.2$, $\varepsilon = 0.05$ (nonlinear damping)
——— = saddle-node; – – – – = Hopf

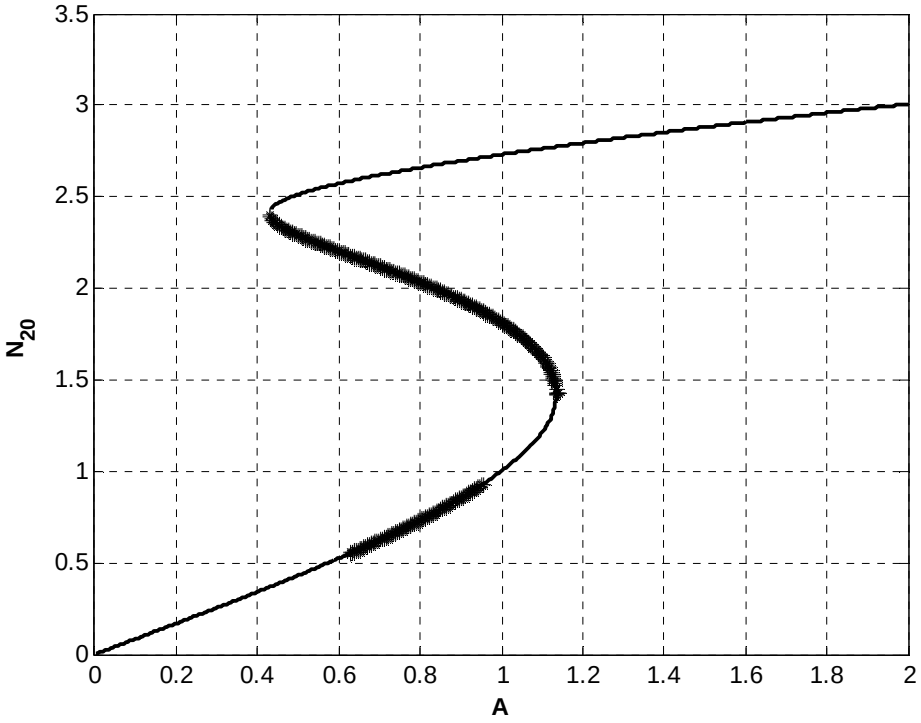


Figure 3-18. Amplitude-response for $\sigma = 1.2$, $\varepsilon = 0.05$, $\lambda = 0.2$ (nonlinear damping)
——— = stable; ********* = unstable

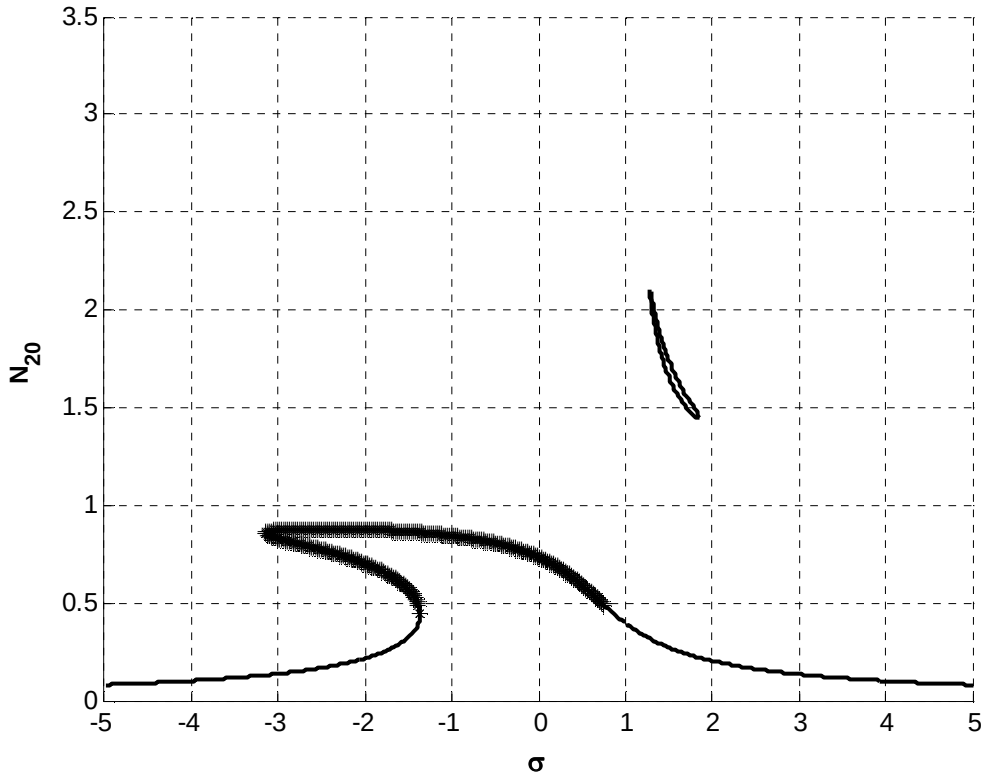


Figure 3-19. Frequency-response for $A = 0.4$, $\varepsilon = 0.05$, $\lambda = 0.2$ (nonlinear damping)
——— = stable; ********* = unstable

3.3.4 Discussion of Results for Hopf Bifurcations

Figures 3-10 and 3-15 show the boundary of the Hopf bifurcation in λ-A space for $\sigma = 0.5$ for the linearly damped and nonlinearly damped systems, respectively. The unstable region is bounded by the curve, and the stable region is the area outside the boundary. As shown in Figures 3-11 and 3-16, the regions of stability are confirmed in A-N_{20} space. Note that N_{20} is related to the response of the system since $|\varphi_{20}|=N_{20}$. To determine stability in Figure 3-11, the sign of the real part of the eigenvalues was evaluated. Positive real parts indicated unstable regions, and negative real parts indicated stable regions.

Similarly, Figures 3-12 and 3-17 show regions of stability of the Hopf bifurcation, but now the saddle-node bifurcation is plotted as well. In this case, the frequency detuning parameter, σ, is equal to 1.2. For the parameters chosen, the saddle-node bifurcation also represents a boundary of stability. Verification is shown in Figures 3-13 and 3-18 for $\lambda = 0.2$. As before, the values of the Hopf bifurcation from Figures 3-12 and 3-17 at $\lambda = 0.2$ represent the unstable regions shown in Figures 3-13 and 3-18, respectively. The same is true for the saddle-node bifurcation shown in Figures 3-12 and 3-17, which depicts another unstable region on the response curve in Figures 3-13 and 3-18, respectively. Similar to the previous case, stability was assessed by checking the sign of the real parts of the eigenvalues.

Dependence of the response on the frequency detuning parameter is depicted in Figures 3-14 and 3-19. Eigenvalues were used again to evaluate stability as shown on the response curve. Note that this frequency-response curve can be spot-checked for chosen parameters. For example, in the nonlinearly damped system, taking a point in the unstable region from Figures 3-15 and 3-16, say $\lambda = 0.2$, $\sigma = 0.5$, and $A = 0.4$, this location corresponds to an instability on the curve in Figure 3-19. Similarly, taking the stable point $\lambda = 0.2$, $\sigma = 1.2$, and $A = 0.4$ from Figures 3-17 and 3-18, this point is represented as a stable region of the response shown in Figure 3-19 as well. Due to these spot-checks, agreement between the bifurcation diagrams, amplitude-response, and frequency-response has been validated for the chosen parameters. The spot-check can be applied in a similar manner to the plots shown for the linearly damped system.

The Hopf bifurcations of the linearly damped system appear qualitatively different from those of the nonlinearly damped system. Comparing Figure 3-10 with

Figure 3-15, it is evident that although both plots begin with the same amplitude at $\lambda = 0$, the shapes of the plots are clearly different. The plot given in Figure 3-10 appears roughly symmetrical about approximately $A = 0.65$, while the plot for the nonlinearly damped system given in Figure 3-15 shows a more decreasing trend of the upper branch and a more leveling trend of the lower branch. In addition, the nonlinearly damped system bifurcation point occurs at a higher value of λ than that of the linearly damped system.

Comparing Figure 3-11 with Figure 3-16, there is no apparent change in the response between the linearly damped and nonlinearly damped systems for the chosen parameters. However, the boundary of the bifurcations shown in Figures 3-12 and 3-17 are qualitatively different. In the linearly damped system, it can be seen that the amplitude for the Hopf bifurcation is contained within the amplitude of the saddle-node bifurcation. However, in the nonlinearly damped system, the amplitude of the Hopf bifurcation extends past the boundary of the saddle-node bifurcation. This change is attributed more to a change in the saddle-node bifurcation than the Hopf bifurcation since in the nonlinearly damped system, the saddle-node bifurcation point occurs at a lower value of λ than that for the linearly damped system.

The response, N_{20}, shown in Figure 3-13 is qualitatively similar to that shown in Figure 3-18. For the chosen parameters, the only apparent difference in the plots is the value of the amplitude, A, at which the upper part of the response changes direction. In the nonlinearly damped system, this change of direction occurs at a higher amplitude than that for the linearly damped system.

The frequency response plots for both systems (shown in Figures 3-14 and 3-19, respectively) are qualitatively similar as well. The notable changes in the response characteristics are that for the nonlinearly damped system, the upper part of the lower response changes direction at a lower value of σ as opposed to the linearly damped system. Additionally, the upper response of the nonlinearly damped system has a lower range of response, N_{20}, and σ than that for the linearly damped system.

3.4 Conclusions

As demonstrated in this chapter, the nonlinearly damped system gives qualitatively different results than the linearly damped system. By simply cubing the velocity difference in the linearly damped system equations, the system has been modified to produce clearly different results for chosen parameters. The most notable differences appear to be the location of the bifurcation points, which set the trend for the plots. However, the modification of the linear velocity difference to the cubic velocity difference also changed the appearance of the plots for the saddle-node bifurcations and chosen parameters for the Hopf bifurcation.

3.A Appendix

3.A.1 Saddle-Node Bifurcation Equation Derivations (Linear Damping)

The system is given by

$$\begin{gathered}\ddot{y_1} + \varepsilon\lambda(\dot{y_1} - \dot{y_2}) + (1 + \varepsilon\sigma)y_1 + \frac{4}{3}\varepsilon(y_1 - y_2)^3 = \varepsilon A\cos t \\ \varepsilon\ddot{y_2} + \varepsilon\lambda(\dot{y_2} - \dot{y_1}) + \frac{4}{3}\varepsilon(y_2 - y_1)^3 = 0.\end{gathered} \tag{3.64}$$

Let

$$v = y_1 + \varepsilon y_2, \quad w = y_1 - y_2. \tag{3.65}$$

Making substitutions of the second equation of equations (3.65) into equations (3.64) gives

$$\begin{gathered}\ddot{y_1} + \varepsilon\lambda\dot{w} + (1 + \varepsilon\sigma)y_1 + \frac{4}{3}\varepsilon w^3 = \varepsilon A\cos t \\ \varepsilon\ddot{y_2} - \varepsilon\lambda\dot{w} - \frac{4}{3}\varepsilon w^3 = 0.\end{gathered} \tag{3.66}$$

From the second equation of equations (3.65) and the second equation of equations (3.66), the following can be written:

$$\ddot{y_1} = \ddot{w} + \ddot{y_2}, \quad \ddot{y_2} = \lambda\dot{w} + \frac{4}{3}w^3. \tag{3.67}$$

Combining equations (3.67), gives

$$\ddot{y_1} = \ddot{w} + \lambda\dot{w} + \frac{4}{3}w^3. \tag{3.68}$$

From the first equation of equations (3.65) and the first equation of equations (3.66), we can write

$$\varepsilon\ddot{y_2} = \ddot{v} - \ddot{y_1}, \quad \ddot{y_1} = \varepsilon A\cos t - \varepsilon\lambda\dot{w} - (1 + \varepsilon\sigma)y_1 - \frac{4}{3}\varepsilon w^3. \tag{3.69}$$

Combining equations (3.69) results in

$$\varepsilon\ddot{y_2} = \ddot{v} - \varepsilon A\cos t + \varepsilon\lambda\dot{w} + (1 + \varepsilon\sigma)y_1 + \frac{4}{3}\varepsilon w^3. \tag{3.70}$$

In order to eliminate the y_1 term from equation (3.70), equations (3.65) can be combined, resulting in

$$y_1 = \frac{v+\varepsilon w}{1+\varepsilon}, \tag{3.71}$$

and substitution of equation (3.71) into equation (3.70) leads to

$$\varepsilon \ddot{y}_2 = \ddot{v} - \varepsilon A \cos t + \varepsilon\lambda\dot{w} + (1+\varepsilon\sigma)\frac{v+\varepsilon w}{1+\varepsilon} + \frac{4}{3}\varepsilon w^3. \tag{3.72}$$

Substituting equations (3.68), (3.71), and (3.72) into equation (3.66) results in

$$\begin{aligned} \ddot{v} + (1+\varepsilon\sigma)\frac{v+\varepsilon w}{1+\varepsilon} &= \varepsilon A \cos t \\ \ddot{w} + (1+\varepsilon\sigma)\frac{v+\varepsilon w}{1+\varepsilon} + (1+\varepsilon)\lambda\dot{w} + \frac{4}{3}(1+\varepsilon)w^3 &= \varepsilon A \cos t. \end{aligned} \tag{3.73}$$

Let

$$\varphi_1 e^{it} = \dot{v} + iv, \quad \varphi_2 e^{it} = \dot{w} + iw. \tag{3.74}$$

Rearranging equation (3.74) for φ_1 and φ_2 gives

$$\varphi_1 = (\dot{v} + iv)e^{-it} \quad \text{and} \quad \varphi_2 = (\dot{w} + iw)e^{-it}, \tag{3.75}$$

with the complex conjugates

$$\varphi_1^* = (\dot{v} - iv)e^{it} \quad \text{and} \quad \varphi_2^* = (\dot{w} - iw)e^{it}. \tag{3.76}$$

Also, it easily can be verified that

$$\dot{\varphi}_1 = (\ddot{v} + v)e^{-it}, \quad \dot{\varphi}_2 = (\ddot{w} + w)e^{-it}. \tag{3.77}$$

Expanding the first equation of equations (3.73) and rewriting the cosine term on the right-hand side of the equation gives

$$\ddot{v} + \frac{(1+\varepsilon\sigma)}{1+\varepsilon}v + \frac{(1+\varepsilon\sigma)}{1+\varepsilon}\varepsilon w = \frac{\varepsilon A}{2}(e^{it} + e^{-it}). \tag{3.78}$$

Further manipulation of the right-hand side of the equation gives

$$\ddot{v} + \frac{(1+\varepsilon\sigma)}{1+\varepsilon}v + \frac{(1+\varepsilon\sigma)}{1+\varepsilon}\varepsilon w = \frac{\varepsilon A}{2}(1 + e^{-2it})e^{it}. \tag{3.79}$$

Multiplying both sides of equation (3.79) by e^{-it} gives

$$\ddot{v}e^{-it} + \frac{(1+\varepsilon\sigma)}{1+\varepsilon} v e^{-it} + \frac{(1+\varepsilon\sigma)}{1+\varepsilon} \varepsilon w e^{-it} = \frac{\varepsilon A}{2}(1 + e^{-2it}). \tag{3.80}$$

Equation (3.80) can be rewritten as

$$(\ddot{v} + v)e^{-it} + \left[\frac{(1+\varepsilon\sigma)}{1+\varepsilon} - 1\right] v e^{-it} + \frac{(1+\varepsilon\sigma)}{1+\varepsilon} \varepsilon w e^{-it} = \frac{\varepsilon A}{2}(1 + e^{-2it}). \tag{3.81}$$

Substituting the first equation of equation (3.77) into equation (3.81) and rewriting terms gives

$$\dot{\varphi}_1 + \left[\frac{(1+\varepsilon\sigma)}{1+\varepsilon} - \frac{1+\varepsilon}{1+\varepsilon}\right] v e^{-it} + \left[\frac{\varepsilon}{1+\varepsilon} + \frac{\varepsilon^2\sigma}{1+\varepsilon}\right] \varepsilon w e^{-it} = \frac{\varepsilon A}{2}(1 + e^{-2it}), \tag{3.82}$$

which reduces to

$$\dot{\varphi}_1 + \frac{(\varepsilon\sigma-\varepsilon)}{1+\varepsilon} v e^{-it} + \left[\frac{\varepsilon}{1+\varepsilon} + \frac{\varepsilon^2\sigma}{1+\varepsilon}\right] w e^{-it} = \frac{\varepsilon A}{2}(1 + e^{-2it}). \tag{3.83}$$

Equation (3.83) can be rewritten as

$$\dot{\varphi}_1 + \frac{\varepsilon\sigma}{1+\varepsilon}(v + \varepsilon w)e^{-it} + \frac{\varepsilon}{1+\varepsilon}(w - v)e^{-it} = \frac{\varepsilon A}{2}(1 + e^{-2it}). \tag{3.84}$$

Making use of equations (3.75) and (3.76) to eliminate *v* and *w* from equation (3.84) results in

$$\begin{aligned} &\dot{\varphi}_1 + \frac{i\varepsilon\sigma}{2(1+\varepsilon)}\left(-\varphi_1 - \varepsilon\varphi_2 + \varphi_1^* e^{-2it} + \varepsilon\varphi_2^* e^{-2it}\right) \\ &+ \frac{i\varepsilon}{2(1+\varepsilon)}\left(\varphi_1 - \varphi_2 - \varphi_1^* e^{-2it} + \varphi_2^* e^{-2it}\right) \\ &= \frac{\varepsilon A}{2}(1 + e^{-2it}). \end{aligned} \tag{3.85}$$

Expanding the second equation of equation (3.73) and rewriting the cosine term on the right-hand side of the equation gives

$$\ddot{w} + \frac{(1+\varepsilon\sigma)}{1+\varepsilon} v + \frac{(1+\varepsilon\sigma)}{1+\varepsilon} \varepsilon w + (1+\varepsilon)\lambda\dot{w} + \frac{4}{3}(1+\varepsilon)w^3 = \frac{\varepsilon A}{2}(1 + e^{-2it})e^{it}. \tag{3.86}$$

Using equation (3.77), the second derivative of *w* can be eliminated as

$$\left(\dot{\varphi}_2 e^{it} - w\right) + \frac{(1+\varepsilon\sigma)}{1+\varepsilon} v + \frac{(1+\varepsilon\sigma)}{1+\varepsilon} \varepsilon w + (1+\varepsilon)\lambda \dot{w} + \frac{4}{3}(1+\varepsilon)w^3 = \frac{\varepsilon A}{2}(1+e^{-2it})e^{it}. \tag{3.87}$$

Multiplying both sides of equation (3.87) by e^{-it} and rewriting gives

$$\dot{\varphi}_2 - we^{-it} + \frac{1}{1+\varepsilon}(v+\varepsilon w)e^{-it} + \frac{\varepsilon\sigma}{1+\varepsilon}(v+\varepsilon w)e^{-it} + (1+\varepsilon)\lambda \dot{w} e^{-it} + \frac{4}{3}(1+\varepsilon)w^3 e^{-it} = \frac{\varepsilon A}{2}(1+e^{-2it}). \tag{3.88}$$

Combining terms in equation (3.88) gives

$$\dot{\varphi}_2 + \frac{1}{1+\varepsilon}[v+\varepsilon w - w(1+\varepsilon)]e^{-it} + \frac{\varepsilon\sigma}{1+\varepsilon}(v+\varepsilon w)e^{-it} + (1+\varepsilon)\lambda \dot{w} e^{-it} + \frac{4}{3}(1+\varepsilon)w^3 e^{-it} = \frac{\varepsilon A}{2}(1+e^{-2it}). \tag{3.89}$$

Equation (3.89) can be reduced to

$$\dot{\varphi}_2 + \frac{1}{1+\varepsilon}[v-w]e^{-it} + \frac{\varepsilon\sigma}{1+\varepsilon}(v+\varepsilon w)e^{-it} + (1+\varepsilon)\lambda \dot{w} e^{-it} + \frac{4}{3}(1+\varepsilon)w^3 e^{-it} = \frac{\varepsilon A}{2}(1+e^{-2it}). \tag{3.90}$$

Making use of equations (3.75) and (3.76) to eliminate *v* and *w* from the second and third terms in equation (3.90) results in

$$\dot{\varphi}_2 + \frac{i}{2(1+\varepsilon)}\left(\varphi_2 - \varphi_1 + \varphi_1^* e^{-2it} - \varphi_2^* e^{-2it}\right) - \frac{i\varepsilon\sigma}{2(1+\varepsilon)}\left(\varphi_1 + \varepsilon\varphi_2 - \varphi_1^* e^{-2it} - \varepsilon\varphi_2^* e^{-2it}\right) + (1+\varepsilon)\lambda \dot{w} e^{-it} + \frac{4}{3}(1+\varepsilon)w^3 e^{-it} = \frac{\varepsilon A}{2}\left(1+e^{-2it}\right). \tag{3.91}$$

But

$$\dot{w}e^{-it} = \frac{1}{2}(\varphi_2 + \varphi_2^* e^{-2it}). \tag{3.92}$$

Also,

$$we^{-it} = -\frac{i}{2}(\varphi_2 - \varphi_2^* e^{-2it}), \tag{3.93}$$

and

$$\left(we^{-it}\right)^3 = \left[-\frac{i}{2}(\varphi_2 - \varphi_2^* e^{-2it})\right]^3. \tag{3.94}$$

Thus,

$$\left(we^{-it}\right)^3 = \frac{i}{8}({\varphi_2}^3 - {\varphi_2^*}^3 e^{-6it} - 3{\varphi_2}^2\varphi_2^* e^{-2it} + 3\varphi_2{\varphi_2^*}^2 e^{-4it}). \tag{3.95}$$

Noting that

$$w^3 e^{-it} = \left(we^{-it}\right)^3 e^{2it}, \tag{3.96}$$

we now have

$$w^3 e^{-it} = \frac{i}{8}({\varphi_2}^3 e^{2it} - {\varphi_2^*}^3 e^{-4it} - 3{\varphi_2}^2\varphi_2^* + 3\varphi_2{\varphi_2^*}^2 e^{-2it}). \tag{3.97}$$

Substituting equations (3.92) and (3.97) into equation (3.91) gives

$$\dot{\varphi}_2 + \frac{i}{2(1+\varepsilon)}\left(\varphi_2 - \varphi_1 + \varphi_1^* e^{-2it} - \varphi_2^* e^{-2it}\right) \tag{3.98}$$

$$-\frac{i\varepsilon\sigma}{2(1+\varepsilon)}\left(\varphi_1 + \varepsilon\varphi_2 - \varphi_1^* e^{-2it} - \varepsilon\varphi_2^* e^{-2it}\right) + (1+\varepsilon)\lambda\left[\frac{1}{2}(\varphi_2 + \varphi_2^* e^{-2it})\right]$$

$$+\frac{4}{3}(1+\varepsilon)\left[\frac{i}{8}({\varphi_2}^3 e^{2it} - {\varphi_2^*}^3 e^{-4it} - 3{\varphi_2}^2\varphi_2^* + 3\varphi_2{\varphi_2^*}^2 e^{-2it})\right] = \frac{\varepsilon A}{2}\left(1 + e^{-2it}\right).$$

Omitting the exponential terms from equations (3.85) and (3.98) based on the justification in Section 3.A.3.1 gives the averaged system

$$\begin{gathered}\dot{\varphi}_1 + \frac{i\varepsilon}{2(1+\varepsilon)}(\varphi_1 - \varphi_2) - \frac{i\varepsilon\sigma}{2(1+\varepsilon)}(\varphi_1 + \varepsilon\varphi_2) = \frac{\varepsilon A}{2} \\ \dot{\varphi}_2 + \frac{\lambda(1+\varepsilon)}{2}\varphi_2 + \frac{i}{2(1+\varepsilon)}(\varphi_2 - \varphi_1) - \frac{i\varepsilon\sigma}{2(1+\varepsilon)}(\varphi_1 + \varepsilon\varphi_2) \\ -\frac{i}{2}(1+\varepsilon)|\varphi_2|^2\varphi_2 = \frac{\varepsilon A}{2}.\end{gathered} \tag{3.99}$$

Setting the derivatives of equation (3.99) to zero gives

$$\begin{gathered}\frac{i\varepsilon}{2(1+\varepsilon)}(\varphi_{10}-\varphi_{20})-\frac{i\varepsilon\sigma}{2(1+\varepsilon)}(\varphi_{10}+\varepsilon\varphi_{20})=\frac{\varepsilon A}{2}\\ \frac{\lambda(1+\varepsilon)}{2}\varphi_{20}+\frac{i}{2(1+\varepsilon)}(\varphi_{20}-\varphi_{10})-\frac{i\varepsilon\sigma}{2(1+\varepsilon)}(\varphi_{10}+\varepsilon\varphi_{20})\\ -\frac{i}{2}(1+\varepsilon)|\varphi_{20}|^2\varphi_{20}=\frac{\varepsilon A}{2}.\end{gathered} \tag{3.100}$$

We can rewrite the first equation of equations (3.100) as

$$\left[\frac{i\varepsilon}{2(1+\varepsilon)}-\frac{i\varepsilon\sigma}{2(1+\varepsilon)}\right]\varphi_{10}-\left[\frac{i\varepsilon}{2(1+\varepsilon)}+\frac{i\varepsilon^2\sigma}{2(1+\varepsilon)}\right]\varphi_{20}=\frac{\varepsilon A}{2}. \tag{3.101}$$

Then, after reducing equation (3.101) and rearranging, we have

$$\frac{i}{2(1+\varepsilon)}(1-\sigma)\varphi_{10}=\frac{i}{2(1+\varepsilon)}(1+\varepsilon\sigma)\varphi_{20}+\frac{A}{2}. \tag{3.102}$$

Dividing equation (3.102) by *(1-σ)* gives

$$\frac{i}{2(1+\varepsilon)}\varphi_{10}=\frac{1}{1-\sigma}\left[\frac{i}{2(1+\varepsilon)}(1+\varepsilon\sigma)\varphi_{20}+\frac{A}{2}\right]. \tag{3.103}$$

Multiplying both sides of equation (3.103) by *2(1 + ε) / i* gives

$$\varphi_{10}=\frac{(1+\varepsilon\sigma)\varphi_{20}-i(1+\varepsilon)A}{1-\sigma}. \tag{3.104}$$

Substituting Eq. (3.103) into the second equation of equations (3.100) gives

$$\begin{gathered}\frac{\lambda(1+\varepsilon)}{2}\varphi_{20}+\frac{i}{2(1+\varepsilon)}\varphi_{20}-\frac{1}{1-\sigma}\left[\frac{i}{2(1+\varepsilon)}(1+\varepsilon\sigma)\varphi_{20}+\frac{A}{2}\right]\\ -\frac{\varepsilon\sigma}{1-\sigma}\left[\frac{i}{2(1+\varepsilon)}(1+\varepsilon\sigma)\varphi_{20}+\frac{A}{2}\right]-\frac{i\varepsilon^2\sigma}{2(1+\varepsilon)}\varphi_{20}-\frac{i}{2}(1+\varepsilon)|\varphi_{20}|^2\varphi_{20}=\frac{\varepsilon A}{2}.\end{gathered} \tag{3.105}$$

Multiplying both sides of equation (3.105) by 2 and separating real and imaginary coefficients of φ_{20} gives

$$\begin{gathered}\lambda(1+\varepsilon)\varphi_{20}+i\left[\frac{1}{1+\varepsilon}-\frac{1+\varepsilon\sigma}{(1-\sigma)(1+\varepsilon)}-\frac{\varepsilon\sigma(1+\varepsilon\sigma)}{(1-\sigma)(1+\varepsilon)}-\frac{\varepsilon^2\sigma}{1+\varepsilon}-(1+\varepsilon)|\varphi_{20}|^2\right]\varphi_{20}\\ -\frac{A}{1-\sigma}-\frac{A\varepsilon\sigma}{1-\sigma}=\varepsilon A\end{gathered} \tag{3.106}$$

Equation (3.106) may be further reduced to

$$\begin{aligned}\lambda(1+\varepsilon)\varphi_{20} + i\left[\frac{1-\varepsilon\sigma^2}{1+\varepsilon} - \frac{1+\varepsilon\sigma+\varepsilon\sigma(1+\varepsilon\sigma)}{(1-\sigma)(1+\varepsilon)} - (1+\varepsilon)|\varphi_{20}|^2\right]\varphi_{20}, \\ = \varepsilon A + \frac{A}{1-\sigma} + \frac{A\varepsilon\sigma}{1-\sigma},\end{aligned} \tag{3.107}$$

then to

$$\begin{aligned}\lambda(1+\varepsilon)\varphi_{20} + i\left[\frac{(1-\varepsilon\sigma^2)(1-\sigma)-1-2\varepsilon\sigma-\varepsilon^2\sigma^2}{(1+\varepsilon)(1-\sigma)} - (1+\varepsilon)|\varphi_{20}|^2\right]\varphi_{20}, \\ = A\left[\varepsilon + \frac{1}{1-\sigma} + \frac{\varepsilon\sigma}{1-\sigma}\right],\end{aligned} \tag{3.108}$$

followed by

$$\lambda(1+\varepsilon)\varphi_{20} + i\left[\frac{-\varepsilon^2\sigma-2\varepsilon\sigma-\sigma}{(1+\varepsilon)(1-\sigma)} - (1+\varepsilon)|\varphi_{20}|^2\right]\varphi_{20} = A\left[\varepsilon + \frac{1+\varepsilon\sigma}{1-\sigma}\right]. \tag{3.109}$$

Continuing with the reduction,

$$\lambda(1+\varepsilon)\varphi_{20} + i\left[\frac{-\sigma(1+\varepsilon)^2}{(1+\varepsilon)(1-\sigma)} - (1+\varepsilon)|\varphi_{20}|^2\right]\varphi_{20} = A\left[\frac{\varepsilon(1-\sigma)+(1+\varepsilon\sigma)}{(1-\sigma)}\right]. \tag{3.110}$$

Dividing by (1 + ε) gives

$$\lambda\varphi_{20} + i\left[\frac{-\sigma(1+\varepsilon)^2}{(1+\varepsilon)^2(1-\sigma)} - |\varphi_{20}|^2\right]\varphi_{20} = A\left[\frac{\varepsilon+1}{(1-\sigma)(1+\varepsilon)}\right]. \tag{3.111}$$

Thus,

$$\lambda\varphi_{20} - i\left[\frac{\sigma}{(1-\sigma)} + |\varphi_{20}|^2\right]\varphi_{20} = \frac{A}{1-\sigma}. \tag{3.112}$$

Referring to the derivation in Section 3.A.3.2, we can write

$$\left[\lambda^2 + \frac{\sigma^2}{(1-\sigma)^2}\right]|\varphi_{20}|^2 + \frac{2\sigma}{(1-\sigma)}|\varphi_{20}|^4 + |\varphi_{20}|^6 = \frac{A^2}{(1-\sigma)^2}. \tag{3.113}$$

Equation (3.113) can be rewritten as

$$\alpha_1 Z + \alpha_2 Z^2 + \alpha_3 Z^3 + \alpha_4 = 0, \tag{3.114}$$

where

$$|\varphi_{20}|^2 = Z, \quad \alpha_1 = \lambda^2 + \frac{\sigma^2}{(1-\sigma)^2}$$
$$\alpha_2 = \frac{2\sigma}{1-\sigma}, \quad \alpha_3 = 1, \quad \alpha_4 = \frac{-A^2}{(1-\sigma)^2}. \tag{3.115}$$

Taking the derivative of equation (3.114) with respect to Z gives

$$3\alpha_3 Z^2 + 2\alpha_2 Z + \alpha_1 = 0. \tag{3.116}$$

Next, equation (3.116) is solved for Z using the quadratic formula:

$$Z = \frac{-\alpha_2}{3\alpha_3} \pm \frac{\sqrt{{\alpha_2}^2 - 3\alpha_1\alpha_3}}{3\alpha_3}. \tag{3.117}$$

Substituting this expression for Z into equation (3.114) gives the equation representing the saddle-node bifurcation boundary:

$$\alpha_4 = -\alpha_1\left(\frac{-\alpha_2}{3\alpha_3} \pm \frac{\sqrt{{\alpha_2}^2 - 3\alpha_1\alpha_3}}{3\alpha_3}\right) - \alpha_2\left(\frac{-\alpha_2}{3\alpha_3} \pm \frac{\sqrt{{\alpha_2}^2 - 3\alpha_1\alpha_3}}{3\alpha_3}\right)^2$$
$$-\alpha_3\left(\frac{-\alpha_2}{3\alpha_3} \pm \frac{\sqrt{{\alpha_2}^2 - 3\alpha_1\alpha_3}}{3\alpha_3}\right)^3. \tag{3.118}$$

3.A.2 Saddle-Node Bifurcation Equation Derivations (Nonlinear Damping)

The nonlinearly damped system is given by

$$\ddot{y_1} + \varepsilon\lambda(\dot{y_1} - \dot{y_2})^3 + (1 + \varepsilon\sigma)y_1 + \frac{4}{3}\varepsilon(y_1 - y_2)^3 = \varepsilon A\cos t$$
$$\varepsilon\ddot{y_2} + \varepsilon\lambda(\dot{y_2} - \dot{y_1})^3 + \frac{4}{3}\varepsilon(y_2 - y_1)^3 = 0. \tag{3.119}$$

Let

$$v = y_1 + \varepsilon y_2, \quad w = y_1 - y_2. \tag{3.120}$$

Making substitutions of the second equation of equations (3.120) into equation (3.119) gives

$$\begin{aligned} \ddot{y_1} + \varepsilon\lambda\dot{w}^3 + (1+\varepsilon\sigma)y_1 + \tfrac{4}{3}\varepsilon w^3 &= \varepsilon A\cos t \\ \varepsilon\ddot{y_2} - \varepsilon\lambda\dot{w^3} - \tfrac{4}{3}\varepsilon w^3 &= 0. \end{aligned} \tag{3.121}$$

From the second equation of equations (3.120) and the second equation of equations (3.121), we can write

$$\ddot{y_1} = \ddot{w} + \ddot{y_2}, \qquad \ddot{y_2} = \lambda\dot{w^3} + \tfrac{4}{3}w^3. \tag{3.122}$$

Combining equations (3.122), we have

$$\ddot{y_1} = \ddot{w} + \lambda\dot{w^3} + \tfrac{4}{3}w^3. \tag{3.123}$$

From the first equation of equations (3.120) and the first equation of equations (3.121), we can write

$$\varepsilon\ddot{y_2} = \ddot{v} - \ddot{y_1}, \qquad \ddot{y_1} = \varepsilon A\cos t - \varepsilon\lambda\dot{w}^3 - (1+\varepsilon\sigma)y_1 - \tfrac{4}{3}\varepsilon w^3. \tag{3.124}$$

Combining equations (3.124), we have

$$\varepsilon\ddot{y_2} = \ddot{v} - \varepsilon A\cos t + \varepsilon\lambda\dot{w}^3 + (1+\varepsilon\sigma)y_1 + \tfrac{4}{3}\varepsilon w^3. \tag{3.125}$$

In order to eliminate the y_1 term from equation (3.125), equations (3.120) can be combined, resulting in

$$y_1 = \tfrac{v+\varepsilon w}{1+\varepsilon}, \tag{3.126}$$

and substitution of equation (3.126) into equation (3.125) leads to

$$\varepsilon\ddot{y_2} = \ddot{v} - \varepsilon A\cos t + \varepsilon\lambda\dot{w}^3 + (1+\varepsilon\sigma)\tfrac{v+\varepsilon w}{1+\varepsilon} + \tfrac{4}{3}\varepsilon w^3. \tag{3.127}$$

Substituting equations (3.123), (3.126), and (3.127) into equations (3.121) results in

$$\ddot{v} + (1 + \varepsilon\sigma)\frac{v+\varepsilon w}{1+\varepsilon} = \varepsilon A \cos t$$
$$\ddot{w} + (1 + \varepsilon\sigma)\frac{v+\varepsilon w}{1+\varepsilon} + (1 + \varepsilon)\lambda\dot{w}^3 + \frac{4}{3}(1 + \varepsilon)w^3 = \varepsilon A \cos t. \quad (3.128)$$

Let

$$\varphi_1 e^{it} = \dot{v} + iv, \quad \varphi_2 e^{it} = \dot{w} + iw. \quad (3.129)$$

Rearranging equations (3.129) for φ_1 and φ_2 gives

$$\varphi_1 = (\dot{v} + iv)e^{-it}, \quad \varphi_2 = (\dot{w} + iw)e^{-it} \quad (3.130)$$

with the complex conjugates

$$\varphi_1^* = (\dot{v} - iv)e^{it}, \quad \varphi_2^* = (\dot{w} - iw)e^{it}. \quad (3.131)$$

Also, it easily can be verified that

$$\dot{\varphi}_1 = (\ddot{v} + v)e^{-it}, \quad \dot{\varphi}_2 = (\ddot{w} + w)e^{-it}. \quad (3.132)$$

Expanding the first equation of equations (3.128) and rewriting the cosine term on the right-hand side of the equation gives

$$\ddot{v} + \frac{(1+\varepsilon\sigma)}{1+\varepsilon}v + \frac{(1+\varepsilon\sigma)}{1+\varepsilon}\varepsilon w = \frac{\varepsilon A}{2}(e^{it} + e^{-it}). \quad (3.133)$$

Further manipulation of the right-hand side of the equation gives

$$\ddot{v} + \frac{(1+\varepsilon\sigma)}{1+\varepsilon}v + \frac{(1+\varepsilon\sigma)}{1+\varepsilon}\varepsilon w = \frac{\varepsilon A}{2}(1 + e^{-2it})e^{it}. \quad (3.134)$$

Multiplying both sides of equation (3.134) by e^{-it} gives

$$\ddot{v}e^{-it} + \frac{(1+\varepsilon\sigma)}{1+\varepsilon}ve^{-it} + \frac{(1+\varepsilon\sigma)}{1+\varepsilon}\varepsilon we^{-it} = \frac{\varepsilon A}{2}(1 + e^{-2it}). \quad (3.135)$$

Equation (3.135) can be rewritten as

$$(\ddot{v} + v)e^{-it} + \left[\frac{(1+\varepsilon\sigma)}{1+\varepsilon} - 1\right]ve^{-it} + \frac{(1+\varepsilon\sigma)}{1+\varepsilon}\varepsilon we^{-it} = \frac{\varepsilon A}{2}(1 + e^{-2it}). \quad (3.136)$$

Substituting the first equation of equations (3.132) into equations (3.136) and rewriting terms gives

$$\dot{\varphi_1} + \left[\frac{(1+\varepsilon\sigma)}{1+\varepsilon} - \frac{1+\varepsilon}{1+\varepsilon}\right] v e^{-it} + \left[\frac{\varepsilon}{1+\varepsilon} + \frac{\varepsilon^2\sigma}{1+\varepsilon}\right] \varepsilon w e^{-it} = \frac{\varepsilon A}{2}(1 + e^{-2it}), \tag{3.137}$$

which reduces to

$$\dot{\varphi_1} + \frac{(\varepsilon\sigma-\varepsilon)}{1+\varepsilon} v e^{-it} + \left[\frac{\varepsilon}{1+\varepsilon} + \frac{\varepsilon^2\sigma}{1+\varepsilon}\right] w e^{-it} = \frac{\varepsilon A}{2}(1 + e^{-2it}). \tag{3.138}$$

Equation (3.138) can be rewritten as

$$\dot{\varphi_1} + \frac{\varepsilon\sigma}{1+\varepsilon}(v + \varepsilon w)e^{-it} + \frac{\varepsilon}{1+\varepsilon}(w - v)e^{-it} = \frac{\varepsilon A}{2}(1 + e^{-2it}). \tag{3.139}$$

Making use of equations (3.130) and (3.131) to eliminate *v* and *w* from equation (3.139) results in

$$\begin{aligned} &\dot{\varphi_1} + \frac{i\varepsilon\sigma}{2(1+\varepsilon)}\left(-\varphi_1 - \varepsilon\varphi_2 + \varphi_1^* e^{-2it} + \varepsilon\varphi_2^* e^{-2it}\right) \\ &+ \frac{i\varepsilon}{2(1+\varepsilon)}\left(\varphi_1 - \varphi_2 - \varphi_1^* e^{-2it} + \varphi_2^* e^{-2it}\right) = \frac{\varepsilon A}{2}(1 + e^{-2it}). \end{aligned} \tag{3.140}$$

Expanding the second equation of equations (3.128) and rewriting the cosine term on the right-hand side of the equation gives

$$\begin{aligned} &\ddot{w} + \frac{(1+\varepsilon\sigma)}{1+\varepsilon} v + \frac{(1+\varepsilon\sigma)}{1+\varepsilon} \varepsilon w + (1 + \varepsilon)\lambda \dot{w}^3 + \\ &\frac{4}{3}(1 + \varepsilon)w^3 = \frac{\varepsilon A}{2}(1 + e^{-2it})e^{it}. \end{aligned} \tag{3.141}$$

Using equation (3.132), the second derivative of *w* can be eliminated as

$$\begin{aligned} &\left(\dot{\varphi}_2 e^{it} - w\right) + \frac{(1+\varepsilon\sigma)}{1+\varepsilon} v + \frac{(1+\varepsilon\sigma)}{1+\varepsilon} \varepsilon w + (1 + \varepsilon)\lambda \dot{w}^3 + \frac{4}{3}(1 + \varepsilon)w^3 \\ &= \frac{\varepsilon A}{2}(1 + e^{-2it})e^{it}. \end{aligned} \tag{3.142}$$

Multiplying both sides of equation (3.142) by e^{-it} and rewriting gives

$$\dot{\varphi}_2 - we^{-it} + \frac{1}{1+\varepsilon}(v + \varepsilon w)e^{-it} + \frac{\varepsilon\sigma}{1+\varepsilon}(v + \varepsilon w)e^{-it} + (1 + \varepsilon)\lambda\dot{w^3}e^{-it} + \frac{4}{3}(1 + \varepsilon)w^3e^{-it} = \frac{\varepsilon A}{2}(1 + e^{-2it}). \tag{3.143}$$

Combining terms in equation (3.143) gives

$$\dot{\varphi}_2 + \frac{1}{1+\varepsilon}[v + \varepsilon w - w(1 + \varepsilon)]e^{-it} + \frac{\varepsilon\sigma}{1+\varepsilon}(v + \varepsilon w)e^{-it} + (1 + \varepsilon)\lambda\dot{w^3}e^{-it} + \frac{4}{3}(1 + \varepsilon)w^3e^{-it} = \frac{\varepsilon A}{2}(1 + e^{-2it}). \tag{3.144}$$

Equation (3.144) can be reduced to

$$\dot{\varphi}_2 + \frac{1}{1+\varepsilon}[v - w]e^{-it} + \frac{\varepsilon\sigma}{1+\varepsilon}(v + \varepsilon w)e^{-it} + (1 + \varepsilon)\lambda\dot{w^3}e^{-it} + \frac{4}{3}(1 + \varepsilon)w^3e^{-it} = \frac{\varepsilon A}{2}(1 + e^{-2it}). \tag{3.145}$$

Making use of equations (3.130) and (3.131) to eliminate *v* and *w* from the second and third terms in equation (3.145) results in

$$\dot{\varphi}_2 + \frac{i}{2(1+\varepsilon)}\left(\varphi_2 - \varphi_1 + \varphi_1^*e^{-2it} - \varphi_2^*e^{-2it}\right) - \frac{i\varepsilon\sigma}{2(1+\varepsilon)}\left(\varphi_1 + \varepsilon\varphi_2 - \varphi_1^*e^{-2it} - \varepsilon\varphi_2^*e^{-2it}\right) + (1 + \varepsilon)\lambda\dot{w^3}e^{-it} + \frac{4}{3}(1 + \varepsilon)w^3e^{-it} = \frac{\varepsilon A}{2}\left(1 + e^{-2it}\right). \tag{3.146}$$

But

$$\dot{w}e^{-it} = \frac{1}{2}(\varphi_2 + \varphi_2^*e^{-2it}), \tag{3.147}$$

and

$$(\dot{w}e^{-it})^3 = \left[\frac{1}{2}(\varphi_2 + \varphi_2^*e^{-2it})\right]^3. \tag{3.148}$$

Thus,

$$(\dot{w}e^{-it})^3 = \frac{1}{8}(\varphi_2{}^3 + \varphi_2^{*3}e^{-6it} + 3\varphi_2{}^2\varphi_2^*e^{-2it} + 3\varphi_2\varphi_2^{*2}e^{-4it}). \tag{3.149}$$

Noting that

$$\dot{w}^3 e^{-it} = \left(\dot{w}e^{-it}\right)^3 e^{2it}, \tag{3.150}$$

we now have

$$\dot{w}^3 e^{-it} = \frac{1}{8}(\varphi_2{}^3 e^{2it} + \varphi_2^{*3} e^{-4it} + 3\varphi_2{}^2\varphi_2^* + 3\varphi_2\varphi_2^{*2} e^{-2it}). \tag{3.151}$$

Also,

$$we^{-it} = -\frac{i}{2}(\varphi_2 - \varphi_2^* e^{-2it}), \tag{3.152}$$

and

$$\left(we^{-it}\right)^3 = \left[-\frac{i}{2}(\varphi_2 - \varphi_2^* e^{-2it})\right]^3. \tag{3.153}$$

Thus,

$$\left(we^{-it}\right)^3 = \frac{i}{8}(\varphi_2{}^3 - \varphi_2^{*3} e^{-6it} - 3\varphi_2{}^2\varphi_2^* e^{-2it} + 3\varphi_2\varphi_2^{*2} e^{-4it}). \tag{3.154}$$

Noting that

$$w^3 e^{-it} = \left(we^{-it}\right)^3 e^{2it}, \tag{3.155}$$

we now have

$$w^3 e^{-it} = \frac{i}{8}(\varphi_2{}^3 e^{2it} - \varphi_2^{*3} e^{-4it} - 3\varphi_2{}^2\varphi_2^* + 3\varphi_2\varphi_2^{*2} e^{-2it}). \tag{3.156}$$

Substituting equations (3.151) and (3.156) into equation (3.146) gives

$$\begin{aligned}
&\dot{\varphi}_2 + \frac{i}{2(1+\varepsilon)}\left(\varphi_2 - \varphi_1 + \varphi_1^* e^{-2it} - \varphi_2^* e^{-2it}\right) \\
&-\frac{i\varepsilon\sigma}{2(1+\varepsilon)}\left(\varphi_1 + \varepsilon\varphi_2 - \varphi_1^* e^{-2it} - \varepsilon\varphi_2^* e^{-2it}\right) \\
&+(1+\varepsilon)\lambda\left[\frac{1}{8}\left(\varphi_2{}^3 e^{2it} + \varphi_2^{*3} e^{-4it} + 3\varphi_2{}^2\varphi_2^* + 3\varphi_2\varphi_2^{*2} e^{-2it}\right)\right] \\
&+\frac{4}{3}(1+\varepsilon)\left[\frac{i}{8}\left(\varphi_2{}^3 e^{2it} - \varphi_2^{*3} e^{-4it} - 3\varphi_2{}^2\varphi_2^* + 3\varphi_2\varphi_2^{*2} e^{-2it}\right)\right] \\
&= \frac{\varepsilon A}{2}\left(1 + e^{-2it}\right).
\end{aligned} \tag{3.157}$$

Omitting the exponential terms from equations (3.140) and (3.157) based on the justification in Section 3.A.3.1 gives the averaged system

$$\begin{gathered}\dot{\varphi_1}+\frac{i\varepsilon}{2(1+\varepsilon)}(\varphi_1-\varphi_2)-\frac{i\varepsilon\sigma}{2(1+\varepsilon)}(\varphi_1+\varepsilon\varphi_2)=\frac{\varepsilon A}{2}\\ \dot{\varphi_2}+\frac{3}{8}\lambda(1+\varepsilon)|\varphi_2|^2\varphi_2+\frac{i}{2(1+\varepsilon)}(\varphi_2-\varphi_1)-\frac{i\varepsilon\sigma}{2(1+\varepsilon)}(\varphi_1+\varepsilon\varphi_2)\\ -\frac{i}{2}(1+\varepsilon)|\varphi_2|^2\varphi_2=\frac{\varepsilon A}{2}.\end{gathered} \tag{3.158}$$

Setting the derivatives of equations (3.158) to zero gives

$$\begin{gathered}\frac{i\varepsilon}{2(1+\varepsilon)}(\varphi_{10}-\varphi_{20})-\frac{i\varepsilon\sigma}{2(1+\varepsilon)}(\varphi_{10}+\varepsilon\varphi_{20})=\frac{\varepsilon A}{2}\\ \frac{3}{8}\lambda(1+\varepsilon)|\varphi_{20}|^2\varphi_{20}+\frac{i}{2(1+\varepsilon)}(\varphi_{20}-\varphi_{10})-\frac{i\varepsilon\sigma}{2(1+\varepsilon)}(\varphi_{10}+\varepsilon\varphi_{20})\\ -\frac{i}{2}(1+\varepsilon)|\varphi_{20}|^2\varphi_{20}=\frac{\varepsilon A}{2}.\end{gathered} \tag{3.159}$$

We can rewrite the first equation of equations (3.159) as

$$\left[\frac{i\varepsilon}{2(1+\varepsilon)}-\frac{i\varepsilon\sigma}{2(1+\varepsilon)}\right]\varphi_{10}-\left[\frac{i\varepsilon}{2(1+\varepsilon)}+\frac{i\varepsilon^2\sigma}{2(1+\varepsilon)}\right]\varphi_{20}=\frac{\varepsilon A}{2}. \tag{3.160}$$

Then, after reducing equation (3.160) and rearranging, we have

$$\frac{i}{2(1+\varepsilon)}(1-\sigma)\varphi_{10}=\frac{i}{2(1+\varepsilon)}(1+\varepsilon\sigma)\varphi_{20}+\frac{A}{2}. \tag{3.161}$$

Dividing equation (3.161) by *(1-σ)* gives

$$\frac{i}{2(1+\varepsilon)}\varphi_{10}=\frac{1}{1-\sigma}\left[\frac{i}{2(1+\varepsilon)}(1+\varepsilon\sigma)\varphi_{20}+\frac{A}{2}\right]. \tag{3.162}$$

Multiplying both sides of equation (3.162) by *2(1 + ε) / i* gives

$$\varphi_{10}=\frac{(1+\varepsilon\sigma)\varphi_{20}-i(1+\varepsilon)A}{1-\sigma}. \tag{3.163}$$

Substituting equation (3.162) into the second equation of equations (3.159) gives

$$\begin{gathered}\frac{3}{8}\lambda(1+\varepsilon)|\varphi_{20}|^2\varphi_{20}+\frac{i}{2(1+\varepsilon)}\varphi_{20}-\frac{1}{1-\sigma}\left[\frac{i}{2(1+\varepsilon)}(1+\varepsilon\sigma)\varphi_{20}+\frac{A}{2}\right]\\ -\frac{\varepsilon\sigma}{1-\sigma}\left[\frac{i}{2(1+\varepsilon)}(1+\varepsilon\sigma)\varphi_{20}+\frac{A}{2}\right]-\frac{i\varepsilon^2\sigma}{2(1+\varepsilon)}\varphi_{20}-\frac{i}{2}(1+\varepsilon)|\varphi_{20}|^2\varphi_{20}=\frac{\varepsilon A}{2}.\end{gathered} \tag{3.164}$$

Multiplying both sides of equation (3.164) by 2 and separating real and imaginary coefficients of φ_{20} gives

$$\frac{3}{4}\lambda(1+\varepsilon)|\varphi_{20}|^2\varphi_{20} - \frac{A}{1-\sigma} - \frac{A\varepsilon\sigma}{1-\sigma} + i\left[\frac{1}{1+\varepsilon} - \frac{1+\varepsilon\sigma}{(1-\sigma)(1+\varepsilon)} - \frac{\varepsilon\sigma(1+\varepsilon\sigma)}{(1-\sigma)(1+\varepsilon)} - \frac{\varepsilon^2\sigma}{1+\varepsilon} - (1+\varepsilon)|\varphi_{20}|^2\right]\varphi_{20} = \varepsilon A \tag{3.165}$$

Equation (3.165) may be further reduced to

$$\frac{3}{4}\lambda(1+\varepsilon)|\varphi_{20}|^2\varphi_{20} + i\left[\frac{1-\varepsilon\sigma^2}{1+\varepsilon} - \frac{1+\varepsilon\sigma+\varepsilon\sigma(1+\varepsilon\sigma)}{(1-\sigma)(1+\varepsilon)} - (1+\varepsilon)|\varphi_{20}|^2\right]\varphi_{20} = \varepsilon A + \frac{A}{1-\sigma} + \frac{A\varepsilon\sigma}{1-\sigma}, \tag{3.166}$$

then to

$$\frac{3}{4}\lambda(1+\varepsilon)|\varphi_{20}|^2\varphi_{20} + i\left[\frac{(1-\varepsilon\sigma^2)(1-\sigma)-1-2\varepsilon\sigma-\varepsilon^2\sigma^2}{(1+\varepsilon)(1-\sigma)} - (1+\varepsilon)|\varphi_{20}|^2\right]\varphi_{20} = A\left[\varepsilon + \frac{1}{1-\sigma} + \frac{\varepsilon\sigma}{1-\sigma}\right], \tag{3.167}$$

followed by

$$\frac{3}{4}\lambda(1+\varepsilon)|\varphi_{20}|^2\varphi_{20} + i\left[\frac{-\varepsilon^2\sigma-2\varepsilon\sigma-\sigma}{(1+\varepsilon)(1-\sigma)} - (1+\varepsilon)|\varphi_{20}|^2\right]\varphi_{20} = A\left[\varepsilon + \frac{1+\varepsilon\sigma}{1-\sigma}\right]. \tag{3.168}$$

Continuing with the reduction,

$$\frac{3}{4}\lambda(1+\varepsilon)|\varphi_{20}|^2\varphi_{20} + i\left[\frac{-\sigma(1+\varepsilon)^2}{(1+\varepsilon)(1-\sigma)} - (1+\varepsilon)|\varphi_{20}|^2\right]\varphi_{20} = A\left[\frac{\varepsilon(1-\sigma)+(1+\varepsilon\sigma)}{(1-\sigma)}\right]. \tag{3.169}$$

Dividing by (1 + ε) gives

$$\frac{3}{4}\lambda|\varphi_{20}|^2\varphi_{20} + i\left[\frac{-\sigma(1+\varepsilon)^2}{(1+\varepsilon)^2(1-\sigma)} - |\varphi_{20}|^2\right]\varphi_{20} = A\left[\frac{\varepsilon+1}{(1-\sigma)(1+\varepsilon)}\right]. \tag{3.170}$$

Thus,

$$\frac{3}{4}\lambda|\varphi_{20}|^2\varphi_{20} - i\left[\frac{\sigma}{(1-\sigma)} + |\varphi_{20}|^2\right]\varphi_{20} = \frac{A}{1-\sigma}. \tag{3.171}$$

Referring to the derivation in Section 3.A.3.3, we can write

$$\frac{\sigma^2}{(1-\sigma)^2}|\varphi_{20}|^2 + \frac{2\sigma}{(1-\sigma)}|\varphi_{20}|^4 + \left[\frac{9}{16}\lambda^2 + 1\right]|\varphi_{20}|^6 = \frac{A^2}{(1-\sigma)^2}. \tag{3.172}$$

Referring to equation (3.172), we now have

$$\begin{gathered} |\varphi_{20}|^2 = Z, \quad \alpha_1 = \frac{\sigma^2}{(1-\sigma)^2}, \quad \alpha_2 = \frac{2\sigma}{1-\sigma}, \\ \alpha_3 = \frac{9}{16}\lambda^2 + 1, \quad \alpha_4 = \frac{-A^2}{(1-\sigma)^2} \end{gathered} \tag{3.173}$$

Thus, equation (3.172) can be rewritten as

$$\alpha_1 Z + \alpha_2 Z^2 + \alpha_3 Z^3 + \alpha_4 = 0 \tag{3.174}$$

The derivative of equation (3.174) with respect to Z is

$$3\alpha_3 Z^2 + 2\alpha_2 Z + \alpha_1 = 0. \tag{3.175}$$

Next, equation (3.175) is solved for Z using the quadratic formula:

$$Z = \frac{-\alpha_2}{3\alpha_3} \pm \frac{\sqrt{{\alpha_2}^2 - 3\alpha_1\alpha_3}}{3\alpha_3}. \tag{3.176}$$

Substituting this expression for Z into equation (3.174) gives the equation representing the saddle-node bifurcation boundary:

$$\begin{aligned} \alpha_4 = &-\alpha_1\left(\frac{-\alpha_2}{3\alpha_3} \pm \frac{\sqrt{{\alpha_2}^2 - 3\alpha_1\alpha_3}}{3\alpha_3}\right) - \alpha_2\left(\frac{-\alpha_2}{3\alpha_3} \pm \frac{\sqrt{{\alpha_2}^2 - 3\alpha_1\alpha_3}}{3\alpha_3}\right)^2 \\ &-\alpha_3\left(\frac{-\alpha_2}{3\alpha_3} \pm \frac{\sqrt{{\alpha_2}^2 - 3\alpha_1\alpha_3}}{3\alpha_3}\right)^3. \end{aligned} \tag{3.177}$$

3.A.3 Saddle-Node Bifurcation Analysis Digressions

3.A.3.1 Rationale for Averaged Systems

Equations (3.99) and (3.158) are obtained by taking the average of equations (3.85) and (3.98) for the linearly damped system and (3.140) and (3.157) for the nonlinearly damped system. First, understanding the following exponential average should be established. Over one period, the average value of a function (Stewart, 1999) is given by:

$$\bar{f} = \frac{1}{2\pi}\int_0^{2\pi} f(t)dt. \tag{3.178}$$

If

$$f(t) = e^{nit}, \tag{3.179}$$

then

$$\bar{f} = \frac{1}{2\pi}\int_0^{2\pi} e^{nit}dt \tag{3.180}$$

$$= \frac{1}{2\pi}\frac{1}{ni}e^{nit}\Big|_0^{2\pi} \tag{3.181}$$

$$= \frac{1}{2\pi ni}\left(e^{2\pi ni} - e^{0}\right) \tag{3.182}$$

$$= \frac{1}{2\pi ni}\left(e^{2\pi ni} - 1\right) \tag{3.183}$$

But

$$e^{2\pi ni} = \cos 2\pi n + sgn(n)i \sin 2\pi n \tag{3.184}$$

For

$$|n| = 0, 2, 4, 6, ..., \tag{3.185}$$

we can write

$$e^{2\pi ni} = 1 + sgn(n)i \cdot 0. \tag{3.186}$$

Hence

$$e^{2\pi ni} = 1. \tag{3.187}$$

Thus,

$$\bar{f} = \frac{1}{2\pi ni}(1-1), \tag{3.188}$$

which means

$$\bar{f} = 0. \tag{3.189}$$

Hence, the averages of e^{-4it}, e^{-2it}, and e^{2it} are all zero.

3.A.3.2 Derivation of Equation (3.113) (Linear Damping)

Beginning with equation (3.112):

$$\lambda\varphi_{20} - i\left[\frac{\sigma}{(1-\sigma)} + |\varphi_{20}|^2\right]\varphi_{20} = \frac{A}{1-\sigma}, \tag{3.190}$$

let

$$\varphi_{20} = a + ib. \tag{3.191}$$

Substituting equation (3.191) into equation (3.190) gives

$$\lambda(a+ib) - i\left[\frac{\sigma}{(1-\sigma)} + |\varphi_{20}|^2\right](a+ib) = \frac{A}{1-\sigma}. \tag{3.192}$$

Collecting real and imaginary terms gives

$$\left\{\lambda a + \left[\frac{\sigma}{(1-\sigma)} + |\varphi_{20}|^2\right]b\right\} + i\left\{\lambda b - \left[\frac{\sigma}{(1-\sigma)} + |\varphi_{20}|^2\right]a\right\} = \frac{A}{1-\sigma}. \tag{3.193}$$

Taking the magnitude of both sides gives

$$\left\{\lambda a + \left[\frac{\sigma}{(1-\sigma)} + |\varphi_{20}|^2\right]b\right\}^2 + \left\{\lambda b - \left[\frac{\sigma}{(1-\sigma)} + |\varphi_{20}|^2\right]a\right\}^2 = \frac{A^2}{(1-\sigma)^2}. \tag{3.194}$$

Expanding gives

$$\lambda^2 a^2 + 2\lambda ab\left[\frac{\sigma}{(1-\sigma)} + |\varphi_{20}|^2\right] + \left[\frac{\sigma}{(1-\sigma)} + |\varphi_{20}|^2\right]^2 b^2 + \lambda^2 b^2$$
$$-2\lambda ab\left[\frac{\sigma}{(1-\sigma)} + |\varphi_{20}|^2\right] + \left[\frac{\sigma}{(1-\sigma)} + |\varphi_{20}|^2\right]^2 a^2 = \frac{A^2}{(1-\sigma)^2}. \quad (3.195)$$

Reducing yields

$$\lambda^2(a^2 + b^2) + \left[\frac{\sigma}{(1-\sigma)} + |\varphi_{20}|^2\right]^2 (a^2 + b^2) = \frac{A^2}{(1-\sigma)^2}. \quad (3.196)$$

But

$$(a^2 + b^2) = |\varphi_{20}|^2, \quad (3.197)$$

so making this substitution gives

$$\lambda^2|\varphi_{20}|^2 + \left[\frac{\sigma}{(1-\sigma)} + |\varphi_{20}|^2\right]^2 |\varphi_{20}|^2 = \frac{A^2}{(1-\sigma)^2}. \quad (3.198)$$

Expanding again yields

$$\lambda^2|\varphi_{20}|^2 + \left[\frac{\sigma^2}{(1-\sigma)^2} + \frac{2\sigma}{(1-\sigma)}|\varphi_{20}|^2 + |\varphi_{20}|^4\right]|\varphi_{20}|^2 = \frac{A^2}{(1-\sigma)^2}. \quad (3.199)$$

Finally, reducing results in

$$\left[\lambda^2 + \frac{\sigma^2}{(1-\sigma)^2}\right]|\varphi_{20}|^2 + \frac{2\sigma}{(1-\sigma)}|\varphi_{20}|^4 + |\varphi_{20}|^6 = \frac{A^2}{(1-\sigma)^2}. \quad (3.200)$$

3.A.3.3 Derivation of Equation (3.172) (Nonlinear Damping)

Beginning with equation (3.171):

$$\frac{3}{4}\lambda|\varphi_{20}|^2\varphi_{20} - i\left[\frac{\sigma}{(1-\sigma)} + |\varphi_{20}|^2\right]\varphi_{20} = \frac{A}{1-\sigma}, \quad (3.201)$$

let

$$\varphi_{20} = a + ib. \quad (3.202)$$

Substituting equation (3.202) into equation (3.201) gives

$$\frac{3}{4}\lambda|\varphi_{20}|^2(a + ib) - i\left[\frac{\sigma}{(1-\sigma)} + |\varphi_{20}|^2\right](a + ib) = \frac{A}{1-\sigma}. \quad (3.203)$$

Collecting real and imaginary terms gives

$$\left\{\frac{3}{4}\lambda|\varphi_{20}|^2 a + \left[\frac{\sigma}{(1-\sigma)} + |\varphi_{20}|^2\right] b\right\} + i\left\{\frac{3}{4}\lambda|\varphi_{20}|^2 b - \left[\frac{\sigma}{(1-\sigma)} + |\varphi_{20}|^2\right] a\right\} = \frac{A}{1-\sigma}. \tag{3.204}$$

Taking the magnitude of both sides gives

$$\left\{\frac{3}{4}\lambda|\varphi_{20}|^2 a + \left[\frac{\sigma}{(1-\sigma)} + |\varphi_{20}|^2\right] b\right\}^2 + \left\{\frac{3}{4}\lambda|\varphi_{20}|^2 b - \left[\frac{\sigma}{(1-\sigma)} + |\varphi_{20}|^2\right] a\right\}^2 = \frac{A^2}{(1-\sigma)^2}. \tag{3.205}$$

Expanding gives

$$\frac{9}{16}\lambda^2|\varphi_{20}|^4 a^2 + \frac{3}{2}\lambda|\varphi_{20}|^2 ab\left[\frac{\sigma}{(1-\sigma)} + |\varphi_{20}|^2\right] + \left[\frac{\sigma}{(1-\sigma)} + |\varphi_{20}|^2\right]^2 b^2 + \frac{9}{16}\lambda^2|\varphi_{20}|^4 b^2 - \frac{3}{2}\lambda|\varphi_{20}|^2 ab\left[\frac{\sigma}{(1-\sigma)} + |\varphi_{20}|^2\right] + \left[\frac{\sigma}{(1-\sigma)} + |\varphi_{20}|^2\right]^2 a^2 = \frac{A^2}{(1-\sigma)^2}. \tag{3.206}$$

Reducing yields

$$\frac{9}{16}\lambda^2|\varphi_{20}|^4(a^2 + b^2) + \left[\frac{\sigma}{(1-\sigma)} + |\varphi_{20}|^2\right]^2 (a^2 + b^2) = \frac{A^2}{(1-\sigma)^2}. \tag{3.207}$$

But

$$(a^2 + b^2) = |\varphi_{20}|^2, \tag{3.208}$$

so making this substitution gives

$$\frac{9}{16}\lambda^2|\varphi_{20}|^6 + \left[\frac{\sigma}{(1-\sigma)} + |\varphi_{20}|^2\right]^2 |\varphi_{20}|^2 = \frac{A^2}{(1-\sigma)^2}. \tag{3.209}$$

Expanding again yields

$$\frac{9}{16}\lambda^2|\varphi_{20}|^6 + \left[\frac{\sigma^2}{(1-\sigma)^2} + \frac{2\sigma}{(1-\sigma)}|\varphi_{20}|^2 + |\varphi_{20}|^4\right]|\varphi_{20}|^2 = \frac{A^2}{(1-\sigma)^2}. \tag{3.210}$$

Finally, reducing results in

$$\frac{\sigma^2}{(1-\sigma)^2}|\varphi_{20}|^2 + \frac{2\sigma}{(1-\sigma)}|\varphi_{20}|^4 + \left[\frac{9}{16}\lambda^2 + 1\right]|\varphi_{20}|^6 = \frac{A^2}{(1-\sigma)^2}. \tag{3.211}$$

3.A.4 Hopf Bifurcation Equation Derivations (Linear Damping)

Let

$$\varphi_1 = \varphi_{10} + \delta_1, \quad \varphi_2 = \varphi_{20} + \delta_2. \tag{3.212}$$

Substituting equation (3.212) into the first equation of equation (3.99) gives

$$\begin{gathered}\dot{\varphi_{10}} + \dot{\delta_1} + \frac{i\varepsilon}{2(1+\varepsilon)}(\varphi_{10} + \delta_1 - \varphi_{20} - \delta_2) \\ -\frac{i\varepsilon\sigma}{2(1+\varepsilon)}(\varphi_{10} + \delta_1 + \varepsilon\varphi_{20} + \varepsilon\delta_2) = \frac{\varepsilon A}{2},\end{gathered} \tag{3.213}$$

but the derivative of the fixed point φ_{10} equals zero. Reducing equation (3.213) further gives

$$\begin{gathered}\dot{\delta_1} + \frac{i\varepsilon}{2(1+\varepsilon)}(\delta_1 - \delta_2) - \frac{i\varepsilon\sigma}{2(1+\varepsilon)}(\delta_1 + \varepsilon\delta_2) \\ + \frac{i\varepsilon}{2(1+\varepsilon)}(\varphi_{10} - \varphi_{20}) - \frac{i\varepsilon\sigma}{2(1+\varepsilon)}(\varphi_{10} + \varepsilon\varphi_{20}) = \frac{\varepsilon A}{2}.\end{gathered} \tag{3.214}$$

Introducing the first equation of equations (3.100) into equation (3.214) and rearranging gives

$$\dot{\delta_1} = -\frac{i\varepsilon}{2(1+\varepsilon)}(\delta_1 - \delta_2) + \frac{i\varepsilon\sigma}{2(1+\varepsilon)}(\delta_1 + \varepsilon\delta_2). \tag{3.215}$$

Substituting equation (3.212) into the second equation of equation (3.99) gives

$$\begin{gathered}\dot{\varphi_{20}} + \dot{\delta_2} + \frac{\lambda(1+\varepsilon)}{2}(\varphi_{20} + \delta_2) + \frac{i}{2(1+\varepsilon)}(\varphi_{20} + \delta_2 - \varphi_{10} - \delta_1) \\ -\frac{i\varepsilon\sigma}{2(1+\varepsilon)}(\varphi_{10} + \delta_1 + \varepsilon\varphi_{20} + \varepsilon\delta_2) - \frac{i}{2}(1+\varepsilon)|\varphi_{20} + \delta_2|^2(\varphi_{20} + \delta_2) = \frac{\varepsilon A}{2},\end{gathered} \tag{3.216}$$

but the derivative of the fixed point φ_{20} equals zero, and by the derivation in Section 3.A.6.1,

$$|\varphi_{20}+\delta_2|^2(\varphi_{20}+\delta_2) \approx |\varphi_{20}|^2\varphi_{20} + 2\delta_2|\varphi_{20}|^2 + \delta_2^*{\varphi_{20}}^2. \tag{3.217}$$

Thus, equation (3.216) can be reduced to

$$\begin{gathered}\dot{\delta_2} + \tfrac{\lambda(1+\varepsilon)}{2}\delta_2 + \tfrac{i}{2(1+\varepsilon)}(\delta_2-\delta_1) - \tfrac{i\varepsilon\sigma}{2(1+\varepsilon)}(\delta_1+\varepsilon\delta_2) \\ -\tfrac{i}{2}(1+\varepsilon)[|\varphi_{20}|^2\varphi_{20} + 2\delta_2|\varphi_{20}|^2 + \delta_2^*{\varphi_{20}}^2] + \tfrac{\lambda(1+\varepsilon)}{2}\varphi_{20} \\ +\tfrac{i}{2(1+\varepsilon)}(\varphi_{20}-\varphi_{10}) - \tfrac{i\varepsilon\sigma}{2(1+\varepsilon)}(\varphi_{10}+\varepsilon\varphi_{20}) = \tfrac{\varepsilon A}{2}.\end{gathered} \tag{3.218}$$

Equation (3.218) can be further reduced to

$$\begin{gathered}\dot{\delta_2} + \tfrac{\lambda(1+\varepsilon)}{2}\delta_2 + \tfrac{i}{2(1+\varepsilon)}(\delta_2-\delta_1) - \tfrac{i\varepsilon\sigma}{2(1+\varepsilon)}(\delta_1+\varepsilon\delta_2) \\ -i(1+\varepsilon)\delta_2|\varphi_{20}|^2 - \tfrac{i}{2}(1+\varepsilon)\delta_2^*{\varphi_{20}}^2 + \tfrac{\lambda(1+\varepsilon)}{2}\varphi_{20} \\ +\tfrac{i}{2(1+\varepsilon)}(\varphi_{20}-\varphi_{10}) - \tfrac{i\varepsilon\sigma}{2(1+\varepsilon)}(\varphi_{10}+\varepsilon\varphi_{20}) - \tfrac{i}{2}(1+\varepsilon)|\varphi_{20}|^2\varphi_{20} = \tfrac{\varepsilon A}{2}.\end{gathered} \tag{3.219}$$

Introducing the second equation of equations (3.100) into equation (3.219) and rearranging gives

$$\begin{gathered}\dot{\delta_2} = -\tfrac{\lambda(1+\varepsilon)}{2}\delta_2 - \tfrac{i}{2(1+\varepsilon)}(\delta_2-\delta_1) + \tfrac{i\varepsilon\sigma}{2(1+\varepsilon)}(\delta_1+\varepsilon\delta_2) \\ +i(1+\varepsilon)|\varphi_{20}|^2\delta_2 + \tfrac{i(1+\varepsilon)}{2}{\varphi_{20}}^2\delta_2^*.\end{gathered} \tag{3.220}$$

From equations (3.215) and (3.220), we can write

$$\dot{\delta_1^*} = \tfrac{i\varepsilon}{2(1+\varepsilon)}(\delta_1^*-\delta_2^*) - \tfrac{i\varepsilon\sigma}{2(1+\varepsilon)}(\delta_1^*+\varepsilon\delta_2^*) \tag{3.221}$$

and

$$\begin{gathered}\dot{\delta_2^*} = -\tfrac{\lambda(1+\varepsilon)}{2}\delta_2^* + \tfrac{i}{2(1+\varepsilon)}(\delta_2^*-\delta_1^*) - \tfrac{i\varepsilon\sigma}{2(1+\varepsilon)}(\delta_1^*+\varepsilon\delta_2^*) \\ -i(1+\varepsilon)|\varphi_{20}|^2\delta_2^* - \tfrac{i(1+\varepsilon)}{2}{\varphi_{20}^*}^2\delta_2.\end{gathered} \tag{3.222}$$

Now we can construct the matrix $F = [\delta_1; \delta_1^*; \delta_2; \delta_2^*]$. Using MATLAB, the Jacobian matrix was computed. From the MATLAB output and substituting in

$$|\varphi_{20}| = N_{20}, \quad {N_{20}}^4 = {\varphi_{20}}^2 {\varphi_{20}^*}^2, \tag{3.223}$$

we develop the relations

$$\begin{gathered}
\gamma_1 = \lambda(1+\varepsilon), \\
\gamma_2 = \left(\tfrac{3}{2}\varepsilon + \tfrac{3}{4} + \tfrac{3}{4}\varepsilon^2\right){N_{20}}^4 + (\varepsilon^2\sigma - 1){N_{20}}^2 + \tfrac{1}{4}\lambda^2(\varepsilon+1)^2 + \tfrac{1}{4}(\varepsilon^2\sigma^2+1), \\
\gamma_3 = \tfrac{1}{4}\lambda\varepsilon(\varepsilon\sigma^2+1), \\
\gamma_4 = \tfrac{3}{16}\varepsilon^2(1-\sigma)^2{N_{20}}^4 + \tfrac{1}{4}\varepsilon^2\sigma(1-\sigma){N_{20}}^2 + \tfrac{1}{16}\varepsilon^2[(1-\sigma)^2\lambda^2 + \sigma^2],
\end{gathered} \tag{3.224}$$

in which the characteristic polynomial is given by

$$\mu^4 + \gamma_1\mu^3 + \gamma_2\mu^2 + \gamma_3\mu + \gamma_4 = 0. \tag{3.225}$$

For the Hopf bifurcation to occur, we must have

$$\mu = \pm\Omega, \tag{3.226}$$

in which Ω is a real number. Substituting equation (3.226) into equation (3.225) and separating into real and imaginary parts gives

$$\Omega^4 - \gamma_2\Omega^2 + \gamma_4 = 0, \quad \Omega(\gamma_1\Omega^2 - \gamma_3) = 0 \Rightarrow \Omega^2 = \frac{\gamma_3}{\gamma_1}. \tag{3.227}$$

Substituting the relation for Ω^2 into the first equation of equations (3.227) and reducing gives

$${\gamma_3}^2 - \gamma_2\gamma_3\gamma_1 + \gamma_4{\gamma_1}^2 = 0. \tag{3.228}$$

MATLAB was used to determine the coefficients, v_i, for

$$v_1 z^2 + v_2 z + v_3 = 0, \tag{3.229}$$

based on equation (3.228). Solving for z in equation (3.229) gives

$$z_{1,2} = \frac{-v_2 \mp \sqrt{{v_2}^2 - 4v_3v_1}}{2v_1}, \tag{3.230}$$

and from equation (3.114), we have the boundaries of stability, given by

$$\alpha_1 z_i + \alpha_2 {z_i}^2 + \alpha_3 {z_i}^3 + \alpha_4 = 0; \quad z_i = Z; \quad i = 1,2.$$

The remainder of this analysis is carried out using MATLAB.

3.A.5 Hopf Bifurcation Equation Derivations (Nonlinear Damping)

Let

$$\varphi_1 = \varphi_{10} + \delta_1, \quad \varphi_2 = \varphi_{20} + \delta_2. \tag{3.231}$$

Substituting equation (3.231) into the first equation of equations (3.158) gives

$$\begin{aligned} &\dot{\varphi_{10}} + \dot{\delta_1} + \frac{i\varepsilon}{2(1+\varepsilon)}(\varphi_{10} + \delta_1 - \varphi_{20} - \delta_2) \\ &-\frac{i\varepsilon\sigma}{2(1+\varepsilon)}(\varphi_{10} + \delta_1 + \varepsilon\varphi_{20} + \varepsilon\delta_2) = \frac{\varepsilon A}{2}, \end{aligned} \tag{3.232}$$

but the derivative of the fixed point φ_{10} equals zero. Reducing equation (3.232) further gives

$$\begin{aligned} &\dot{\delta_1} + \frac{i\varepsilon}{2(1+\varepsilon)}(\delta_1 - \delta_2) - \frac{i\varepsilon\sigma}{2(1+\varepsilon)}(\delta_1 + \varepsilon\delta_2) \\ &+\frac{i\varepsilon}{2(1+\varepsilon)}(\varphi_{10} - \varphi_{20}) - \frac{i\varepsilon\sigma}{2(1+\varepsilon)}(\varphi_{10} + \varepsilon\varphi_{20}) = \frac{\varepsilon A}{2}. \end{aligned} \tag{3.233}$$

Introducing the first equation of equations (3.159) into equation (3.233) and rearranging gives

$$\dot{\delta_1} = -\frac{i\varepsilon}{2(1+\varepsilon)}(\delta_1 - \delta_2) + \frac{i\varepsilon\sigma}{2(1+\varepsilon)}(\delta_1 + \varepsilon\delta_2). \tag{3.234}$$

Substituting equations (3.231) into the second equation of equations (3.158) gives

$$\begin{aligned} &\dot{\varphi_{20}} + \dot{\delta_2} + \frac{3\lambda(1+\varepsilon)}{8}|\varphi_{20} + \delta_2|^2(\varphi_{20} + \delta_2) + \frac{i}{2(1+\varepsilon)}(\varphi_{20} + \delta_2 - \varphi_{10} - \delta_1) \\ &-\frac{i\varepsilon\sigma}{2(1+\varepsilon)}(\varphi_{10} + \delta_1 + \varepsilon\varphi_{20} + \varepsilon\delta_2) - \frac{i}{2}(1+\varepsilon)|\varphi_{20} + \delta_2|^2(\varphi_{20} + \delta_2) = \frac{\varepsilon A}{2}, \end{aligned} \tag{3.235}$$

but the derivative of the fixed point φ_{20} equals zero, and by the derivation in Section 3.A.6.1,

$$|\varphi_{20}+\delta_2|^2(\varphi_{20}+\delta_2) \approx |\varphi_{20}|^2\varphi_{20} + 2\delta_2|\varphi_{20}|^2 + \delta_2^*{\varphi_{20}}^2. \tag{3.236}$$

Thus, equation (3.235) can be reduced to

$$\begin{aligned}&\dot{\delta_2} + \frac{i}{2(1+\varepsilon)}(\delta_2-\delta_1) - \frac{i\varepsilon\sigma}{2(1+\varepsilon)}(\delta_1+\varepsilon\delta_2)\\ &+\left[\frac{3\lambda(1+\varepsilon)}{8} - \frac{i(1+\varepsilon)}{2}\right][|\varphi_{20}|^2\varphi_{20} + 2\delta_2|\varphi_{20}|^2 + \delta_2^*{\varphi_{20}}^2]\\ &+\frac{i}{2(1+\varepsilon)}(\varphi_{20}-\varphi_{10}) - \frac{i\varepsilon\sigma}{2(1+\varepsilon)}(\varphi_{10}+\varepsilon\varphi_{20}) = \frac{\varepsilon A}{2}.\end{aligned} \tag{3.237}$$

Equation (3.237) can be further reduced to

$$\begin{aligned}&\dot{\delta_2} + \frac{i}{2(1+\varepsilon)}(\delta_2-\delta_1) - \frac{i\varepsilon\sigma}{2(1+\varepsilon)}(\delta_1+\varepsilon\delta_2)\\ &+\left[\frac{3\lambda(1+\varepsilon)}{4} - i(1+\varepsilon)\right]\delta_2|\varphi_{20}|^2 + \left[\frac{3\lambda(1+\varepsilon)}{8} - \frac{i(1+\varepsilon)}{2}\right]\delta_2^*{\varphi_{20}}^2\\ &+\frac{i}{2(1+\varepsilon)}(\varphi_{20}-\varphi_{10}) - \frac{i\varepsilon\sigma}{2(1+\varepsilon)}(\varphi_{10}+\varepsilon\varphi_{20}) + \left[\frac{3\lambda(1+\varepsilon)}{8} - \frac{i(1+\varepsilon)}{2}\right]|\varphi_{20}|^2\varphi_{20} = \frac{\varepsilon A}{2}.\end{aligned} \tag{3.238}$$

Introducing the second equation of equations (3.159) into equation (3.238) and rearranging gives

$$\begin{aligned}\dot{\delta_2} = &-\frac{i}{2(1+\varepsilon)}(\delta_2-\delta_1) + \frac{i\varepsilon\sigma}{2(1+\varepsilon)}(\delta_1+\varepsilon\delta_2)\\ &-\frac{3\lambda(1+\varepsilon)}{4}|\varphi_{20}|^2\delta_2 + i(1+\varepsilon)|\varphi_{20}|^2\delta_2 - \frac{3\lambda(1+\varepsilon)}{8}{\varphi_{20}}^2\delta_2^* + \frac{i(1+\varepsilon)}{2}{\varphi_{20}}^2\delta_2^*.\end{aligned} \tag{3.239}$$

From equations (3.234) and (3.239), we can write

$$\dot{\delta_1^*} = \frac{i\varepsilon}{2(1+\varepsilon)}(\delta_1^*-\delta_2^*) - \frac{i\varepsilon\sigma}{2(1+\varepsilon)}(\delta_1^*+\varepsilon\delta_2^*) \tag{3.240}$$

and

$$\begin{aligned}\dot{\delta_2^*} = &\frac{i}{2(1+\varepsilon)}(\delta_2^*-\delta_1^*) - \frac{i\varepsilon\sigma}{2(1+\varepsilon)}(\delta_1^*+\varepsilon\delta_2^*)\\ &-\frac{3\lambda(1+\varepsilon)}{4}|\varphi_{20}|^2\delta_2^* - i(1+\varepsilon)|\varphi_{20}|^2\delta_2^* - \frac{3\lambda(1+\varepsilon)}{8}{\varphi_{20}^*}^2\delta_2 - \frac{i(1+\varepsilon)}{2}{\varphi_{20}^*}^2\delta_2.\end{aligned} \tag{3.241}$$

Now we can construct the matrix $F = [\delta_1; \delta_1^*; \delta_2; \delta_2^*]$. Using MATLAB, the Jacobian matrix was computed. From the MATLAB output and substituting in

$$|\varphi_{20}| = N_{20}, \quad {N_{20}}^4 = {\varphi_{20}}^2 {\varphi_{20}^*}^2, \tag{3.242}$$

we develop the relations

$$\begin{aligned}
\gamma_1 &= \tfrac{3\lambda}{2}(1+\varepsilon){N_{20}}^2, \\
\gamma_2 &= \left(\tfrac{27\varepsilon^2\lambda^2}{64} + \tfrac{27\varepsilon\lambda^2}{32} + \tfrac{27\lambda^2}{64} + \tfrac{3\varepsilon^2}{4} + \tfrac{3\varepsilon}{2} + \tfrac{3}{4}\right){N_{20}}^4 + \\
&\qquad (\varepsilon^2\sigma - 1){N_{20}}^2 + \frac{1}{4}(\varepsilon^2\sigma^2 + 1), \\
\gamma_3 &= \tfrac{3\varepsilon\lambda}{8}(1+\varepsilon\sigma^2){N_{20}}^2, \\
\gamma_4 &= \left(\tfrac{27\varepsilon^2\lambda^2\sigma^2}{256} - \tfrac{27\varepsilon^2\lambda^2\sigma}{128} + \tfrac{27\varepsilon^2\lambda^2}{256} + \tfrac{3\varepsilon^2\sigma^2}{16} - \tfrac{3\varepsilon^2\sigma}{8} + \tfrac{3\varepsilon^2}{16}\right){N_{20}}^4 \\
&\qquad + \tfrac{1}{4}\varepsilon^2\sigma(1-\sigma){N_{20}}^2 + \tfrac{\varepsilon^2\sigma^2}{16},
\end{aligned} \tag{3.243}$$

in which the characteristic polynomial is given by

$$\mu^4 + \gamma_1\mu^3 + \gamma_2\mu^2 + \gamma_3\mu + \gamma_4 = 0. \tag{3.244}$$

For the Hopf bifurcation to occur, we must have

$$\mu = \pm i\Omega, \tag{3.245}$$

in which Ω is a real number. Substituting equation (3.245) into equation (3.244) and separating into real and imaginary parts gives

$$\Omega^4 - \gamma_2\Omega^2 + \gamma_4 = 0, \quad \Omega(\gamma_1\Omega^2 - \gamma_3) = 0 \Rightarrow \Omega^2 = \tfrac{\gamma_3}{\gamma_1}. \tag{3.246}$$

Substituting the relation for Ω^2 into the first equation of equations (3.246) and reducing gives

$${\gamma_3}^2 - \gamma_2\gamma_3\gamma_1 + \gamma_4{\gamma_1}^2 = 0. \tag{3.247}$$

Letting $z = N_{20}^2$, equations (3.243) can be rewritten as

$$\gamma_1 = \frac{3\lambda}{2}(1+\varepsilon)z, \tag{3.248}$$

$$\gamma_2 = \left(\frac{27\varepsilon^2\lambda^2}{64} + \frac{27\varepsilon\lambda^2}{32} + \frac{27\lambda^2}{64} + \frac{3\varepsilon^2}{4} + \frac{3\varepsilon}{2} + \frac{3}{4}\right)z^2 + (\varepsilon^2\sigma - 1)z + \frac{1}{4}(\varepsilon^2\sigma^2 + 1),$$

$$\gamma_3 = \frac{3\varepsilon\lambda}{8}(1+\varepsilon\sigma^2)z,$$

$$\gamma_4 = \left(\frac{27\varepsilon^2\lambda^2\sigma^2}{256} - \frac{27\varepsilon^2\lambda^2\sigma}{128} + \frac{27\varepsilon^2\lambda^2}{256} + \frac{3\varepsilon^2\sigma^2}{16} - \frac{3\varepsilon^2\sigma}{8} + \frac{3\varepsilon^2}{16}\right)z^2 + \frac{1}{4}\varepsilon^2\sigma(1-\sigma)z + \frac{\varepsilon^2\sigma^2}{16}.$$

MATLAB was used to determine the coefficients

$$v_1 = -\frac{27}{1024}\varepsilon\lambda^2(9\lambda^2+16)(\varepsilon\sigma+1)^2(\varepsilon+1)^2, \tag{3.249}$$

$$v_2 = -\frac{9}{16}\varepsilon\lambda^2(\varepsilon\sigma-1)(\varepsilon\sigma+1)^2(\varepsilon+1),$$

$$v_3 = -\frac{9}{64}\varepsilon\lambda^2(\varepsilon^2\sigma^2-1)^2,$$

for

$$v_1 z^2 + v_2 z + v_3 = 0, \tag{3.250}$$

based on equation (3.247). Solving for z in equation (3.250) gives

$$z_{1,2} = \frac{-v_2 \mp \sqrt{{v_2}^2 - 4v_3v_1}}{2v_1}, \tag{3.251}$$

and from equation (3.174), we have the boundaries of stability, given by

$$\alpha_1 z_i + \alpha_2 {z_i}^2 + \alpha_3 {z_i}^3 + \alpha_4 = 0; \quad z_i = Z; \quad i = 1,2. \tag{3.252}$$

The remainder of this analysis is carried out using MATLAB.

3.A.6 Hopf Bifurcation Analysis Digressions

3.A.6.1 Derivation of Equations (3.217) and (3.236)

Let

$$\varphi_{20} = a + bi, \quad \delta_2 = c + di \tag{3.253}$$

Then,

$$\varphi_{20}{}^{*} = a - bi. \tag{3.254}$$

$$\varphi_{20}\varphi_{20}{}^{*} = (a + bi)(a - bi) \tag{3.255}$$

$$= a^2 + b^2 = |\varphi_{20}|^2 \tag{3.256}$$

Thus,

$$\varphi_{20}\varphi_{20}{}^{*} = |\varphi_{20}|^2, \tag{3.257}$$

and by similarity,

$$\delta_2\delta_2{}^{*} = |\delta_2|^2. \tag{3.258}$$

Additionally,

$$\begin{aligned} |\varphi_{20} + \delta_2|^2 &= |a + bi + c + di|^2 \\ &= |(a + c) + i(b + d)|^2 \\ &= (a + c)^2 + (b + d)^2, \end{aligned} \tag{3.259}$$

and

$$\begin{aligned} (\varphi_{20} + \delta_2)(\varphi_{20}{}^{*} + \delta_2{}^{*}) &= (a + bi + c + di)(a - bi + c - di) \\ &= [(a + c) + i(b + d)][(a + c) - i(b + d)] \\ &= (a + c)^2 + (b + d)^2 \\ &= |(a + c) + i(b + d)|^2. \end{aligned} \tag{3.260}$$

Thus,

$$|\varphi_{20} + \delta_2|^2 = (\varphi_{20} + \delta_2)(\varphi_{20}{}^* + \delta_2{}^*). \tag{3.261}$$

Continuing with the derivation,

$$|\varphi_{20} + \delta_2|^2(\varphi_{20} + \delta_2) = (\varphi_{20} + \delta_2)(\varphi_{20}{}^* + \delta_2{}^*)(\varphi_{20} + \delta_2) \tag{3.262}$$

$$= (\varphi_{20}\varphi_{20}{}^* + \delta_2{}^*\varphi_{20} + \delta_2\varphi_{20}{}^* + \delta_2\delta_2{}^*)(\varphi_{20} + \delta_2) \tag{3.263}$$

$$= (|\varphi_{20}|^2 + \delta_2{}^*\varphi_{20} + \delta_2\varphi_{20}{}^* + |\delta_2|^2)(\varphi_{20} + \delta_2) \tag{3.264}$$

$$\begin{aligned} &= |\varphi_{20}|^2\varphi_{20} + \delta_2|\varphi_{20}|^2 + \delta_2{}^*\varphi_{20}{}^2 + \delta_2\delta_2{}^*\varphi_{20} \\ &+\delta_2\varphi_{20}\varphi_{20}{}^* + \delta_2{}^2\varphi_{20}{}^* + |\delta_2|^2(\varphi_{20} + \delta_2) \end{aligned} \tag{3.265}$$

$$\begin{aligned} &= |\varphi_{20}|^2\varphi_{20} + \delta_2|\varphi_{20}|^2 + \delta_2{}^*\varphi_{20}{}^2 + |\delta_2|^2\varphi_{20} + \\ &\delta_2|\varphi_{20}|^2 + \delta_2{}^2\varphi_{20}{}^* + |\delta_2|^2(\varphi_{20} + \delta_2) \end{aligned} \tag{3.266}$$

$$\begin{aligned} &= |\varphi_{20}|^2\varphi_{20} + 2\delta_2|\varphi_{20}|^2 + \delta_2{}^*\varphi_{20}{}^2 + \\ &|\delta_2|^2\varphi_{20} + \delta_2{}^2\varphi_{20}{}^* + |\delta_2|^2(\varphi_{20} + \delta_2) \end{aligned} \tag{3.267}$$

Omitting nonlinear terms (specifically, terms containing $|\delta_2|^2$ and $\delta_2{}^2$), the equation is reduced to

$$|\varphi_{20} + \delta_2|^2(\varphi_{20} + \delta_2) = |\varphi_{20}|^2\varphi_{20} + 2\delta_2|\varphi_{20}|^2 + \delta_2{}^*\varphi_{20}{}^2. \tag{3.268}$$

3.A.6.2 Derivation of Equivalent Expression for N_{20}^4

Let

$$\varphi_{20} = a + bi, \quad \varphi_{20}^* = a - bi. \tag{3.269}$$

Then,

$$\varphi_{20}{}^2 \varphi_{20}^*{}^2 = (a + bi)^2 (a - bi)^2 \tag{3.270}$$

$$= a^4 + 2a^2 b^2 + b^4 \tag{3.271}$$

$$= (a^2 + b^2)^2 \tag{3.272}$$

$$= |\varphi_{20}|^4. \tag{3.273}$$

Since $|\varphi_{20}| = N_{20}$, then $|\varphi_{20}|^4 = N_{20}{}^4$. Thus,

$$\varphi_{20}{}^2 \varphi_{20}^*{}^2 = N_{20}{}^4. \tag{3.274}$$

3.A.7 MATLAB Code

3.A.7.1 Saddle-Node Example

```
close all
clear all

mu = [0:0.01:5];
xp = sqrt(mu);
xn = -sqrt(mu);

figure
plot(mu,xp,'-k',mu,xn,'--k','LineWidth',2)
xlim([-2 2]); ylim([-2 2]);
xlabel('\fontsize{12}\bf\mu'); ylabel('\bfx'); grid on;
legend('\bfstable', '\bfunstable', 'Location', 'NorthWest')
```

3.A.7.2 Solving for Z for Saddle-Node Plots

```
close all
clear all

syms a1 a2 a3 a4 Z

z = solve('a1*Z + a2*Z^2 + a3*Z^3 + a4 = 0', Z);

z1 = simple(z(1))
z2 = simple(z(2))
z3 = simple(z(3))
```

3.A.7.3 Saddle-Node Plots (Linear Damping)

```
%Saddle-Node Bifurcation - Linear Damping
%(boundaries of number of real periodic solutions)

close all
clear all

n_sigma = 0; k2 = 0;

for sigma = [1.2 2 5 7 -0.5 -2 -5 -7 3];
    k2 = k2+1; sigma_vec(k2) = sigma; k = 0; n_sigma = n_sigma + 1;

    for lambda = 0:.001:4
        k = k+1;
        a1 = lambda^2 + sigma^2/(1 - sigma)^2;
        a2 = 2*sigma/(1 - sigma);
        a3 = 1;
```

```
    %Solving for a4:
    a4_p = -a1*(-a2/(3*a3) + sqrt(a2^2 - 3*a1*a3)/(3*a3)) - ...
        a2*(-a2/(3*a3) + sqrt(a2^2 - 3*a1*a3)/(3*a3))^2 - ...
        a3*(-a2/(3*a3) + sqrt(a2^2 - 3*a1*a3)/(3*a3))^3;

    a4_m = -a1*(-a2/(3*a3) - sqrt(a2^2 - 3*a1*a3)/(3*a3)) - ...
        a2*(-a2/(3*a3) - sqrt(a2^2 - 3*a1*a3)/(3*a3))^2 - ...
        a3*(-a2/(3*a3) - sqrt(a2^2 - 3*a1*a3)/(3*a3))^3;

    %Solving for A from a4:

    A_p = sqrt(-a4_p*(1-sigma)^2); A_m = sqrt(-a4_m*(1-sigma)^2);
    lambda_vec(k) = lambda; A_p_vec(k) = A_p; A_m_vec(k) = A_m;
end

m = k; g = ones(1,k); h = ones(1,k);

for n = 1:k
    if abs(A_p_vec(n)- A_m_vec(n)) < .0001
        g(n) = 0; h(n) = 0;
    elseif n > 1 && g(n-1) == 0 && h(n-1) == 0
        m = m-1; g(n) = 0; h(n) = 0;
    else
    end
end

lambda_vec_2 = zeros(1,m);
A_p_vec_2 = zeros(1,m); A_m_vec_2 = zeros(1,m);

for q = 1:m
    lambda_vec_2(q) = lambda_vec(q);
    A_p_vec_2(q) = A_p_vec(q);
    A_m_vec_2(q) = A_m_vec(q);
end

if n_sigma == 1                         %sigma = 1.2
    figure
    xlabel('\fontsize{12}\bf\lambda'); ylabel('\bfA');
    hold on;
    axis([0 3.5 0 4.5])
    plot(lambda_vec_2, A_p_vec_2,':k','LineWidth',2)
    plot(lambda_vec_2, A_m_vec_2,':k','LineWidth',2)
    text(3, 1.2, ['\fontsize{14}\bf\sigma\fontsize{12} = ', ...
        num2str(sigma_vec(n_sigma))])

elseif n_sigma == 2                      %sigma = 2
    plot(lambda_vec_2, A_p_vec_2,'-k','LineWidth',2)
    plot(lambda_vec_2, A_m_vec_2,'-k','LineWidth',2)
    text(1.2, 1.5, ['\fontsize{14}\bf\sigma\fontsize{12} = ', ...
        num2str(sigma_vec(n_sigma))])

elseif n_sigma == 3                      %sigma = 5
    plot(lambda_vec_2, A_p_vec_2,'--k','LineWidth',2)
    plot(lambda_vec_2, A_m_vec_2,'--k','LineWidth',2)
```

```
        text(0.7, 2.5, ['\fontsize{14}\bf\sigma\fontsize{12} = ', ...
            num2str(sigma_vec(n_sigma))])

    elseif n_sigma == 4                     %sigma = 7
        plot(lambda_vec_2, A_p_vec_2,'-.k','LineWidth',2)
        plot(lambda_vec_2, A_m_vec_2,'-.k','LineWidth',2)
        text(.7, 3.7, ['\fontsize{14}\bf\sigma\fontsize{12} = ', ...
            num2str(sigma_vec(n_sigma))])

    elseif n_sigma == 5                     %sigma = -0.5
        figure
        xlabel('\fontsize{12}\bf\lambda'); ylabel('\bfA');
        hold on;
        axis([0 0.8 0 4])
        plot(lambda_vec_2, A_p_vec_2,':k','LineWidth',2)
        plot(lambda_vec_2, A_m_vec_2,':k','LineWidth',2)
        text(0.21, 0.18, ['\fontsize{14}\bf\sigma\fontsize{12} = ', ...
            num2str(sigma_vec(n_sigma))])

    elseif n_sigma == 6                      %sigma = -2
        plot(lambda_vec_2, A_p_vec_2,'-k','LineWidth',2)
        plot(lambda_vec_2, A_m_vec_2,'-k','LineWidth',2)
        text(0.4, 0.8, ['\fontsize{14}\bf\sigma\fontsize{12} = ', ...
            num2str(sigma_vec(n_sigma))])

    elseif n_sigma == 7                      %sigma = -5
        plot(lambda_vec_2, A_p_vec_2,'--k','LineWidth',2)
        plot(lambda_vec_2, A_m_vec_2,'--k','LineWidth',2)
        text(0.48, 2.2, ['\fontsize{14}\bf\sigma\fontsize{12} = ', ...
            num2str(sigma_vec(n_sigma))])

    elseif n_sigma == 8                      %sigma = -7
        plot(lambda_vec_2, A_p_vec_2,'-.k','LineWidth',2)
        plot(lambda_vec_2, A_m_vec_2,'-.k','LineWidth',2)
        text(0.5, 3.25, ['\fontsize{14}\bf\sigma\fontsize{12} = ', ...
            num2str(sigma_vec(n_sigma))])
    else
    end

    if n_sigma == 9                     %sigma = 3
        figure
        axis([0 1 0 2.5])
        xlabel('\fontsize{12}\bf\lambda'); ylabel('\bfA'); hold on;
        plot(lambda_vec_2, A_p_vec_2,'-k','LineWidth',2)
        plot(lambda_vec_2, A_m_vec_2,'-k','LineWidth',2)
    else
    end
end


%**************************************************************************
***
%**************************************************************************
***
```

```
%*********************************************************************
***


%Check number of solutions within boundaries by substitution:

clear all

n = 0;

for sigma = [3 1.2 2 5 7 -0.5 -2 -5 -7];
    if sigma == 3;
        lambda = 0.3;
        for A = [0.2, 1, 1.7];
            n = n + 1;
            a1v(n) = lambda^2 + sigma^2/(1 - sigma)^2;
            a2v(n) = 2*sigma/(1 - sigma);
            a3v(n) = 1;
            a4v(n) = -A^2/(1-sigma)^2;
            sigma_vec(n) = sigma; A_vec(n) = A; lam_vec(n) = lambda;
        end
    elseif sigma == 1.2
        lambda = 0.3;
        for A = [0.2, 0.75, 1.6];
            n = n + 1;
            a1v(n) = lambda^2 + sigma^2/(1 - sigma)^2;
            a2v(n) = 2*sigma/(1 - sigma);
            a3v(n) = 1;
            a4v(n) = -A^2/(1-sigma)^2;
            sigma_vec(n) = sigma; A_vec(n) = A; lam_vec(n) = lambda;
        end
    elseif sigma == 2
        lambda = 0.3;
        for A = [0.2, 0.75, 1.6];
            n = n + 1;
            a1v(n) = lambda^2 + sigma^2/(1 - sigma)^2;
            a2v(n) = 2*sigma/(1 - sigma);
            a3v(n) = 1;
            a4v(n) = -A^2/(1-sigma)^2;
            sigma_vec(n) = sigma; A_vec(n) = A; lam_vec(n) = lambda;
        end
    elseif sigma == 5
        lambda = 0.3;
        for A = [0.2, 1.6, 2.5];
            n = n + 1;
            a1v(n) = lambda^2 + sigma^2/(1 - sigma)^2;
            a2v(n) = 2*sigma/(1 - sigma);
            a3v(n) = 1;
            a4v(n) = -A^2/(1-sigma)^2;
            sigma_vec(n) = sigma; A_vec(n) = A; lam_vec(n) = lambda;
        end
    elseif sigma == 7
        lambda = 0.3;
        for A = [0.5, 2.5, 3.5];
            n = n + 1;
            a1v(n) = lambda^2 + sigma^2/(1 - sigma)^2;
```

```
            a2v(n) = 2*sigma/(1 - sigma);
            a3v(n) = 1;
            a4v(n) = -A^2/(1-sigma)^2;
            sigma_vec(n) = sigma; A_vec(n) = A; lam_vec(n) = lambda;
        end
    elseif sigma == -0.5
        lambda = 0.3;
        for A = [0.01, 0.1, 0.5];
            n = n + 1;
            a1v(n) = lambda^2 + sigma^2/(1 - sigma)^2;
            a2v(n) = 2*sigma/(1 - sigma);
            a3v(n) = 1;
            a4v(n) = -A^2/(1-sigma)^2;
            sigma_vec(n) = sigma; A_vec(n) = A; lam_vec(n) = lambda;
        end
    elseif sigma == -2
        lambda = 0.3;
        for A = [0.1, 0.5, 1];
            n = n + 1;
            a1v(n) = lambda^2 + sigma^2/(1 - sigma)^2;
            a2v(n) = 2*sigma/(1 - sigma);
            a3v(n) = 1;
            a4v(n) = -A^2/(1-sigma)^2;
            sigma_vec(n) = sigma; A_vec(n) = A; lam_vec(n) = lambda;
        end
    elseif sigma == -5
        lambda = 0.3;
        for A = [0.5, 1.5, 2];
            n = n + 1;
            a1v(n) = lambda^2 + sigma^2/(1 - sigma)^2;
            a2v(n) = 2*sigma/(1 - sigma);
            a3v(n) = 1;
            a4v(n) = -A^2/(1-sigma)^2;
            sigma_vec(n) = sigma; A_vec(n) = A; lam_vec(n) = lambda;
        end
    elseif sigma == -7
        lambda = 0.3;
        for A = [0.5, 2, 3];
            n = n + 1;
            a1v(n) = lambda^2 + sigma^2/(1 - sigma)^2;
            a2v(n) = 2*sigma/(1 - sigma);
            a3v(n) = 1;
            a4v(n) = -A^2/(1-sigma)^2;
            sigma_vec(n) = sigma; A_vec(n) = A; lam_vec(n) = lambda;
        end
    else
    end
end

for k = 1:n

    %Values for Z1, Z2, and Z3 were obtained by solving equation
(3.15):

    a1 = a1v(k); a2 = a2v(k); a3 = a3v(k); a4 = a4v(k);
```

```
    Z1 = ((a1^3/(27*a3^3) + a4^2/(4*a3^2) + (a2^3*a4)/(27*a3^4) - ...
        (a1^2*a2^2)/(108*a3^4) - (a1*a2*a4)/(6*a3^3))^(1/2) - ...
        a4/(2*a3) - a2^3/(27*a3^3) + (a1*a2)/(6*a3^2))^(1/3) - ...
        a2/(3*a3) - (a1/(3*a3) - a2^2/(9*a3^2))/((a1^3/(27*a3^3) + ...
        a4^2/(4*a3^2) + (a2^3*a4)/(27*a3^4) - (a1^2*a2^2)/(108*a3^4) -
...
        (a1*a2*a4)/(6*a3^3))^(1/2) - a4/(2*a3) - a2^3/(27*a3^3) + ...
        (a1*a2)/(6*a3^2))^(1/3);

    Z2 = -(a3^2*(3*3^(1/2)*1i + 3)*(-(27*a3^2*a4 - 54*a3^3*((4*a1^3*a3
- ...
        a1^2*a2^2 - 18*a1*a2*a3*a4 + 4*a2^3*a4 + ...
        27*a3^2*a4^2)/(108*a3^4))^(1/2) + 2*a2^3 - ...
        9*a1*a2*a3)/(54*a3^3))^(2/3) - a2^2*((3^(1/2)*1i)/3 - 1/3) +
...
        a1*a3*(3^(1/2)*1i - 1) + 2*a2*a3*(-(27*a3^2*a4 - ...
        54*a3^3*((4*a1^3*a3 - a1^2*a2^2 - 18*a1*a2*a3*a4 + 4*a2^3*a4 +
...
        27*a3^2*a4^2)/(108*a3^4))^(1/2) + 2*a2^3 - ...
        9*a1*a2*a3)/(54*a3^3))^(1/3))/(6*a3^2*(-(27*a3^2*a4 - ...
        54*a3^3*((4*a1^3*a3 - a1^2*a2^2 - 18*a1*a2*a3*a4 + 4*a2^3*a4 +
...
        27*a3^2*a4^2)/(108*a3^4))^(1/2) + 2*a2^3 - ...
        9*a1*a2*a3)/(54*a3^3))^(1/3));

    Z3 = -(a2^2*((3^(1/2)*1i)/3 + 1/3) - a3^2*(3*3^(1/2)*1i - ...
        3)*(-(27*a3^2*a4 - 54*a3^3*((4*a1^3*a3 - a1^2*a2^2 - ...
        18*a1*a2*a3*a4 + 4*a2^3*a4 + 27*a3^2*a4^2)/(108*a3^4))^(1/2) +
...
        2*a2^3 - 9*a1*a2*a3)/(54*a3^3))^(2/3) - a1*a3*(3^(1/2)*1i + ...
        1) + 2*a2*a3*(-(27*a3^2*a4 - 54*a3^3*((4*a1^3*a3 - a1^2*a2^2 -
...
        18*a1*a2*a3*a4 + 4*a2^3*a4 + 27*a3^2*a4^2)/(108*a3^4))^(1/2) +
...
        2*a2^3 - 9*a1*a2*a3)/(54*a3^3))^(1/3))/(6*a3^2*(-(27*a3^2*a4 -
...
        54*a3^3*((4*a1^3*a3 - a1^2*a2^2 - 18*a1*a2*a3*a4 + 4*a2^3*a4 +
...
        27*a3^2*a4^2)/(108*a3^4))^(1/2) + 2*a2^3 - ...
        9*a1*a2*a3)/(54*a3^3))^(1/3));

    %Eliminate approximation errors:

    if abs(imag(Z1)) < 0.00001
        Z1 = real(Z1);
    else
    end

    if abs(imag(Z2)) < 0.00001
        Z2 = real(Z2);
    else
    end

    if abs(imag(Z3)) < 0.00001
        Z3 = real(Z3);
    else
```

```
    end

    if k == 1
        plot(lam_vec(k), A_vec(k), '*k')
        text(lam_vec(k) + 0.03,A_vec(k)+0.24,['\fontsize{12}\bf\lambda'
...
            '\fontsize{10} = ', num2str(lam_vec(k)), ', A = ', ...
            num2str(A_vec(k))],'FontSize',10)
        text(lam_vec(k) + 0.03, A_vec(k)+0.12, ['\bfZ_1 = ',
num2str(Z1)])
        text(lam_vec(k) + 0.03, A_vec(k), ['\bfZ_2 = ', num2str(Z2)])
        text(lam_vec(k) + 0.03, A_vec(k)-0.12, ['\bfZ_3 = ',
num2str(Z3)])
    elseif k == 2
        plot(lam_vec(k), A_vec(k), '*k')
        text(lam_vec(k) - 0.21,A_vec(k)+0.24,['\fontsize{12}\bf\lambda'
...
            '\fontsize{10} = ', num2str(lam_vec(k)), ', A = ', ...
            num2str(A_vec(k))],'FontSize',10)
        text(lam_vec(k) - 0.21, A_vec(k)+0.12, ['\bfZ_1 = ',
num2str(Z1)])
        text(lam_vec(k) - 0.21, A_vec(k), ['\bfZ_2 = ', num2str(Z2)])
        text(lam_vec(k) - 0.21, A_vec(k)-0.12, ['\bfZ_3 = ',
num2str(Z3)])
    elseif k == 3
        plot(lam_vec(k), A_vec(k), '*k')
        text(lam_vec(k) + 0.03,A_vec(k)+0.48,['\fontsize{12}\bf\lambda'
...
            '\fontsize{10} = ', num2str(lam_vec(k)), ', A = ', ...
            num2str(A_vec(k))],'FontSize',10)
        text(lam_vec(k) + 0.03, A_vec(k)+0.36, ['\bfZ_1 = ',
num2str(Z1)])
        text(lam_vec(k) + 0.03, A_vec(k)+.24, ['\bfZ_2 = ',
num2str(Z2)])
        text(lam_vec(k) + 0.03, A_vec(k)+.12, ['\bfZ_3 = ',
num2str(Z3)])
    else
    end

    %Check number of real periodic solutions:

    if abs(imag(Z1)) > 0 && abs(imag(Z2)) > 0 && abs(imag(Z3)) > 0
        num_sol(k) = 0; sigma_out(k) = sigma_vec(k);
        lambda_out(k) = lam_vec(k); A_out(k) = A_vec(k);
    elseif abs(imag(Z1)) == 0 && abs(imag(Z2)) > 0 && abs(imag(Z3)) > 0
        num_sol(k) = 1; sigma_out(k) = sigma_vec(k);
        lambda_out(k) = lam_vec(k); A_out(k) = A_vec(k);
    elseif abs(imag(Z1)) > 0 && abs(imag(Z2)) == 0 && abs(imag(Z3)) > 0
        num_sol(k) = 1; sigma_out(k) = sigma_vec(k);
        lambda_out(k) = lam_vec(k); A_out(k) = A_vec(k);
    elseif abs(imag(Z1)) > 0 && abs(imag(Z2)) > 0 && abs(imag(Z3)) == 0
        num_sol(k) = 1; sigma_out(k) = sigma_vec(k);
        lambda_out(k) = lam_vec(k); A_out(k) = A_vec(k);
    elseif abs(imag(Z1)) == 0 && abs(imag(Z2)) == 0 && abs(imag(Z3)) ==
0
        num_sol(k) = 3; sigma_out(k) = sigma_vec(k);
```

```
        lambda_out(k) = lam_vec(k); A_out(k) = A_vec(k);
    else
        num_sol(k) = 2; sigma_out(k) = sigma_vec(k);
        lambda_out(k) = lam_vec(k); A_out(k) = A_vec(k);
    end
end
```

3.A.7.4 Saddle-Node Plots (Nonlinear Damping)

```
%Saddle-Node Bifurcation - Nonlinear Damping
%(boundaries of number of real periodic solutions)

close all
clear all

n_sigma = 0; k2 = 0;

for sigma = [1.2 2 5 7 -0.5 -2 -5 -7 3];
    k2 = k2+1; sigma_vec(k2) = sigma; k = 0; n_sigma = n_sigma + 1;

    for lambda = 0:.001:4
        k = k+1;
        a1 = sigma^2/(1 - sigma)^2;
        a2 = 2*sigma/(1 - sigma);
        a3 = (9/16)*lambda^2 + 1;

        %Solving for a4:
        a4_p = -a1*(-a2/(3*a3) + sqrt(a2^2 - 3*a1*a3)/(3*a3)) - ...
            a2*(-a2/(3*a3) + sqrt(a2^2 - 3*a1*a3)/(3*a3))^2 - ...
            a3*(-a2/(3*a3) + sqrt(a2^2 - 3*a1*a3)/(3*a3))^3;

        a4_m = -a1*(-a2/(3*a3) - sqrt(a2^2 - 3*a1*a3)/(3*a3)) - ...
            a2*(-a2/(3*a3) - sqrt(a2^2 - 3*a1*a3)/(3*a3))^2 - ...
            a3*(-a2/(3*a3) - sqrt(a2^2 - 3*a1*a3)/(3*a3))^3;

        %Solving for A from a4:

        A_p = sqrt(-a4_p*(1-sigma)^2); A_m = sqrt(-a4_m*(1-sigma)^2);
        lambda_vec(k) = lambda; A_p_vec(k) = A_p; A_m_vec(k) = A_m;
    end

    m = k; g = ones(1,k); h = ones(1,k);

    for n = 1:k
        if abs(A_p_vec(n)- A_m_vec(n)) < .0001
            g(n) = 0; h(n) = 0;
        elseif n > 1 && g(n-1) == 0 && h(n-1) == 0
            m = m-1; g(n) = 0; h(n) = 0;
        else
        end
    end
```

```
lambda_vec_2 = zeros(1,m);
A_p_vec_2 = zeros(1,m); A_m_vec_2 = zeros(1,m);

for q = 1:m
    lambda_vec_2(q) = lambda_vec(q);
    A_p_vec_2(q) = A_p_vec(q);
    A_m_vec_2(q) = A_m_vec(q);
end

if n_sigma == 1                          %sigma = 1.2
    figure
    xlabel('\fontsize{12}\bf\lambda'); ylabel('\bfA');
    hold on;
    axis([0 3.5 0 4.5])
    plot(lambda_vec_2, A_p_vec_2,':k','LineWidth',2)
    plot(lambda_vec_2, A_m_vec_2,':k','LineWidth',2)
    text(0.7, 1.4, ['\fontsize{14}\bf\sigma\fontsize{12} = ', ...
        num2str(sigma_vec(n_sigma))])

elseif n_sigma == 2                       %sigma = 2
    plot(lambda_vec_2, A_p_vec_2,'-k','LineWidth',2)
    plot(lambda_vec_2, A_m_vec_2,'-k','LineWidth',2)
    text(0.7, 0.9, ['\fontsize{14}\bf\sigma\fontsize{12} = ', ...
        num2str(sigma_vec(n_sigma))])

elseif n_sigma == 3                       %sigma = 5
    plot(lambda_vec_2, A_p_vec_2,'--k','LineWidth',2)
    plot(lambda_vec_2, A_m_vec_2,'--k','LineWidth',2)
    text(0.7, 2.5, ['\fontsize{14}\bf\sigma\fontsize{12} = ', ...
        num2str(sigma_vec(n_sigma))])

elseif n_sigma == 4                       %sigma = 7
    plot(lambda_vec_2, A_p_vec_2,'-.k','LineWidth',2)
    plot(lambda_vec_2, A_m_vec_2,'-.k','LineWidth',2)
    text(0.7, 3.3, ['\fontsize{14}\bf\sigma\fontsize{12} = ', ...
        num2str(sigma_vec(n_sigma))])

elseif n_sigma == 5                       %sigma = -0.5
    figure
    xlabel('\fontsize{12}\bf\lambda'); ylabel('\bfA');
    hold on;
    axis([0 0.8 0 4])
    plot(lambda_vec_2, A_p_vec_2,':k','LineWidth',2)
    plot(lambda_vec_2, A_m_vec_2,':k','LineWidth',2)
    text(0.7, 0.25, ['\fontsize{14}\bf\sigma\fontsize{12} = ', ...
        num2str(sigma_vec(n_sigma))])

elseif n_sigma == 6                        %sigma = -2
    plot(lambda_vec_2, A_p_vec_2,'-k','LineWidth',2)
    plot(lambda_vec_2, A_m_vec_2,'-k','LineWidth',2)
    text(0.7, 0.8, ['\fontsize{14}\bf\sigma\fontsize{12} = ', ...
        num2str(sigma_vec(n_sigma))])

elseif n_sigma == 7                        %sigma = -5
```

```
        plot(lambda_vec_2, A_p_vec_2,'--k','LineWidth',2)
        plot(lambda_vec_2, A_m_vec_2,'--k','LineWidth',2)
        text(0.7, 2, ['\fontsize{14}\bf\sigma\fontsize{12} = ', ...
            num2str(sigma_vec(n_sigma))])

    elseif n_sigma == 8                     %sigma = -7
        plot(lambda_vec_2, A_p_vec_2,'-.k','LineWidth',2)
        plot(lambda_vec_2, A_m_vec_2,'-.k','LineWidth',2)
        text(0.7, 2.8, ['\fontsize{14}\bf\sigma\fontsize{12} = ', ...
            num2str(sigma_vec(n_sigma))])
    else
    end

    if n_sigma == 9                     %sigma = 3
        figure
        axis([0 1 0 2.5])
        xlabel('\fontsize{12}\bf\lambda'); ylabel('\bfA'); hold on;
        plot(lambda_vec_2, A_p_vec_2,'-k','LineWidth',2)
        plot(lambda_vec_2, A_m_vec_2,'-k','LineWidth',2)
    else
    end
end


%**************************************************************************
***
%**************************************************************************
***
%**************************************************************************
***


%Check number of solutions within boundaries by substitution:

clear all

n = 0;

for sigma = [3 1.2 2 5 7 -0.5 -2 -5 -7];
    if sigma == 3;
        lambda = 0.3;
        for A = [0.2, 1, 1.7];
            n = n + 1;
            a1v(n) = sigma^2/(1 - sigma)^2;
            a2v(n) = 2*sigma/(1 - sigma);
            a3v(n) = (9/16)*lambda^2 + 1;
            a4v(n) = -A^2/(1-sigma)^2;
            sigma_vec(n) = sigma; A_vec(n) = A; lam_vec(n) = lambda;
        end
    elseif sigma == 1.2
        lambda = 0.3;
        for A = [0.2, 0.75, 1.6];
            n = n + 1;
            a1v(n) = sigma^2/(1 - sigma)^2;
            a2v(n) = 2*sigma/(1 - sigma);
```

```
        a3v(n) = (9/16)*lambda^2 + 1;
        a4v(n) = -A^2/(1-sigma)^2;
        sigma_vec(n) = sigma; A_vec(n) = A; lam_vec(n) = lambda;
    end
elseif sigma == 2
    lambda = 0.3;
    for A = [0.2, 0.75, 1.6];
        n = n + 1;
        a1v(n) = sigma^2/(1 - sigma)^2;
        a2v(n) = 2*sigma/(1 - sigma);
        a3v(n) = (9/16)*lambda^2 + 1;
        a4v(n) = -A^2/(1-sigma)^2;
        sigma_vec(n) = sigma; A_vec(n) = A; lam_vec(n) = lambda;
    end
elseif sigma == 5
    lambda = 0.3;
    for A = [0.2, 1.6, 2.5];
        n = n + 1;
        a1v(n) = sigma^2/(1 - sigma)^2;
        a2v(n) = 2*sigma/(1 - sigma);
        a3v(n) = (9/16)*lambda^2 + 1;
        a4v(n) = -A^2/(1-sigma)^2;
        sigma_vec(n) = sigma; A_vec(n) = A; lam_vec(n) = lambda;
    end
elseif sigma == 7
    lambda = 0.3;
    for A = [0.5, 2.5, 3.5];
        n = n + 1;
        a1v(n) = sigma^2/(1 - sigma)^2;
        a2v(n) = 2*sigma/(1 - sigma);
        a3v(n) = (9/16)*lambda^2 + 1;
        a4v(n) = -A^2/(1-sigma)^2;
        sigma_vec(n) = sigma; A_vec(n) = A; lam_vec(n) = lambda;
    end
elseif sigma == -0.5
    lambda = 0.3;
    for A = [0.01, 0.1, 0.5];
        n = n + 1;
        a1v(n) = sigma^2/(1 - sigma)^2;
        a2v(n) = 2*sigma/(1 - sigma);
        a3v(n) = (9/16)*lambda^2 + 1;
        a4v(n) = -A^2/(1-sigma)^2;
        sigma_vec(n) = sigma; A_vec(n) = A; lam_vec(n) = lambda;
    end
elseif sigma == -2
    lambda = 0.3;
    for A = [0.1, 0.5, 1];
        n = n + 1;
        a1v(n) = sigma^2/(1 - sigma)^2;
        a2v(n) = 2*sigma/(1 - sigma);
        a3v(n) = (9/16)*lambda^2 + 1;
        a4v(n) = -A^2/(1-sigma)^2;
        sigma_vec(n) = sigma; A_vec(n) = A; lam_vec(n) = lambda;
    end
elseif sigma == -5
    lambda = 0.3;
    for A = [0.5, 1.5, 2];
```

```
            n = n + 1;
            a1v(n) = sigma^2/(1 - sigma)^2;
            a2v(n) = 2*sigma/(1 - sigma);
            a3v(n) = (9/16)*lambda^2 + 1;
            a4v(n) = -A^2/(1-sigma)^2;
            sigma_vec(n) = sigma; A_vec(n) = A; lam_vec(n) = lambda;
        end
    elseif sigma == -7
        lambda = 0.3;
        for A = [0.5, 2, 3];
            n = n + 1;
            a1v(n) = sigma^2/(1 - sigma)^2;
            a2v(n) = 2*sigma/(1 - sigma);
            a3v(n) = (9/16)*lambda^2 + 1;
            a4v(n) = -A^2/(1-sigma)^2;
            sigma_vec(n) = sigma; A_vec(n) = A; lam_vec(n) = lambda;
        end
    else
    end
end

for k = 1:n

    %Values for Z1, Z2, and Z3 were obtained by solving equation
(3.24).

    a1 = a1v(k); a2 = a2v(k); a3 = a3v(k); a4 = a4v(k);

    Z1 = ((a1^3/(27*a3^3) + a4^2/(4*a3^2) + (a2^3*a4)/(27*a3^4) - ...
        (a1^2*a2^2)/(108*a3^4) - (a1*a2*a4)/(6*a3^3))^(1/2) - ...
        a4/(2*a3) - a2^3/(27*a3^3) + (a1*a2)/(6*a3^2))^(1/3) - ...
        a2/(3*a3) - (a1/(3*a3) - a2^2/(9*a3^2))/((a1^3/(27*a3^3) + ...
        a4^2/(4*a3^2) + (a2^3*a4)/(27*a3^4) - (a1^2*a2^2)/(108*a3^4) -
...
        (a1*a2*a4)/(6*a3^3))^(1/2) - a4/(2*a3) - a2^3/(27*a3^3) + ...
        (a1*a2)/(6*a3^2))^(1/3);

    Z2 = -(a3^2*(3*3^(1/2)*1i + 3)*(-(27*a3^2*a4 - 54*a3^3*((4*a1^3*a3
- ...
        a1^2*a2^2 - 18*a1*a2*a3*a4 + 4*a2^3*a4 + ...
        27*a3^2*a4^2)/(108*a3^4))^(1/2) + 2*a2^3 - ...
        9*a1*a2*a3)/(54*a3^3))^(2/3) - a2^2*((3^(1/2)*1i)/3 - 1/3) +
...
        a1*a3*(3^(1/2)*1i - 1) + 2*a2*a3*(-(27*a3^2*a4 - ...
        54*a3^3*((4*a1^3*a3 - a1^2*a2^2 - 18*a1*a2*a3*a4 + 4*a2^3*a4 +
...
        27*a3^2*a4^2)/(108*a3^4))^(1/2) + 2*a2^3 - ...
        9*a1*a2*a3)/(54*a3^3))^(1/3))/(6*a3^2*(-(27*a3^2*a4 - ...
        54*a3^3*((4*a1^3*a3 - a1^2*a2^2 - 18*a1*a2*a3*a4 + 4*a2^3*a4 +
...
        27*a3^2*a4^2)/(108*a3^4))^(1/2) + 2*a2^3 - ...
        9*a1*a2*a3)/(54*a3^3))^(1/3));

    Z3 = -(a2^2*((3^(1/2)*1i)/3 + 1/3) - a3^2*(3*3^(1/2)*1i - ...
        3)*(-(27*a3^2*a4 - 54*a3^3*((4*a1^3*a3 - a1^2*a2^2 - ...
```

```
        18*a1*a2*a3*a4 + 4*a2^3*a4 + 27*a3^2*a4^2)/(108*a3^4))^(1/2) +
...
        2*a2^3 - 9*a1*a2*a3)/(54*a3^3))^(2/3) - a1*a3*(3^(1/2)*1i + ...
        1) + 2*a2*a3*(-(27*a3^2*a4 - 54*a3^3*((4*a1^3*a3 - a1^2*a2^2 -
...
        18*a1*a2*a3*a4 + 4*a2^3*a4 + 27*a3^2*a4^2)/(108*a3^4))^(1/2) +
...
        2*a2^3 - 9*a1*a2*a3)/(54*a3^3))^(1/3))/(6*a3^2*(-(27*a3^2*a4 -
...
        54*a3^3*((4*a1^3*a3 - a1^2*a2^2 - 18*a1*a2*a3*a4 + 4*a2^3*a4 +
...
        27*a3^2*a4^2)/(108*a3^4))^(1/2) + 2*a2^3 - ...
        9*a1*a2*a3)/(54*a3^3))^(1/3));

    %Eliminate approximation errors:

    if abs(imag(Z1)) < 0.00001
        Z1 = real(Z1);
    else
    end

    if abs(imag(Z2)) < 0.00001
        Z2 = real(Z2);
    else
    end

    if abs(imag(Z3)) < 0.00001
        Z3 = real(Z3);
    else
    end

    if k == 1
        plot(lam_vec(k), A_vec(k), '*k')
        text(lam_vec(k) + 0.03,A_vec(k)+0.24,['\fontsize{12}\bf\lambda'
...
            '\fontsize{10} = ', num2str(lam_vec(k)), ', A = ', ...
            num2str(A_vec(k))],'FontSize',10)
        text(lam_vec(k) + 0.03, A_vec(k)+0.12, ['\bfZ_1 = ',
num2str(Z1)])
        text(lam_vec(k) + 0.03, A_vec(k), ['\bfZ_2 = ', num2str(Z2)])
        text(lam_vec(k) + 0.03, A_vec(k)-0.12, ['\bfZ_3 = ',
num2str(Z3)])
    elseif k == 2
        plot(lam_vec(k), A_vec(k), '*k')
        text(lam_vec(k) - 0.18,A_vec(k)+0.24,['\fontsize{12}\bf\lambda'
...
            '\fontsize{10} = ', num2str(lam_vec(k)), ', A = ', ...
            num2str(A_vec(k))],'FontSize',10)
        text(lam_vec(k) - 0.18, A_vec(k)+0.12, ['\bfZ_1 = ',
num2str(Z1)])
        text(lam_vec(k) - 0.18, A_vec(k), ['\bfZ_2 = ', num2str(Z2)])
        text(lam_vec(k) - 0.18, A_vec(k)-0.12, ['\bfZ_3 = ',
num2str(Z3)])
    elseif k == 3
        plot(lam_vec(k), A_vec(k), '*k')
```

```
        text(lam_vec(k) + 0.03,A_vec(k)+0.24,['\fontsize{12}\bf\lambda'
...
            '\fontsize{10} = ', num2str(lam_vec(k)), ', A = ', ...
            num2str(A_vec(k))],'FontSize',10)
        text(lam_vec(k) + 0.03, A_vec(k)+0.12, ['\bfZ_1 = ',
num2str(Z1)])
        text(lam_vec(k) + 0.03, A_vec(k), ['\bfZ_2 = ', num2str(Z2)])
        text(lam_vec(k) + 0.03, A_vec(k)-0.12, ['\bfZ_3 = ',
num2str(Z3)])
    else
    end

    %Check number of real periodic solutions:

    if abs(imag(Z1)) > 0 && abs(imag(Z2)) > 0 && abs(imag(Z3)) > 0
        num_sol(k) = 0; sigma_out(k) = sigma_vec(k);
        lambda_out(k) = lam_vec(k); A_out(k) = A_vec(k);
    elseif abs(imag(Z1)) == 0 && abs(imag(Z2)) > 0 && abs(imag(Z3)) > 0
        num_sol(k) = 1; sigma_out(k) = sigma_vec(k);
        lambda_out(k) = lam_vec(k); A_out(k) = A_vec(k);
    elseif abs(imag(Z1)) > 0 && abs(imag(Z2)) == 0 && abs(imag(Z3)) > 0
        num_sol(k) = 1; sigma_out(k) = sigma_vec(k);
        lambda_out(k) = lam_vec(k); A_out(k) = A_vec(k);
    elseif abs(imag(Z1)) > 0 && abs(imag(Z2)) > 0 && abs(imag(Z3)) == 0
        num_sol(k) = 1; sigma_out(k) = sigma_vec(k);
        lambda_out(k) = lam_vec(k); A_out(k) = A_vec(k);
    elseif abs(imag(Z1)) == 0 && abs(imag(Z2)) == 0 && abs(imag(Z3)) ==
0
        num_sol(k) = 3; sigma_out(k) = sigma_vec(k);
        lambda_out(k) = lam_vec(k); A_out(k) = A_vec(k);
    else
        num_sol(k) = 2; sigma_out(k) = sigma_vec(k);
        lambda_out(k) = lam_vec(k); A_out(k) = A_vec(k);
    end
end
```

3.A.7.5 Hopf Bifurcation Example

```
close all
clear all

mu_span = 4; mu_step = 0.0001;
r_trivial = 0;

for alpha = [-1 1]
    ks = 0; ku = 0; k_ts = 0; k_tu = 0;
    mu_stable = zeros(1,2); mu_unstable = mu_stable;

    for mu = -mu_span/2:mu_step:mu_span/2

        if mu <= 0
            k_ts = k_ts + 1;
            r_trivial_stable(k_ts) = r_trivial;
```

```
            mu_trivial_stable(k_ts) = mu;
        elseif mu > 0
            k_tu = k_tu + 1;
            r_trivial_unstable(k_tu) = r_trivial;
            mu_trivial_unstable(k_tu) = mu;
        else
        end

        r_pos = real(1i*sqrt(mu/alpha));
        r_neg = real(-1i*sqrt(mu/alpha));

        lambda = -2*mu;

        if lambda <= 0
            if abs(r_pos) > .00001
                ks = ks + 1;
                r_pos_stable(ks) = r_pos;
                r_neg_stable(ks) = r_neg;
                mu_stable(ks) = mu;
            else
            end
        elseif lambda > 0
            if abs(r_pos) > .00001
                ku = ku + 1;
                r_pos_unstable(ku) = r_pos;
                r_neg_unstable(ku) = r_neg;
                mu_unstable(ku) = mu;
            else
            end
        else
        end
    end

    figure
    hold on; grid on;
    if length(mu_stable) > 5
        plot(mu_stable,r_pos_stable, '-k','LineWidth',2)
        plot(mu_stable, r_neg_stable, '-k','LineWidth',2)
    else
    end
    if length(mu_unstable) > 5
        plot(mu_unstable, r_pos_unstable, '--k','LineWidth',2)
        plot(mu_unstable, r_neg_unstable, '--k','LineWidth',2)
    else
    end
    plot(mu_trivial_stable,r_trivial_stable,'-k','LineWidth',2)
    plot(mu_trivial_unstable,r_trivial_unstable,'--k','LineWidth',2)
    xlabel('\bf\mu'); ylabel('\bfr');
end
```

3.A.7.6 Hopf Bifurcation Analysis - Solving for Coefficients, γ_i (Linear Damping)

```
%Calculation of the coefficients of the characteristic polynomial
%(linear damping case)

close all
clear all

syms del1 del1_conj del2 del2_conj lambda e sigma N20 phi20 phi20_conj
mu

%From equations (3.45):

del1_dot = -1i*e/(2*(1+e))*(del1-del2) + ...
    1i*e*sigma*(del1+e*del2)/(2*(1+e));

del1_conj_dot = 1i*e/(2*(1+e))*(del1_conj-del2_conj) - ...
    1i*e*sigma*(del1_conj+e*del2_conj)/(2*(1+e));

del2_dot = -lambda*(1+e)/2*del2 - 1i/(2*(1+e))*(del2-del1) + ...
    1i*e*sigma/(2*(1+e))*(del1+e*del2) + 1i*(1+e)*N20^2*del2 + ...
    1i*(1+e)/2*phi20^2*del2_conj;

del2_conj_dot = -lambda*(1+e)/2*del2_conj + ...
    1i/(2*(1+e))*(del2_conj-del1_conj) - ...
    1i*e*sigma/(2*(1+e))*(del1_conj+e*del2_conj) - ...
    1i*(1+e)*N20^2*del2_conj - 1i*(1+e)/2*phi20_conj^2*del2;

f = [del1_dot; del1_conj_dot; del2_dot; del2_conj_dot];
v = [del1 del1_conj del2 del2_conj];
DxF = jacobian(f,v);
CHAR = DxF - mu*eye(4);
C_poly = det(CHAR);
C_coeffs = coeffs(C_poly, mu);

gamma1 = simple(collect(C_coeffs(4), N20))
gamma2 = simple(collect(C_coeffs(3), N20))
gamma3 = simple(collect(C_coeffs(2), N20))
gamma4 = simple(collect(C_coeffs(1), N20))
```

3.A.7.7 Hopf Bifurcation Analysis - Simplification of Coefficients, γ_i Based on the MATLAB Output (Linear Damping)

The MATLAB output for the coefficients of the characteristic polynomial is as follows:

gamma1 =

lambda*(e + 1)

gamma2 =

(4*N20^4*e^2 + 8*N20^4*e + 4*N20^4 + 4*N20^2*e^2*sigma - 4*N20^2 + e^2*lambda^2 - e^2*phi20^2*phi20_conj^2 + e^2*sigma^2 + 2*e*lambda^2 - 2*e*phi20^2*phi20_conj^2 + lambda^2 - phi20^2*phi20_conj^2 + 1)/4

gamma3 =

(e*lambda*(e*sigma^2 + 1))/4

gamma4 =

(e^2*(4*N20^4*sigma^2 - 8*N20^4*sigma + 4*N20^4 - 4*N20^2*sigma^2 + 4*N20^2*sigma + lambda^2*sigma^2 - 2*lambda^2*sigma + lambda^2 - phi20^2*phi20_conj^2*sigma^2 + 2*phi20^2*phi20_conj^2*sigma - phi20^2*phi20_conj^2 + sigma^2))/16

Rewriting this output into a more reader-friendly form yields

$$\gamma_1 = \lambda(1+\varepsilon),$$

$$\gamma_2 = \frac{1}{4}\big(4{N_{20}}^4\varepsilon^2 + 8{N_{20}}^4\varepsilon + 4{N_{20}}^4 + 4{N_{20}}^2\varepsilon^2\sigma - 4{N_{20}}^2 + \varepsilon^2\lambda^2 - \varepsilon^2{\varphi_{20}}^2{\varphi_{20}^*}^2 + \varepsilon^2\sigma^2 + 2\varepsilon\lambda^2 - 2\varepsilon{\varphi_{20}}^2{\varphi_{20}^*}^2 + \lambda^2 - {\varphi_{20}}^2{\varphi_{20}^*}^2 + 1\big)$$

$$\gamma_3 = \tfrac{1}{4}\lambda\varepsilon(\varepsilon\sigma^2 + 1),$$

$$\gamma_4 = \frac{\varepsilon^2}{16}\big(4{N_{20}}^4\sigma^2 - 8{N_{20}}^4\sigma + 4{N_{20}}^4 - 4{N_{20}}^2\sigma^2 + 4{N_{20}}^2\sigma + \lambda^2\sigma^2 - 2\lambda^2\sigma + \lambda^2 - {\varphi_{20}}^2{\varphi_{20}^*}^2\sigma^2 + 2{\varphi_{20}}^2{\varphi_{20}^*}^2\sigma - {\varphi_{20}}^2{\varphi_{20}^*}^2 + \sigma^2\big)$$

Noting that

$${\varphi_{20}}^2{\varphi_{20}^*}^2 = {N_{20}}^4,$$

as shown in Section 3.A.6.2, γ_2 and γ_4 can be reduced as follows

$$\gamma_2 = \frac{1}{4}\left(4{N_{20}}^4\varepsilon^2 + 8{N_{20}}^4\varepsilon + 4{N_{20}}^4 + 4{N_{20}}^2\varepsilon^2\sigma - 4{N_{20}}^2 + \varepsilon^2\lambda^2 - \varepsilon^2{N_{20}}^4 + \varepsilon^2\sigma^2 + 2\varepsilon\lambda^2 - 2\varepsilon{N_{20}}^4 + \lambda^2 - {N_{20}}^4 + 1\right)$$

$$\gamma_4 = \frac{\varepsilon^2}{16}\left(4{N_{20}}^4\sigma^2 - 8{N_{20}}^4\sigma + 4{N_{20}}^4 - 4{N_{20}}^2\sigma^2 + 4{N_{20}}^2\sigma + \lambda^2\sigma^2 - 2\lambda^2\sigma + \lambda^2 - {N_{20}}^4\sigma^2 + 2{N_{20}}^4\sigma - {N_{20}}^4 + \sigma^2\right)$$

Finally, γ_2 and γ_4 can be reduced further to

$$\gamma_2 = \left(\frac{3}{2}\varepsilon + \frac{3}{4} + \frac{3}{4}\varepsilon^2\right){N_{20}}^4 + (\varepsilon^2\sigma - 1){N_{20}}^2 + \frac{1}{4}\lambda^2(\varepsilon + 1)^2 + \frac{1}{4}(\varepsilon^2\sigma^2 + 1),$$

$$\gamma_4 = \tfrac{3}{16}\varepsilon^2(1-\sigma)^2{N_{20}}^4 + \tfrac{1}{4}\varepsilon^2\sigma(1-\sigma){N_{20}}^2 + \tfrac{1}{16}\varepsilon^2[(1-\sigma)^2\lambda^2 + \sigma^2].$$

3.A.7.8 Hopf Bifurcation Analysis – Solving for Coefficients, v_i (Linear Damping)

```
%Calculation of the coefficients of v1*z^2 + v2*z + v3 = 0
%(linear damping case)

close all
clear all

syms lambda e sigma z

gamma1 = lambda*(e + 1);

gamma2 = (4*z^2*e^2 + 8*z^2*e + 4*z^2 + 4*z*e^2*sigma - 4*z + ...
    e^2*lambda^2 - e^2*z^2 + e^2*sigma^2 + 2*e*lambda^2 - ...
    2*e*z^2 + lambda^2 - z^2 + 1)/4;

gamma3 = (e*lambda*(e*sigma^2 + 1))/4;

gamma4 = (e^2*(4*z^2*sigma^2 - 8*z^2*sigma + 4*z^2 - 4*z*sigma^2 + ...
    4*z*sigma + lambda^2*sigma^2 - 2*lambda^2*sigma + lambda^2 - ...
    z^2*sigma^2 + 2*z^2*sigma - z^2 + sigma^2))/16;

V = collect(gamma3^2 - gamma2*gamma3*gamma1 + gamma4*gamma1^2, z);
C = coeffs(V,z);

v1 = simple(C(3))
v2 = simple(C(2))
v3 = simple(C(1))
```

3.A.7.9 Hopf Bifurcation Analysis - Simplification of Coefficients v_i Based on the MATLAB Output (Linear Damping)

The MATLAB output for the coefficients is as follows:

v1 =

-(3*e*lambda^2*(e*sigma + 1)^2*(e + 1)^2)/16

v2 =

-(e*lambda^2*(e*sigma - 1)*(e*sigma + 1)^2*(e + 1))/4

v3 =

-(e*lambda^2*(e*sigma + 1)^2*(e^2*lambda^2 + e^2*sigma^2 + 2*e*lambda^2 - 2*e*sigma + lambda^2 + 1))/16

3.A.7.10 Hopf Bifurcation Plots (Linear Damping)

```
%Hopf Bifurcation (linear damping)

%Generation of the First Plot

close all
clear all

sigma = 0.5; e = 0.05; k = 0;

for lambda = 0:0.001:2
    k = k+1;

    v1 = -(3*e*lambda^2*(e*sigma + 1)^2*(e + 1)^2)/16;

    v2 = -(e*lambda^2*(e*sigma - 1)*(e*sigma + 1)^2*(e + 1))/4;

    v3 = -(e*lambda^2*(e*sigma + 1)^2*(e^2*lambda^2 + e^2*sigma^2 + ...
        2*e*lambda^2 - 2*e*sigma + lambda^2 + 1))/16;

    a1 = lambda^2 + sigma^2/(1 - sigma)^2;    %alpha 1
    a2 = 2*sigma/(1 - sigma);                 %alpha 2
    a3 = 1;                                   %alpha 3

    z1 = (-v2 - sqrt(v2^2 - 4*v3*v1))/(2*v1); %Boundary of stability
    z2 = (-v2 + sqrt(v2^2 - 4*v3*v1))/(2*v1); %Boundary of stability
```

```
    A1p(k) = (1-sigma)*sqrt(a1*z1 + a2*z1^2 + a3*z1^3);  %corresponds
to z1
    A1m(k) = -(1-sigma)*sqrt(a1*z1 + a2*z1^2 + a3*z1^3); %corresponds
to z1
    A2p(k) = (1-sigma)*sqrt(a1*z2 + a2*z2^2 + a3*z2^3);  %corresponds
to z2
    A2m(k) = -(1-sigma)*sqrt(a1*z2 + a2*z2^2 + a3*z2^3); %corresponds
to z2

    lam_vec(k) = lambda;
end

%Truncating the plot:

k2 = 0;
for m = 1:k
    if lam_vec(m) < 0.54
        k2 = k2 + 1;
        A1p_2(k2) = A1p(m); lam_vec_2(k2) = lam_vec(m);
        A2p_2(k2) = A2p(m); lam_vec_2(k2) = lam_vec(m);
    else
    end
end

figure
plot(lam_vec_2,A1p_2,'k',lam_vec_2,A2p_2,'-k','LineWidth',2)
xlabel('\fontsize{12}\bf\lambda'); ylabel('\bfA'); grid on;
axis([0 0.8 0.2 1]);


%**************************************************************************
**
%**************************************************************************
**
%**************************************************************************
**


%Generation of Second Plot

%The next two lines that are commented out were used to find z.
%syms z
%z_solved = solve('a1*z + a2*z^2 + a3*z^3 +a4 = 0', z)

clear lambda k k2 z1 z2

lambda = 0.2; k = 0; k2 = 0;

for A = 0.2:0.01:2
    k = k+1;

    %From Eq. (6.1-58):
```

```
    a1 = lambda^2 + sigma^2/(1 - sigma)^2;    %alpha 1
    a2 = 2*sigma/(1 - sigma);                 %alpha 2
    a3 = 1;                                   %alpha 3
    a4 = -A^2/(1-sigma)^2;                    %alpha 4

    %values for z, z1, and z2 were obtained by using solve command

    z1 = ((a1^3/(27*a3^3) + a4^2/(4*a3^2) + (a2^3*a4)/(27*a3^4) - ...
        (a1^2*a2^2)/(108*a3^4) - (a1*a2*a4)/(6*a3^3))^(1/2) - ...
        a4/(2*a3) - a2^3/(27*a3^3) + (a1*a2)/(6*a3^2))^(1/3) - ...
        a2/(3*a3) - (a1/(3*a3) - a2^2/(9*a3^2))/((a1^3/(27*a3^3) + ...
        a4^2/(4*a3^2) + (a2^3*a4)/(27*a3^4) - (a1^2*a2^2)/(108*a3^4) -
...
        (a1*a2*a4)/(6*a3^3))^(1/2) - a4/(2*a3) - a2^3/(27*a3^3) + ...
        (a1*a2)/(6*a3^2))^(1/3);

    z2 = (a1/(3*a3) - a2^2/(9*a3^2))/(2*((a1^3/(27*a3^3) + ...
        a4^2/(4*a3^2) + (a2^3*a4)/(27*a3^4) - (a1^2*a2^2)/(108*a3^4) -
...
        (a1*a2*a4)/(6*a3^3))^(1/2) - a4/(2*a3) - a2^3/(27*a3^3) + ...
        (a1*a2)/(6*a3^2))^(1/3)) - ((a1^3/(27*a3^3) + a4^2/(4*a3^2) +
...
        (a2^3*a4)/(27*a3^4) - (a1^2*a2^2)/(108*a3^4) - ...
        (a1*a2*a4)/(6*a3^3))^(1/2) - a4/(2*a3) - a2^3/(27*a3^3) + ...
        (a1*a2)/(6*a3^2))^(1/3)/2 - a2/(3*a3) - ...
        (3^(1/2)*1i*(((a1^3/(27*a3^3) + a4^2/(4*a3^2) + ...
        (a2^3*a4)/(27*a3^4) - (a1^2*a2^2)/(108*a3^4) - ...
        (a1*a2*a4)/(6*a3^3))^(1/2) - a4/(2*a3) - a2^3/(27*a3^3) + ...
        (a1*a2)/(6*a3^2))^(1/3) + (a1/(3*a3) - ...
        a2^2/(9*a3^2))/((a1^3/(27*a3^3) + a4^2/(4*a3^2) + ...
        (a2^3*a4)/(27*a3^4) - (a1^2*a2^2)/(108*a3^4) - ...
        (a1*a2*a4)/(6*a3^3))^(1/2) - a4/(2*a3) - a2^3/(27*a3^3) + ...
        (a1*a2)/(6*a3^2))^(1/3)))/2;

    z3 = (a1/(3*a3) - a2^2/(9*a3^2))/(2*((a1^3/(27*a3^3) + ...
        a4^2/(4*a3^2) + (a2^3*a4)/(27*a3^4) - (a1^2*a2^2)/(108*a3^4) -
...
        (a1*a2*a4)/(6*a3^3))^(1/2) - a4/(2*a3) - a2^3/(27*a3^3) + ...
        (a1*a2)/(6*a3^2))^(1/3)) - ((a1^3/(27*a3^3) + a4^2/(4*a3^2) +
...
        (a2^3*a4)/(27*a3^4) - (a1^2*a2^2)/(108*a3^4) - ...
        (a1*a2*a4)/(6*a3^3))^(1/2) - a4/(2*a3) - a2^3/(27*a3^3) + ...
        (a1*a2)/(6*a3^2))^(1/3)/2 - a2/(3*a3) + ...
        (3^(1/2)*1i*(((a1^3/(27*a3^3) + a4^2/(4*a3^2) + ...
        (a2^3*a4)/(27*a3^4) - (a1^2*a2^2)/(108*a3^4) - ...
        (a1*a2*a4)/(6*a3^3))^(1/2) - a4/(2*a3) - a2^3/(27*a3^3) + ...
        (a1*a2)/(6*a3^2))^(1/3) + (a1/(3*a3) - ...
        a2^2/(9*a3^2))/((a1^3/(27*a3^3) + a4^2/(4*a3^2) + ...
        (a2^3*a4)/(27*a3^4) - (a1^2*a2^2)/(108*a3^4) - ...
        (a1*a2*a4)/(6*a3^3))^(1/2) - a4/(2*a3) - a2^3/(27*a3^3) + ...
        (a1*a2)/(6*a3^2))^(1/3)))/2;

    %Since z = N20^2, the following solves for N20:

    N20_z1p(k) = sqrt(z1); N20_z1m(k) = -sqrt(z1);
    N20_z2p(k) = sqrt(z2); N20_z2m(k) = -sqrt(z2);
```

```
    N20_z3p(k) = sqrt(z3); N20_z3m(k) = -sqrt(z3);

    A_vec(k) = A;
end

%*********************************************************************
*

%Find unstable locations:

k_u = 0; k_check = 0;
for ind = 1:length(N20_z1p)
    N20 = N20_z1p(ind);
    z = N20^2;

    gamma1 = lambda*(e + 1);

    gamma2 = (4*z^2*e^2 + 8*z^2*e + 4*z^2 + 4*z*e^2*sigma - 4*z + ...
        e^2*lambda^2 - e^2*z^2 + e^2*sigma^2 + 2*e*lambda^2 - ...
        2*e*z^2 + lambda^2 - z^2 + 1)/4;

    gamma3 = (e*lambda*(e*sigma^2 + 1))/4;

    gamma4 = (e^2*(4*z^2*sigma^2 - 8*z^2*sigma + 4*z^2 - 4*z*sigma^2 +
...
        4*z*sigma + lambda^2*sigma^2 - 2*lambda^2*sigma + lambda^2 -
...
        z^2*sigma^2 + 2*z^2*sigma - z^2 + sigma^2))/16;

    %Characteristic equation
    CHAR = [1 gamma1 gamma2 gamma3 gamma4];

    R = roots(CHAR);

    %Looking for positive real parts of the eigenvalues:

    for kR = 1:length(R)
        if real(R(kR)) > 0
            k_u = k_u + 1;
            z_unstable(k_u) = N20;
            A_unstable(k_u) = A_vec(ind);
            k_check = k_check + 1;
        else
        end
    end
end

if k_check == 0
    No_Plot = 0;
else
    No_Plot = 1;
end

figure
```

```
plot(A_vec,N20_z1p,'-k','LineWidth',2)
hold on
if No_Plot == 1;
plot(A_unstable, z_unstable, '*k')
else
end
xlabel('\bfA'); ylabel('\bfN_2_0'); grid on;
axis([0.2 2 0.2 1.6]);


%**************************************************************************
**
%**************************************************************************
**
%**************************************************************************
**


%Generation of the third plot

clear all

sigma = 1.2; e = 0.05; k = 0;

for lambda = 0:0.001:4

    k = k+1;

    v1 = -(3*e*lambda^2*(e*sigma + 1)^2*(e + 1)^2)/16;

    v2 = -(e*lambda^2*(e*sigma - 1)*(e*sigma + 1)^2*(e + 1))/4;

    v3 = -(e*lambda^2*(e*sigma + 1)^2*(e^2*lambda^2 + e^2*sigma^2 + ...
        2*e*lambda^2 - 2*e*sigma + lambda^2 + 1))/16;

    a1 = lambda^2 + sigma^2/(1 - sigma)^2;    %alpha 1
    a2 = 2*sigma/(1 - sigma);                 %alpha 2
    a3 = 1;                                   %alpha 3

    z1 = (-v2 - sqrt(v2^2 - 4*v3*v1))/(2*v1); %Boundary of stability
    z2 = (-v2 + sqrt(v2^2 - 4*v3*v1))/(2*v1); %Boundary of stability

    A1p = (1-sigma)*sqrt(a1*z1 + a2*z1^2 + a3*z1^3);  %corresponds to
z1
    A1m = -(1-sigma)*sqrt(a1*z1 + a2*z1^2 + a3*z1^3); %corresponds to
z1
    A2p = (1-sigma)*sqrt(a1*z2 + a2*z2^2 + a3*z2^3);  %corresponds to
z2
    A2m = -(1-sigma)*sqrt(a1*z2 + a2*z2^2 + a3*z2^3); %corresponds to
z2


    A11(k) = A1p; A12(k) = A1m; %corresponds to z1
    A21(k) = A2p; A22(k) = A2m; %corresponds to z2
    lam_vec(k) = lambda;
```

```
    if lambda == 0.001
        A12_0 = A12(k); A22_0 = A22(k);
    else
    end
end

k2 = 0;
for n = 1:k
    if lam_vec(n) < .525
        k2 = k2 + 1;
        A11_2(k2) = A11(n); A12_2(k2) = A12(n);
        A21_2(k2) = A21(n); A22_2(k2) = A22(n);
        lam_vec_2(k2) = lam_vec(n);
    else
    end
end

k_SN = 0;
for lambda_SN = 0:.001:4
    k_SN = k_SN+1;

    a1 = lambda_SN^2 + sigma^2/(1 - sigma)^2;    %alpha 1
    a2 = 2*sigma/(1 - sigma);                    %alpha 2
    a3 = 1;                                      %alpha 3

    %Solving for a4:
    a4_p = -a1*(-a2/(3*a3) + sqrt(a2^2 - 3*a1*a3)/(3*a3)) - ...
        a2*(-a2/(3*a3) + sqrt(a2^2 - 3*a1*a3)/(3*a3))^2 - ...
        a3*(-a2/(3*a3) + sqrt(a2^2 - 3*a1*a3)/(3*a3))^3;

    a4_m = -a1*(-a2/(3*a3) - sqrt(a2^2 - 3*a1*a3)/(3*a3)) - ...
        a2*(-a2/(3*a3) - sqrt(a2^2 - 3*a1*a3)/(3*a3))^2 - ...
        a3*(-a2/(3*a3) - sqrt(a2^2 - 3*a1*a3)/(3*a3))^3;

    %Solving for A from a4:
    A_p = sqrt(-a4_p*(1-sigma)^2); A_m = sqrt(-a4_m*(1-sigma)^2);

    lambda_SN_vec(k_SN) = lambda_SN;
    A_p_vec_SN(k_SN) = A_p;
    A_m_vec_SN(k_SN) = A_m;

    if lambda_SN == 0.001
        A12_SN = A_p_vec_SN(k_SN);
        A22_SN = A_m_vec_SN(k_SN);
    else
    end
end

%Truncating the Saddle-Node Plots:

m_SN = k_SN; g_SN = ones(1,k_SN); h_SN = ones(1,k_SN);

for n_SN = 1:k_SN
```

```
    if abs(A_p_vec_SN(n_SN)- A_m_vec_SN(n_SN)) < .0001
        g_SN(n_SN) = 0; h_SN(n_SN) = 0;
    elseif n_SN > 1 && g_SN(n_SN-1) == 0 && h_SN(n_SN-1) == 0
        m_SN = m_SN-1; g_SN(n_SN) = 0; h_SN(n_SN) = 0;
    else
    end
end

lambda_SN_vec_2 = zeros(1,m_SN);
A_p_vec_2_SN = zeros(1,m_SN); A_m_vec_2_SN = zeros(1,m_SN);

for q = 1:m_SN
    lambda_SN_vec_2(q) = lambda_SN_vec(q);
    A_p_vec_2_SN(q) = A_p_vec_SN(q);
    A_m_vec_2_SN(q) = A_m_vec_SN(q);
end

figure
hold on; grid on;
plot(lam_vec_2,A12_2,'k--',lam_vec_2,A22_2,'k--','LineWidth',2)
plot(lambda_SN_vec_2, A_p_vec_2_SN,'-k','LineWidth',2)
plot(lambda_SN_vec_2, A_m_vec_2_SN,'-k','LineWidth',2)
xlabel('\fontsize{12}\bf\lambda'); ylabel('\bfA');
axis([0 3.5 0 1.8]);


%**************************************************************************
**
%**************************************************************************
**
%**************************************************************************
**


%Generation of the Fourth Plot

clear lambda k k2 z1 z2

lambda = 0.2; k = 0; k5 = 0; k7 = 0;

for A = 0:0.001:2
    k = k+1;

    a1 = lambda^2 + sigma^2/(1 - sigma)^2;    %alpha 1
    a2 = 2*sigma/(1 - sigma);                 %alpha 2
    a3 = 1;                                   %alpha 3
    a4 = -A^2/(1-sigma)^2;                    %alpha 4

    %values for z, z1, and z2 were obtained by using solve command

    z1 = ((a1^3/(27*a3^3) + a4^2/(4*a3^2) + (a2^3*a4)/(27*a3^4) - ...
        (a1^2*a2^2)/(108*a3^4) - (a1*a2*a4)/(6*a3^3))^(1/2) - ...
        a4/(2*a3) - a2^3/(27*a3^3) + (a1*a2)/(6*a3^2))^(1/3) - ...
        a2/(3*a3) - (a1/(3*a3) - a2^2/(9*a3^2))/((a1^3/(27*a3^3) + ...
```

```
        a4^2/(4*a3^2) + (a2^3*a4)/(27*a3^4) - (a1^2*a2^2)/(108*a3^4) -
...
        (a1*a2*a4)/(6*a3^3))^(1/2) - a4/(2*a3) - a2^3/(27*a3^3) + ...
        (a1*a2)/(6*a3^2))^(1/3);

    z2 = (a1/(3*a3) - a2^2/(9*a3^2))/(2*((a1^3/(27*a3^3) + ...
        a4^2/(4*a3^2) + (a2^3*a4)/(27*a3^4) - (a1^2*a2^2)/(108*a3^4) -
...
        (a1*a2*a4)/(6*a3^3))^(1/2) - a4/(2*a3) - a2^3/(27*a3^3) + ...
        (a1*a2)/(6*a3^2))^(1/3)) - ((a1^3/(27*a3^3) + a4^2/(4*a3^2) +
...
        (a2^3*a4)/(27*a3^4) - (a1^2*a2^2)/(108*a3^4) - ...
        (a1*a2*a4)/(6*a3^3))^(1/2) - a4/(2*a3) - a2^3/(27*a3^3) + ...
        (a1*a2)/(6*a3^2))^(1/3)/2 - a2/(3*a3) - ...
        (3^(1/2)*1i*(((a1^3/(27*a3^3) + a4^2/(4*a3^2) + ...
        (a2^3*a4)/(27*a3^4) - (a1^2*a2^2)/(108*a3^4) - ...
        (a1*a2*a4)/(6*a3^3))^(1/2) - a4/(2*a3) - a2^3/(27*a3^3) + ...
        (a1*a2)/(6*a3^2))^(1/3) + (a1/(3*a3) - ...
        a2^2/(9*a3^2))/((a1^3/(27*a3^3) + a4^2/(4*a3^2) + ...
        (a2^3*a4)/(27*a3^4) - (a1^2*a2^2)/(108*a3^4) - ...
        (a1*a2*a4)/(6*a3^3))^(1/2) - a4/(2*a3) - a2^3/(27*a3^3) + ...
        (a1*a2)/(6*a3^2))^(1/3)))/2;

    z3 = (a1/(3*a3) - a2^2/(9*a3^2))/(2*((a1^3/(27*a3^3) + ...
        a4^2/(4*a3^2) + (a2^3*a4)/(27*a3^4) - (a1^2*a2^2)/(108*a3^4) -
...
        (a1*a2*a4)/(6*a3^3))^(1/2) - a4/(2*a3) - a2^3/(27*a3^3) + ...
        (a1*a2)/(6*a3^2))^(1/3)) - ((a1^3/(27*a3^3) + a4^2/(4*a3^2) +
...
        (a2^3*a4)/(27*a3^4) - (a1^2*a2^2)/(108*a3^4) - ...
        (a1*a2*a4)/(6*a3^3))^(1/2) - a4/(2*a3) - a2^3/(27*a3^3) + ...
        (a1*a2)/(6*a3^2))^(1/3)/2 - a2/(3*a3) + ...
        (3^(1/2)*1i*(((a1^3/(27*a3^3) + a4^2/(4*a3^2) + ...
        (a2^3*a4)/(27*a3^4) - (a1^2*a2^2)/(108*a3^4) - ...
        (a1*a2*a4)/(6*a3^3))^(1/2) - a4/(2*a3) - a2^3/(27*a3^3) + ...
        (a1*a2)/(6*a3^2))^(1/3) + (a1/(3*a3) - ...
        a2^2/(9*a3^2))/((a1^3/(27*a3^3) + a4^2/(4*a3^2) + ...
        (a2^3*a4)/(27*a3^4) - (a1^2*a2^2)/(108*a3^4) - ...
        (a1*a2*a4)/(6*a3^3))^(1/2) - a4/(2*a3) - a2^3/(27*a3^3) + ...
        (a1*a2)/(6*a3^2))^(1/3)))/2;

    %Since z = N20^2, the following solves for N20:

    N20_z1p(k) = sqrt(z1); N20_z1m(k) = -sqrt(z1);
    N20_z2p(k) = sqrt(z2); N20_z2m(k) = -sqrt(z2);
    N20_z3p(k) = sqrt(z3); N20_z3m(k) = -sqrt(z3);

    A_vec(k) = A;
end

%Truncating the response:

k2 = 0; k3 = 0; k4 = 0;
for n = 1:k
    if A_vec(n) > 0.095
```

```
        k2 = k2 + 1;
        N20_1(k2) = N20_z1p(n); A_vec_11(k2) = A_vec(n);
        N20_2a(k2) = N20_z2p(n); A_vec_21(k2) = A_vec(n);
    else
    end
end

for m = 1:k2
    if A_vec_11(m) < 1.14
        k3 = k3 + 1;
        N20_2(k3) = N20_2a(m); A_vec_21_2(k3) = A_vec_21(m);
    else
    end
end

for p = 1:k
    if A_vec(p) < 1.14
        k4 = k4 + 1;
        N20_3(k4) = N20_z3p(p); A_vec_31(k4) = A_vec(p);
    else
    end
end

%*************************************************************************
*

%Find unstable locations:

k_u1 = 0; k_check_1 = 0;
for ind1 = 1:length(N20_1)
    N20 = N20_1(ind1);
    z = N20^2;

    gamma1 = lambda*(e + 1);

    gamma2 = (4*z^2*e^2 + 8*z^2*e + 4*z^2 + 4*z*e^2*sigma - 4*z + ...
        e^2*lambda^2 - e^2*z^2 + e^2*sigma^2 + 2*e*lambda^2 - ...
        2*e*z^2 + lambda^2 - z^2 + 1)/4;

    gamma3 = (e*lambda*(e*sigma^2 + 1))/4;

    gamma4 = (e^2*(4*z^2*sigma^2 - 8*z^2*sigma + 4*z^2 - 4*z*sigma^2 +
...
        4*z*sigma + lambda^2*sigma^2 - 2*lambda^2*sigma + lambda^2 -
...
        z^2*sigma^2 + 2*z^2*sigma - z^2 + sigma^2))/16;

    %Characteristic equation
    CHAR1 = [1 gamma1 gamma2 gamma3 gamma4];

    R1 = roots(CHAR1);

    %Looking for positive real parts of the eigenvalues:
```

```
    for kR1 = 1:length(R1)
        if real(R1(kR1)) > 0
            k_u1 = k_u1 + 1;
            z_unstable_Hopf1(k_u1) = N20;
            A_unstable_Hopf1(k_u1) = A_vec_11(ind1);
            k_check_1 = k_check_1 + 1;
        else
        end
    end
end

k_u2 = 0; k_check_2 = 0;
for ind2 = 1:length(N20_2)
    N20 = N20_2(ind2);
    z = N20^2;

    gamma1 = lambda*(e + 1);

    gamma2 = (4*z^2*e^2 + 8*z^2*e + 4*z^2 + 4*z*e^2*sigma - 4*z + ...
        e^2*lambda^2 - e^2*z^2 + e^2*sigma^2 + 2*e*lambda^2 - ...
        2*e*z^2 + lambda^2 - z^2 + 1)/4;

    gamma3 = (e*lambda*(e*sigma^2 + 1))/4;

    gamma4 = (e^2*(4*z^2*sigma^2 - 8*z^2*sigma + 4*z^2 - 4*z*sigma^2 +
...
        4*z*sigma + lambda^2*sigma^2 - 2*lambda^2*sigma + lambda^2 -
...
        z^2*sigma^2 + 2*z^2*sigma - z^2 + sigma^2))/16;

    %Characteristic equation
    CHAR2 = [1 gamma1 gamma2 gamma3 gamma4];

    R2 = roots(CHAR2);

    %Looking for positive real parts of the eigenvalues

    for kR2 = 1:length(R2)
        if real(R2(kR2)) > 0
            k_u2 = k_u2 + 1;
            z_unstable_Hopf2(k_u2) = N20;
            A_unstable_Hopf2(k_u2) = A_vec_21_2(ind2);
            k_check_2 = k_check_2 + 1;
        else
        end
    end
end

k_u3 = 0; k_check_3 = 0;
for ind3 = 1:length(N20_3)
    N20 = N20_3(ind3);
    z = N20^2;

    gamma1 = lambda*(e + 1);
```

```
    gamma2 = (4*z^2*e^2 + 8*z^2*e + 4*z^2 + 4*z*e^2*sigma - 4*z + ...
        e^2*lambda^2 - e^2*z^2 + e^2*sigma^2 + 2*e*lambda^2 - ...
        2*e*z^2 + lambda^2 - z^2 + 1)/4;

    gamma3 = (e*lambda*(e*sigma^2 + 1))/4;

    gamma4 = (e^2*(4*z^2*sigma^2 - 8*z^2*sigma + 4*z^2 - 4*z*sigma^2 +
...
        4*z*sigma + lambda^2*sigma^2 - 2*lambda^2*sigma + lambda^2 -
...
        z^2*sigma^2 + 2*z^2*sigma - z^2 + sigma^2))/16;

    %Characteristic equation
    CHAR3 = [1 gamma1 gamma2 gamma3 gamma4];

    R3 = roots(CHAR3);

    %Looking for positive real parts of the eigenvalues

    for kR3 = 1:length(R3)
        if real(R3(kR3)) > 0
            k_u3 = k_u3 + 1;
            z_unstable_Hopf3(k_u3) = N20;
            A_unstable_Hopf3(k_u3) = A_vec_31(ind3);
            k_check_3 = k_check_3 + 1;
        else
        end
    end
end

%If k_check == 0, then that means A_unstable_Hopf and
%Z_unstable_Hopf are undefined, and thus the solution is stable.
%Here it is determined if that is the case, and a decision is made
%whether to plot based on the stability of the solution.

if k_check_1 == 0
    No_Plot1 = 0;
else
    No_Plot1 = 1;
end

if k_check_2 == 0
    No_Plot2 = 0;
else
    No_Plot2 = 1;
end

if k_check_3 == 0
    No_Plot3 = 0;
else
    No_Plot3 = 1;
end
```

```
%*********************************************************************

figure
plot(A_vec_11,N20_1,'-k',A_vec_21_2,N20_2,'-k',A_vec_31,N20_3,'-k', ...
    'LineWidth',2)
hold on; grid on;

if No_Plot1 == 1;
    plot(A_unstable_Hopf1, z_unstable_Hopf1, '*k');
else
end

if No_Plot2 == 1;
    plot(A_unstable_Hopf2, z_unstable_Hopf2, '*k');
else
end

if No_Plot3 == 1;
    plot(A_unstable_Hopf3, z_unstable_Hopf3, '*k');
else
end

xlabel('\bfA'); ylabel('\bfN_2_0');
axis([0 2 0 3.5]);


%*********************************************************************
**
%*********************************************************************
**
%*********************************************************************
**


%Generation of the fifth plot

clear all

lambda = 0.2; e = 0.05; A = 0.4; k = 0;
syms sigma

for sigma = -5:0.001:5
    k = k+1;

    v1 = -(27*e*lambda^2*(9*lambda^2 + 16)*(e*sigma + 1)^2*(e +
1)^2)/1024;

    v2 = -(9*e*lambda^2*(e*sigma - 1)*(e*sigma + 1)^2*(e + 1))/16;

    v3 = -(9*e*lambda^2*(e^2*sigma^2 - 1)^2)/64;

    a1 = lambda^2 + sigma^2/(1 - sigma)^2;    %alpha 1
    a2 = 2*sigma/(1 - sigma);                 %alpha 2
    a3 = 1;                                   %alpha 3
```

```
    a4 = -A^2/(1-sigma)^2;                    %alpha 4

    %values for z, z1, and z2 were obtained by using solve command

    z1 = ((a1^3/(27*a3^3) + a4^2/(4*a3^2) + (a2^3*a4)/(27*a3^4) - ...
        (a1^2*a2^2)/(108*a3^4) - (a1*a2*a4)/(6*a3^3))^(1/2) - ...
        a4/(2*a3) - a2^3/(27*a3^3) + (a1*a2)/(6*a3^2))^(1/3) - ...
        a2/(3*a3) - (a1/(3*a3) - a2^2/(9*a3^2))/((a1^3/(27*a3^3) + ...
        a4^2/(4*a3^2) + (a2^3*a4)/(27*a3^4) - (a1^2*a2^2)/(108*a3^4) -
...
        (a1*a2*a4)/(6*a3^3))^(1/2) - a4/(2*a3) - a2^3/(27*a3^3) + ...
        (a1*a2)/(6*a3^2))^(1/3);

    z2 = (a1/(3*a3) - a2^2/(9*a3^2))/(2*((a1^3/(27*a3^3) + ...
        a4^2/(4*a3^2) + (a2^3*a4)/(27*a3^4) - (a1^2*a2^2)/(108*a3^4) -
...
        (a1*a2*a4)/(6*a3^3))^(1/2) - a4/(2*a3) - a2^3/(27*a3^3) + ...
        (a1*a2)/(6*a3^2))^(1/3)) - ((a1^3/(27*a3^3) + a4^2/(4*a3^2) +
...
        (a2^3*a4)/(27*a3^4) - (a1^2*a2^2)/(108*a3^4) - ...
        (a1*a2*a4)/(6*a3^3))^(1/2) - a4/(2*a3) - a2^3/(27*a3^3) + ...
        (a1*a2)/(6*a3^2))^(1/3)/2 - a2/(3*a3) - ...
        (3^(1/2)*1i*(((a1^3/(27*a3^3) + a4^2/(4*a3^2) + ...
        (a2^3*a4)/(27*a3^4) - (a1^2*a2^2)/(108*a3^4) - ...
        (a1*a2*a4)/(6*a3^3))^(1/2) - a4/(2*a3) - a2^3/(27*a3^3) + ...
        (a1*a2)/(6*a3^2))^(1/3) + (a1/(3*a3) - ...
        a2^2/(9*a3^2))/((a1^3/(27*a3^3) + a4^2/(4*a3^2) + ...
        (a2^3*a4)/(27*a3^4) - (a1^2*a2^2)/(108*a3^4) - ...
        (a1*a2*a4)/(6*a3^3))^(1/2) - a4/(2*a3) - a2^3/(27*a3^3) + ...
        (a1*a2)/(6*a3^2))^(1/3)))/2;

    z3 = (a1/(3*a3) - a2^2/(9*a3^2))/(2*((a1^3/(27*a3^3) + ...
        a4^2/(4*a3^2) + (a2^3*a4)/(27*a3^4) - (a1^2*a2^2)/(108*a3^4) -
...
        (a1*a2*a4)/(6*a3^3))^(1/2) - a4/(2*a3) - a2^3/(27*a3^3) + ...
        (a1*a2)/(6*a3^2))^(1/3)) - ((a1^3/(27*a3^3) + a4^2/(4*a3^2) +
...
        (a2^3*a4)/(27*a3^4) - (a1^2*a2^2)/(108*a3^4) - ...
        (a1*a2*a4)/(6*a3^3))^(1/2) - a4/(2*a3) - a2^3/(27*a3^3) + ...
        (a1*a2)/(6*a3^2))^(1/3)/2 - a2/(3*a3) + ...
        (3^(1/2)*1i*(((a1^3/(27*a3^3) + a4^2/(4*a3^2) + ...
        (a2^3*a4)/(27*a3^4) - (a1^2*a2^2)/(108*a3^4) - ...
        (a1*a2*a4)/(6*a3^3))^(1/2) - a4/(2*a3) - a2^3/(27*a3^3) + ...
        (a1*a2)/(6*a3^2))^(1/3) + (a1/(3*a3) - ...
        a2^2/(9*a3^2))/((a1^3/(27*a3^3) + a4^2/(4*a3^2) + ...
        (a2^3*a4)/(27*a3^4) - (a1^2*a2^2)/(108*a3^4) - ...
        (a1*a2*a4)/(6*a3^3))^(1/2) - a4/(2*a3) - a2^3/(27*a3^3) + ...
        (a1*a2)/(6*a3^2))^(1/3)))/2;

    N20_z11(k) = sqrt(z1); N20_z12(k) = -sqrt(z1);
    N20_z21(k) = sqrt(z2); N20_z22(k) = -sqrt(z2);
    N20_z31(k) = sqrt(z3); N20_z32(k) = -sqrt(z3);

    sigma_vec(k) = sigma;
end
```

```
%Truncating the response:

k2 = 0; k3 = 0; k4 = 0; k5 = 0; k6 = 0; k7 = 0;
for n = 1:k
    if N20_z11(n) < 3.1
        k2 = k2 + 1;
        z11_2(k2) = N20_z11(n); sigma_vec_11(k2) = sigma_vec(n);
        z21_2(k2) = N20_z21(n); sigma_vec_21(k2) = sigma_vec(n);
    else
    end
end

for m = 1:k2
    if sigma_vec_21(m) < -1.21 || sigma_vec_21(m) > 1
        k3 = k3 + 1;
        z21_2_2(k3) = z21_2(m); sigma_vec_21_2(k3) = sigma_vec_21(m);
    else
    end
end

for q = 1:k3
    if sigma_vec_21_2(q) < 2.57 && sigma_vec_21_2(q) > -1.6
        k6 = k6 + 1;
        z21_2_3(k6) = z21_2_2(q); sigma_vec_21_3(k6) =
sigma_vec_21_2(q);
    else
    end
end

for h = 1:k2
    if sigma_vec_11(h) > -1.6 && sigma_vec_11(h) < 2.57
        k5 = k5 + 1;
        z11_2_2(k5) = z11_2(h); sigma_vec_11_2(k5) = sigma_vec_11(h);
    else
    end
end

for p = 1:k
    if sigma_vec(p) < -1.21
        k4 = k4 + 1;
        z31_2(k4) = N20_z31(p); sigma_vec_31(k4) = sigma_vec(p);
    elseif sigma_vec(p) > 1
        k4 = k4 + 1;
        z31_2(k4) = N20_z31(p); sigma_vec_31(k4) = sigma_vec(p);
    else
    end
end

%Since the response jumps from one location to another at some
locations,
%this part of the program separates those jumps to discrete branches to
be
%used later for determining stability:
```

```
step_z11_1 = 0; step_z11_2 = 0; plot_thresh = 0.01; test_val = 0;
for step1 = 2:length(z11_2_2)
    if abs(z11_2_2(step1) - z11_2_2(step1 - 1)) > plot_thresh && ...
            abs(sigma_vec_11_2(step1) - sigma_vec_11_2(step1 - 1)) ...
            > plot_thresh
        test_val = 2;
    else
    end

    if test_val < 1
        step_z11_1 = step_z11_1 + 1;
        z11_final1(step_z11_1) = z11_2_2(step1);
        sigma_z11_final1(step_z11_1) = sigma_vec_11_2(step1);
    else
        step_z11_2 = step_z11_2 + 1;
        z11_final2(step_z11_2) = z11_2_2(step1);
        sigma_z11_final2(step_z11_2) = sigma_vec_11_2(step1);
    end
end

step_z21_1 = 0; step_z21_2 = 0; plot_thresh = 0.01; test_val = 0;
for step2 = 2:length(z21_2_3)
    if abs(z21_2_3(step2) - z21_2_3(step2 - 1)) > plot_thresh && ...
            abs(sigma_vec_21_3(step2) - sigma_vec_21_3(step2 - 1)) ...
            > plot_thresh
        test_val = 2;
    else
    end

    if test_val < 1
        step_z21_1 = step_z21_1 + 1;
        z21_final1(step_z21_1) = z21_2_3(step2);
        sigma_z21_final1(step_z21_1) = sigma_vec_21_3(step2);
    else
        step_z21_2 = step_z21_2 + 1;
        z21_final2(step_z21_2) = z21_2_3(step2);
        sigma_z21_final2(step_z21_2) = sigma_vec_21_3(step2);
    end
end

step_z31_1 = 0; step_z31_2 = 0; plot_thresh = 0.01; test_val = 0;
for step3 = 2:length(z31_2)
    if abs(z31_2(step3) - z31_2(step3 - 1)) > plot_thresh && ...
            abs(sigma_vec_31(step3) - sigma_vec_31(step3 - 1)) ...
            > plot_thresh
        test_val = 2;
    else
    end

    if test_val < 1
        step_z31_1 = step_z31_1 + 1;
        z31_final1(step_z31_1) = z31_2(step3);
        sigma_z31_final1(step_z31_1) = sigma_vec_31(step3);
    else
        step_z31_2 = step_z31_2 + 1;
        z31_final2(step_z31_2) = z31_2(step3);
```

```
        sigma_z31_final2(step_z31_2) = sigma_vec_31(step3);
    end
end

figure
hold on; grid on;
plot(sigma_z11_final1,z11_final1,'-k','LineWidth',2)
plot(sigma_z11_final2,z11_final2,'-k','LineWidth',2)
plot(sigma_z21_final1,z21_final1,'-k','LineWidth',2)
plot(sigma_z21_final2,z21_final2,'-k','LineWidth',2)
plot(sigma_z31_final1,z31_final1,'-k','LineWidth',2)
plot(sigma_z31_final2,z31_final2,'-k','LineWidth',2)
xlabel('\fontsize{12}\bf\sigma'); ylabel('\bfN_2_0');

%**********************************************************************
*

%Find unstable locations:

k_u1 = 0; k_check_1 = 0;
for ind1 = 1:length(z11_final1)
    N20 = z11_final1(ind1);
    z = N20^2;

    gamma1 = lambda*(e + 1);

    gamma2 = (4*z^2*e^2 + 8*z^2*e + 4*z^2 + 4*z*e^2*sigma - 4*z + ...
        e^2*lambda^2 - e^2*z^2 + e^2*sigma^2 + 2*e*lambda^2 - ...
        2*e*z^2 + lambda^2 - z^2 + 1)/4;

    gamma3 = (e*lambda*(e*sigma^2 + 1))/4;

    gamma4 = (e^2*(4*z^2*sigma^2 - 8*z^2*sigma + 4*z^2 - 4*z*sigma^2 +
...
        4*z*sigma + lambda^2*sigma^2 - 2*lambda^2*sigma + lambda^2 -
...
        z^2*sigma^2 + 2*z^2*sigma - z^2 + sigma^2))/16;

    %Characteristic equation
    CHAR1 = [1 gamma1 gamma2 gamma3 gamma4];

    R1 = roots(CHAR1);

    %Looking for positive real parts of the eigenvalues:

    for kR1 = 1:length(R1)
        if real(R1(kR1)) > 0
            k_u1 = k_u1 + 1;
            z_unstable_Hopf1(k_u1) = N20;
            sigma_unstable_Hopf1(k_u1) = sigma_z11_final1(ind1);
            k_check_1 = k_check_1 + 1;
        else
        end
    end
```

```
end

k_u2 = 0; k_check_2 = 0;
for ind2 = 1:length(z11_final2)
    N20 = z11_final2(ind2);
    z = N20^2;

    gamma1 = lambda*(e + 1);

    gamma2 = (4*z^2*e^2 + 8*z^2*e + 4*z^2 + 4*z*e^2*sigma - 4*z + ...
        e^2*lambda^2 - e^2*z^2 + e^2*sigma^2 + 2*e*lambda^2 - ...
        2*e*z^2 + lambda^2 - z^2 + 1)/4;

    gamma3 = (e*lambda*(e*sigma^2 + 1))/4;

    gamma4 = (e^2*(4*z^2*sigma^2 - 8*z^2*sigma + 4*z^2 - 4*z*sigma^2 +
...
        4*z*sigma + lambda^2*sigma^2 - 2*lambda^2*sigma + lambda^2 -
...
        z^2*sigma^2 + 2*z^2*sigma - z^2 + sigma^2))/16;

    %Characteristic equation
    CHAR2 = [1 gamma1 gamma2 gamma3 gamma4];

    R2 = roots(CHAR2);

    %Looking for positive real parts of the eigenvalues

    for kR2 = 1:length(R2)
        if real(R2(kR2)) > 0
            k_u2 = k_u2 + 1;
            z_unstable_Hopf2(k_u2) = N20;
            sigma_unstable_Hopf2(k_u2) = sigma_z11_final2(ind2);
            k_check_2 = k_check_2 + 1;
        else
        end
    end
end

k_u3 = 0; k_check_3 = 0;
for ind3 = 1:length(z21_final1)
    N20 = z21_final1(ind3);
    z = N20^2;

    gamma1 = lambda*(e + 1);

    gamma2 = (4*z^2*e^2 + 8*z^2*e + 4*z^2 + 4*z*e^2*sigma - 4*z + ...
        e^2*lambda^2 - e^2*z^2 + e^2*sigma^2 + 2*e*lambda^2 - ...
        2*e*z^2 + lambda^2 - z^2 + 1)/4;

    gamma3 = (e*lambda*(e*sigma^2 + 1))/4;

    gamma4 = (e^2*(4*z^2*sigma^2 - 8*z^2*sigma + 4*z^2 - 4*z*sigma^2 +
...
```

```
        4*z*sigma + lambda^2*sigma^2 - 2*lambda^2*sigma + lambda^2 -
...
        z^2*sigma^2 + 2*z^2*sigma - z^2 + sigma^2))/16;

    %Characteristic equation
    CHAR3 = [1 gamma1 gamma2 gamma3 gamma4];

    R3 = roots(CHAR3);

    %Looking for positive real parts of the eigenvalues

    for kR3 = 1:length(R3)
        if real(R3(kR3)) > 0
            k_u3 = k_u3 + 1;
            z_unstable_Hopf3(k_u3) = N20;
            sigma_unstable_Hopf3(k_u3) = sigma_z21_final1(ind3);
            k_check_3 = k_check_3 + 1;
        else
        end
    end
end

k_u4 = 0; k_check_4 = 0;
for ind4 = 1:length(z21_final2)
    N20 = z21_final2(ind4);
    z = N20^2;

    gamma1 = lambda*(e + 1);

    gamma2 = (4*z^2*e^2 + 8*z^2*e + 4*z^2 + 4*z*e^2*sigma - 4*z + ...
        e^2*lambda^2 - e^2*z^2 + e^2*sigma^2 + 2*e*lambda^2 - ...
        2*e*z^2 + lambda^2 - z^2 + 1)/4;

    gamma3 = (e*lambda*(e*sigma^2 + 1))/4;

    gamma4 = (e^2*(4*z^2*sigma^2 - 8*z^2*sigma + 4*z^2 - 4*z*sigma^2 +
...
        4*z*sigma + lambda^2*sigma^2 - 2*lambda^2*sigma + lambda^2 -
...
        z^2*sigma^2 + 2*z^2*sigma - z^2 + sigma^2))/16;

    %Characteristic equation
    CHAR4 = [1 gamma1 gamma2 gamma3 gamma4];

    R4 = roots(CHAR4);

    %Looking for positive real parts of the eigenvalues

    for kR4 = 1:length(R4)
        if real(R4(kR4)) > 0
            k_u4 = k_u4 + 1;
            z_unstable_Hopf4(k_u4) = N20;
            sigma_unstable_Hopf4(k_u4) = sigma_z21_final2(ind4);
            k_check_4 = k_check_4 + 1;
```

```
        else
        end
    end
end

k_u5 = 0; k_check_5 = 0;
for ind5 = 1:length(z31_final1)
    N20 = z31_final1(ind5);
    z = N20^2;

    gamma1 = lambda*(e + 1);

    gamma2 = (4*z^2*e^2 + 8*z^2*e + 4*z^2 + 4*z*e^2*sigma - 4*z + ...
        e^2*lambda^2 - e^2*z^2 + e^2*sigma^2 + 2*e*lambda^2 - ...
        2*e*z^2 + lambda^2 - z^2 + 1)/4;

    gamma3 = (e*lambda*(e*sigma^2 + 1))/4;

    gamma4 = (e^2*(4*z^2*sigma^2 - 8*z^2*sigma + 4*z^2 - 4*z*sigma^2 +
...
        4*z*sigma + lambda^2*sigma^2 - 2*lambda^2*sigma + lambda^2 -
...
        z^2*sigma^2 + 2*z^2*sigma - z^2 + sigma^2))/16;

    %Characteristic equation
    CHAR5 = [1 gamma1 gamma2 gamma3 gamma4];

    R5 = roots(CHAR5);

    %Looking for positive real parts of the eigenvalues

    for kR5 = 1:length(R5)
        if real(R5(kR5)) > 0
            k_u5 = k_u5 + 1;
            z_unstable_Hopf5(k_u5) = N20;
            sigma_unstable_Hopf5(k_u5) = sigma_z31_final1(ind5);
            k_check_5 = k_check_5 + 1;
        else
        end
    end
end

%***
k_u6 = 0; k_check_6 = 0;
for ind6 = 1:length(z31_final2)
    N20 = z31_final2(ind6);
    z = N20^2;

    gamma1 = lambda*(e + 1);

    gamma2 = (4*z^2*e^2 + 8*z^2*e + 4*z^2 + 4*z*e^2*sigma - 4*z + ...
        e^2*lambda^2 - e^2*z^2 + e^2*sigma^2 + 2*e*lambda^2 - ...
        2*e*z^2 + lambda^2 - z^2 + 1)/4;
```

```
    gamma3 = (e*lambda*(e*sigma^2 + 1))/4;

    gamma4 = (e^2*(4*z^2*sigma^2 - 8*z^2*sigma + 4*z^2 - 4*z*sigma^2 +
...
        4*z*sigma + lambda^2*sigma^2 - 2*lambda^2*sigma + lambda^2 -
...
        z^2*sigma^2 + 2*z^2*sigma - z^2 + sigma^2))/16;

    %Characteristic equation
    CHAR6 = [1 gamma1 gamma2 gamma3 gamma4];

    R6 = roots(CHAR6);

    %Looking for positive real parts of the eigenvalues

    for kR6 = 1:length(R6)
        if real(R6(kR6)) > 0
            k_u6 = k_u6 + 1;
            z_unstable_Hopf6(k_u6) = N20;
            sigma_unstable_Hopf6(k_u6) = sigma_z31_final2(ind6);
            k_check_6 = k_check_6 + 1;
        else
        end
    end
end

%If k_check == 0, then that means A_unstable_Hopf and
%Z_unstable_Hopf are undefined, and thus the solution is stable.
%Here it is determined if that is the case, and a decision is made
%whether to plot based on the stability of the solution.

if k_check_1 == 0
    No_Plot1 = 0;
else
    No_Plot1 = 1;
end

if k_check_2 == 0
    No_Plot2 = 0;
else
    No_Plot2 = 1;
end

if k_check_3 == 0
    No_Plot3 = 0;
else
    No_Plot3 = 1;
end

if k_check_4 == 0
    No_Plot4 = 0;
else
    No_Plot4 = 1;
end
```

```
if k_check_5 == 0
    No_Plot5 = 0;
else
    No_Plot5 = 1;
end

if k_check_6 == 0
    No_Plot6 = 0;
else
    No_Plot6 = 1;
end

if No_Plot1 == 1;
    plot(sigma_unstable_Hopf1, z_unstable_Hopf1, '*k');
else
end

if No_Plot2 == 1;
    plot(sigma_unstable_Hopf2, z_unstable_Hopf2, '*k');
else
end

if No_Plot3 == 1;
    plot(sigma_unstable_Hopf3, z_unstable_Hopf3, '*k');
else
end

if No_Plot4 == 1;
    plot(sigma_unstable_Hopf4, z_unstable_Hopf4, '*k');
else
end

if No_Plot5 == 1;
    plot(sigma_unstable_Hopf5, z_unstable_Hopf5, '*k');
else
end

if No_Plot6 == 1;
    plot(sigma_unstable_Hopf6, z_unstable_Hopf6, '*k');
else
end

axis([-5 5 0 3.5]);
```

3.A.7.11 Hopf Bifurcation Analysis - Solving for Coefficients, γ_i (Nonlinear Damping)

```
%Calculation of the coefficients of the characteristic polynomial
%(nonlinear damping case)

close all
clear all
```

```
syms del1 del1_conj del2 del2_conj lambda e sigma N20 phi20 phi20_conj
mu

%From equations (3.55):

del1_dot = -1i*e/(2*(1+e))*(del1-del2) + ...
    1i*e*sigma*(del1+e*del2)/(2*(1+e));

del1_conj_dot = 1i*e/(2*(1+e))*(del1_conj-del2_conj) - ...
    1i*e*sigma*(del1_conj+e*del2_conj)/(2*(1+e));

del2_dot = -1i/(2*(1+e))*(del2-del1) + ...
    1i*e*sigma/(2*(1+e))*(del1+e*del2) - ...
    3*lambda*(1+e)/4*N20^2*del2 + 1i*(1+e)*N20^2*del2 - ...
    3*lambda*(1+e)/8*phi20^2*del2_conj + 1i*(1+e)/2*phi20^2*del2_conj;

del2_conj_dot = 1i/(2*(1+e))*(del2_conj-del1_conj) - ...
    1i*e*sigma/(2*(1+e))*(del1_conj+e*del2_conj) - ...
    3*lambda*(1+e)/4*N20^2*del2_conj - 1i*(1+e)*N20^2*del2_conj - ...
    3*lambda*(1+e)/8*phi20_conj^2*del2 - 1i*(1+e)/2*phi20_conj^2*del2;

f = [del1_dot; del1_conj_dot; del2_dot; del2_conj_dot];
v = [del1 del1_conj del2 del2_conj];
DxF = jacobian(f,v);
CHAR = DxF - mu*eye(4);
C_poly = det(CHAR);
C_coeffs = coeffs(C_poly, mu);

gamma1 = simple(collect(C_coeffs(4), N20))
gamma2 = simple(collect(C_coeffs(3), N20))
gamma3 = simple(collect(C_coeffs(2), N20))
gamma4 = simple(collect(C_coeffs(1), N20))
```

3.A.7.12 Hopf Bifurcation Analysis - Simplification of Coefficients, γ_i Based on the MATLAB Output (Nonlinear Damping)

The MATLAB output for the coefficients of the characteristic polynomial is as follows:

gamma1 =

(3*N20^2*lambda*(e + 1))/2

gamma2 =

N20^2*(e^2*sigma - 1) + N20^4*((9*e^2*lambda^2)/16 + e^2 + (9*e*lambda^2)/8 + 2*e + (9*lambda^2)/16 + 1) + (e^2*sigma^2)/4 - (phi20^2*phi20_conj^2)/4 - (e*phi20^2*phi20_conj^2)/2 - (e^2*phi20^2*phi20_conj^2)/4 - (9*lambda^2*phi20^2*phi20_conj^2)/64 - (9*e*lambda^2*phi20^2*phi20_conj^2)/32 - (9*e^2*lambda^2*phi20^2*phi20_conj^2)/64 + 1/4

gamma3 =

(3*N20^2*e*lambda*(e*sigma^2 + 1))/8

gamma4 =

(e^2*(36*N20^4*lambda^2*sigma^2 - 72*N20^4*lambda^2*sigma + 36*N20^4*lambda^2 + 64*N20^4*sigma^2 - 128*N20^4*sigma + 64*N20^4 - 64*N20^2*sigma^2 + 64*N20^2*sigma - 9*lambda^2*phi20^2*phi20_conj^2*sigma^2 + 18*lambda^2*phi20^2*phi20_conj^2*sigma - 9*lambda^2*phi20^2*phi20_conj^2 - 16*phi20^2*phi20_conj^2*sigma^2 + 32*phi20^2*phi20_conj^2*sigma - 16*phi20^2*phi20_conj^2 + 16*sigma^2))/256

Rewriting this output into a more reader-friendly form yields

$$\gamma_1 = \frac{3\lambda}{2}(1+\varepsilon){N_{20}}^2,$$

$$\gamma_2 = (\varepsilon^2\sigma - 1){N_{20}}^2 + \left(\frac{9}{16}\varepsilon^2\lambda^2 + \varepsilon^2 + \frac{9}{8}\varepsilon\lambda^2 + 2\varepsilon + \frac{9}{16}\lambda^2 + 1\right){N_{20}}^4 + \frac{1}{4}\varepsilon^2\sigma^2 - \frac{1}{4}{\varphi_{20}}^2{\varphi_{20}^*}^2 - \frac{1}{2}\varepsilon{\varphi_{20}}^2{\varphi_{20}^*}^2 - \frac{1}{4}\varepsilon^2{\varphi_{20}}^2{\varphi_{20}^*}^2 - \frac{9}{64}\lambda^2{\varphi_{20}}^2{\varphi_{20}^*}^2 - \frac{9}{32}\varepsilon\lambda^2{\varphi_{20}}^2{\varphi_{20}^*}^2 - \frac{9}{64}\varepsilon^2\lambda^2{\varphi_{20}}^2{\varphi_{20}^*}^2 + \frac{1}{4},$$

$$\gamma_3 = \frac{3\varepsilon\lambda}{8}(1+\varepsilon\sigma^2){N_{20}}^2,$$

$$\gamma_4 = \frac{\varepsilon^2}{256}\left(36\lambda^2\sigma^2{N_{20}}^4 - 72\lambda^2\sigma{N_{20}}^4 + 36\lambda^2{N_{20}}^4 + 64\sigma^2{N_{20}}^4 - 128\sigma{N_{20}}^4 + 64{N_{20}}^4 - 64\sigma^2{N_{20}}^2 + 64\sigma{N_{20}}^2 - 9\lambda^2\sigma^2{\varphi_{20}}^2{\varphi_{20}^*}^2 + 18\lambda^2\sigma{\varphi_{20}}^2{\varphi_{20}^*}^2 - 9\lambda^2{\varphi_{20}}^2{\varphi_{20}^*}^2 - 16\sigma^2{\varphi_{20}}^2{\varphi_{20}^*}^2 + 32\sigma{\varphi_{20}}^2{\varphi_{20}^*}^2 - 16{\varphi_{20}}^2{\varphi_{20}^*}^2 + 16\sigma^2\right).$$

Noting that

$${\varphi_{20}}^2{\varphi_{20}^*}^2 = {N_{20}}^4,$$

as shown in Section 3.A.6.2, γ_2 and γ_4 can be reduced as follows

$$\gamma_2 = (\varepsilon^2\sigma - 1){N_{20}}^2 + \left(\frac{9}{16}\varepsilon^2\lambda^2 + \varepsilon^2 + \frac{9}{8}\varepsilon\lambda^2 + 2\varepsilon + \frac{9}{16}\lambda^2 + 1\right){N_{20}}^4 + \frac{1}{4}\varepsilon^2\sigma^2 - \frac{1}{4}{N_{20}}^4 - \frac{1}{2}\varepsilon{N_{20}}^4 - \frac{1}{4}\varepsilon^2{N_{20}}^4 - \frac{9}{64}\lambda^2{N_{20}}^4 - \frac{9}{32}\varepsilon\lambda^2{N_{20}}^4 - \frac{9}{64}\varepsilon^2\lambda^2{N_{20}}^4 + \frac{1}{4}$$

$$\gamma_4 = \frac{\varepsilon^2}{256}\left(36\lambda^2\sigma^2{N_{20}}^4 - 72\lambda^2\sigma{N_{20}}^4 + 36\lambda^2{N_{20}}^4 + 64\sigma^2{N_{20}}^4 - 128\sigma{N_{20}}^4 + 64{N_{20}}^4 - 64\sigma^2{N_{20}}^2 + 64\sigma{N_{20}}^2 - 9\lambda^2\sigma^2{N_{20}}^4 + 18\lambda^2\sigma{N_{20}}^4 - 9\lambda^2{N_{20}}^4 - 16\sigma^2{N_{20}}^4 + 32\sigma{N_{20}}^4 - 16{N_{20}}^4 + 16\sigma^2\right).$$

Finally, γ_2 and γ_4 can be reduced further to

$$\gamma_2 = \left(\frac{27\varepsilon^2\lambda^2}{64} + \frac{27\varepsilon\lambda^2}{32} + \frac{27\lambda^2}{64} + \frac{3\varepsilon^2}{4} + \frac{3\varepsilon}{2} + \frac{3}{4}\right){N_{20}}^4 + (\varepsilon^2\sigma - 1){N_{20}}^2 + \frac{1}{4}(\varepsilon^2\sigma^2 + 1),$$

$$\gamma_4 = \left(\frac{27\varepsilon^2\lambda^2\sigma^2}{256} - \frac{27\varepsilon^2\lambda^2\sigma}{128} + \frac{27\varepsilon^2\lambda^2}{256} + \frac{3\varepsilon^2\sigma^2}{16} - \frac{3\varepsilon^2\sigma}{8} + \frac{3\varepsilon^2}{16}\right){N_{20}}^4 + \frac{1}{4}\varepsilon^2\sigma(1-\sigma){N_{20}}^2 + \frac{\varepsilon^2\sigma^2}{16}.$$

3.A.7.13 Hopf Bifurcation Analysis – Solving for Coefficients, v_i (Nonlinear Damping)

```
%Calculation of the coefficients of v1*z^2 + v2*z + v3 = 0
%(nonlinear damping case)

close all
clear all

syms lambda e sigma z

gamma1 = (3*lambda/2)*(1 + e)*z;

gamma2 = (27*e^2*lambda^2/64 + 27*e*lambda^2/32 + 27*lambda^2/64 + ...
    3*e^2/4 + 3*e/2 + 3/4)*z^2 + (e^2*sigma - 1)*z + ...
    (1/4)*(e^2*sigma^2 + 1);

gamma3 = (3*e*lambda/8)*(1 + e*sigma^2)*z;

gamma4 = (27*e^2*lambda^2*sigma^2/256 - 27*e^2*lambda^2*sigma/128 + ...
    27*e^2*lambda^2/256 + 3*e^2*sigma^2/16 - 3*e^2*sigma/8 + ...
    3*e^2/16)*z^2 + (1/4)*e^2*sigma*(1-sigma)*z + e^2*sigma^2/16;
```

```
V = collect(gamma3^2 - gamma2*gamma3*gamma1 + gamma4*gamma1^2, z);
C = coeffs(V,z);

v1 = simple(C(3))
v2 = simple(C(2))
v3 = simple(C(1))
```

3.A.7.14 Hopf Bifurcation Analysis - Simplification of Coefficients v_i Based on the MATLAB Output (Nonlinear Damping)

The MATLAB output for the coefficients of Eq. (6.2-20) is as follows:

v1 =

-(27*e*lambda^2*(9*lambda^2 + 16)*(e*sigma + 1)^2*(e + 1)^2)/1024

v2 =

-(9*e*lambda^2*(e*sigma - 1)*(e*sigma + 1)^2*(e + 1))/16

v3 =

-(9*e*lambda^2*(e^2*sigma^2 - 1)^2)/64

Rewriting this output into a more reader-friendly form yields

$$v_1 = -\frac{27}{1024}\varepsilon\lambda^2(9\lambda^2 + 16)(\varepsilon\sigma + 1)^2(\varepsilon + 1)^2$$

$$v_2 = -\frac{9}{16}\varepsilon\lambda^2(\varepsilon\sigma - 1)(\varepsilon\sigma + 1)^2(\varepsilon + 1)$$

$$v_3 = -\frac{9}{64}\varepsilon\lambda^2(\varepsilon^2\sigma^2 - 1)^2$$

3.A.7.15 Hopf Bifurcation Plots (Nonlinear Damping)

```
%Hopf Bifurcation (nonlinear damping)

%Generation of the First Plot

close all
clear all
```

```
sigma = 0.5; e = 0.05; k = 0;

for lambda = 0:0.001:2
    k = k+1;

    v1 = -(27*e*lambda^2*(9*lambda^2 + 16)*(e*sigma + 1)^2*(e +
1)^2)/1024;

    v2 = -(9*e*lambda^2*(e*sigma - 1)*(e*sigma + 1)^2*(e + 1))/16;

    v3 = -(9*e*lambda^2*(e^2*sigma^2 - 1)^2)/64;

    a1 = sigma^2/(1 - sigma)^2;    %alpha 1
    a2 = 2*sigma/(1 - sigma);      %alpha 2
    a3 = 9/16*lambda^2 + 1;        %alpha 3

    z1 = (-v2 - sqrt(v2^2 - 4*v3*v1))/(2*v1); %Boundary of stability
    z2 = (-v2 + sqrt(v2^2 - 4*v3*v1))/(2*v1); %Boundary of stability

    A1p(k) = (1-sigma)*sqrt(a1*z1 + a2*z1^2 + a3*z1^3);  %corresponds
to z1
    A1m(k) = -(1-sigma)*sqrt(a1*z1 + a2*z1^2 + a3*z1^3); %corresponds
to z1
    A2p(k) = (1-sigma)*sqrt(a1*z2 + a2*z2^2 + a3*z2^3);  %corresponds
to z2
    A2m(k) = -(1-sigma)*sqrt(a1*z2 + a2*z2^2 + a3*z2^3); %corresponds
to z2

    lam_vec(k) = lambda;
end

%Truncating the plot:

k2 = 0;
for m = 1:k
    if lam_vec(m) < 0.772
        k2 = k2 + 1;
        A1p_2(k2) = A1p(m); lam_vec_2(k2) = lam_vec(m);
        A2p_2(k2) = A2p(m); lam_vec_2(k2) = lam_vec(m);
    else
    end
end

figure
plot(lam_vec_2,A1p_2,'k',lam_vec_2,A2p_2,'-k','LineWidth',2)
xlabel('\fontsize{12}\bf\lambda'); ylabel('\bfA'); grid on;
axis([0 0.8 0.2 1]);


%**************************************************************************
**
%**************************************************************************
**
```

```
%*************************************************************************
**


%Generation of Second Plot

%The next two lines that are commented out were used to find z.
%syms z
%z_solved = solve('a1*z + a2*z^2 + a3*z^3 +a4 = 0', z)

clear lambda k k2 z1 z2

lambda = 0.2; k = 0; k2 = 0;

for A = 0.2:0.01:2
    k = k+1;

    a1 = sigma^2/(1 - sigma)^2;  %alpha 1
    a2 = 2*sigma/(1 - sigma);    %alpha 2
    a3 = 9/16*lambda^2 + 1;     %alpha 3
    a4 = -A^2/(1-sigma)^2;      %alpha 4

    %values for z, z1, and z2 were obtained by using solve command

    z1 = ((a1^3/(27*a3^3) + a4^2/(4*a3^2) + (a2^3*a4)/(27*a3^4) - ...
        (a1^2*a2^2)/(108*a3^4) - (a1*a2*a4)/(6*a3^3))^(1/2) - ...
        a4/(2*a3) - a2^3/(27*a3^3) + (a1*a2)/(6*a3^2))^(1/3) - ...
        a2/(3*a3) - (a1/(3*a3) - a2^2/(9*a3^2))/((a1^3/(27*a3^3) + ...
        a4^2/(4*a3^2) + (a2^3*a4)/(27*a3^4) - (a1^2*a2^2)/(108*a3^4) -
...
        (a1*a2*a4)/(6*a3^3))^(1/2) - a4/(2*a3) - a2^3/(27*a3^3) + ...
        (a1*a2)/(6*a3^2))^(1/3);

    z2 = (a1/(3*a3) - a2^2/(9*a3^2))/(2*((a1^3/(27*a3^3) + ...
        a4^2/(4*a3^2) + (a2^3*a4)/(27*a3^4) - (a1^2*a2^2)/(108*a3^4) -
...
        (a1*a2*a4)/(6*a3^3))^(1/2) - a4/(2*a3) - a2^3/(27*a3^3) + ...
        (a1*a2)/(6*a3^2))^(1/3)) - ((a1^3/(27*a3^3) + a4^2/(4*a3^2) +
...
        (a2^3*a4)/(27*a3^4) - (a1^2*a2^2)/(108*a3^4) - ...
        (a1*a2*a4)/(6*a3^3))^(1/2) - a4/(2*a3) - a2^3/(27*a3^3) + ...
        (a1*a2)/(6*a3^2))^(1/3)/2 - a2/(3*a3) - ...
        (3^(1/2)*1i*(((a1^3/(27*a3^3) + a4^2/(4*a3^2) + ...
        (a2^3*a4)/(27*a3^4) - (a1^2*a2^2)/(108*a3^4) - ...
        (a1*a2*a4)/(6*a3^3))^(1/2) - a4/(2*a3) - a2^3/(27*a3^3) + ...
        (a1*a2)/(6*a3^2))^(1/3) + (a1/(3*a3) - ...
        a2^2/(9*a3^2))/((a1^3/(27*a3^3) + a4^2/(4*a3^2) + ...
        (a2^3*a4)/(27*a3^4) - (a1^2*a2^2)/(108*a3^4) - ...
        (a1*a2*a4)/(6*a3^3))^(1/2) - a4/(2*a3) - a2^3/(27*a3^3) + ...
        (a1*a2)/(6*a3^2))^(1/3)))/2;

    z3 = (a1/(3*a3) - a2^2/(9*a3^2))/(2*((a1^3/(27*a3^3) + ...
        a4^2/(4*a3^2) + (a2^3*a4)/(27*a3^4) - (a1^2*a2^2)/(108*a3^4) -
...
        (a1*a2*a4)/(6*a3^3))^(1/2) - a4/(2*a3) - a2^3/(27*a3^3) + ...
```

```
        (a1*a2)/(6*a3^2))^(1/3)) - ((a1^3/(27*a3^3) + a4^2/(4*a3^2) +
...
        (a2^3*a4)/(27*a3^4) - (a1^2*a2^2)/(108*a3^4) - ...
        (a1*a2*a4)/(6*a3^3))^(1/2) - a4/(2*a3) - a2^3/(27*a3^3) + ...
        (a1*a2)/(6*a3^2))^(1/3)/2 - a2/(3*a3) + ...
        (3^(1/2)*1i*(((a1^3/(27*a3^3) + a4^2/(4*a3^2) + ...
        (a2^3*a4)/(27*a3^4) - (a1^2*a2^2)/(108*a3^4) - ...
        (a1*a2*a4)/(6*a3^3))^(1/2) - a4/(2*a3) - a2^3/(27*a3^3) + ...
        (a1*a2)/(6*a3^2))^(1/3) + (a1/(3*a3) - ...
        a2^2/(9*a3^2))/((a1^3/(27*a3^3) + a4^2/(4*a3^2) + ...
        (a2^3*a4)/(27*a3^4) - (a1^2*a2^2)/(108*a3^4) - ...
        (a1*a2*a4)/(6*a3^3))^(1/2) - a4/(2*a3) - a2^3/(27*a3^3) + ...
        (a1*a2)/(6*a3^2))^(1/3)))/2;

    %Since z = N20^2, the following solves for N20:

    N20_z1p(k) = sqrt(z1); N20_z1m(k) = -sqrt(z1);
    N20_z2p(k) = sqrt(z2); N20_z2m(k) = -sqrt(z2);
    N20_z3p(k) = sqrt(z3); N20_z3m(k) = -sqrt(z3);

    A_vec(k) = A;
end

%*********************************************************************
*

%Find unstable locations:

k_u = 0; k_check = 0;
for ind = 1:length(N20_z1p)
    N20 = N20_z1p(ind);

    gamma1 = (3*lambda/2)*(1 + e)*N20^2;

    gamma2 = (27*e^2*lambda^2/64 + 27*e*lambda^2/32 + 27*lambda^2/64 +
...
        3*e^2/4 + 3*e/2 + 3/4)*N20^4 + (e^2*sigma - 1)*N20^2 + ...
        (1/4)*(e^2*sigma^2 + 1);

    gamma3 = (3*e*lambda/8)*(1 + e*sigma^2)*N20^2;

    gamma4 = (27*e^2*lambda^2*sigma^2/256 - 27*e^2*lambda^2*sigma/128 +
...
    27*e^2*lambda^2/256 + 3*e^2*sigma^2/16 - 3*e^2*sigma/8 + ...
    3*e^2/16)*N20^4 + (1/4)*e^2*sigma*(1-sigma)*N20^2 + e^2*sigma^2/16;

    %Characteristic equation
    CHAR = [1 gamma1 gamma2 gamma3 gamma4];

    R = roots(CHAR);

    %Looking for positive real parts of the eigenvalues:

    for kR = 1:length(R)
```

```
        if real(R(kR)) > 0
            k_u = k_u + 1;
            z_unstable(k_u) = N20;
            A_unstable(k_u) = A_vec(ind);
            k_check = k_check + 1;
        else
        end
    end
end

if k_check == 0
    No_Plot = 0;
else
    No_Plot = 1;
end

figure
plot(A_vec,N20_z1p,'-k','LineWidth',2)
hold on
if No_Plot == 1;
plot(A_unstable, z_unstable, '*k')
else
end
xlabel('\bfA'); ylabel('\bfN_2_0'); grid on;
axis([0.2 2 0.2 1.6]);


%**************************************************************************
**
%**************************************************************************
**
%**************************************************************************
**


%Generation of the third plot

clear all

sigma = 1.2; e = 0.05; k = 0;

for lambda = 0:0.001:4

    k = k+1;

    v1 = -(27*e*lambda^2*(9*lambda^2 + 16)*(e*sigma + 1)^2*(e +
1)^2)/1024;

    v2 = -(9*e*lambda^2*(e*sigma - 1)*(e*sigma + 1)^2*(e + 1))/16;

    v3 = -(9*e*lambda^2*(e^2*sigma^2 - 1)^2)/64;

    a1 = sigma^2/(1 - sigma)^2;  %alpha 1
    a2 = 2*sigma/(1 - sigma);    %alpha 2
```

```
    a3 = 9/16*lambda^2 + 1;      %alpha 3

    z1 = (-v2 - sqrt(v2^2 - 4*v3*v1))/(2*v1); %Boundary of stability
    z2 = (-v2 + sqrt(v2^2 - 4*v3*v1))/(2*v1); %Boundary of stability

    A1p = (1-sigma)*sqrt(a1*z1 + a2*z1^2 + a3*z1^3);  %corresponds to
z1
    A1m = -(1-sigma)*sqrt(a1*z1 + a2*z1^2 + a3*z1^3); %corresponds to
z1
    A2p = (1-sigma)*sqrt(a1*z2 + a2*z2^2 + a3*z2^3);  %corresponds to
z2
    A2m = -(1-sigma)*sqrt(a1*z2 + a2*z2^2 + a3*z2^3); %corresponds to
z2


    A11(k) = A1p; A12(k) = A1m; %corresponds to z1
    A21(k) = A2p; A22(k) = A2m; %corresponds to z2
    lam_vec(k) = lambda;

    if lambda == 0.001
        A12_0 = A12(k); A22_0 = A22(k);
    else
    end
end

k2 = 0;
for n = 1:k
    if lam_vec(n) < .78
        k2 = k2 + 1;
        A11_2(k2) = A11(n); A12_2(k2) = A12(n);
        A21_2(k2) = A21(n); A22_2(k2) = A22(n);
        lam_vec_2(k2) = lam_vec(n);
    else
    end
end

k_SN = 0;
for lambda_SN = 0:.001:4
    k_SN = k_SN+1;

    a1 = sigma^2/(1 - sigma)^2;
    a2 = 2*sigma/(1 - sigma);
    a3 = (9/16)*lambda_SN^2 + 1;

    %Solving for a4:
    a4_p = -a1*(-a2/(3*a3) + sqrt(a2^2 - 3*a1*a3)/(3*a3)) - ...
        a2*(-a2/(3*a3) + sqrt(a2^2 - 3*a1*a3)/(3*a3))^2 - ...
        a3*(-a2/(3*a3) + sqrt(a2^2 - 3*a1*a3)/(3*a3))^3;

    a4_m = -a1*(-a2/(3*a3) - sqrt(a2^2 - 3*a1*a3)/(3*a3)) - ...
        a2*(-a2/(3*a3) - sqrt(a2^2 - 3*a1*a3)/(3*a3))^2 - ...
        a3*(-a2/(3*a3) - sqrt(a2^2 - 3*a1*a3)/(3*a3))^3;

    %Solving for A from a4:
    A_p = sqrt(-a4_p*(1-sigma)^2); A_m = sqrt(-a4_m*(1-sigma)^2);
```

```
    lambda_SN_vec(k_SN) = lambda_SN;
    A_p_vec_SN(k_SN) = A_p;
    A_m_vec_SN(k_SN) = A_m;

    if lambda_SN == 0.001
        A12_SN = A_p_vec_SN(k_SN);
        A22_SN = A_m_vec_SN(k_SN);
    else
    end
end

%Truncating the Saddle-Node Plots:

m_SN = k_SN; g_SN = ones(1,k_SN); h_SN = ones(1,k_SN);

for n_SN = 1:k_SN
    if abs(A_p_vec_SN(n_SN)- A_m_vec_SN(n_SN)) < .0001
        g_SN(n_SN) = 0; h_SN(n_SN) = 0;
    elseif n_SN > 1 && g_SN(n_SN-1) == 0 && h_SN(n_SN-1) == 0
        m_SN = m_SN-1; g_SN(n_SN) = 0; h_SN(n_SN) = 0;
    else
    end
end

lambda_SN_vec_2 = zeros(1,m_SN);
A_p_vec_2_SN = zeros(1,m_SN); A_m_vec_2_SN = zeros(1,m_SN);

for q = 1:m_SN
    lambda_SN_vec_2(q) = lambda_SN_vec(q);
    A_p_vec_2_SN(q) = A_p_vec_SN(q);
    A_m_vec_2_SN(q) = A_m_vec_SN(q);
end

figure
hold on; grid on;
plot(lam_vec_2,A12_2,'k--',lam_vec_2,A22_2,'k--','LineWidth',2)
plot(lambda_SN_vec_2, A_p_vec_2_SN,'k','LineWidth',2)
plot(lambda_SN_vec_2, A_m_vec_2_SN,'k','LineWidth',2)
xlabel('\fontsize{12}\bf\lambda'); ylabel('\bfA');
axis([0 3.5 0 1.8]);


%**************************************************************************
**
%**************************************************************************
**
%**************************************************************************
**


%Generation of the Fourth Plot

clear lambda k k2 z1 z2
```

```
lambda = 0.2; k = 0; k5 = 0; k7 = 0;

for A = 0:0.001:2
    k = k+1;

    a1 = sigma^2/(1 - sigma)^2;
    a2 = 2*sigma/(1 - sigma);
    a3 = 9/16*lambda^2 + 1;
    a4 = -A^2/(1-sigma)^2;

    %values for z, z1, and z2 were obtained by using solve command

    z1 = ((a1^3/(27*a3^3) + a4^2/(4*a3^2) + (a2^3*a4)/(27*a3^4) - ...
        (a1^2*a2^2)/(108*a3^4) - (a1*a2*a4)/(6*a3^3))^(1/2) - ...
        a4/(2*a3) - a2^3/(27*a3^3) + (a1*a2)/(6*a3^2))^(1/3) - ...
        a2/(3*a3) - (a1/(3*a3) - a2^2/(9*a3^2))/((a1^3/(27*a3^3) + ...
        a4^2/(4*a3^2) + (a2^3*a4)/(27*a3^4) - (a1^2*a2^2)/(108*a3^4) -
...
        (a1*a2*a4)/(6*a3^3))^(1/2) - a4/(2*a3) - a2^3/(27*a3^3) + ...
        (a1*a2)/(6*a3^2))^(1/3);

    z2 = (a1/(3*a3) - a2^2/(9*a3^2))/(2*((a1^3/(27*a3^3) + ...
        a4^2/(4*a3^2) + (a2^3*a4)/(27*a3^4) - (a1^2*a2^2)/(108*a3^4) -
...
        (a1*a2*a4)/(6*a3^3))^(1/2) - a4/(2*a3) - a2^3/(27*a3^3) + ...
        (a1*a2)/(6*a3^2))^(1/3)) - ((a1^3/(27*a3^3) + a4^2/(4*a3^2) +
...
        (a2^3*a4)/(27*a3^4) - (a1^2*a2^2)/(108*a3^4) - ...
        (a1*a2*a4)/(6*a3^3))^(1/2) - a4/(2*a3) - a2^3/(27*a3^3) + ...
        (a1*a2)/(6*a3^2))^(1/3)/2 - a2/(3*a3) - ...
        (3^(1/2)*1i*(((a1^3/(27*a3^3) + a4^2/(4*a3^2) + ...
        (a2^3*a4)/(27*a3^4) - (a1^2*a2^2)/(108*a3^4) - ...
        (a1*a2*a4)/(6*a3^3))^(1/2) - a4/(2*a3) - a2^3/(27*a3^3) + ...
        (a1*a2)/(6*a3^2))^(1/3) + (a1/(3*a3) - ...
        a2^2/(9*a3^2))/((a1^3/(27*a3^3) + a4^2/(4*a3^2) + ...
        (a2^3*a4)/(27*a3^4) - (a1^2*a2^2)/(108*a3^4) - ...
        (a1*a2*a4)/(6*a3^3))^(1/2) - a4/(2*a3) - a2^3/(27*a3^3) + ...
        (a1*a2)/(6*a3^2))^(1/3)))/2;

    z3 = (a1/(3*a3) - a2^2/(9*a3^2))/(2*((a1^3/(27*a3^3) + ...
        a4^2/(4*a3^2) + (a2^3*a4)/(27*a3^4) - (a1^2*a2^2)/(108*a3^4) -
...
        (a1*a2*a4)/(6*a3^3))^(1/2) - a4/(2*a3) - a2^3/(27*a3^3) + ...
        (a1*a2)/(6*a3^2))^(1/3)) - ((a1^3/(27*a3^3) + a4^2/(4*a3^2) +
...
        (a2^3*a4)/(27*a3^4) - (a1^2*a2^2)/(108*a3^4) - ...
        (a1*a2*a4)/(6*a3^3))^(1/2) - a4/(2*a3) - a2^3/(27*a3^3) + ...
        (a1*a2)/(6*a3^2))^(1/3)/2 - a2/(3*a3) + ...
        (3^(1/2)*1i*(((a1^3/(27*a3^3) + a4^2/(4*a3^2) + ...
        (a2^3*a4)/(27*a3^4) - (a1^2*a2^2)/(108*a3^4) - ...
        (a1*a2*a4)/(6*a3^3))^(1/2) - a4/(2*a3) - a2^3/(27*a3^3) + ...
        (a1*a2)/(6*a3^2))^(1/3) + (a1/(3*a3) - ...
        a2^2/(9*a3^2))/((a1^3/(27*a3^3) + a4^2/(4*a3^2) + ...
        (a2^3*a4)/(27*a3^4) - (a1^2*a2^2)/(108*a3^4) - ...
        (a1*a2*a4)/(6*a3^3))^(1/2) - a4/(2*a3) - a2^3/(27*a3^3) + ...
        (a1*a2)/(6*a3^2))^(1/3)))/2;
```

```
    %Since z = N20^2, the following solves for N20:

    N20_z1p(k) = sqrt(z1); N20_z1m(k) = -sqrt(z1);
    N20_z2p(k) = sqrt(z2); N20_z2m(k) = -sqrt(z2);
    N20_z3p(k) = sqrt(z3); N20_z3m(k) = -sqrt(z3);

    A_vec(k) = A;
end

%Truncating the response:

k2 = 0; k3 = 0; k4 = 0;
for n = 1:k
    if A_vec(n) > 0.43
        k2 = k2 + 1;
        N20_1(k2) = N20_z1p(n); A_vec_11(k2) = A_vec(n);
        N20_2a(k2) = N20_z2p(n); A_vec_21(k2) = A_vec(n);
    else
    end
end

for m = 1:k2
    if A_vec_11(m) < 1.14
        k3 = k3 + 1;
        N20_2(k3) = N20_2a(m); A_vec_21_2(k3) = A_vec_21(m);
    else
    end
end

for p = 1:k
    if A_vec(p) < 1.14
        k4 = k4 + 1;
        N20_3(k4) = N20_z3p(p); A_vec_31(k4) = A_vec(p);
    else
    end
end

%*********************************************************************
*

%Find unstable locations:

k_u1 = 0; k_check_1 = 0;
for ind1 = 1:length(N20_1)
    N20 = N20_1(ind1);

    gamma1 = (3*lambda/2)*(1 + e)*N20^2;

    gamma2 = (27*e^2*lambda^2/64 + 27*e*lambda^2/32 + 27*lambda^2/64 +
...
        3*e^2/4 + 3*e/2 + 3/4)*N20^4 + (e^2*sigma - 1)*N20^2 + ...
        (1/4)*(e^2*sigma^2 + 1);

    gamma3 = (3*e*lambda/8)*(1 + e*sigma^2)*N20^2;
```

```
    gamma4 = (27*e^2*lambda^2*sigma^2/256 - 27*e^2*lambda^2*sigma/128 +
...
    27*e^2*lambda^2/256 + 3*e^2*sigma^2/16 - 3*e^2*sigma/8 + ...
    3*e^2/16)*N20^4 + (1/4)*e^2*sigma*(1-sigma)*N20^2 + e^2*sigma^2/16;

    %Characteristic equation
    CHAR1 = [1 gamma1 gamma2 gamma3 gamma4];

    R1 = roots(CHAR1);

    %Looking for positive real parts of the eigenvalues:

    for kR1 = 1:length(R1)
        if real(R1(kR1)) > 0
            k_u1 = k_u1 + 1;
            z_unstable_Hopf1(k_u1) = N20;
            A_unstable_Hopf1(k_u1) = A_vec_11(ind1);
            k_check_1 = k_check_1 + 1;
        else
        end
    end
end

k_u2 = 0; k_check_2 = 0;
for ind2 = 1:length(N20_2)
    N20 = N20_2(ind2);

    gamma1 = (3*lambda/2)*(1 + e)*N20^2;

    gamma2 = (27*e^2*lambda^2/64 + 27*e*lambda^2/32 + 27*lambda^2/64 +
...
        3*e^2/4 + 3*e/2 + 3/4)*N20^4 + (e^2*sigma - 1)*N20^2 + ...
        (1/4)*(e^2*sigma^2 + 1);

    gamma3 = (3*e*lambda/8)*(1 + e*sigma^2)*N20^2;

    gamma4 = (27*e^2*lambda^2*sigma^2/256 - 27*e^2*lambda^2*sigma/128 +
...
    27*e^2*lambda^2/256 + 3*e^2*sigma^2/16 - 3*e^2*sigma/8 + ...
    3*e^2/16)*N20^4 + (1/4)*e^2*sigma*(1-sigma)*N20^2 + e^2*sigma^2/16;

    %Characteristic equation
    CHAR2 = [1 gamma1 gamma2 gamma3 gamma4];

    R2 = roots(CHAR2);

    %Looking for positive real parts of the eigenvalues

    for kR2 = 1:length(R2)
        if real(R2(kR2)) > 0
            k_u2 = k_u2 + 1;
            z_unstable_Hopf2(k_u2) = N20;
            A_unstable_Hopf2(k_u2) = A_vec_21_2(ind2);
            k_check_2 = k_check_2 + 1;
```

```
        else
        end
    end
end

k_u3 = 0; k_check_3 = 0;
for ind3 = 1:length(N20_3)
    N20 = N20_3(ind3);

    gamma1 = (3*lambda/2)*(1 + e)*N20^2;

    gamma2 = (27*e^2*lambda^2/64 + 27*e*lambda^2/32 + 27*lambda^2/64 +
...
        3*e^2/4 + 3*e/2 + 3/4)*N20^4 + (e^2*sigma - 1)*N20^2 + ...
        (1/4)*(e^2*sigma^2 + 1);

    gamma3 = (3*e*lambda/8)*(1 + e*sigma^2)*N20^2;

    gamma4 = (27*e^2*lambda^2*sigma^2/256 - 27*e^2*lambda^2*sigma/128 +
...
    27*e^2*lambda^2/256 + 3*e^2*sigma^2/16 - 3*e^2*sigma/8 + ...
    3*e^2/16)*N20^4 + (1/4)*e^2*sigma*(1-sigma)*N20^2 + e^2*sigma^2/16;

    %Characteristic equation
    CHAR3 = [1 gamma1 gamma2 gamma3 gamma4];

    R3 = roots(CHAR3);

    %Looking for positive real parts of the eigenvalues

    for kR3 = 1:length(R3)
        if real(R3(kR3)) > 0
            k_u3 = k_u3 + 1;
            z_unstable_Hopf3(k_u3) = N20;
            A_unstable_Hopf3(k_u3) = A_vec_31(ind3);
            k_check_3 = k_check_3 + 1;
        else
        end
    end
end

%If k_check == 0, then that means A_unstable_Hopf and
%Z_unstable_Hopf are undefined, and thus the solution is stable.
%Here it is determined if that is the case, and a decision is made
%whether to plot based on the stability of the solution.

if k_check_1 == 0
    No_Plot1 = 0;
else
    No_Plot1 = 1;
end

if k_check_2 == 0
    No_Plot2 = 0;
```

```
else
    No_Plot2 = 1;
end

if k_check_3 == 0
    No_Plot3 = 0;
else
    No_Plot3 = 1;
end

%*********************************************************************

figure
plot(A_vec_11,N20_1,'-k',A_vec_21_2,N20_2,'-k',A_vec_31,N20_3,'-k', ...
    'LineWidth',2)
hold on; grid on;

if No_Plot1 == 1;
    plot(A_unstable_Hopf1, z_unstable_Hopf1, '*k');
else
end

if No_Plot2 == 1;
    plot(A_unstable_Hopf2, z_unstable_Hopf2, '*k');
else
end

if No_Plot3 == 1;
    plot(A_unstable_Hopf3, z_unstable_Hopf3, '*k');
else
end

xlabel('\bfA'); ylabel('\bfN_2_0');
axis([0 2 0 3.5]);


%*********************************************************************
**
%*********************************************************************
**
%*********************************************************************
**


%Generation of the fifth plot

clear all

lambda = 0.2; e = 0.05; A = 0.4; k = 0;
syms sigma

for sigma = -5:0.001:5
    k = k+1;
```

```
    v1 = -(27*e*lambda^2*(9*lambda^2 + 16)*(e*sigma + 1)^2*(e +
1)^2)/1024;

    v2 = -(9*e*lambda^2*(e*sigma - 1)*(e*sigma + 1)^2*(e + 1))/16;

    v3 = -(9*e*lambda^2*(e^2*sigma^2 - 1)^2)/64;

    a1 = sigma^2/(1 - sigma)^2;
    a2 = 2*sigma/(1 - sigma);
    a3 = 9/16*lambda^2 + 1;
    a4 = -A^2/(1-sigma)^2;

    %values for z, z1, and z2 were obtained by using solve command

    z1 = ((a1^3/(27*a3^3) + a4^2/(4*a3^2) + (a2^3*a4)/(27*a3^4) - ...
        (a1^2*a2^2)/(108*a3^4) - (a1*a2*a4)/(6*a3^3))^(1/2) - ...
        a4/(2*a3) - a2^3/(27*a3^3) + (a1*a2)/(6*a3^2))^(1/3) - ...
        a2/(3*a3) - (a1/(3*a3) - a2^2/(9*a3^2))/((a1^3/(27*a3^3) + ...
        a4^2/(4*a3^2) + (a2^3*a4)/(27*a3^4) - (a1^2*a2^2)/(108*a3^4) -
...
        (a1*a2*a4)/(6*a3^3))^(1/2) - a4/(2*a3) - a2^3/(27*a3^3) + ...
        (a1*a2)/(6*a3^2))^(1/3);

    z2 = (a1/(3*a3) - a2^2/(9*a3^2))/(2*((a1^3/(27*a3^3) + ...
        a4^2/(4*a3^2) + (a2^3*a4)/(27*a3^4) - (a1^2*a2^2)/(108*a3^4) -
...
        (a1*a2*a4)/(6*a3^3))^(1/2) - a4/(2*a3) - a2^3/(27*a3^3) + ...
        (a1*a2)/(6*a3^2))^(1/3)) - ((a1^3/(27*a3^3) + a4^2/(4*a3^2) +
...
        (a2^3*a4)/(27*a3^4) - (a1^2*a2^2)/(108*a3^4) - ...
        (a1*a2*a4)/(6*a3^3))^(1/2) - a4/(2*a3) - a2^3/(27*a3^3) + ...
        (a1*a2)/(6*a3^2))^(1/3)/2 - a2/(3*a3) - ...
        (3^(1/2)*1i*(((a1^3/(27*a3^3) + a4^2/(4*a3^2) + ...
        (a2^3*a4)/(27*a3^4) - (a1^2*a2^2)/(108*a3^4) - ...
        (a1*a2*a4)/(6*a3^3))^(1/2) - a4/(2*a3) - a2^3/(27*a3^3) + ...
        (a1*a2)/(6*a3^2))^(1/3) + (a1/(3*a3) - ...
        a2^2/(9*a3^2))/((a1^3/(27*a3^3) + a4^2/(4*a3^2) + ...
        (a2^3*a4)/(27*a3^4) - (a1^2*a2^2)/(108*a3^4) - ...
        (a1*a2*a4)/(6*a3^3))^(1/2) - a4/(2*a3) - a2^3/(27*a3^3) + ...
        (a1*a2)/(6*a3^2))^(1/3)))/2;

    z3 = (a1/(3*a3) - a2^2/(9*a3^2))/(2*((a1^3/(27*a3^3) + ...
        a4^2/(4*a3^2) + (a2^3*a4)/(27*a3^4) - (a1^2*a2^2)/(108*a3^4) -
...
        (a1*a2*a4)/(6*a3^3))^(1/2) - a4/(2*a3) - a2^3/(27*a3^3) + ...
        (a1*a2)/(6*a3^2))^(1/3)) - ((a1^3/(27*a3^3) + a4^2/(4*a3^2) +
...
        (a2^3*a4)/(27*a3^4) - (a1^2*a2^2)/(108*a3^4) - ...
        (a1*a2*a4)/(6*a3^3))^(1/2) - a4/(2*a3) - a2^3/(27*a3^3) + ...
        (a1*a2)/(6*a3^2))^(1/3)/2 - a2/(3*a3) + ...
        (3^(1/2)*1i*(((a1^3/(27*a3^3) + a4^2/(4*a3^2) + ...
        (a2^3*a4)/(27*a3^4) - (a1^2*a2^2)/(108*a3^4) - ...
        (a1*a2*a4)/(6*a3^3))^(1/2) - a4/(2*a3) - a2^3/(27*a3^3) + ...
        (a1*a2)/(6*a3^2))^(1/3) + (a1/(3*a3) - ...
        a2^2/(9*a3^2))/((a1^3/(27*a3^3) + a4^2/(4*a3^2) + ...
```

```
        (a2^3*a4)/(27*a3^4) - (a1^2*a2^2)/(108*a3^4) - ...
        (a1*a2*a4)/(6*a3^3))^(1/2) - a4/(2*a3) - a2^3/(27*a3^3) + ...
        (a1*a2)/(6*a3^2))^(1/3)))/2;

    N20_z11(k) = sqrt(z1); N20_z12(k) = -sqrt(z1);
    N20_z21(k) = sqrt(z2); N20_z22(k) = -sqrt(z2);
    N20_z31(k) = sqrt(z3); N20_z32(k) = -sqrt(z3);

    sigma_vec(k) = sigma;
end

%Truncating the response:

k2 = 0; k3 = 0; k4 = 0; k5 = 0; k6 = 0; k7 = 0;
for n = 1:k
    if N20_z11(n) < 3.1
        k2 = k2 + 1;
        z11_2(k2) = N20_z11(n); sigma_vec_11(k2) = sigma_vec(n);
        z21_2(k2) = N20_z21(n); sigma_vec_21(k2) = sigma_vec(n);
    else
    end
end

for m = 1:k2
    if sigma_vec_21(m) < -1.36 || sigma_vec_21(m) > 1.28
        k3 = k3 + 1;
        z21_2_2(k3) = z21_2(m); sigma_vec_21_2(k3) = sigma_vec_21(m);
    else
    end
end

for q = 1:k3
    if sigma_vec_21_2(q) < 1.86 && sigma_vec_21_2(q) > -3.14
        k6 = k6 + 1;
        z21_2_3(k6) = z21_2_2(q); sigma_vec_21_3(k6) =
sigma_vec_21_2(q);
    else
    end
end

for h = 1:k2
    if sigma_vec_11(h) > -3.14 && sigma_vec_11(h) < 1.86
        k5 = k5 + 1;
        z11_2_2(k5) = z11_2(h); sigma_vec_11_2(k5) = sigma_vec_11(h);
    else
    end
end

for w = 1:k5
    if sigma_vec_11_2(w) > 1.28 || sigma_vec_11_2(w) < 1
        k7 = k7 + 1;
        z11_2_3(k7) = z11_2_2(w); sigma_vec_11_3(k7) =
sigma_vec_11_2(w);
    else
    end
```

```
end

for p = 1:k
    if sigma_vec(p) < -1.36
        k4 = k4 + 1;
        z31_2(k4) = N20_z31(p); sigma_vec_31(k4) = sigma_vec(p);
    elseif sigma_vec(p) > 1
        k4 = k4 + 1;
        z31_2(k4) = N20_z31(p); sigma_vec_31(k4) = sigma_vec(p);
    else
    end
end

%Since the response jumps from one location to another at some
locations,
%this part of the program separates those jumps to discrete branches to
be
%used later for determining stability:

step_z11_1 = 0; step_z11_2 = 0; plot_thresh = 0.01; test_val = 0;
for step1 = 2:length(z11_2_3)
    if abs(z11_2_3(step1) - z11_2_3(step1 - 1)) > plot_thresh && ...
            abs(sigma_vec_11_3(step1) - sigma_vec_11_3(step1 - 1)) ...
            > plot_thresh
        test_val = 2;
    else
    end

    if test_val < 1
        step_z11_1 = step_z11_1 + 1;
        z11_final1(step_z11_1) = z11_2_3(step1);
        sigma_z11_final1(step_z11_1) = sigma_vec_11_3(step1);
    else
        step_z11_2 = step_z11_2 + 1;
        z11_final2(step_z11_2) = z11_2_3(step1);
        sigma_z11_final2(step_z11_2) = sigma_vec_11_3(step1);
    end
end

step_z21_1 = 0; step_z21_2 = 0; plot_thresh = 0.01; test_val = 0;
for step2 = 2:length(z21_2_3)
    if abs(z21_2_3(step2) - z21_2_3(step2 - 1)) > plot_thresh && ...
            abs(sigma_vec_21_3(step2) - sigma_vec_21_3(step2 - 1)) ...
            > plot_thresh
        test_val = 2;
    else
    end

    if test_val < 1
        step_z21_1 = step_z21_1 + 1;
        z21_final1(step_z21_1) = z21_2_3(step2);
        sigma_z21_final1(step_z21_1) = sigma_vec_21_3(step2);
    else
        step_z21_2 = step_z21_2 + 1;
        z21_final2(step_z21_2) = z21_2_3(step2);
        sigma_z21_final2(step_z21_2) = sigma_vec_21_3(step2);
```

```
    end
end

step_z31_1 = 0; step_z31_2 = 0; plot_thresh = 0.01; test_val = 0;
for step3 = 2:length(z31_2)
    if abs(z31_2(step3) - z31_2(step3 - 1)) > plot_thresh && ...
            abs(sigma_vec_31(step3) - sigma_vec_31(step3 - 1)) ...
            > plot_thresh
        test_val = 2;
    else
    end

    if test_val < 1
        step_z31_1 = step_z31_1 + 1;
        z31_final1(step_z31_1) = z31_2(step3);
        sigma_z31_final1(step_z31_1) = sigma_vec_31(step3);
    else
        step_z31_2 = step_z31_2 + 1;
        z31_final2(step_z31_2) = z31_2(step3);
        sigma_z31_final2(step_z31_2) = sigma_vec_31(step3);
    end
end

figure
hold on; grid on;
plot(sigma_z11_final1,z11_final1,'-k','LineWidth',2)
plot(sigma_z11_final2,z11_final2,'-k','LineWidth',2)
plot(sigma_z21_final1,z21_final1,'-k','LineWidth',2)
plot(sigma_z21_final2,z21_final2,'-k','LineWidth',2)
plot(sigma_z31_final1,z31_final1,'-k','LineWidth',2)
plot(sigma_z31_final2,z31_final2,'-k','LineWidth',2)
xlabel('\fontsize{12}\bf\sigma'); ylabel('\bfN_2_0');

%**************************************************************************
*

%Find unstable locations:

k_u1 = 0; k_check_1 = 0;
for ind1 = 1:length(z11_final1)
    N20 = z11_final1(ind1);

    gamma1 = (3*lambda/2)*(1 + e)*N20^2;

    gamma2 = (27*e^2*lambda^2/64 + 27*e*lambda^2/32 + 27*lambda^2/64 +
...
        3*e^2/4 + 3*e/2 + 3/4)*N20^4 + (e^2*sigma - 1)*N20^2 + ...
        (1/4)*(e^2*sigma^2 + 1);

    gamma3 = (3*e*lambda/8)*(1 + e*sigma^2)*N20^2;

    gamma4 = (27*e^2*lambda^2*sigma^2/256 - 27*e^2*lambda^2*sigma/128 +
...
    27*e^2*lambda^2/256 + 3*e^2*sigma^2/16 - 3*e^2*sigma/8 + ...
    3*e^2/16)*N20^4 + (1/4)*e^2*sigma*(1-sigma)*N20^2 + e^2*sigma^2/16;
```

```
    %Characteristic equation
    CHAR1 = [1 gamma1 gamma2 gamma3 gamma4];

    R1 = roots(CHAR1);

    %Looking for positive real parts of the eigenvalues:

    for kR1 = 1:length(R1)
        if real(R1(kR1)) > 0
            k_u1 = k_u1 + 1;
            z_unstable_Hopf1(k_u1) = N20;
            sigma_unstable_Hopf1(k_u1) = sigma_z11_final1(ind1);
            k_check_1 = k_check_1 + 1;
        else
        end
    end
end

k_u2 = 0; k_check_2 = 0;
for ind2 = 1:length(z11_final2)
    N20 = z11_final2(ind2);

    gamma1 = (3*lambda/2)*(1 + e)*N20^2;

    gamma2 = (27*e^2*lambda^2/64 + 27*e*lambda^2/32 + 27*lambda^2/64 +
...
        3*e^2/4 + 3*e/2 + 3/4)*N20^4 + (e^2*sigma - 1)*N20^2 + ...
        (1/4)*(e^2*sigma^2 + 1);

    gamma3 = (3*e*lambda/8)*(1 + e*sigma^2)*N20^2;

    gamma4 = (27*e^2*lambda^2*sigma^2/256 - 27*e^2*lambda^2*sigma/128 +
...
    27*e^2*lambda^2/256 + 3*e^2*sigma^2/16 - 3*e^2*sigma/8 + ...
    3*e^2/16)*N20^4 + (1/4)*e^2*sigma*(1-sigma)*N20^2 + e^2*sigma^2/16;

    %Characteristic equation
    CHAR2 = [1 gamma1 gamma2 gamma3 gamma4];

    R2 = roots(CHAR2);

    %Looking for positive real parts of the eigenvalues

    for kR2 = 1:length(R2)
        if real(R2(kR2)) > 0
            k_u2 = k_u2 + 1;
            z_unstable_Hopf2(k_u2) = N20;
            sigma_unstable_Hopf2(k_u2) = sigma_z11_final2(ind2);
            k_check_2 = k_check_2 + 1;
        else
        end
    end
end
```

```
k_u3 = 0; k_check_3 = 0;
for ind3 = 1:length(z21_final1)
    N20 = z21_final1(ind3);

    gamma1 = (3*lambda/2)*(1 + e)*N20^2;

    gamma2 = (27*e^2*lambda^2/64 + 27*e*lambda^2/32 + 27*lambda^2/64 +
...
        3*e^2/4 + 3*e/2 + 3/4)*N20^4 + (e^2*sigma - 1)*N20^2 + ...
        (1/4)*(e^2*sigma^2 + 1);

    gamma3 = (3*e*lambda/8)*(1 + e*sigma^2)*N20^2;

    gamma4 = (27*e^2*lambda^2*sigma^2/256 - 27*e^2*lambda^2*sigma/128 +
...
    27*e^2*lambda^2/256 + 3*e^2*sigma^2/16 - 3*e^2*sigma/8 + ...
    3*e^2/16)*N20^4 + (1/4)*e^2*sigma*(1-sigma)*N20^2 + e^2*sigma^2/16;

    %Characteristic equation
    CHAR3 = [1 gamma1 gamma2 gamma3 gamma4];

    R3 = roots(CHAR3);

    %Looking for positive real parts of the eigenvalues

    for kR3 = 1:length(R3)
        if real(R3(kR3)) > 0
            k_u3 = k_u3 + 1;
            z_unstable_Hopf3(k_u3) = N20;
            sigma_unstable_Hopf3(k_u3) = sigma_z21_final1(ind3);
            k_check_3 = k_check_3 + 1;
        else
        end
    end
end

k_u4 = 0; k_check_4 = 0;
for ind4 = 1:length(z21_final2)
    N20 = z21_final2(ind4);

    gamma1 = (3*lambda/2)*(1 + e)*N20^2;

    gamma2 = (27*e^2*lambda^2/64 + 27*e*lambda^2/32 + 27*lambda^2/64 +
...
        3*e^2/4 + 3*e/2 + 3/4)*N20^4 + (e^2*sigma - 1)*N20^2 + ...
        (1/4)*(e^2*sigma^2 + 1);

    gamma3 = (3*e*lambda/8)*(1 + e*sigma^2)*N20^2;

    gamma4 = (27*e^2*lambda^2*sigma^2/256 - 27*e^2*lambda^2*sigma/128 +
...
    27*e^2*lambda^2/256 + 3*e^2*sigma^2/16 - 3*e^2*sigma/8 + ...
    3*e^2/16)*N20^4 + (1/4)*e^2*sigma*(1-sigma)*N20^2 + e^2*sigma^2/16;
```

```
    %Characteristic equation
    CHAR4 = [1 gamma1 gamma2 gamma3 gamma4];

    R4 = roots(CHAR4);

    %Looking for positive real parts of the eigenvalues

    for kR4 = 1:length(R4)
        if real(R4(kR4)) > 0
            k_u4 = k_u4 + 1;
            z_unstable_Hopf4(k_u4) = N20;
            sigma_unstable_Hopf4(k_u4) = sigma_z21_final2(ind4);
            k_check_4 = k_check_4 + 1;
        else
        end
    end
end

k_u5 = 0; k_check_5 = 0;
for ind5 = 1:length(z31_final1)
    N20 = z31_final1(ind5);

    gamma1 = (3*lambda/2)*(1 + e)*N20^2;

    gamma2 = (27*e^2*lambda^2/64 + 27*e*lambda^2/32 + 27*lambda^2/64 +
...
        3*e^2/4 + 3*e/2 + 3/4)*N20^4 + (e^2*sigma - 1)*N20^2 + ...
        (1/4)*(e^2*sigma^2 + 1);

    gamma3 = (3*e*lambda/8)*(1 + e*sigma^2)*N20^2;

    gamma4 = (27*e^2*lambda^2*sigma^2/256 - 27*e^2*lambda^2*sigma/128 +
...
    27*e^2*lambda^2/256 + 3*e^2*sigma^2/16 - 3*e^2*sigma/8 + ...
    3*e^2/16)*N20^4 + (1/4)*e^2*sigma*(1-sigma)*N20^2 + e^2*sigma^2/16;

    %Characteristic equation
    CHAR5 = [1 gamma1 gamma2 gamma3 gamma4];

    R5 = roots(CHAR5);

    %Looking for positive real parts of the eigenvalues

    for kR5 = 1:length(R5)
        if real(R5(kR5)) > 0
            k_u5 = k_u5 + 1;
            z_unstable_Hopf5(k_u5) = N20;
            sigma_unstable_Hopf5(k_u5) = sigma_z31_final1(ind5);
            k_check_5 = k_check_5 + 1;
        else
        end
    end
end
```

```
%***
k_u6 = 0; k_check_6 = 0;
for ind6 = 1:length(z31_final2)
    N20 = z31_final2(ind6);

    gamma1 = (3*lambda/2)*(1 + e)*N20^2;

    gamma2 = (27*e^2*lambda^2/64 + 27*e*lambda^2/32 + 27*lambda^2/64 +
...
        3*e^2/4 + 3*e/2 + 3/4)*N20^4 + (e^2*sigma - 1)*N20^2 + ...
        (1/4)*(e^2*sigma^2 + 1);

    gamma3 = (3*e*lambda/8)*(1 + e*sigma^2)*N20^2;

    gamma4 = (27*e^2*lambda^2*sigma^2/256 - 27*e^2*lambda^2*sigma/128 +
...
    27*e^2*lambda^2/256 + 3*e^2*sigma^2/16 - 3*e^2*sigma/8 + ...
    3*e^2/16)*N20^4 + (1/4)*e^2*sigma*(1-sigma)*N20^2 + e^2*sigma^2/16;

    %Characteristic equation
    CHAR6 = [1 gamma1 gamma2 gamma3 gamma4];

    R6 = roots(CHAR6);

    %Looking for positive real parts of the eigenvalues

    for kR6 = 1:length(R6)
        if real(R6(kR6)) > 0
            k_u6 = k_u6 + 1;
            z_unstable_Hopf6(k_u6) = N20;
            sigma_unstable_Hopf6(k_u6) = sigma_z31_final2(ind6);
            k_check_6 = k_check_6 + 1;
        else
        end
    end
end

%If k_check == 0, then that means A_unstable_Hopf and
%Z_unstable_Hopf are undefined, and thus the solution is stable.
%Here it is determined if that is the case, and a decision is made
%whether to plot based on the stability of the solution.

if k_check_1 == 0
    No_Plot1 = 0;
else
    No_Plot1 = 1;
end

if k_check_2 == 0
    No_Plot2 = 0;
else
    No_Plot2 = 1;
end
```

```
if k_check_3 == 0
    No_Plot3 = 0;
else
    No_Plot3 = 1;
end

if k_check_4 == 0
    No_Plot4 = 0;
else
    No_Plot4 = 1;
end

if k_check_5 == 0
    No_Plot5 = 0;
else
    No_Plot5 = 1;
end

if k_check_6 == 0
    No_Plot6 = 0;
else
    No_Plot6 = 1;
end

if No_Plot1 == 1;
    plot(sigma_unstable_Hopf1, z_unstable_Hopf1, '*k');
else
end

if No_Plot2 == 1;
    plot(sigma_unstable_Hopf2, z_unstable_Hopf2, '*k');
else
end

if No_Plot3 == 1;
    plot(sigma_unstable_Hopf3, z_unstable_Hopf3, '*k');
else
end

if No_Plot4 == 1;
    plot(sigma_unstable_Hopf4, z_unstable_Hopf4, '*k');
else
end

if No_Plot5 == 1;
    plot(sigma_unstable_Hopf5, z_unstable_Hopf5, '*k');
else
end

if No_Plot6 == 1;
    plot(sigma_unstable_Hopf6, z_unstable_Hopf6, '*k');
else
end

axis([-5 5 0 3.5]);
```

4.0 Strongly Modulated Response (SMR)

4.1 SMR Introduction

As discussed by Starosvetsky and Gendelman (2008b), "the combination of essential nonlinearity and strong mass asymmetry brings about a possibility of response regimes qualitatively different from steady-state and weakly modulated responses existing in the vicinities of averaged flow equations in conditions of 1:1 resonance". These responses, referred to as "strongly modulated response" (SMR), can be periodic, quasiperiodic, or chaotic, hence the term "modulated". The goal of this section is to determine the frequency range for the existence of the SMR.

4.2 SMR Analysis for the System with Linear Damping

4.2.1 Slow Invariant Manifold (SIM) Projection (Linear Damping)

Manifolds are defined as "smooth and continuous surfaces", and "can be thought of as generalized surfaces" (Nayfeh and Balachandran, 1995). For the purpose of this thesis, a manifold is considered to be the space in which the system response occurs. The SIM refers to the space in which the response is dependent on the slow time scale. The following analysis presents the derivation of the SIM in terms of system parameters.

Equations (3.11),

$$\dot{\varphi_1} + \frac{i\varepsilon}{2(1+\varepsilon)}(\varphi_1 - \varphi_2) - \frac{i\varepsilon\sigma}{2(1+\varepsilon)}(\varphi_1 + \varepsilon\varphi_2) = \frac{\varepsilon A}{2}$$

and (4.1)

$$\dot{\varphi_2} + \frac{\lambda(1+\varepsilon)}{2}\varphi_2 + \frac{i}{2(1+\varepsilon)}(\varphi_2 - \varphi_1) - \frac{i\varepsilon\sigma}{2(1+\varepsilon)}(\varphi_1 + \varepsilon\varphi_2) - \frac{i}{2}(1+\varepsilon)|\varphi_2|^2\varphi_2 = \frac{\varepsilon A}{2},$$

can be combined into the following second-order ODE:

$$\frac{d^2\varphi_2}{dt^2} + \frac{d}{dt}\left[-i\frac{(1+\varepsilon)}{2}|\varphi_2|^2\varphi_2 - i\frac{i\lambda(1+\varepsilon)^2+\varepsilon^2\sigma-1-\varepsilon(1-\sigma)}{2(1+\varepsilon)}\varphi_2\right] - i\frac{\varepsilon^2 A(1-\sigma)}{4(1+\varepsilon)} + i\frac{\lambda\varepsilon(1-\sigma)}{4}\varphi_2 + \frac{\varepsilon(1-\sigma)}{4}|\varphi_2|^2\varphi_2 - \frac{\varepsilon\sigma(\varepsilon^2\sigma-1)-\varepsilon^3\sigma+\varepsilon}{4(1+\varepsilon)^2}\varphi_2 + \frac{\varepsilon(1+\varepsilon\sigma)^2}{4(1+\varepsilon)^2}\varphi_2 = i\frac{\varepsilon A(1+\varepsilon\sigma)}{4(1+\varepsilon)}. \quad (4.2)$$

Since

$$\varphi_2 = \varphi_2(\tau_0, \tau_1, \ldots),\ \tau_k = \varepsilon^k t,\ k = 0,1,\ldots,$$
$$\frac{d}{dt} = \frac{\partial}{\partial\tau_0} + \varepsilon\frac{\partial}{\partial\tau_1} + \cdots, \quad (4.3)$$

making substitutions into equation (4.2) and separating based on powers of ε gives

$$\varepsilon^0:\ {D_0}^2\varphi_2 + D_0\left[-\frac{i}{2}|\varphi_2|^2\varphi_2 + \frac{\lambda+i}{2}\varphi_2\right] = 0, \quad (4.4)$$

and

$$\varepsilon^1:\ 2{D_0}^2\varphi_2 + 2D_0D_1\varphi_2 + D_0\left[-\frac{3i}{2}|\varphi_2|^2\varphi_2 + \frac{3\lambda+i(2-\sigma)}{2}\varphi_2\right] + D_1\left[-\frac{i}{2}|\varphi_2|^2\varphi_2 + \frac{\lambda+i}{2}\varphi_2\right] + i\frac{\lambda(1-\sigma)}{4}\varphi_2 + \frac{(1-\sigma)}{4}|\varphi_2|^2\varphi_2 + \frac{\sigma}{4}\varphi_2 = i\frac{A}{4}. \quad (4.5)$$

Note that τ_0 and τ_1 are referred to as the fast and slow time scales, respectively, since by equation (4.3), $\tau_0 = t$, and $\tau_1 = \varepsilon t$, where $\varepsilon << 1$.

Integrating equation (4.4) with respect to τ_0 gives

$$\frac{\partial}{\partial\tau_0}\varphi_2 - \frac{i}{2}|\varphi_2|^2\varphi_2 + \frac{\lambda+i}{2}\varphi_2 = C(\tau_1, \ldots), \quad (4.6)$$

where $C(\tau_1, \ldots)$ is a result of the integration. The equation for the fixed points was obtained by omitting the derivative term, thus also yielding the equation for the SIM:

$$-\frac{i}{2}|\varphi_2|^2\varphi_2 + \frac{\lambda+i}{2}\varphi_2 = C(\tau_1). \quad (4.7)$$

Since the analysis is concerned only with studying the equation with respect to τ_0 and τ_1, note that the fixed points φ_2 are functions only of τ_1. Substituting

$$\Phi(\tau_1) = \varphi_2 \quad (4.8)$$

and rearranging, equation (4.7) can be rewritten as

$$-\frac{i}{2}|\Phi|^2\Phi + \frac{\lambda+i}{2}\Phi = C(\tau_1). \tag{4.9}$$

Letting

$$\Phi(\tau_1) = N(\tau_1)e^{i\theta(\tau_1)}, \tag{4.10}$$

and making the substitution, equation (4.9) becomes

$$-\frac{i}{2}N^3e^{i\theta} + \frac{\lambda+i}{2}Ne^{i\theta} = C(\tau_1), \tag{4.11}$$

Through algebraic manipulations and taking the magnitude of equation (4.11), the following relation was obtained:

$$N^6 - 2N^4 + (\lambda^2+1)N^2 = 4|C(\tau_1)|^2. \tag{4.12}$$

Making the substitution,

$$Z(\tau_1) = [N(\tau_1)]^2, \tag{4.13}$$

yields

$$Z^3 - 2Z^2 + (\lambda^2+1)Z = 4|C(\tau_1)|^2. \tag{4.14}$$

Taking the derivative of the left hand side with respect to Z and setting equal to zero gives

$$3Z^2 - 4Z + (\lambda^2+1) = 0. \tag{4.15}$$

The derivative has the following roots:

$$Z_{1,2} = \frac{2\mp\sqrt{1-3\lambda^2}}{3}. \tag{4.16}$$

Since

$$N(\tau_1) = \pm\sqrt{Z(\tau_1)}, \tag{4.17}$$

the positive values of N corresponding to the roots $Z_{1,2}$ are given as

$$N_{1,2} = \sqrt{\frac{2\mp\sqrt{1-3\lambda^2}}{3}}, \tag{4.18}$$

where N_1 and N_2 define the fold lines. Physically, fold lines correspond to the locations on the SIM at which the trajectories of the SIM may jump between stable branches.

Equation (4.12) was used to generate the SIM projection on the (N, $4|C|^2$) plane shown in Figure 4-1. As described by Starosvetsky and Gendelman (2008b), the fold lines from equation (4.18) are plotted to show the locations for the jumps from one stable branch of the SIM to another. Refer to Section 4.A.1.1 for detailed derivations of these equations and Section 4.A.4.1 for the MATLAB code used for generation of Figure 4-1.

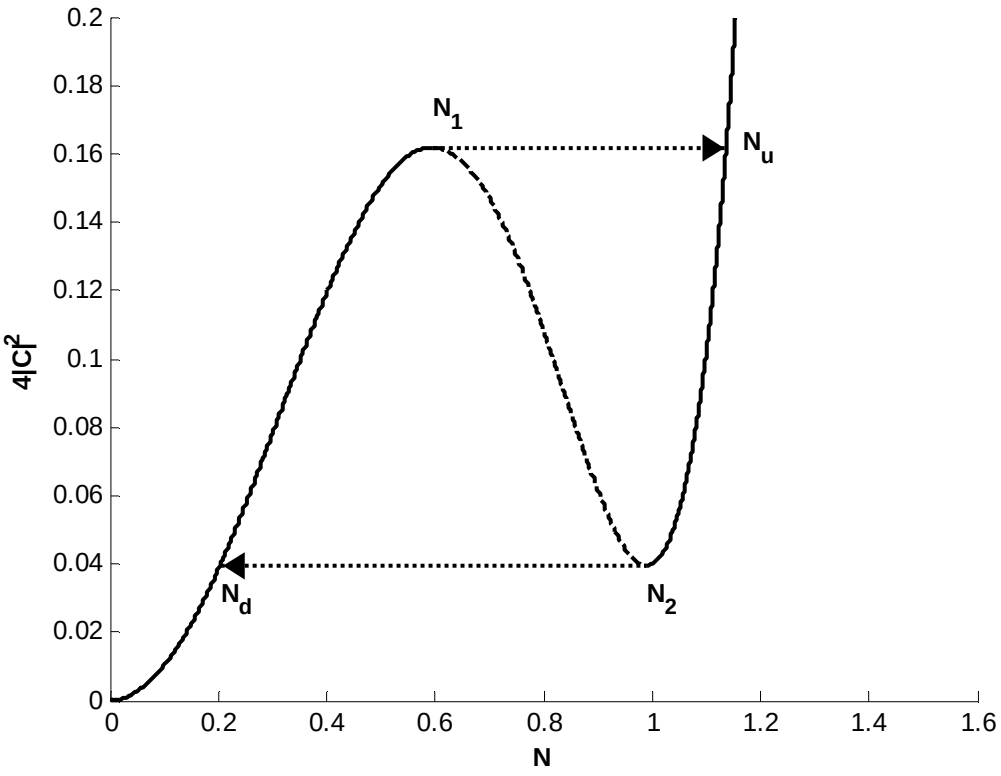


Figure 4-1. SIM projection for $\lambda = 0.2$ (linear damping)
——— = stable; - - - - - - - - = unstable

4.2.2 Phase Portraits (Linear Damping)

Continuing the analysis with equation (4.5), taking the limit as τ_0 approaches $+\infty$ and substituting

$$\Phi(\tau_1) = \varphi_2, \tag{4.19}$$

the equation becomes

$$\frac{\partial}{\partial\tau_1}\left[-\frac{i}{2}|\Phi|^2\Phi + \frac{\lambda+i}{2}\Phi\right] + \frac{(1-\sigma)}{4}|\Phi|^2\Phi + \left[\frac{\sigma}{4} + i\frac{\lambda(1-\sigma)}{4}\right]\Phi - i\frac{A}{4} = 0. \tag{4.20}$$

Note that the derivatives with respect to τ_0 drop out since a finite function is constant with respect to an infinite time.

Let

$$G = -\frac{(1-\sigma)}{4}|\Phi|^2\Phi - \left[\frac{\sigma}{4} + i\frac{\lambda(1-\sigma)}{4}\right]\Phi + i\frac{A}{4}, \tag{4.21}$$

then equation (4.20) simplifies to

$$\left[-i|\Phi|^2 + \frac{\lambda+i}{2}\right]\frac{\partial\Phi}{\partial\tau_1} - \frac{i}{2}\Phi^2\frac{\partial\Phi^*}{\partial\tau_1} = G. \tag{4.22}$$

Taking the complex conjugate of equation (4.22), making substitutions, and reducing gives

$$\frac{\partial\Phi}{\partial\tau_1} = \frac{2G\left(2i|\Phi|^2+\lambda-i\right)+2i\Phi^2G^*}{3|\Phi|^4-4|\Phi|^2+\lambda^2+1}. \tag{4.23}$$

Substituting

$$\Phi(\tau_1) = N(\tau_1)e^{i\theta(\tau_1)} \tag{4.24}$$

into equation (4.23) and performing manipulations, the following relations are derived:

$$\frac{\partial N}{\partial\tau_1} = \frac{-AN^2\cos\theta-\lambda N+\lambda A\sin\theta+A\cos\theta}{2(3N^4-4N^2+\lambda^2+1)}, \tag{4.25}$$

and

$$\frac{\partial\theta}{\partial\tau_1} = \frac{-3(1-\sigma)N^4+(1-4\sigma)N^2+3AN\sin\theta+\left[\sigma-\lambda^2(1-\sigma)\right]+(\lambda A\cos\theta-A\sin\theta)/N}{2(3N^4-4N^2+\lambda^2+1)}. \tag{4.26}$$

Letting *g(N)* represent the denominator in equations (4.25) and (4.26), the fold lines occur when *g(N)* = 0. Thus, the equations can be rescaled by *g(N)* to avoid singularities as follows:

$$N' = -AN^2 \cos\theta - \lambda N + \lambda A \sin\theta + A\cos\theta \tag{4.27}$$

and

$$\theta' = -3(1-\sigma)N^4 + (1-4\sigma)N^2 + 3AN\sin\theta + [\sigma - \lambda^2(1-\sigma)] + (\lambda A\cos\theta - A\sin\theta)/N. \tag{4.28}$$

Locations on the lower fold line at which θ is unchanging (and thus θ'=0) are denoted by Θ_1 and Θ_2. The interval Θ_1- Θ_2 was derived in MATLAB by determining the values of θ that satisfy these conditions. Refer to Section 4.A.4.2 for the code used for this computation.

In addition, MATLAB was used to numerically integrate equations (4.27) and (4.28) in order to generate the phase portraits shown in Figures 4-2, 4-3, and 4-4. Due to the rescaling, *N* refers to *N'*, and θ refers to θ' in these plots. The phase portraits were generated for time up to five seconds. Refer to Section 4.A.1.2 for details on the equation derivations and Section 4.A.4.4 for the MATLAB code. The phase portraits only show stable trajectories on the SIM. Arrows denote the direction of the trajectories with increasing time.

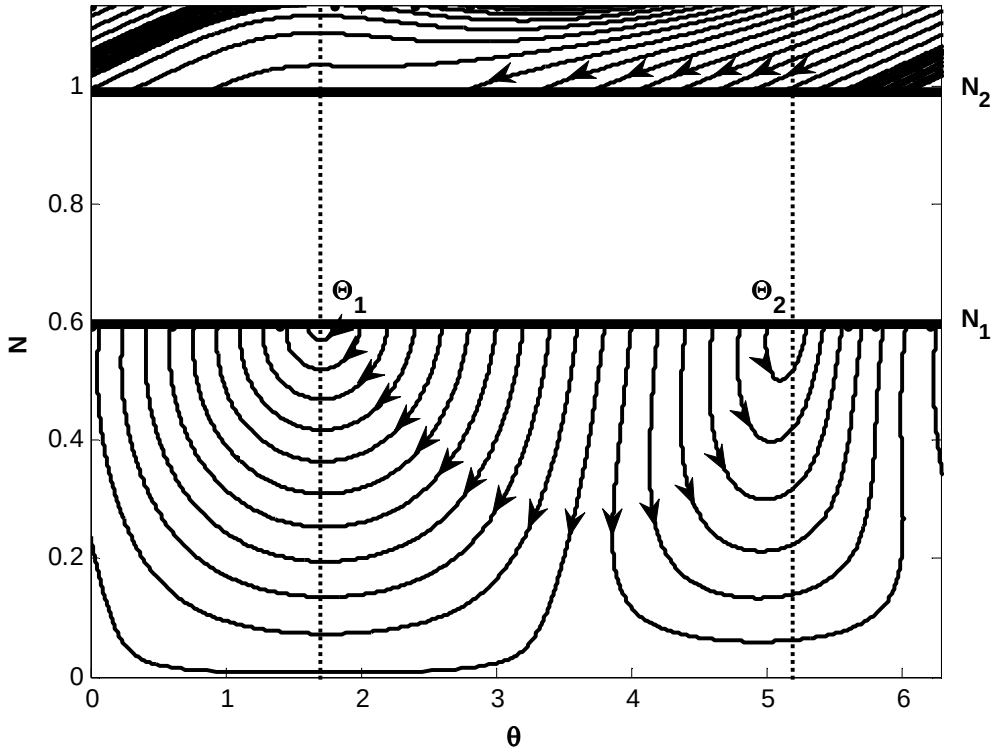


Figure 4-2. Phase portrait of the SIM for $A = 1$, $\lambda = 0.2$, and $\sigma = -2$ (linear damping)

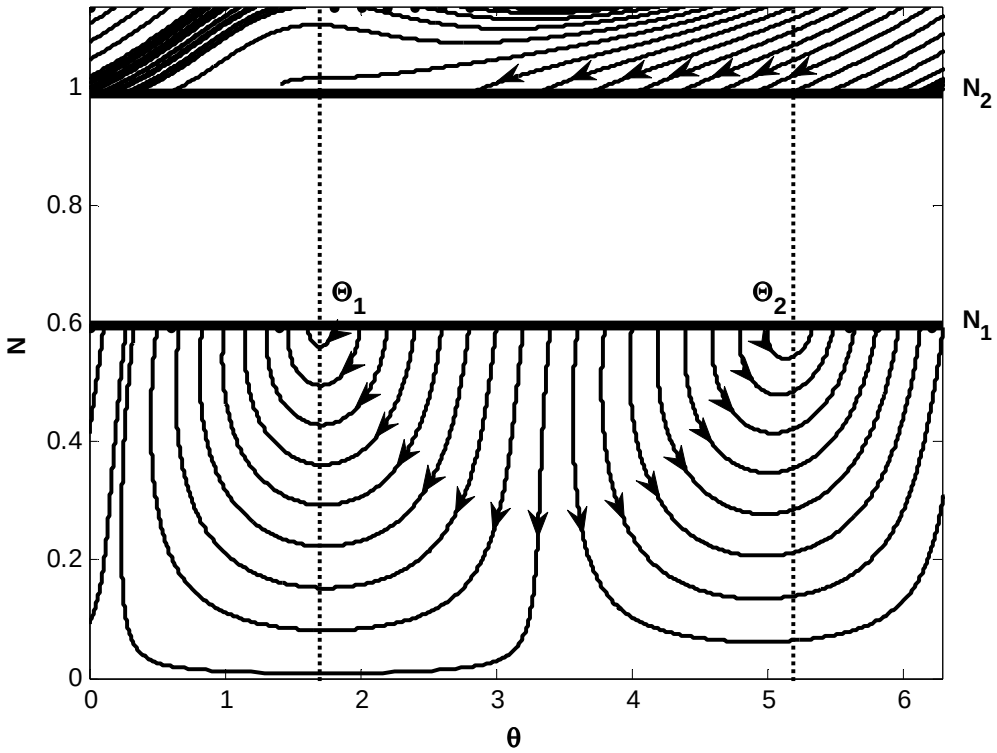


Figure 4-3. Phase portrait of the SIM for $A = 1$, $\lambda = 0.2$, and $\sigma = 0$ (linear damping)

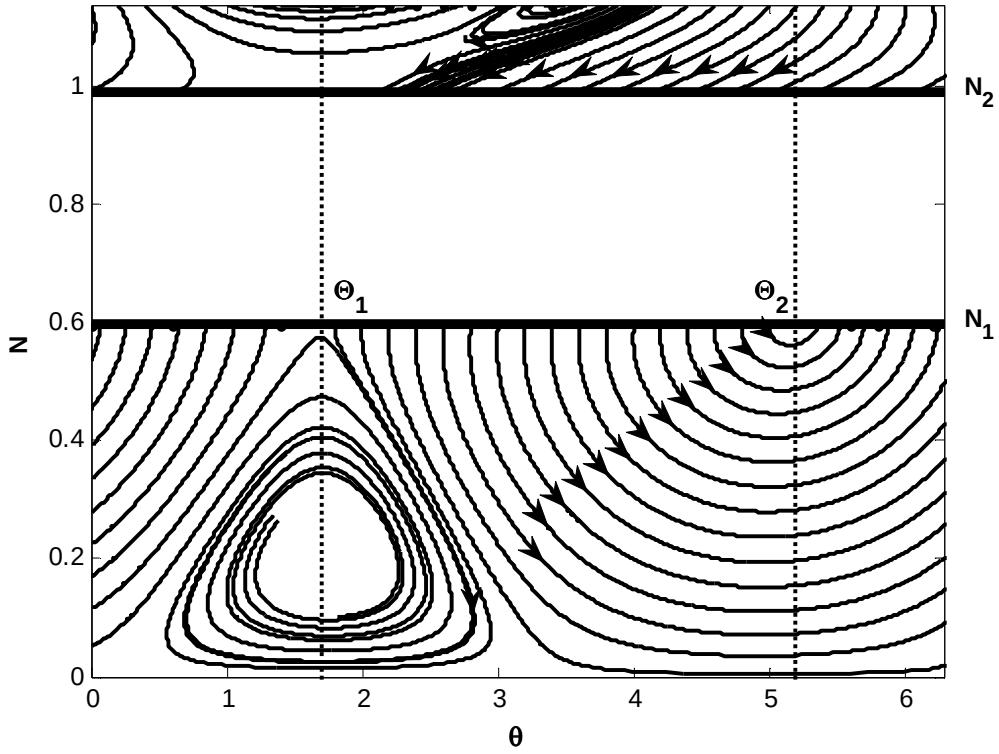


Figure 4-4. Phase portrait of the SIM for $A = 1$, $\lambda = 0.2$, and $\sigma = 5$ (linear damping)

4.2.3 1-D Mapping (Linear Damping)

Since $C(\tau_1)$ is constant, equation (4.14) may be rewritten as

$$Z_{1,2}{}^3 - 2Z_{1,2}{}^2 + (\lambda^2 + 1)Z_{1,2} = Z_{u,d}{}^3 - 2Z_{u,d}{}^2 + (\lambda^2 + 1)Z_{u,d}. \qquad (4.29)$$

MATLAB was used to determine $Z_{u,d}$ (see Section 4.A.4.3 for the MATLAB code used). Since one solution for $Z_{1,2}$ (the fold lines) was already determined, the MATLAB code returns the solutions that are not equivalent to $Z_{1,2}$. Interpretation of the MATLAB output gives

$$N_u = \sqrt{\frac{2}{3}\left(1 + \sqrt{1 - 3\lambda^2}\right)}, \qquad (4.30)$$

and

$$N_d = \sqrt{\frac{2}{3}\left(1 - \sqrt{1 - 3\lambda^2}\right)}. \tag{4.31}$$

From equation (4.11), the phase angle of the fixed point is found to be

$$\theta(\tau_1) = \arg C(\tau_1) - \tan^{-1}\left[\frac{\left(1-N^2\right)}{\lambda}\right]. \tag{4.32}$$

Taking into account that C(τ_1) is constant, manipulations between equations (4.30) and (4.32) give the phase angle at N_u on the upper stable branch from the jump at N_1:

$$\theta_u = \theta_{01} + \tan^{-1}\left[\frac{9\lambda\sqrt{1-3\lambda^2}}{15\lambda^2-1-\sqrt{1-3\lambda^2}}\right]. \tag{4.33}$$

Additionally, equations (4.31) and (4.32) give the phase angle at N_d on the lower stable branch from the jump at N_2:

$$\theta_d = \theta_{02} - \tan^{-1}\left[\frac{9\lambda\sqrt{1-3\lambda^2}}{15\lambda^2-1+\sqrt{1-3\lambda^2}}\right]. \tag{4.34}$$

Figures 4-5 through 4-9 were generated using MATLAB for varying values of σ. Refer to Section 4.A.1.3 for details on the equation derivations, Section 4.A.3 for details on the 1-D map creation, and Section 4.A.4.4 for the MATLAB code. From varying σ and observing when trajectories from the Θ_1-Θ_2 interval no longer returned, the SMR was found to exist in the intervals of σ = [-5.07, 9.11] and σ = [9.73, 10.07].

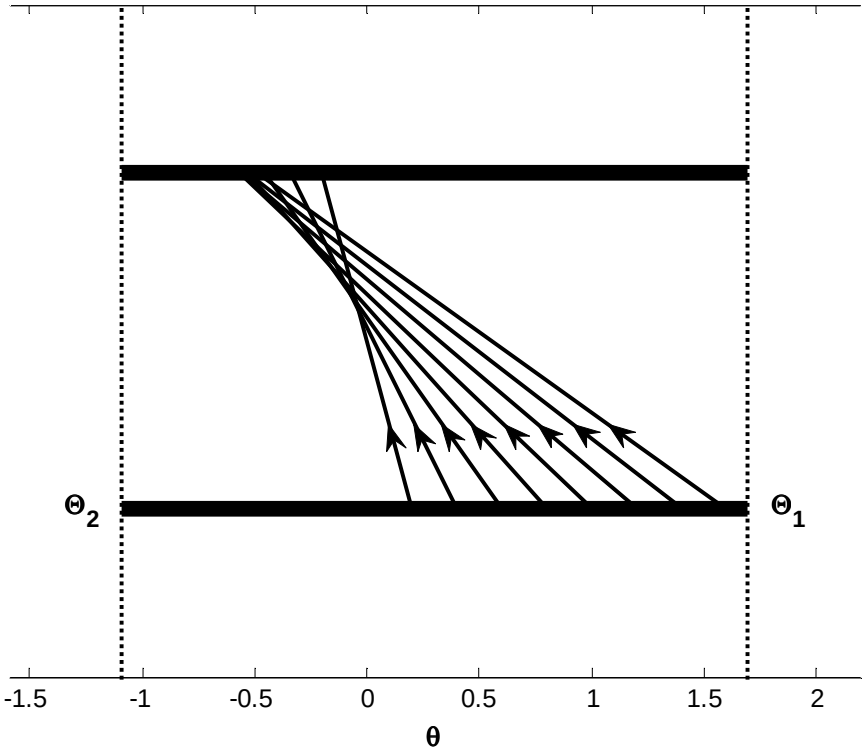


Figure 4-5. 1-D map for $A = 1$, $\lambda = 0.2$, and $\sigma = -2$ (linear damping)

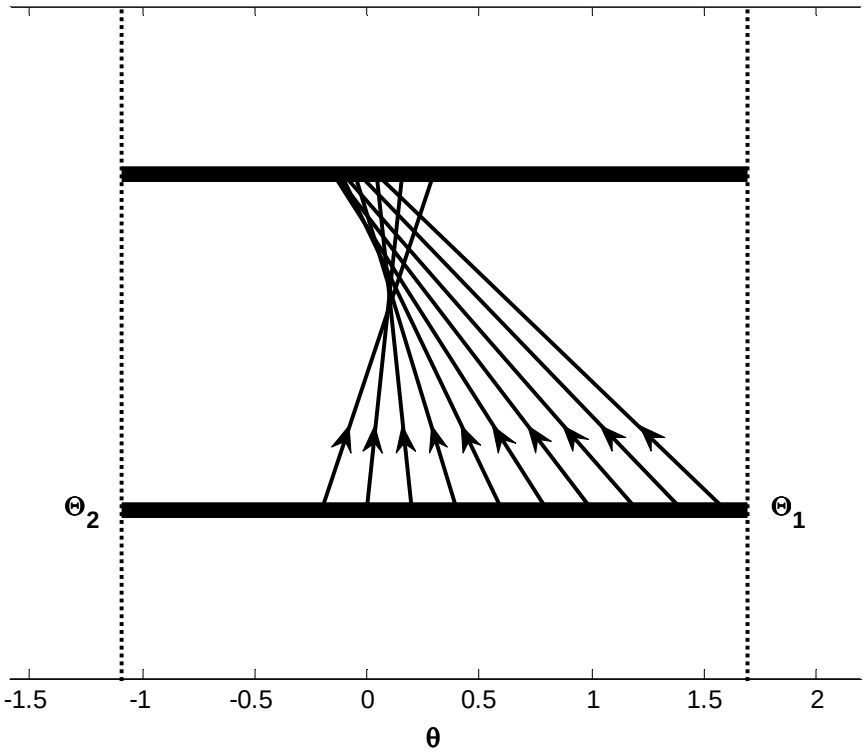


Figure 4-6. 1-D map for $A = 1$, $\lambda = 0.2$, and $\sigma = 0$ (linear damping)

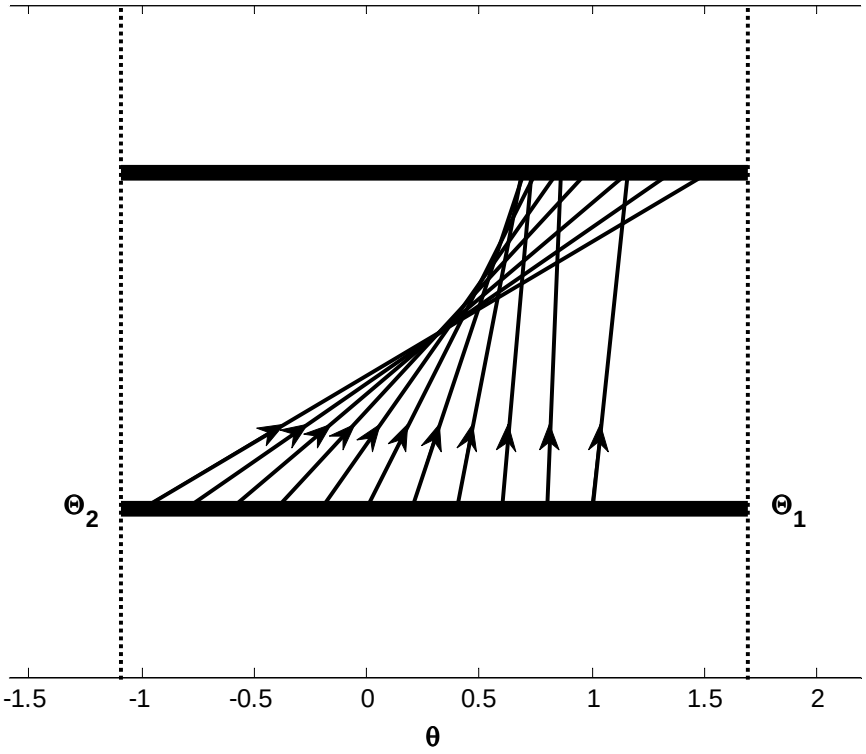


Figure 4-7. 1-D map for $A = 1$, $\lambda = 0.2$, and $\sigma = 5$ (linear damping)

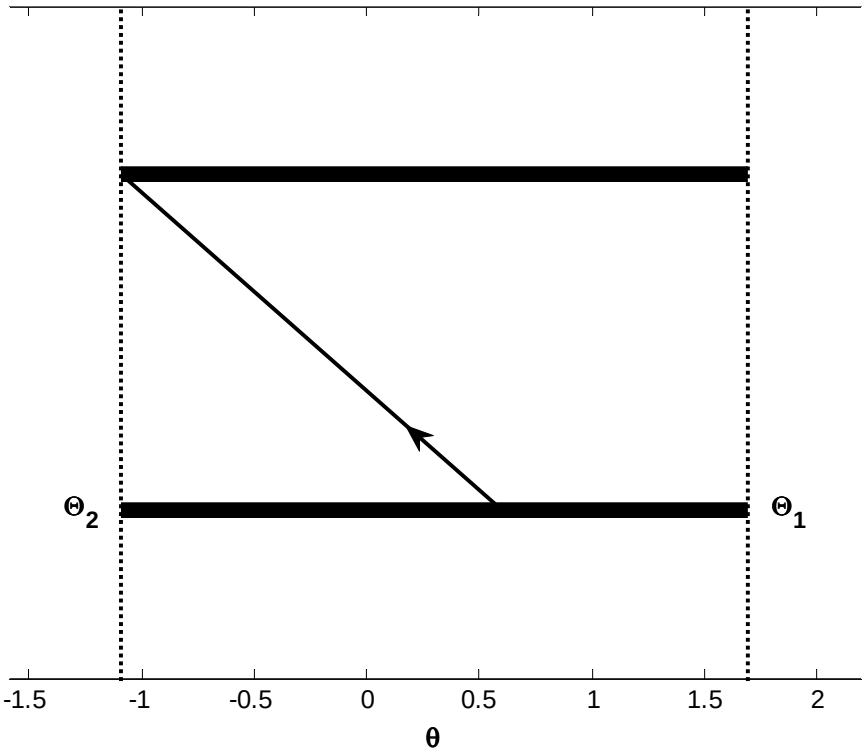


Figure 4-8. 1-D map for $A = 1$, $\lambda = 0.2$, and $\sigma = -5.07$ (linear damping)

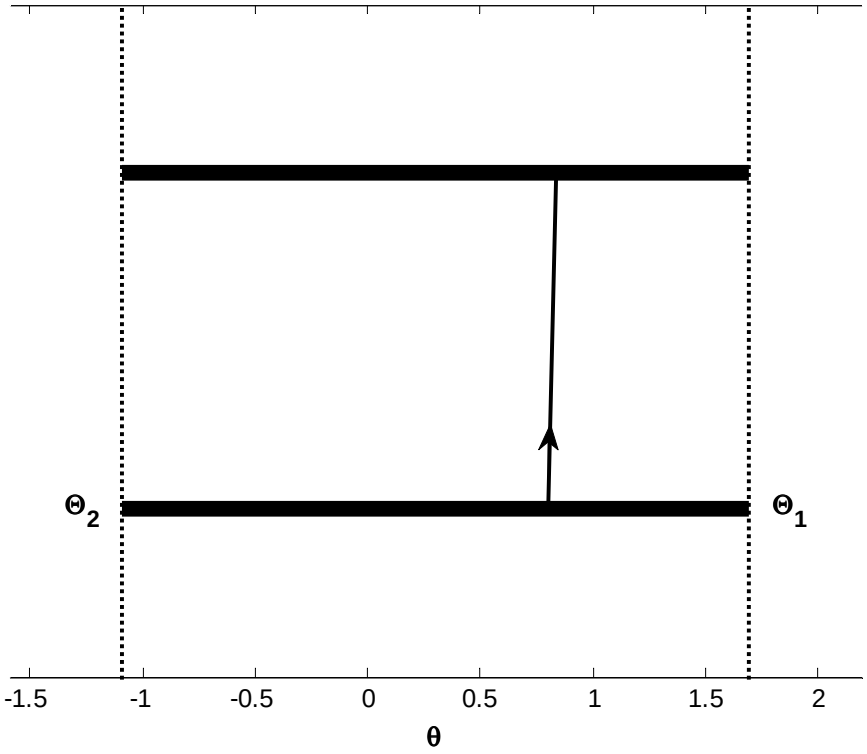


Figure 4-9. 1-D map for $A = 1$, $\lambda = 0.2$, and $\sigma = 9.11$ (linear damping)

4.3 SMR Analysis for the System with Nonlinear Damping

4.3.1 Slow Invariant Manifold (SIM) Projection (Nonlinear Damping)

Equations (3.20),

$$\dot{\varphi_1} + \frac{i\varepsilon}{2(1+\varepsilon)}(\varphi_1 - \varphi_2) - \frac{i\varepsilon\sigma}{2(1+\varepsilon)}(\varphi_1 + \varepsilon\varphi_2) = \frac{\varepsilon A}{2}$$

and (4.35)

$$\dot{\varphi_2} + \frac{3}{8}\lambda(1+\varepsilon)|\varphi_2|^2\varphi_2 + \frac{i}{2(1+\varepsilon)}(\varphi_2 - \varphi_1) -$$

$$\frac{i\varepsilon\sigma}{2(1+\varepsilon)}(\varphi_1 + \varepsilon\varphi_2) - \frac{i}{2}(1+\varepsilon)|\varphi_2|^2\varphi_2 = \frac{\varepsilon A}{2},$$

can be combined into the following second-order ODE:

$$\frac{d^2\varphi_2}{dt^2}+\frac{d}{dt}\left[-i\frac{(3\lambda i+4)(1+\varepsilon)}{8}|\varphi_2|^2\varphi_2+i\frac{\varepsilon(1-\sigma)-\varepsilon^2\sigma+1}{2(1+\varepsilon)}\varphi_2\right]-i\frac{\varepsilon^2A(1-\sigma)}{4(1+\varepsilon)}+$$
$$i\frac{3\lambda\varepsilon(1-\sigma)-i4\varepsilon}{16}|\varphi_2|^2\varphi_2+\frac{\varepsilon(1+\varepsilon\sigma)^2}{4(1+\varepsilon)^2}\varphi_2-\frac{\varepsilon\sigma}{4}|\varphi_2|^2\varphi_2-$$
$$\frac{\varepsilon\sigma(\varepsilon^2\sigma-1)-\varepsilon^3\sigma+\varepsilon}{4(1+\varepsilon)^2}\varphi_2=i\frac{\varepsilon A(1+\varepsilon\sigma)}{4(1+\varepsilon)}. \tag{4.36}$$

Since

$$\varphi_2=\varphi_2(\tau_0,\tau_1,\dots),\ \tau_k=\varepsilon^k t,\ k=0,1,\dots,$$
$$\frac{d}{dt}=\frac{\partial}{\partial\tau_0}+\varepsilon\frac{\partial}{\partial\tau_1}+\cdots, \tag{4.37}$$

making substitutions into equation (4.36) and separating based on powers of ε gives

$$\varepsilon^0:\ {D_0}^2\varphi_2-D_0\left[i\frac{(3\lambda i+4)}{8}|\varphi_2|^2\varphi_2-\frac{i}{2}\varphi_2\right]=0, \tag{4.38}$$

and

$$\varepsilon^1:\ 2{D_0}^2\varphi_2+2D_0D_1\varphi_2-D_0\left[i\frac{3(3\lambda i+4)}{8}|\varphi_2|^2\varphi_2-i\frac{2-\sigma}{2}\varphi_2\right]-$$
$$D_1\left[i\frac{(3\lambda i+4)}{8}|\varphi_2|^2\varphi_2-\frac{i}{2}\varphi_2\right]+i\frac{3\lambda(1-\sigma)-4i}{16}|\varphi_2|^2\varphi_2+\frac{1}{4}\varphi_2-$$
$$\frac{\sigma}{4}|\varphi_2|^2\varphi_2+\frac{\sigma-1}{4}\varphi_2=i\frac{A}{4}. \tag{4.39}$$

Integrating equation (4.38) with respect to τ_0 gives

$$\frac{\partial}{\partial\tau_0}\varphi_2-i\frac{(3\lambda i+4)}{8}|\varphi_2|^2\varphi_2+\frac{i}{2}\varphi_2=C(\tau_1,\dots), \tag{4.40}$$

where $C(\tau_1, \dots)$ is a result of the integration. The equation for the fixed points was obtained by omitting the derivative term, thus also yielding the equation for the SIM:

$$-i\frac{(3\lambda i+4)}{8}|\varphi_2|^2\varphi_2+\frac{i}{2}\varphi_2=C(\tau_1). \tag{4.41}$$

Since the analysis is concerned only with studying the equation with respect to τ_0 and τ_1, note that the fixed points φ_2 are functions only of τ_1.

Substituting

$$\Phi(\tau_1) = \varphi_2 \tag{4.42}$$

and rearranging, equation (4.41) can be rewritten as

$$\frac{(3\lambda-4i)}{8}|\Phi|^2\Phi + \frac{i}{2}\Phi = C(\tau_1). \tag{4.43}$$

Letting

$$\Phi(\tau_1) = N(\tau_1)e^{i\theta(\tau_1)}, \tag{4.44}$$

and making the substitution, equation (4.43) becomes

$$\frac{(3\lambda-4i)}{8}N^3e^{i\theta} + \frac{i}{2}Ne^{i\theta} = C(\tau_1), \tag{4.45}$$

Through algebraic manipulations and taking the magnitude of equation (4.45), the following relation was obtained:

$$(9\lambda^2 + 16)N^6 - 32N^4 + 16N^2 = 64|C(\tau_1)|^2. \tag{4.46}$$

Making the substitution,

$$Z(\tau_1) = [N(\tau_1)]^2, \tag{4.47}$$

yields

$$(9\lambda^2 + 16)Z^3 - 32Z^2 + 16Z = 64|C(\tau_1)|^2. \tag{4.48}$$

Taking the derivative of the left hand side with respect to Z and setting equal to zero gives

$$3(9\lambda^2 + 16)Z^2 - 64Z + 16 = 0. \tag{4.49}$$

The derivative has the following roots:

$$Z_{1,2} = \frac{32 \mp 4\sqrt{16-27\lambda^2}}{27\lambda^2+48}. \tag{4.50}$$

Since

$$N(\tau_1) = \pm\sqrt{Z(\tau_1)}, \tag{4.51}$$

the positive values of N corresponding to the roots $Z_{1,2}$ are given as

$$N_{1,2} = \sqrt{\frac{32 \mp 4\sqrt{16-27\lambda^2}}{27\lambda^2+48}}, \tag{4.52}$$

where N_1 and N_2 define the fold lines.

Equation (4.46) was used to generate the slow invariant manifold (SIM) projection on the (N, $64|C|^2$) plane shown in Figure 4-10. As described by Starosvetsky and Gendelman (2008b), the fold lines from equation (4.52) are plotted to show the locations for the jumps from one stable branch of the SIM to another. Refer to Section 4.A.2.1 for detailed derivations of these equations and Section 4.A.4.5 for the MATLAB code used for generation of Figure 4-10.

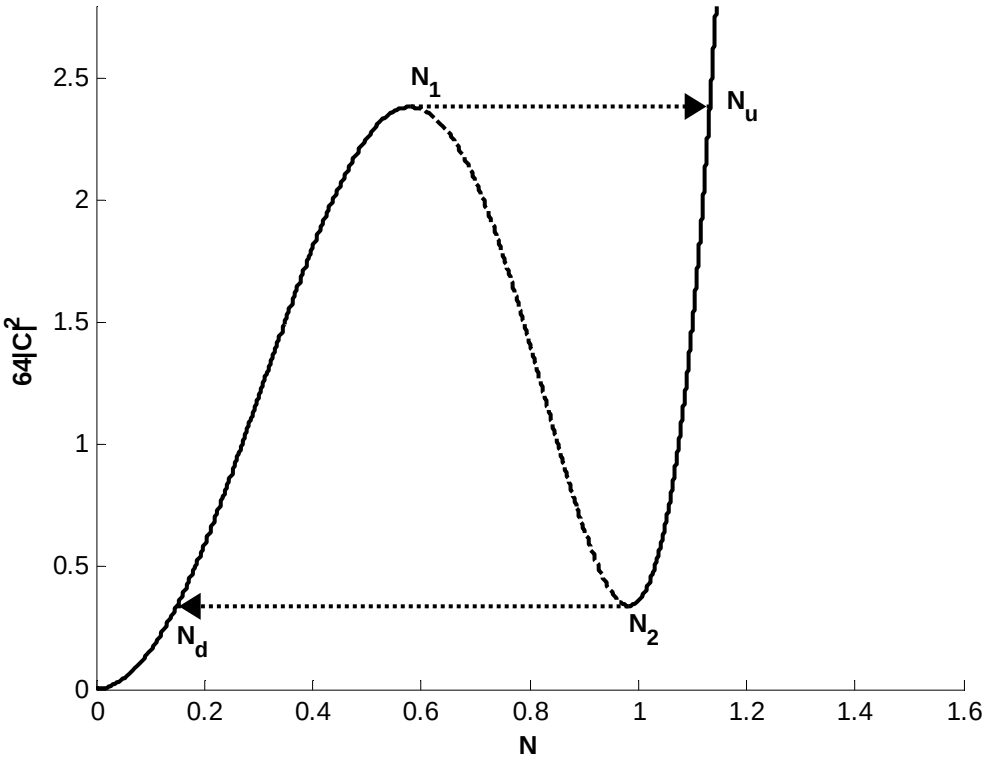


Figure 4-10. SIM projection for $\lambda = 0.2$ (nonlinear damping)
——— = stable; - - - - - - - - = unstable

4.3.2 Phase Portraits (Nonlinear Damping)

Continuing the analysis with equation (4.39), taking the limit as τ_0 approaches $+\infty$ and substituting

$$\Phi(\tau_1) = \varphi_2, \tag{4.53}$$

the equation becomes

$$\frac{\partial}{\partial\tau_1}\left[\frac{i}{2}\Phi + \frac{(3\lambda-4i)}{8}|\Phi|^2\Phi\right] + \frac{(3\lambda i+4)(1-\sigma)}{16}|\Phi|^2\Phi + \frac{\sigma}{4}\Phi - i\frac{A}{4} = 0. \tag{4.54}$$

Note that the derivatives with respect to τ_0 drop out since a finite function is constant with respect to an infinite time. Let

$$G = -\frac{(3\lambda i+4)(1-\sigma)}{16}|\Phi|^2\Phi - \frac{\sigma}{4}\Phi + i\frac{A}{4}, \tag{4.55}$$

then equation (4.54) simplifies to

$$\left[\frac{i}{2} + \frac{(3\lambda-4i)}{4}|\Phi|^2\right]\frac{\partial\Phi}{\partial\tau_1} + \frac{(3\lambda-4i)}{8}\Phi^2\frac{\partial\Phi^*}{\partial\tau_1} = G. \tag{4.56}$$

Taking the complex conjugate of equation (4.56), making substitutions, and reducing gives

$$\frac{\partial\Phi}{\partial\tau_1} = \frac{16G\left[-2i+(3\lambda+4i)|\Phi|^2\right]-8G^*(3\lambda-4i)\Phi^2}{16-64|\Phi|^2+3(9\lambda^2+16)|\Phi|^4}. \tag{4.57}$$

Substituting

$$\Phi(\tau_1) = N(\tau_1)e^{i\theta(\tau_1)} \tag{4.58}$$

into equation (4.57) and performing manipulations, the following relations are derived:

$$\frac{\partial N}{\partial\tau_1} = \frac{-12\sigma\lambda N^3+4A(3\lambda\sin\theta-4\cos\theta)N^2+16A\cos\theta}{2[16-64N^2+3(9\lambda^2+16)N^4]}, \tag{4.59}$$

and

$$\frac{\partial\theta}{\partial\tau_1} = \frac{\left(16-27\lambda^2\right)(1-\sigma)N^4-[(12\lambda-16)(1-\sigma)+48\sigma]N^2+4A(9\lambda\cos\theta+12\sin\theta)N+16\sigma-16A(\sin\theta)/N}{2[16-64N^2+3(9\lambda^2+16)N^4]}. \tag{4.60}$$

Letting *g(N)* represent the denominator in both differential equations, the fold lines occur when *g(N)* = 0. Thus, the equations can be rescaled by *g(N)* to avoid singularities as follows:

$$N' = -12\lambda N^3 + 4A(3\lambda \sin\theta - 4\cos\theta)N^2 + 16A\cos\theta, \tag{4.61}$$

and

$$\theta' = -(48 + 27\lambda^2)(1-\sigma)N^4 + (16 - 64\sigma)N^2 + 4A(9\lambda\cos\theta + 12\sin\theta)N + 16\sigma - (16A\sin\theta)/N. \tag{4.62}$$

Locations on the lower fold line at which θ is unchanging (and thus θ'=0) are denoted by Θ_1 and Θ_2. The interval Θ_1- Θ_2 was derived in MATLAB by determining the values of θ that satisfy these conditions. Refer to Section 4.A.4.6 for the code used for this computation.

In addition, MATLAB was used to numerically integrate equations (4.61) and (4.62) in order to generate the phase portraits shown in Figures 4-11, 4-12, and 4-13. Due to the rescaling, *N* refers to *N'*, and θ refers to θ' in these plots. The phase portraits were generated for time up to five seconds. Refer to Section 4.A.2.2 for details on the equation derivations and Section 4.A.4.8 for the MATLAB code. The phase portraits only show stable trajectories on the SIM. Arrows denote the direction of the trajectories with increasing time.

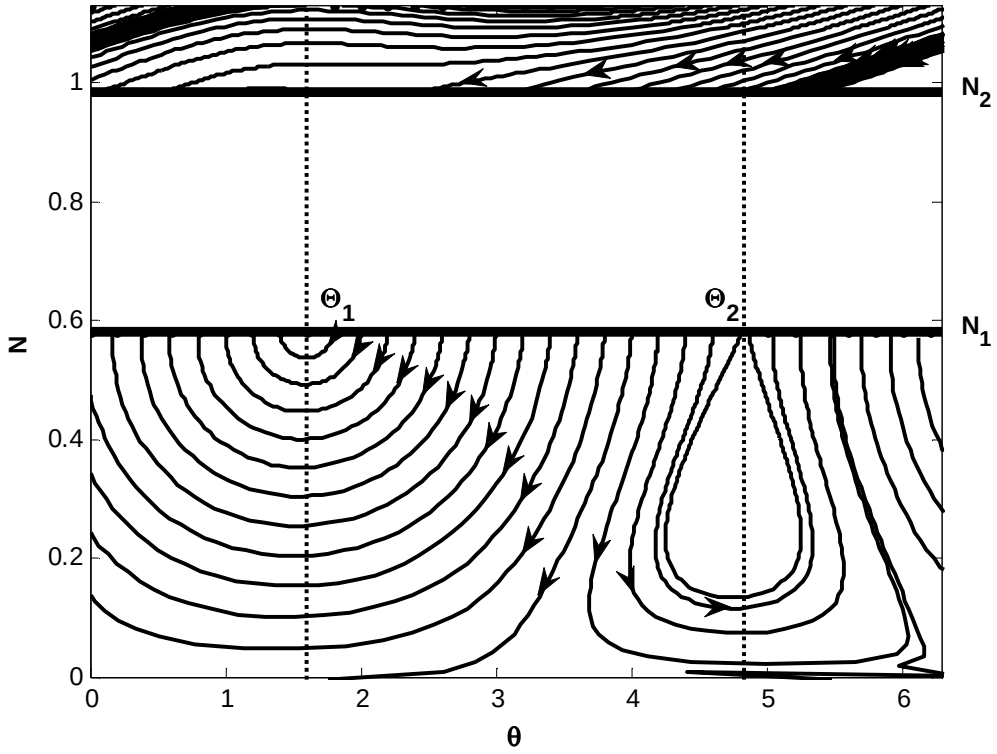


Figure 4-11. Phase portrait of the SIM for $A = 1$, $\lambda = 0.2$, and $\sigma = -4$ (nonlinear damping)

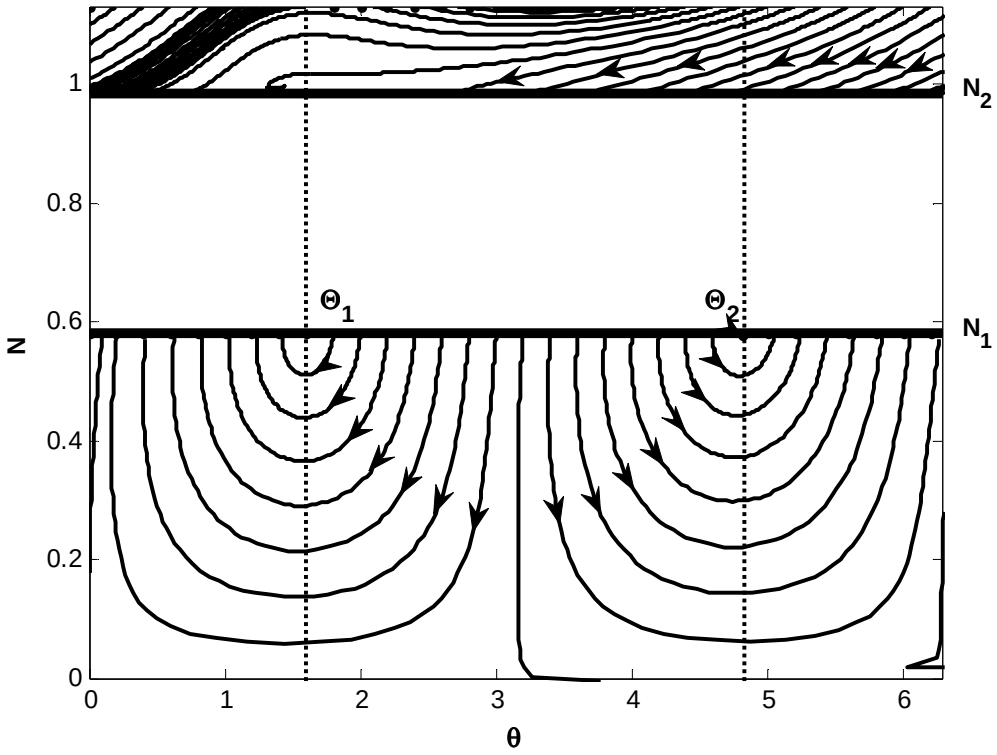


Figure 4-12. Phase portrait of the SIM for $A = 1$, $\lambda = 0.2$, and $\sigma = 0$ (nonlinear damping)

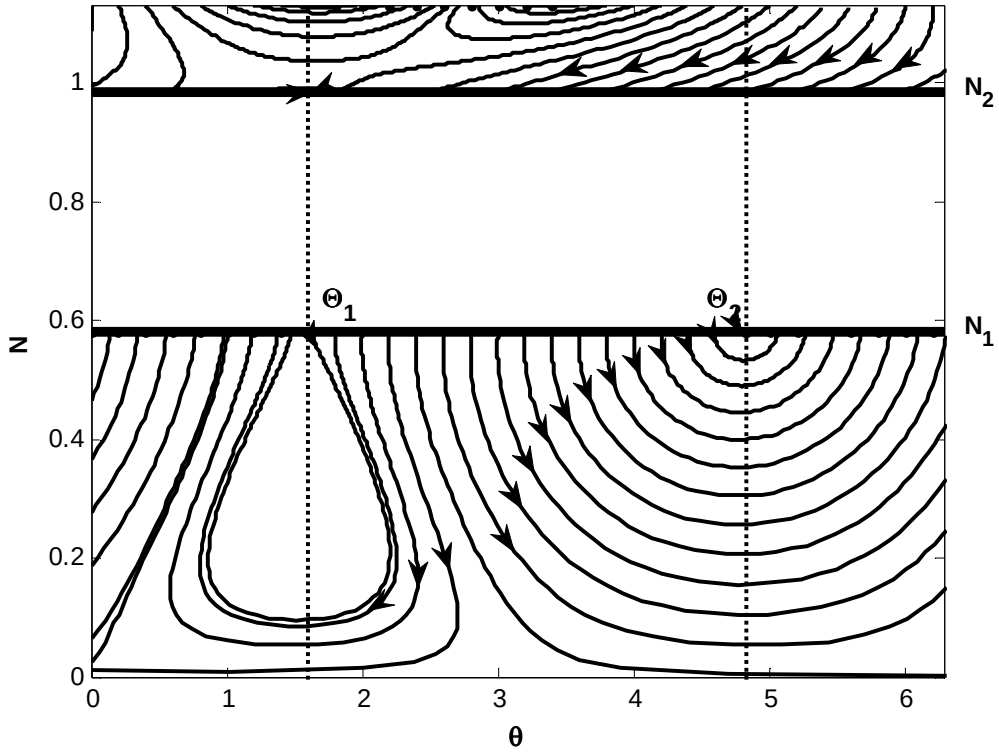


Figure 4-13. Phase portrait of the SIM for $A = 1$, $\lambda = 0.2$, and $\sigma = 4$ (nonlinear damping)

4.3.3 1-D Mapping (Nonlinear Damping)

Since $C(\tau_1)$ is constant, equation (4.48) may be rewritten as

$$(9\lambda^2 + 16){Z_{1,2}}^3 - 32{Z_{1,2}}^2 + 16Z_{1,2} = (9\lambda^2 + 16){Z_{u,d}}^3 - 32{Z_{u,d}}^2 + 16Z_{u,d}. \quad (4.63)$$

MATLAB was used to determine $Z_{u,d}$ (see Section 4.A.4.7 for the MATLAB code used). Since one solution for $Z_{1,2}$ (the fold lines) was already determined, the MATLAB code returns the solutions that are not equivalent to $Z_{1,2}$. Interpretation of the MATLAB output gives

$$N_u = \sqrt{\frac{32+8\sqrt{16-27\lambda^2}}{27\lambda^2+48}}, \quad (4.64)$$

and

$$N_d = \sqrt{\frac{32-8\sqrt{16-27\lambda^2}}{27\lambda^2+48}}. \tag{4.65}$$

From equation (4.45), the phase angle of the fixed point is found to be

$$\theta(\tau_1) = \arg C(\tau_1) - \tan^{-1}\left[\frac{4}{3\lambda}\left(\frac{1}{N^2}-1\right)\right]. \tag{4.66}$$

Taking into account that $C(\tau_1)$ is constant, manipulations between equations (4.64) and (4.66) give the phase angle at N_u on the upper stable branch from the jump at N_1:

$$\theta_u = \theta_{01} + \tan^{-1}\left[\frac{27\lambda(9\lambda^2+16)\sqrt{16-27\lambda^2}}{1215\lambda^4+36\lambda^2(56-\sqrt{16-27\lambda^2})-64\sqrt{16-27\lambda^2}-256}\right]. \tag{4.67}$$

Additionally, equations (4.65) and (4.66) give the phase angle at N_d on the lower stable branch from the jump at N_2:

$$\theta_d = \theta_{02} - \tan^{-1}\left[\frac{27\lambda(9\lambda^2+16)\sqrt{16-27\lambda^2}}{1215\lambda^4+36\lambda^2(56+\sqrt{16-27\lambda^2})+64\sqrt{16-27\lambda^2}-256}\right]. \tag{4.68}$$

Figures 4-14 through 4-18 were generated using MATLAB. Refer to Section 4.A.2.3 for details on the equation derivations, Section 4.A.3 for details on the 1-D map creation, and Section 4.A.4.8 for the MATLAB code. From varying σ and observing when trajectories from the Θ_1-Θ_2 interval no longer returned, the SMR was found to exist in the interval of σ = [-7.17, 8.47].

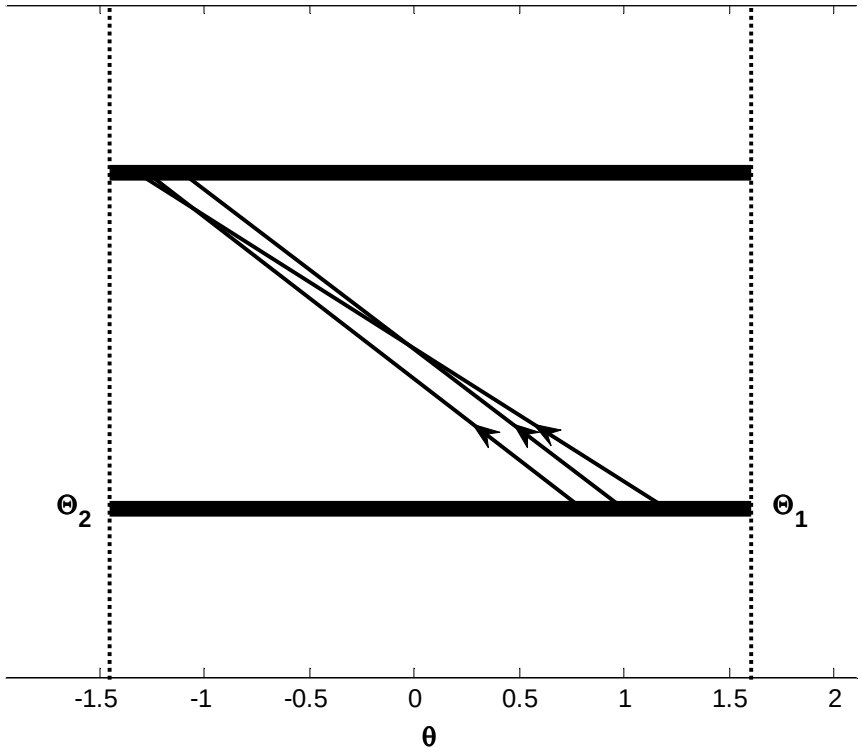


Figure 4-14. 1-D map for $A = 1$, $\lambda = 0.2$, and $\sigma = -4$ (nonlinear damping)

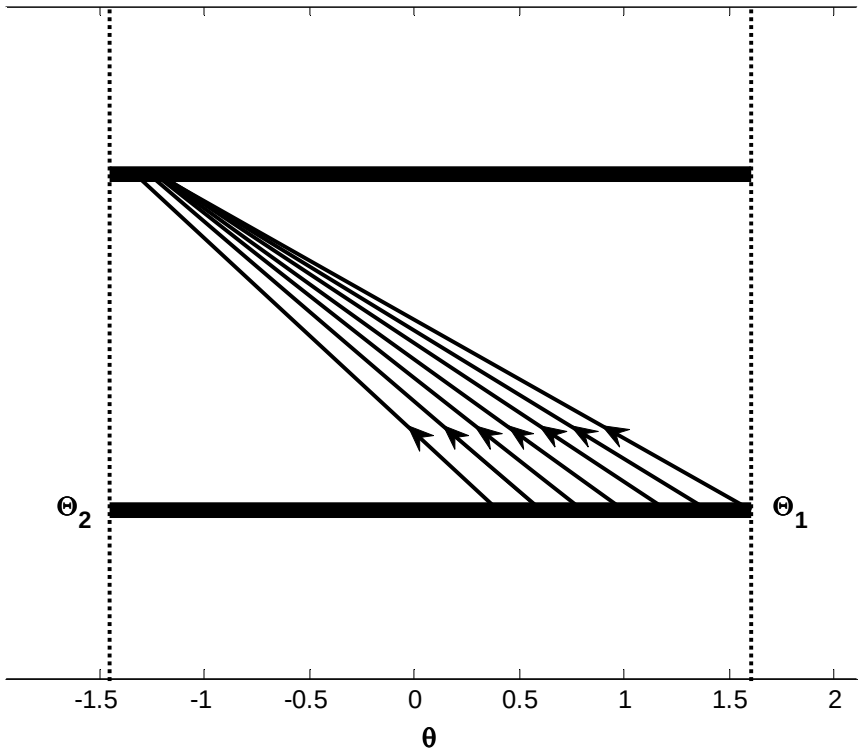


Figure 4-15. 1-D map for $A = 1$, $\lambda = 0.2$, and $\sigma = 0$ (nonlinear damping)

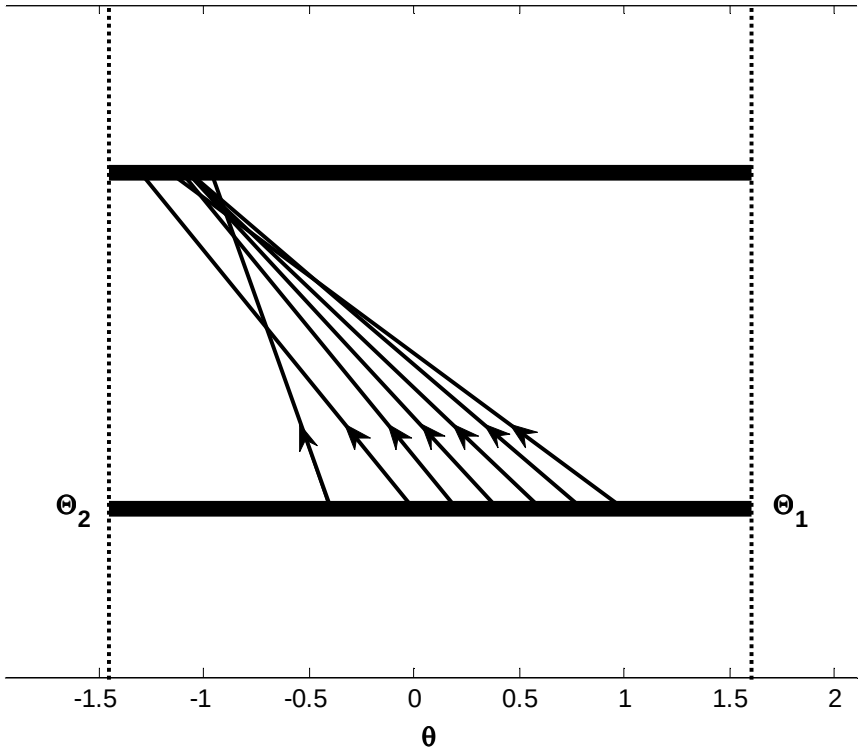


Figure 4-16. 1-D map for $A = 1$, $\lambda = 0.2$, and $\sigma = 4$ (nonlinear damping)

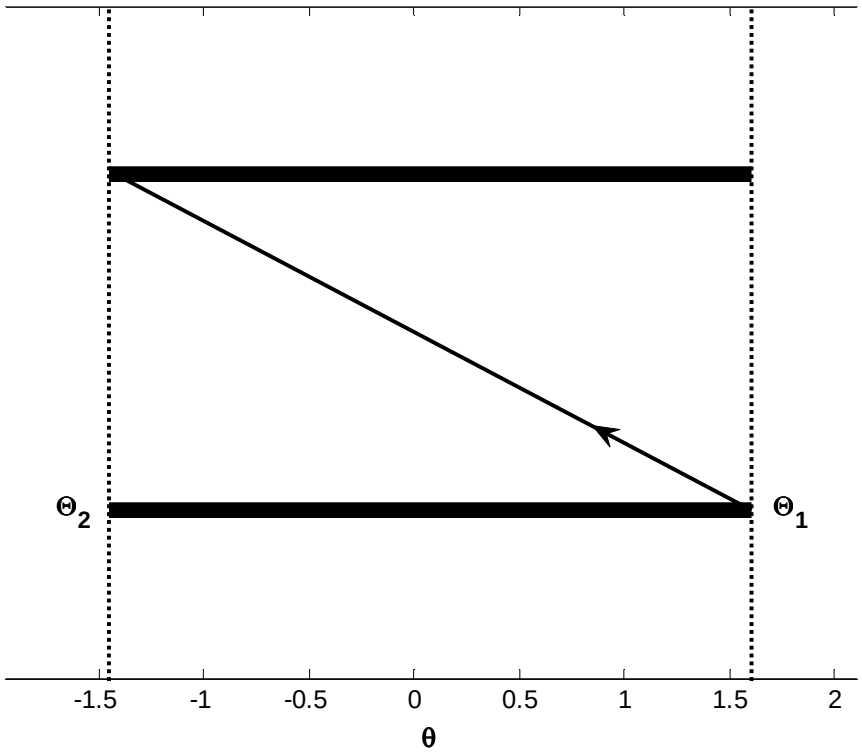


Figure 4-17. 1-D map for $A = 1$, $\lambda = 0.2$, and $\sigma = -7.17$ (nonlinear damping)

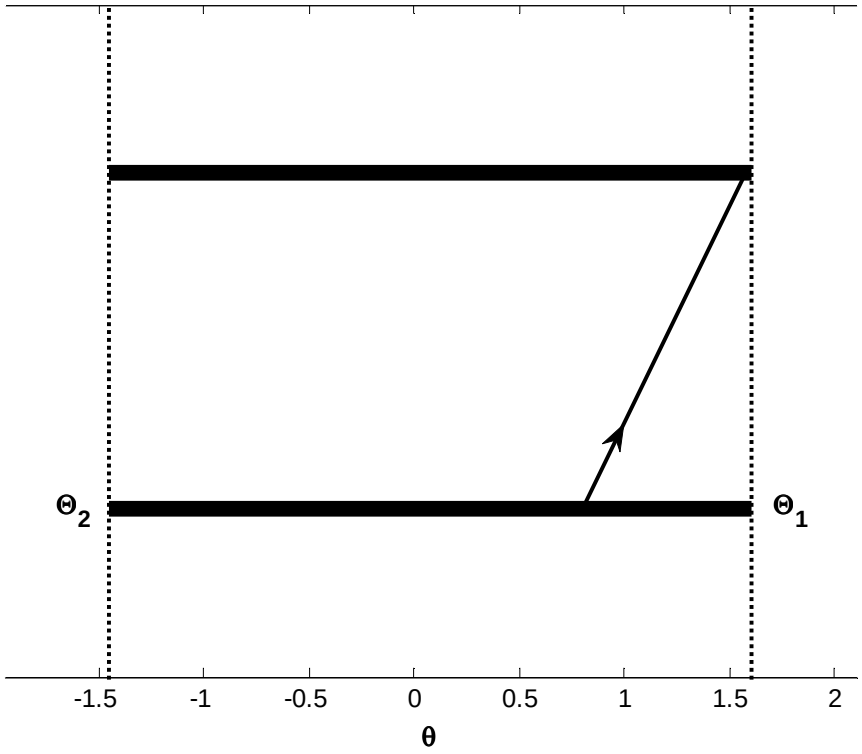


Figure 4-18. 1-D map for $A = 1$, $\lambda = 0.2$, and $\sigma = 8.47$ (nonlinear damping)

4.4 Discussion of Results for SMR

From observation of the preceding plots, there exist qualitative similarities between the linearly and nonlinearly damped systems. First, comparison between the SIM projections reveals that the values of N_1, N_2, N_u, and N_d in the linearly damped system are close to the corresponding values in the nonlinearly damped system. Translating these values over to the phase portraits shows locations of interest to the 1-D mapping. The fold lines are depicted, separating stable and unstable regions. In all phase portraits, the trajectories that leave the Θ_1-Θ_2 interval are evident with arrows denoting the direction of the trajectory with increasing time. Note that in the neighborhood of $\sigma = 0$, the phase portraits of the linearly and nonlinearly damped systems appear very similar. Qualitative similarities are seen between the two systems for increasing magnitudes of σ as well.

From observation of the 1-D maps, it is clear that varying σ has an effect on the location of the return trajectories. When σ is increased or decreased beyond specific values, the 1-D map can no longer be generated since the trajectories no longer return to the Θ_1- Θ_2 interval. Hence, determining these threshold values gives a range of σ for the existence of the SMR. Since σ is directly related to the frequency of the linear oscillator, a frequency band for the SMR existence can be obtained. By equations (3.1) and (3.9), the natural frequency of the linear oscillator in both linearly and nonlinearly damped cases is given by

$$\omega = \sqrt{\frac{stiffness\ of\ linear\ system}{mass\ of\ linear\ oscillator}} = \sqrt{1 + \varepsilon\sigma}. \tag{4.69}$$

Since the forcing frequency (1) in each case is very close to this natural frequency (~1), due to $\varepsilon << 1$, the intervals of σ derived in this analysis also provide an approximate range of forcing frequency for the occurrence of the SMR. For the linearly damped system with ε = 0.05, the SMR exists for frequency ranges of [0.86, 1.21] and [1.22, 1.23]. Again using ε = 0.05, the SMR exists in the nonlinearly damped system for a frequency range of [0.80, 1.19]. Thus, the SMR exists in both linearly and nonlinearly damped systems near 1:1 resonance.

It should be noted that the results presented in this chapter are valid only for an amplitude of A = 1. One may vary the amplitude to obtain results in other cases using the same approach.

4.A Appendix

4.A.1 SMR Equation Derivations (Linear Damping)

4.A.1.1 Derivation of Equations Related to SIM Projection (Linear Damping)

Beginning with equations (3.11),

$$\dot{\varphi_1} + \frac{i\varepsilon}{2(1+\varepsilon)}(\varphi_1 - \varphi_2) - \frac{i\varepsilon\sigma}{2(1+\varepsilon)}(\varphi_1 + \varepsilon\varphi_2) = \frac{\varepsilon A}{2}$$

and (4.70)

$$\dot{\varphi_2} + \frac{\lambda(1+\varepsilon)}{2}\varphi_2 + \frac{i}{2(1+\varepsilon)}(\varphi_2 - \varphi_1) - \frac{i\varepsilon\sigma}{2(1+\varepsilon)}(\varphi_1 + \varepsilon\varphi_2) - \frac{i}{2}(1+\varepsilon)|\varphi_2|^2\varphi_2 = \frac{\varepsilon A}{2}.$$

the second equation can be rewritten as

$$-\frac{i}{2(1+\varepsilon)}\varphi_1 - \frac{i\varepsilon\sigma}{2(1+\varepsilon)}\varphi_1 = \frac{\varepsilon A}{2} - \dot{\varphi_2} - \frac{\lambda(1+\varepsilon)}{2}\varphi_2 + \frac{i}{2}(1+\varepsilon)|\varphi_2|^2\varphi_2 - \frac{i}{2(1+\varepsilon)}\varphi_2 + \frac{i\varepsilon^2\sigma}{2(1+\varepsilon)}\varphi_2. \quad (4.71)$$

The equation can be reduced to

$$-\frac{i(1+\varepsilon\sigma)}{2(1+\varepsilon)}\varphi_1 = \frac{\varepsilon A}{2} - \dot{\varphi_2} - \frac{\lambda(1+\varepsilon)}{2}\varphi_2 + \frac{i}{2}(1+\varepsilon)|\varphi_2|^2\varphi_2 + \frac{i(\varepsilon^2\sigma-1)}{2(1+\varepsilon)}\varphi_2. \quad (4.72)$$

Solving for φ_1 gives

$$\varphi_1 = i\frac{\varepsilon A(1+\varepsilon)}{1+\varepsilon\sigma} - i\frac{2(1+\varepsilon)}{1+\varepsilon\sigma}\dot{\varphi_2} - i\frac{\lambda(1+\varepsilon)^2}{1+\varepsilon\sigma}\varphi_2 - \frac{(1+\varepsilon)^2}{1+\varepsilon\sigma}|\varphi_2|^2\varphi_2 - \frac{\varepsilon^2\sigma-1}{1+\varepsilon\sigma}\varphi_2. \quad (4.73)$$

Substituting into the first of equations (4.70) gives

$$\begin{aligned}&\frac{d}{dt}\left[i\frac{\varepsilon A(1+\varepsilon)}{1+\varepsilon\sigma} - i\frac{2(1+\varepsilon)}{1+\varepsilon\sigma}\dot{\varphi_2} - i\frac{\lambda(1+\varepsilon)^2}{1+\varepsilon\sigma}\varphi_2 - \frac{(1+\varepsilon)^2}{1+\varepsilon\sigma}|\varphi_2|^2\varphi_2 - \frac{\varepsilon^2\sigma-1}{1+\varepsilon\sigma}\varphi_2\right] + \frac{i\varepsilon}{2(1+\varepsilon)}\left[i\frac{\varepsilon A(1+\varepsilon)}{1+\varepsilon\sigma} - \right.\\ &\left. i\frac{2(1+\varepsilon)}{1+\varepsilon\sigma}\dot{\varphi_2} - i\frac{\lambda(1+\varepsilon)^2}{1+\varepsilon\sigma}\varphi_2 - \frac{(1+\varepsilon)^2}{1+\varepsilon\sigma}|\varphi_2|^2\varphi_2 - \frac{\varepsilon^2\sigma-1}{1+\varepsilon\sigma}\varphi_2 - \varphi_2\right]\\ &-\frac{i\varepsilon\sigma}{2(1+\varepsilon)}\left[i\frac{\varepsilon A(1+\varepsilon)}{1+\varepsilon\sigma} - i\frac{2(1+\varepsilon)}{1+\varepsilon\sigma}\dot{\varphi_2} - i\frac{\lambda(1+\varepsilon)^2}{1+\varepsilon\sigma}\varphi_2 - \frac{(1+\varepsilon)^2}{1+\varepsilon\sigma}|\varphi_2|^2\varphi_2 - \frac{\varepsilon^2\sigma-1}{1+\varepsilon\sigma}\varphi_2 + \varepsilon\varphi_2\right] = \frac{\varepsilon A}{2}.\end{aligned} \quad (4.74)$$

This expression can be reduced to

$$-i\frac{2(1+\varepsilon)}{1+\varepsilon\sigma}\frac{d^2\varphi_2}{dt^2}+\frac{d}{dt}\left[i\frac{\varepsilon A(1+\varepsilon)}{1+\varepsilon\sigma}-i\frac{\lambda(1+\varepsilon)^2}{1+\varepsilon\sigma}\varphi_2-\frac{(1+\varepsilon)^2}{1+\varepsilon\sigma}|\varphi_2|^2\varphi_2-\frac{\varepsilon^2\sigma-1}{1+\varepsilon\sigma}\varphi_2\right]+$$
$$\left[-\frac{\varepsilon^2 A}{2(1+\varepsilon\sigma)}+\frac{\varepsilon}{1+\varepsilon\sigma}\dot{\varphi}_2+\frac{\lambda\varepsilon(1+\varepsilon)}{2(1+\varepsilon\sigma)}\varphi_2-i\frac{\varepsilon(1+\varepsilon)}{2(1+\varepsilon\sigma)}|\varphi_2|^2\varphi_2-i\frac{\varepsilon^3\sigma-\varepsilon}{2(1+\varepsilon\sigma)(1+\varepsilon)}\varphi_2-\right. \quad (4.75)$$
$$\left.i\frac{\varepsilon}{2(1+\varepsilon)}\varphi_2\right]+\left[\frac{\varepsilon^2\sigma A}{2(1+\varepsilon\sigma)}-\frac{\varepsilon\sigma}{1+\varepsilon\sigma}\dot{\varphi}_2-\frac{\lambda\varepsilon\sigma(1+\varepsilon)}{2(1+\varepsilon\sigma)}\varphi_2+i\frac{\varepsilon\sigma(1+\varepsilon)}{2(1+\varepsilon\sigma)}|\varphi_2|^2\varphi_2+\right.$$
$$\left.i\frac{\varepsilon\sigma(\varepsilon^2\sigma-1)}{2(1+\varepsilon\sigma)(1+\varepsilon)}\varphi_2-i\frac{\varepsilon^2\sigma}{2(1+\varepsilon)}\varphi_2\right]=\frac{\varepsilon A}{2}.$$

Since $\frac{i\varepsilon A(1+\varepsilon)}{1+\varepsilon\sigma}$ is not dependent on time, the derivative of this term with respect to time is zero. Thus, this term may be removed from the bracketed expression corresponding to the first derivative with respect to time. Taking this simplification into account and further reducing the system gives

$$-i\frac{2(1+\varepsilon)}{1+\varepsilon\sigma}\frac{d^2\varphi_2}{dt^2}+\frac{d}{dt}\left[-\frac{(1+\varepsilon)^2}{1+\varepsilon\sigma}|\varphi_2|^2\varphi_2-\frac{i\lambda(1+\varepsilon)^2+\varepsilon^2\sigma-1}{1+\varepsilon\sigma}\varphi_2\right]-\frac{\varepsilon^2 A(1-\sigma)}{2(1+\varepsilon\sigma)}+$$
$$\frac{\varepsilon(1-\sigma)}{1+\varepsilon\sigma}\dot{\varphi}_2+\frac{\lambda\varepsilon(1+\varepsilon)(1-\sigma)}{2(1+\varepsilon\sigma)}\varphi_2-i\frac{\varepsilon(1-\sigma)(1+\varepsilon)}{2(1+\varepsilon\sigma)}|\varphi_2|^2\varphi_2+i\frac{\varepsilon\sigma(\varepsilon^2\sigma-1)-\varepsilon^3\sigma+\varepsilon}{2(1+\varepsilon\sigma)(1+\varepsilon)}\varphi_2- \quad (4.76)$$
$$i\frac{\varepsilon(1+\varepsilon\sigma)}{2(1+\varepsilon)}\varphi_2=\frac{\varepsilon A}{2}.$$

Taking into account that $\dot{\varphi}_2=\frac{d\varphi_2}{dt}$, and reducing the system further gives

$$-i\frac{2(1+\varepsilon)}{1+\varepsilon\sigma}\frac{d^2\varphi_2}{dt^2}+\frac{d}{dt}\left[-\frac{(1+\varepsilon)^2}{1+\varepsilon\sigma}|\varphi_2|^2\varphi_2-\frac{i\lambda(1+\varepsilon)^2+\varepsilon^2\sigma-1}{1+\varepsilon\sigma}\varphi_2+\frac{\varepsilon(1-\sigma)}{1+\varepsilon\sigma}\varphi_2\right]- \quad (4.77)$$
$$\frac{\varepsilon^2 A(1-\sigma)}{2(1+\varepsilon\sigma)}+\frac{\lambda\varepsilon(1+\varepsilon)(1-\sigma)}{2(1+\varepsilon\sigma)}\varphi_2-i\frac{\varepsilon(1-\sigma)(1+\varepsilon)}{2(1+\varepsilon\sigma)}|\varphi_2|^2\varphi_2+i\frac{\varepsilon\sigma(\varepsilon^2\sigma-1)-\varepsilon^3\sigma+\varepsilon}{2(1+\varepsilon\sigma)(1+\varepsilon)}\varphi_2-$$
$$i\frac{\varepsilon(1+\varepsilon\sigma)}{2(1+\varepsilon)}\varphi_2=\frac{\varepsilon A}{2}.$$

Again reducing the system yields

$$-i\frac{2(1+\varepsilon)}{1+\varepsilon\sigma}\frac{d^2\varphi_2}{dt^2}+\frac{d}{dt}\left[-\frac{(1+\varepsilon)^2}{1+\varepsilon\sigma}|\varphi_2|^2\varphi_2-\frac{i\lambda(1+\varepsilon)^2+\varepsilon^2\sigma-1-\varepsilon(1-\sigma)}{1+\varepsilon\sigma}\varphi_2\right]- \\ \frac{\varepsilon^2A(1-\sigma)}{2(1+\varepsilon\sigma)}+\frac{\lambda\varepsilon(1+\varepsilon)(1-\sigma)}{2(1+\varepsilon\sigma)}\varphi_2-i\frac{\varepsilon(1-\sigma)(1+\varepsilon)}{2(1+\varepsilon\sigma)}|\varphi_2|^2\varphi_2+i\frac{\varepsilon\sigma(\varepsilon^2\sigma-1)-\varepsilon^3\sigma+\varepsilon}{2(1+\varepsilon\sigma)(1+\varepsilon)}\varphi_2- \\ i\frac{\varepsilon(1+\varepsilon\sigma)}{2(1+\varepsilon)}\varphi_2=\frac{\varepsilon A}{2}. \tag{4.78}$$

Multiplying by $\frac{i(1+\varepsilon\sigma)}{2(1+\varepsilon)}$ gives

$$\frac{d^2\varphi_2}{dt^2}+\frac{d}{dt}\left[-i\frac{(1+\varepsilon)}{2}|\varphi_2|^2\varphi_2-i\frac{i\lambda(1+\varepsilon)^2+\varepsilon^2\sigma-1-\varepsilon(1-\sigma)}{2(1+\varepsilon)}\varphi_2\right]-i\frac{\varepsilon^2A(1-\sigma)}{4(1+\varepsilon)}+ \\ i\frac{\lambda\varepsilon(1-\sigma)}{4}\varphi_2+\frac{\varepsilon(1-\sigma)}{4}|\varphi_2|^2\varphi_2-\frac{\varepsilon\sigma(\varepsilon^2\sigma-1)-\varepsilon^3\sigma+\varepsilon}{4(1+\varepsilon)^2}\varphi_2+\frac{\varepsilon(1+\varepsilon\sigma)^2}{4(1+\varepsilon)^2}\varphi_2=i\frac{\varepsilon A(1+\varepsilon\sigma)}{4(1+\varepsilon)}. \tag{4.79}$$

Since

$$\varphi_2=\varphi_2(\tau_0,\tau_1,\dots),\ \tau_k=\varepsilon^k t,\ k=0,1,\dots, \\ \frac{d}{dt}=\frac{\partial}{\partial\tau_0}+\varepsilon\frac{\partial}{\partial\tau_1}+\cdots=D_0+\varepsilon D_1+\cdots, \tag{4.80}$$

and

$$\frac{d^2}{dt^2}=\left(\frac{\partial}{\partial\tau_0}+\varepsilon\frac{\partial}{\partial\tau_1}+\cdots\right)^2=(D_0+\varepsilon D_1+\cdots)^2, \tag{4.81}$$

the equation can be rewritten as

$$(D_0+\varepsilon D_1+\cdots)^2\varphi_2+ \\ (D_0+\varepsilon D_1+\cdots)\left[-i\frac{(1+\varepsilon)}{2}|\varphi_2|^2\varphi_2-i\frac{i\lambda(1+\varepsilon)^2+\varepsilon^2\sigma-1-\varepsilon(1-\sigma)}{2(1+\varepsilon)}\varphi_2\right]- \\ i\frac{\varepsilon^2A(1-\sigma)}{4(1+\varepsilon)}+i\frac{\lambda\varepsilon(1-\sigma)}{4}\varphi_2+\frac{\varepsilon(1-\sigma)}{4}|\varphi_2|^2\varphi_2-\frac{\varepsilon\sigma(\varepsilon^2\sigma-1)-\varepsilon^3\sigma+\varepsilon}{4(1+\varepsilon)^2}\varphi_2+ \\ \frac{\varepsilon(1+\varepsilon\sigma)^2}{4(1+\varepsilon)^2}\varphi_2=i\frac{\varepsilon A(1+\varepsilon\sigma)}{4(1+\varepsilon)}. \tag{4.82}$$

Multiplying both sides by $(1+\varepsilon)^2$ gives

$$(D_0+\varepsilon D_1+\cdots)^2(1+\varepsilon)^2\varphi_2+(D_0+\varepsilon D_1+\cdots)\left[-i\frac{(1+\varepsilon)^3}{2}|\varphi_2|^2\varphi_2-\right. \tag{4.83}$$

$$\left.i\frac{i\lambda(1+\varepsilon)^3+\left(\varepsilon^2\sigma-1-\varepsilon(1-\sigma)\right)(1+\varepsilon)}{2}\varphi_2\right]-i\frac{\varepsilon^2A(1-\sigma)(1+\varepsilon)}{4}+i\frac{\lambda\varepsilon(1-\sigma)(1+\varepsilon)^2}{4}\varphi_2+$$

$$\frac{\varepsilon(1-\sigma)(1+\varepsilon)^2}{4}|\varphi_2|^2\varphi_2-\frac{\varepsilon\sigma\left(\varepsilon^2\sigma-1\right)-\varepsilon^3\sigma+\varepsilon}{4}\varphi_2+\frac{\varepsilon(1+\varepsilon\sigma)^2}{4}\varphi_2=i\frac{\varepsilon A(1+\varepsilon\sigma)(1+\varepsilon)}{4}.$$

Expanding the equation and ignoring terms of ε^2 and above gives

$$\left({D_0}^2+2\varepsilon D_0D_1+\cdots\right)(1+2\varepsilon+\cdots)\varphi_2+ \tag{4.84}$$

$$(D_0+\varepsilon D_1+\cdots)\left[-i\frac{(1+3\varepsilon+\cdots)}{2}|\varphi_2|^2\varphi_2-i\frac{i\lambda(1+3\varepsilon+\cdots)-(1+\varepsilon)-\varepsilon(1-\sigma)+\cdots}{2}\varphi_2\right]+$$

$$i\frac{\lambda\varepsilon(1-\sigma)+\cdots}{4}\varphi_2+\frac{\varepsilon(1-\sigma)+\cdots}{4}|\varphi_2|^2\varphi_2-\frac{-\varepsilon\sigma+\varepsilon+\cdots}{4}\varphi_2+\frac{\varepsilon+\cdots}{4}\varphi_2=i\frac{\varepsilon A+\cdots}{4}.$$

Reducing the equation gives

$$\left({D_0}^2+2\varepsilon D_0D_1+\cdots\right)(1+2\varepsilon+\cdots)\varphi_2+ \tag{4.85}$$

$$(D_0+\varepsilon D_1+\cdots)\left[-i\frac{(1+3\varepsilon+\cdots)}{2}|\varphi_2|^2\varphi_2+\frac{\lambda(1+3\varepsilon+\cdots)+i+i\varepsilon(2-\sigma)+\cdots}{2}\varphi_2\right]+$$

$$i\frac{\lambda\varepsilon(1-\sigma)+\cdots}{4}\varphi_2+\frac{\varepsilon(1-\sigma)+\cdots}{4}|\varphi_2|^2\varphi_2+\frac{\varepsilon\sigma+\cdots}{4}\varphi_2=i\frac{\varepsilon A+\cdots}{4}.$$

Separating the equation based on powers of ε gives

$$\varepsilon^0:\ {D_0}^2\varphi_2+D_0\left[-\frac{i}{2}|\varphi_2|^2\varphi_2+\frac{\lambda+i}{2}\varphi_2\right]=0 \tag{4.86}$$

and

$$\varepsilon^1:\ 2{D_0}^2\varphi_2+2D_0D_1\varphi_2+D_0\left[-\frac{3i}{2}|\varphi_2|^2\varphi_2+\frac{3\lambda+i(2-\sigma)}{2}\varphi_2\right]+ \tag{4.87}$$

$$D_1\left[-\frac{i}{2}|\varphi_2|^2\varphi_2+\frac{\lambda+i}{2}\varphi_2\right]+i\frac{\lambda(1-\sigma)}{4}\varphi_2+\frac{(1-\sigma)}{4}|\varphi_2|^2\varphi_2+\frac{\sigma}{4}\varphi_2=i\frac{A}{4}.$$

Integrating the equation of order ε^0 with respect to τ_0 gives

$$\frac{\partial}{\partial\tau_0}\varphi_2-\frac{i}{2}|\varphi_2|^2\varphi_2+\frac{\lambda+i}{2}\varphi_2=C(\tau_1,\ldots), \tag{4.88}$$

where $C(\tau_1, \ldots)$ is a result of the integration. The equation for the fixed points was obtained by omitting the derivative term, thus yielding

$$-\frac{i}{2}|\varphi_2|^2\varphi_2 + \frac{\lambda+i}{2}\varphi_2 = C(\tau_1). \tag{4.89}$$

Since the analysis is concerned only with studying the equation with respect to τ_0 and τ_1, note that the fixed points φ_2 are functions only of τ_1. Substituting

$$\Phi(\tau_1) = \varphi_2 \tag{4.90}$$

and rearranging, the fixed point equation can be rewritten as

$$-\frac{i}{2}|\Phi|^2\Phi + \frac{\lambda+i}{2}\Phi = C(\tau_1). \tag{4.91}$$

Letting

$$\Phi(\tau_1) = N(\tau_1)e^{i\theta(\tau_1)}, \tag{4.92}$$

and making the substitution, the equation becomes

$$-\frac{i}{2}N^3e^{i\theta} + \frac{\lambda+i}{2}Ne^{i\theta} = C(\tau_1), \tag{4.93}$$

and thus

$$\left[-\frac{i}{2}N^3 + \frac{\lambda+i}{2}N\right]e^{i\theta} = C(\tau_1). \tag{4.94}$$

Using the relation

$$e^{i\theta} = \cos\theta + i\sin\theta \tag{4.95}$$

and substituting, the equation can be rewritten as

$$\left[-\frac{i}{2}N^3 + \frac{\lambda+i}{2}N\right](\cos\theta + i\sin\theta) = C(\tau_1). \tag{4.96}$$

Separating into real and imaginary parts gives

$$\left[\frac{\lambda}{2}N\cos\theta + \left(\frac{1}{2}N^3 - \frac{N}{2}\right)\sin\theta\right] + \tag{4.97}$$

$$i\left[\frac{\lambda}{2}N\sin\theta - \left(\frac{1}{2}N^3 - \frac{N}{2}\right)\cos\theta\right] = C(\tau_1).$$

Taking the magnitude of both sides gives

$$\left[\frac{\lambda}{2}N\cos\theta+\left(\frac{1}{2}N^3-\frac{N}{2}\right)\sin\theta\right]^2+ \tag{4.98}$$

$$\left[\frac{\lambda}{2}N\sin\theta-\left(\frac{1}{2}N^3-\frac{N}{2}\right)\cos\theta\right]^2=|C(\tau_1)|^2.$$

Expanding gives

$$\frac{\lambda^2}{4}N^2\cos^2\theta+\left(\frac{1}{2}N^3-\frac{N}{2}\right)^2\sin^2\theta+\lambda N\left(\frac{1}{2}N^3-\frac{N}{2}\right)\sin\theta\cos\theta+ \tag{4.99}$$

$$\frac{\lambda^2}{4}N^2\sin^2\theta+\left(\frac{1}{2}N^3-\frac{N}{2}\right)^2\cos^2\theta-\lambda N\left(\frac{1}{2}N^3-\frac{N}{2}\right)\sin\theta\cos\theta=|C(\tau_1)|^2.$$

Using trig identities, the equation becomes

$$\frac{\lambda^2}{4}N^2+\left(\frac{1}{2}N^3-\frac{N}{2}\right)^2=|C(\tau_1)|^2. \tag{4.100}$$

Expanding the expression in parenthesis gives

$$\frac{\lambda^2}{4}N^2+\frac{1}{4}N^6-\frac{1}{2}N^4+\frac{1}{4}N^2=|C(\tau_1)|^2. \tag{4.101}$$

Multiplying the equation by 4 gives

$$\lambda^2N^2+N^6-2N^4+N^2=4|C(\tau_1)|^2. \tag{4.102}$$

Finally, the equation is reduced to

$$N^6-2N^4+(\lambda^2+1)N^2=4|C(\tau_1)|^2. \tag{4.103}$$

Making the substitution,

$$Z(\tau_1)=[N(\tau_1)]^2, \tag{4.104}$$

yields

$$Z^3-2Z^2+(\lambda^2+1)Z=4|C(\tau_1)|^2. \tag{4.105}$$

Taking the derivative of the left hand side with respect to Z and setting equal to zero gives

$$3Z^2-4Z+(\lambda^2+1)=0. \tag{4.106}$$

The derivative has the following roots:

$$Z_{1,2} = \frac{4 \mp \sqrt{16-12(\lambda^2+1)}}{6}. \tag{4.107}$$

This equation reduces to

$$Z_{1,2} = \frac{2 \mp \sqrt{4-3(\lambda^2+1)}}{3}, \tag{4.108}$$

which can be further simplified to

$$Z_{1,2} = \frac{2 \mp \sqrt{1-3\lambda^2}}{3}. \tag{4.109}$$

Since

$$N(\tau_1) = \pm\sqrt{Z(\tau_1)}, \tag{4.110}$$

the positive values of N corresponding to the roots $Z_{1,2}$ can be written as

$$N_{1,2} = \sqrt{\frac{2 \mp \sqrt{1-3\lambda^2}}{3}}, \tag{4.111}$$

where N_1 and N_2 define the fold lines.

4.A.1.2 Derivation of Equations Related to Phase Portraits (Linear Damping)

Continuing the analysis with the equation of $O(\varepsilon^1)$, taking the limit as τ_0 approaches $+\infty$ gives

$$D_1\left[-\frac{i}{2}|\varphi_2|^2\varphi_2 + \frac{\lambda+i}{2}\varphi_2\right] + i\frac{\lambda(1-\sigma)}{4}\varphi_2 + \frac{(1-\sigma)}{4}|\varphi_2|^2\varphi_2 + \frac{\sigma}{4}\varphi_2 = i\frac{A}{4}, \tag{4.112}$$

with D_1 being replaced with the original notation. Note that the derivatives with respect to τ_0 drop out since a finite function is constant with respect to an infinite time. Reducing and again substituting

$$\Phi(\tau_1) = \varphi_2, \tag{4.113}$$

the equation becomes

$$\frac{\partial}{\partial\tau_1}\left[-\frac{i}{2}|\Phi|^2\Phi+\frac{\lambda+i}{2}\Phi\right]+i\frac{\lambda(1-\sigma)}{4}\Phi+\frac{(1-\sigma)}{4}|\Phi|^2\Phi+\frac{\sigma}{4}\Phi=i\frac{A}{4}. \quad (4.114)$$

Simplifying yields

$$\frac{\partial}{\partial\tau_1}\left[-\frac{i}{2}|\Phi|^2\Phi+\frac{\lambda+i}{2}\Phi\right]+\frac{(1-\sigma)}{4}|\Phi|^2\Phi+\left[\frac{\sigma}{4}+i\frac{\lambda(1-\sigma)}{4}\right]\Phi-i\frac{A}{4}=0. \quad (4.115)$$

Let

$$G=-\frac{(1-\sigma)}{4}|\Phi|^2\Phi-\left[\frac{\sigma}{4}+i\frac{\lambda(1-\sigma)}{4}\right]\Phi+i\frac{A}{4}, \quad (4.116)$$

then

$$\frac{\partial}{\partial\tau_1}\left[-\frac{i}{2}|\Phi|^2\Phi+\frac{\lambda+i}{2}\Phi\right]=G. \quad (4.117)$$

Distributing the derivative gives

$$\left[-\frac{i}{2}|\Phi|^2+\frac{\lambda+i}{2}\right]\frac{\partial\Phi}{\partial\tau_1}-\frac{i}{2}\Phi\frac{\partial|\Phi|^2}{\partial\tau_1}=G. \quad (4.118)$$

But since

$$|\Phi|^2=\Phi\Phi^*, \quad (4.119)$$

the expression can be rewritten as

$$\left[-\frac{i}{2}|\Phi|^2+\frac{\lambda+i}{2}\right]\frac{\partial\Phi}{\partial\tau_1}-\frac{i}{2}\Phi\left[\frac{\partial}{\partial\tau_1}(\Phi\Phi^*)\right]=G. \quad (4.120)$$

Distributing the derivative gives

$$\left[-\frac{i}{2}|\Phi|^2+\frac{\lambda+i}{2}\right]\frac{\partial\Phi}{\partial\tau_1}-\frac{i}{2}\Phi\left[\Phi^*\frac{\partial\Phi}{\partial\tau_1}+\Phi\frac{\partial\Phi^*}{\partial\tau_1}\right]=G. \quad (4.121)$$

The equation can be reduced to

$$\left[-\frac{i}{2}|\Phi|^2+\frac{\lambda+i}{2}\right]\frac{\partial\Phi}{\partial\tau_1}-\frac{i}{2}|\Phi|^2\frac{\partial\Phi}{\partial\tau_1}-\frac{i}{2}\Phi^2\frac{\partial\Phi^*}{\partial\tau_1}=G. \quad (4.122)$$

Finally, the equation simplifies to

$$\left[-i|\Phi|^2+\frac{\lambda+i}{2}\right]\frac{\partial\Phi}{\partial\tau_1}-\frac{i}{2}\Phi^2\frac{\partial\Phi^*}{\partial\tau_1}=G. \quad (4.123)$$

Taking the complex conjugate gives

$$\left[i|\Phi|^2+\frac{\lambda-i}{2}\right]\frac{\partial\Phi^*}{\partial\tau_1}+\frac{i}{2}\Phi^{*2}\frac{\partial\Phi}{\partial\tau_1}=G^*. \tag{4.124}$$

Solving for the complex conjugate derivative gives

$$\frac{\partial\Phi^*}{\partial\tau_1}=\frac{G^*-\frac{i}{2}\Phi^{*2}\frac{\partial\Phi}{\partial\tau_1}}{i|\Phi|^2+\frac{\lambda-i}{2}}. \tag{4.125}$$

Substituting into the expression above gives

$$\left[-i|\Phi|^2+\frac{\lambda+i}{2}\right]\frac{\partial\Phi}{\partial\tau_1}-\frac{i}{2}\Phi^2\left[\frac{G^*-\frac{i}{2}\Phi^{*2}\frac{\partial\Phi}{\partial\tau_1}}{i|\Phi|^2+\frac{\lambda-i}{2}}\right]=G. \tag{4.126}$$

Multiplying the equation by $\left(i|\Phi|^2+\frac{\lambda-i}{2}\right)$ gives

$$\begin{aligned}&\left(-i|\Phi|^2+\frac{\lambda+i}{2}\right)\left(i|\Phi|^2+\frac{\lambda-i}{2}\right)\frac{\partial\Phi}{\partial\tau_1}-\\&\frac{i}{2}\Phi^2\left(G^*-\frac{i}{2}\Phi^{*2}\frac{\partial\Phi}{\partial\tau_1}\right)=G\left(i|\Phi|^2+\frac{\lambda-i}{2}\right).\end{aligned} \tag{4.127}$$

Further manipulation yields

$$\begin{aligned}&\left[|\Phi|^4+i|\Phi|^2\left(\frac{\lambda+i}{2}-\frac{\lambda-i}{2}\right)+\frac{(\lambda+i)(\lambda-i)}{4}\right]\frac{\partial\Phi}{\partial\tau_1}-\frac{i}{2}\Phi^2G^*\\&-\frac{1}{4}\Phi^2\Phi^{*2}\frac{\partial\Phi}{\partial\tau_1}=G\left(i|\Phi|^2+\frac{\lambda-i}{2}\right).\end{aligned} \tag{4.128}$$

Simplifying gives

$$\left[|\Phi|^4-|\Phi|^2+\frac{\lambda^2+1}{4}-\frac{1}{4}\Phi^2\Phi^{*2}\right]\frac{\partial\Phi}{\partial\tau_1}-\frac{i}{2}\Phi^2G^*=G\left(i|\Phi|^2+\frac{\lambda-i}{2}\right). \tag{4.129}$$

Multiplying the equation by 4 gives

$$\begin{aligned}&\left[4|\Phi|^4-4|\Phi|^2+\lambda^2+1-\Phi^2\Phi^{*2}\right]\frac{\partial\Phi}{\partial\tau_1}-\\&2i\Phi^2G^*=2G(2i|\Phi|^2+\lambda-i).\end{aligned} \tag{4.130}$$

Solving for the derivative term gives

$$\frac{\partial\Phi}{\partial\tau_1}=\frac{2G\left(2i|\Phi|^2+\lambda-i\right)+2i\Phi^2G^*}{4|\Phi|^4-4|\Phi|^2+\lambda^2+1-\Phi^2\Phi^{*2}}. \tag{4.131}$$

Substituting in

$$|\Phi|^4 = \Phi^2\Phi^{*2} \tag{4.132}$$

and reducing gives

$$\frac{\partial\Phi}{\partial\tau_1} = \frac{2G\left(2i|\Phi|^2+\lambda-i\right)+2i\Phi^2G^*}{3|\Phi|^4-4|\Phi|^2+\lambda^2+1}. \tag{4.133}$$

Since

$$\Phi(\tau_1) = N(\tau_1)e^{i\theta(\tau_1)}, \tag{4.134}$$

the following can be written:

$$\frac{\partial\Phi}{\partial\tau_1} = \frac{\partial\mathrm{N}}{\partial\tau_1}e^{i\theta} + iNe^{i\theta}\frac{\partial\theta}{\partial\tau_1}. \tag{4.135}$$

Thus,

$$\frac{\partial\Phi}{\partial\tau_1} = \left(\frac{\partial\mathrm{N}}{\partial\tau_1} + iN\frac{\partial\theta}{\partial\tau_1}\right)e^{i\theta}. \tag{4.136}$$

Setting equal to the expression above gives

$$\left(\frac{\partial\mathrm{N}}{\partial\tau_1} + iN\frac{\partial\theta}{\partial\tau_1}\right)e^{i\theta} = \frac{2G\left(2i|\Phi|^2+\lambda-i\right)+2i\Phi^2G^*}{3|\Phi|^4-4|\Phi|^2+\lambda^2+1}. \tag{4.137}$$

Again substituting

$$\Phi(\tau_1) = N(\tau_1)e^{i\theta(\tau_1)} \tag{4.138}$$

gives

$$\left(\frac{\partial\mathrm{N}}{\partial\tau_1} + iN\frac{\partial\theta}{\partial\tau_1}\right)e^{i\theta} = \frac{2G\left(2iN^2+\lambda-i\right)+2iN^2e^{2i\theta}G^*}{3N^4-4N^2+\lambda^2+1}. \tag{4.139}$$

Moving the exponential term to the right hand side gives

$$\frac{\partial N}{\partial\tau_1} + iN\frac{\partial\theta}{\partial\tau_1} = \frac{2Ge^{-i\theta}\left(2iN^2+\lambda-i\right)+2iN^2G^*e^{i\theta}}{3N^4-4N^2+\lambda^2+1}. \tag{4.140}$$

Substituting

$$\Phi(\tau_1) = N(\tau_1)e^{i\theta(\tau_1)} \tag{4.141}$$

into the expression for G gives

$$G = -\frac{(1-\sigma)}{4}N^3 e^{i\theta} - \left[\frac{\sigma}{4} + i\frac{\lambda(1-\sigma)}{4}\right]Ne^{i\theta} + i\frac{A}{4}. \tag{4.142}$$

Thus,

$$G^* = -\frac{(1-\sigma)}{4}N^3 e^{-i\theta} - \left[\frac{\sigma}{4} - i\frac{\lambda(1-\sigma)}{4}\right]Ne^{-i\theta} - i\frac{A}{4}. \tag{4.143}$$

Therefore,

$$Ge^{-i\theta} = -\frac{(1-\sigma)}{4}N^3 - \left[\frac{\sigma}{4} + i\frac{\lambda(1-\sigma)}{4}\right]N + i\frac{A}{4}e^{-i\theta}, \tag{4.144}$$

and

$$G^* e^{i\theta} = -\frac{(1-\sigma)}{4}N^3 - \left[\frac{\sigma}{4} - i\frac{\lambda(1-\sigma)}{4}\right]N - i\frac{A}{4}e^{i\theta}. \tag{4.145}$$

Substituting into the differential equation gives

$$\frac{\partial N}{\partial \tau_1} + iN\frac{\partial \theta}{\partial \tau_1} = \frac{2\left\{-\frac{(1-\sigma)}{4}N^3 - \left[\frac{\sigma}{4}+i\frac{\lambda(1-\sigma)}{4}\right]N + i\frac{A}{4}e^{-i\theta}\right\}\left(2iN^2+\lambda-i\right)}{3N^4-4N^2+\lambda^2+1} + \frac{2iN^2\left\{-\frac{(1-\sigma)}{4}N^3 - \left[\frac{\sigma}{4}-i\frac{\lambda(1-\sigma)}{4}\right]N - i\frac{A}{4}e^{i\theta}\right\}}{3N^4-4N^2+\lambda^2+1}. \tag{4.146}$$

Manipulating the expression gives

$$\frac{\partial N}{\partial \tau_1} + iN\frac{\partial \theta}{\partial \tau_1} = \frac{\left\{-\frac{(1-\sigma)}{2}N^3 - \left[\frac{\sigma}{2}+i\frac{\lambda(1-\sigma)}{2}\right]N + i\frac{A}{2}e^{-i\theta}\right\}\left(2iN^2+\lambda-i\right)}{3N^4-4N^2+\lambda^2+1} + \frac{\left\{-i\frac{(1-\sigma)}{2}N^5 - \left[i\frac{\sigma}{2}+\frac{\lambda(1-\sigma)}{2}\right]N^3 + \frac{A}{2}e^{i\theta}N^2\right\}}{3N^4-4N^2+\lambda^2+1}. \tag{4.147}$$

Reducing gives

$$\frac{\partial N}{\partial \tau_1} + iN\frac{\partial \theta}{\partial \tau_1} = \frac{\left\{-i(1-\sigma)N^5 - [i\sigma-\lambda(1-\sigma)]N^3 - Ae^{-i\theta}N^2\right\}}{3N^4-4N^2+\lambda^2+1} + \frac{\left\{-\frac{(1-\sigma)}{2}N^3 - \left[\frac{\sigma}{2}+i\frac{\lambda(1-\sigma)}{2}\right]N + i\frac{A}{2}e^{-i\theta}\right\}(\lambda-i)}{3N^4-4N^2+\lambda^2+1} + \frac{\left\{-i\frac{(1-\sigma)}{2}N^5 - \left[i\frac{\sigma}{2}+\frac{\lambda(1-\sigma)}{2}\right]N^3 + \frac{A}{2}e^{i\theta}N^2\right\}}{3N^4-4N^2+\lambda^2+1}. \tag{4.148}$$

Further simplification gives

$$\frac{\partial N}{\partial \tau_1} + iN\frac{\partial \theta}{\partial \tau_1} = \frac{\left\{-\frac{(1-\sigma)}{2}N^3 - \left[\frac{\sigma}{2} + i\frac{\lambda(1-\sigma)}{2}\right]N + i\frac{A}{2}e^{-i\theta}\right\}(\lambda - i)}{3N^4 - 4N^2 + \lambda^2 + 1} + \frac{\left\{-i\frac{3(1-\sigma)}{2}N^5 - \left[i\frac{3\sigma}{2} - \frac{\lambda(1-\sigma)}{2}\right]N^3 + AN^2\left(\frac{1}{2}e^{i\theta} - e^{-i\theta}\right)\right\}}{3N^4 - 4N^2 + \lambda^2 + 1}. \tag{4.149}$$

Multiplying out terms gives

$$\frac{\partial N}{\partial \tau_1} + iN\frac{\partial \theta}{\partial \tau_1} = \frac{\left\{-\frac{\lambda(1-\sigma)}{2}N^3 - \lambda\left[\frac{\sigma}{2} + i\frac{\lambda(1-\sigma)}{2}\right]N + i\frac{\lambda A}{2}e^{-i\theta}\right\}}{3N^4 - 4N^2 + \lambda^2 + 1} + \frac{\left\{i\frac{(1-\sigma)}{2}N^3 - \left[-i\frac{\sigma}{2} + \frac{\lambda(1-\sigma)}{2}\right]N + \frac{A}{2}e^{-i\theta}\right\}}{3N^4 - 4N^2 + \lambda^2 + 1} + \frac{\left\{-i\frac{3(1-\sigma)}{2}N^5 - \left[i\frac{3\sigma}{2} - \frac{\lambda(1-\sigma)}{2}\right]N^3 + AN^2\left(\frac{1}{2}e^{i\theta} - e^{-i\theta}\right)\right\}}{3N^4 - 4N^2 + \lambda^2 + 1}. \tag{4.150}$$

Further reductions give

$$\frac{\partial N}{\partial \tau_1} + iN\frac{\partial \theta}{\partial \tau_1} = \frac{\left\{-\lambda(1-\sigma)N^3 - \lambda[\sigma + i\lambda(1-\sigma)]N + i\lambda Ae^{-i\theta}\right\}}{2(3N^4 - 4N^2 + \lambda^2 + 1)} + \frac{\left\{i(1-\sigma)N^3 - [-i\sigma + \lambda(1-\sigma)]N + Ae^{-i\theta}\right\}}{2(3N^4 - 4N^2 + \lambda^2 + 1)} + \frac{\left\{-i3(1-\sigma)N^5 - [i3\sigma - \lambda(1-\sigma)]N^3 + AN^2\left(e^{i\theta} - 2e^{-i\theta}\right)\right\}}{2(3N^4 - 4N^2 + \lambda^2 + 1)}. \tag{4.151}$$

Substituting

$$e^{i\theta} = \cos\theta + i\sin\theta, \tag{4.152}$$

and

$$e^{-i\theta} = \cos\theta - i\sin\theta \tag{4.153}$$

gives

$$\frac{\partial N}{\partial \tau_1} + iN\frac{\partial \theta}{\partial \tau_1} = \frac{\left\{-\lambda(1-\sigma)N^3 - \lambda[\sigma + i\lambda(1-\sigma)]N + i\lambda A(\cos\theta - i\sin\theta)\right\}}{2(3N^4 - 4N^2 + \lambda^2 + 1)} + \frac{\left\{i(1-\sigma)N^3 - [-i\sigma + \lambda(1-\sigma)]N + A(\cos\theta - i\sin\theta)\right\}}{2(3N^4 - 4N^2 + \lambda^2 + 1)} + \frac{\left\{-i3(1-\sigma)N^5 - [i3\sigma - \lambda(1-\sigma)]N^3 + AN^2(-\cos\theta + 3i\sin\theta)\right\}}{2(3N^4 - 4N^2 + \lambda^2 + 1)} \tag{4.154}$$

Separating into real and imaginary parts gives

$$\frac{\partial N}{\partial \tau_1} + iN\frac{\partial \theta}{\partial \tau_1} = \frac{\{-\lambda(1-\sigma)N^3-\lambda\sigma N+\lambda A\sin\theta-\lambda(1-\sigma)N+A\cos\theta+\lambda(1-\sigma)N^3-AN^2\cos\theta\}}{2(3N^4-4N^2+\lambda^2+1)} + i\frac{\{-\lambda^2(1-\sigma)N+\lambda A\cos\theta+(1-\sigma)N^3+\sigma N-A\sin\theta-3(1-\sigma)N^5-3\sigma N^3+3AN^2\sin\theta\}}{2(3N^4-4N^2+\lambda^2+1)}. \quad (4.155)$$

Reducing gives

$$\frac{\partial N}{\partial \tau_1} + iN\frac{\partial \theta}{\partial \tau_1} = \frac{\{\lambda A\sin\theta-\lambda N+A\cos\theta-AN^2\cos\theta\}}{2(3N^4-4N^2+\lambda^2+1)} + i\frac{\{\lambda A\cos\theta+(1-4\sigma)N^3+[\sigma-\lambda^2(1-\sigma)]N-A\sin\theta-3(1-\sigma)N^5+3AN^2\sin\theta\}}{2(3N^4-4N^2+\lambda^2+1)}. \quad (4.156)$$

Equating real parts of the equation gives

$$\frac{\partial N}{\partial \tau_1} = \frac{\lambda A\sin\theta-\lambda N+A\cos\theta-AN^2\cos\theta}{2(3N^4-4N^2+\lambda^2+1)}, \quad (4.157)$$

and equating imaginary parts gives

$$N\frac{\partial \theta}{\partial \tau_1} = \frac{\lambda A\cos\theta+(1-4\sigma)N^3+[\sigma-\lambda^2(1-\sigma)]N-A\sin\theta-3(1-\sigma)N^5+3AN^2\sin\theta}{2(3N^4-4N^2+\lambda^2+1)}. \quad (4.158)$$

Final manipulations give

$$\frac{\partial N}{\partial \tau_1} = \frac{-AN^2\cos\theta-\lambda N+\lambda A\sin\theta+A\cos\theta}{2(3N^4-4N^2+\lambda^2+1)}, \quad (4.159)$$

and

$$\frac{\partial \theta}{\partial \tau_1} = \frac{-3(1-\sigma)N^4+(1-4\sigma)N^2+3AN\sin\theta+[\sigma-\lambda^2(1-\sigma)]+(\lambda A\cos\theta-A\sin\theta)/N}{2(3N^4-4N^2+\lambda^2+1)}. \quad (4.160)$$

Letting *g(N)* represent the denominator in both differential equations, the fold lines occur when *g(N)* = 0. Thus, the equations can be rescaled by *g(N)* to avoid singularities as follows:

$$N' = -AN^2\cos\theta - \lambda N + \lambda A\sin\theta + A\cos\theta \quad (4.161)$$

and

$$\theta' = -3(1-\sigma)N^4 + (1-4\sigma)N^2 + 3AN\sin\theta + [\sigma - \lambda^2(1-\sigma)] + (\lambda A\cos\theta - A\sin\theta)/N. \quad (4.162)$$

Phase portraits were generated using the rescaled equations. The folded singularities may be obtained by setting $\theta' = 0$ and *g(N)* = 0, thus forcing the situation on the fold lines with unchanging θ. Rewriting gives

$$\theta' = \sigma(3N^4 - 4N^2 + \lambda^2 + 1) - 3N^4 + N^2 + 3AN\sin\theta - \lambda^2 + (\lambda A\cos\theta - A\sin\theta)/N. \quad (4.163)$$

Since

$$\frac{1}{2}g(N) = 3N^4 - 4N^2 + \lambda^2 + 1, \quad (4.164)$$

setting *g(N)* = 0 and $\theta' = 0$ results in

$$-3N^4 + N^2 + 3AN\sin\Theta_{1,2} - \lambda^2 + \left(\lambda A\cos\Theta_{1,2} - A\sin\Theta_{1,2}\right)/N = 0. \quad (4.165)$$

Rewriting gives

$$-3N^4 + N^2 + (3AN - A/N)\sin\Theta_{1,2} - \lambda^2 + \left(\lambda A\cos\Theta_{1,2}\right)/N = 0. \quad (4.166)$$

Since

$$\sin\Theta_{1,2} = \sqrt{1-\cos^2\Theta_{1,2}}, \quad (4.167)$$

the following may be written

$$-3N^4 + N^2 + (3AN - A/N)\sqrt{1-\cos^2\Theta_{1,2}} - \lambda^2 + \left(\lambda A\cos\Theta_{1,2}\right)/N = 0. \quad (4.168)$$

MATLAB was used to solve for Θ_1 and Θ_2 (see Section 4.A.4.2).

4.A.1.3 Derivation of Equations Related to 1-D Mapping (Linear Damping)

Since $C(\tau_1)$ is constant, equation

$$Z^3 - 2Z^2 + (\lambda^2 + 1)Z = 4|C(\tau_1)|^2 \tag{4.169}$$

may be rewritten as

$${Z_{1,2}}^3 - 2{Z_{1,2}}^2 + (\lambda^2 + 1)Z_{1,2} = {Z_{u,d}}^3 - 2{Z_{u,d}}^2 + (\lambda^2 + 1)Z_{u,d}. \tag{4.170}$$

MATLAB was used to determine $Z_{u,d}$ (see Section 4.A.4.3 for the MATLAB code used). Since one solution for $Z_{1,2}$ (the fold lines) was already determined, the MATLAB code returns the solutions that are not equivalent to $Z_{1,2}$. Interpretation of the MATLAB output gives

$$N_u = \sqrt{\frac{2}{3}\left(1 + \sqrt{1 - 3\lambda^2}\right)}, \tag{4.171}$$

and

$$N_d = \sqrt{\frac{2}{3}\left(1 - \sqrt{1 - 3\lambda^2}\right)}. \tag{4.172}$$

From

$$\left[\frac{\lambda}{2}N\cos\theta + \left(\frac{1}{2}N^3 - \frac{N}{2}\right)\sin\theta\right] + i\left[\frac{\lambda}{2}N\sin\theta - \left(\frac{1}{2}N^3 - \frac{N}{2}\right)\cos\theta\right] = C(\tau_1), \tag{4.173}$$

the argument of $C(\tau_1)$ can be expressed as

$$\arg C(\tau_1) = \tan^{-1}\left[\frac{\frac{\lambda}{2}N\sin\theta - \left(\frac{1}{2}N^3 - \frac{N}{2}\right)\cos\theta}{\frac{\lambda}{2}N\cos\theta + \left(\frac{1}{2}N^3 - \frac{N}{2}\right)\sin\theta}\right], \tag{4.174}$$

which can be rewritten as

$$\arg C(\tau_1) = \tan^{-1}\left[\frac{\lambda N\sin\theta - \left(N^3 - N\right)\cos\theta}{\lambda N\cos\theta + (N^3 - N)\sin\theta}\right]. \tag{4.175}$$

Since the coefficient N is common to the numerator and denominator in the inverse tangent expression, N may be factored out, leaving

$$\arg C(\tau_1) = \tan^{-1}\left[\frac{\lambda \sin\theta - (N^2-1)\cos\theta}{\lambda \cos\theta + (N^2-1)\sin\theta}\right]. \tag{4.176}$$

Dividing the numerator and denominator in the inverse tangent expression by $\lambda \cos\theta$ gives

$$\arg C(\tau_1) = \tan^{-1}\left[\frac{\tan\theta - \frac{(N^2-1)}{\lambda}}{1+\frac{(N^2-1)}{\lambda}\tan\theta}\right], \tag{4.177}$$

which can be rewritten as

$$\arg C(\tau_1) = \tan^{-1}\left[\frac{\tan\theta + \frac{(1-N^2)}{\lambda}}{1-\frac{(1-N^2)}{\lambda}\tan\theta}\right]. \tag{4.178}$$

Let

$$q_1 = \tan\theta, \tag{4.179}$$

and

$$q_2 = \frac{(1-N^2)}{\lambda}. \tag{4.180}$$

Then,

$$\arg C(\tau_1) = \tan^{-1}\left[\frac{q_1+q_2}{1-q_1 q_2}\right]. \tag{4.181}$$

From the formula of the "sum and difference of two inverse circular functions" given by Zwillinger (2003), the equation can be rewritten as

$$\arg C(\tau_1) = \tan^{-1} q_1 + \tan^{-1} q_2. \tag{4.182}$$

Thus,

$$\arg C(\tau_1) = \tan^{-1}(\tan\theta) + \tan^{-1}\left[\frac{(1-N^2)}{\lambda}\right]. \tag{4.183}$$

This equation can be reduced to

$$\arg C(\tau_1) = \theta(\tau_1) + \tan^{-1}\left[\frac{(1-N^2)}{\lambda}\right]. \tag{4.184}$$

Finally, the following can be written:

$$\theta(\tau_1) = \arg C(\tau_1) - \tan^{-1}\left[\frac{(1-N^2)}{\lambda}\right]. \tag{4.185}$$

Since

$$Z(\tau_1) = [N(\tau_1)]^2, \tag{4.186}$$

the relation becomes

$$\theta(\tau_1) = \arg C(\tau_1) - \tan^{-1}\left[\frac{1-Z(\tau_1)}{\lambda}\right]. \tag{4.187}$$

Since $C(\tau_1)$ is constant, the equation may be rewritten as

$$\arg C(\tau_1) = \theta_{01,02} + \tan^{-1}\left[\frac{1-Z_{1,2}}{\lambda}\right] = \theta_{u,d} + \tan^{-1}\left[\frac{1-Z_{u,d}}{\lambda}\right]. \tag{4.188}$$

Thus,

$$\theta_{u,d} = \theta_{01,02} + \tan^{-1}\left[\frac{1-Z_{1,2}}{\lambda}\right] - \tan^{-1}\left[\frac{1-Z_{u,d}}{\lambda}\right]. \tag{4.189}$$

Let

$$p_1 = \frac{1-Z_{1,2}}{\lambda}, \tag{4.190}$$

and

$$p_2 = \frac{1-Z_{u,d}}{\lambda}. \tag{4.191}$$

Then,

$$\theta_{u,d} = \theta_{01,02} + \tan^{-1} p_1 - \tan^{-1} p_2. \tag{4.192}$$

From the formula of the "sum and difference of two inverse circular functions" given by Zwillinger (2003), the equation can be rewritten as

$$\theta_{u,d} = \theta_{01,02} + \tan^{-1}\left[\frac{p_1-p_2}{1+p_1p_2}\right]. \tag{4.193}$$

Thus,

$$\theta_{u,d} = \theta_{01,02} + \tan^{-1}\left[\frac{\left(\frac{1-Z_{1,2}}{\lambda}\right)-\left(\frac{1-Z_{u,d}}{\lambda}\right)}{1+\left(\frac{1-Z_{1,2}}{\lambda}\right)\left(\frac{1-Z_{u,d}}{\lambda}\right)}\right]. \quad (4.194)$$

Reducing gives

$$\theta_{u,d} = \theta_{01,02} + \tan^{-1}\left[\frac{\frac{Z_{u,d}-Z_{1,2}}{\lambda}}{1+\frac{1}{\lambda^2}(1-Z_{1,2})(1-Z_{u,d})}\right]. \quad (4.195)$$

The equation reduces to

$$\theta_{u,d} = \theta_{01,02} + \tan^{-1}\left[\frac{(Z_{u,d}-Z_{1,2})\lambda}{\lambda^2+(1-Z_{1,2})(1-Z_{u,d})}\right], \quad (4.196)$$

and thus

$$\theta_{u,d} = \theta_{01,02} + \tan^{-1}\left[\frac{({N_{u,d}}^2-{N_{1,2}}^2)\lambda}{\lambda^2+(1-{N_{1,2}}^2)(1-{N_{u,d}}^2)}\right]. \quad (4.197)$$

Substituting in N_u and N_1 gives

$$\theta_u = \theta_{01} + \tan^{-1}\left[\frac{\left(\frac{2}{3}\left(1+\sqrt{1-3\lambda^2}\right)-\frac{2-\sqrt{1-3\lambda^2}}{3}\right)\lambda}{\lambda^2+\left(1-\frac{2-\sqrt{1-3\lambda^2}}{3}\right)\left(1-\frac{2}{3}\left(1+\sqrt{1-3\lambda^2}\right)\right)}\right]. \quad (4.198)$$

Reducing gives

$$\theta_u = \theta_{01} + \tan^{-1}\left[\frac{\lambda\sqrt{1-3\lambda^2}}{\lambda^2+1-\frac{2}{3}\left(1+\sqrt{1-3\lambda^2}\right)-\frac{2-\sqrt{1-3\lambda^2}}{3}+\frac{2}{3}\left(1+\sqrt{1-3\lambda^2}\right)\left(\frac{2-\sqrt{1-3\lambda^2}}{3}\right)}\right]. \quad (4.199)$$

This equation can be reduced to

$$\theta_u = \theta_{01} + \tan^{-1}\left[\frac{\lambda\sqrt{1-3\lambda^2}}{\lambda^2+1-\frac{4}{3}-\frac{1}{3}\sqrt{1-3\lambda^2}+\frac{2}{9}\left(1+\sqrt{1-3\lambda^2}\right)\left(2-\sqrt{1-3\lambda^2}\right)}\right]. \quad (4.200)$$

Simplification leads to

$$\theta_u = \theta_{01} + \tan^{-1}\left[\frac{\lambda\sqrt{1-3\lambda^2}}{\lambda^2-\frac{1}{3}-\frac{1}{3}\sqrt{1-3\lambda^2}+\frac{2}{9}\left(2+\sqrt{1-3\lambda^2}-1+3\lambda^2\right)}\right]. \quad (4.201)$$

The equation can be rewritten as

$$\theta_u = \theta_{01} + \tan^{-1}\left[\frac{9\lambda\sqrt{1-3\lambda^2}}{9\lambda^2-3-3\sqrt{1-3\lambda^2}+2\left(1+\sqrt{1-3\lambda^2}+3\lambda^2\right)}\right]. \tag{4.202}$$

Simplifying gives

$$\theta_u = \theta_{01} + \tan^{-1}\left[\frac{9\lambda\sqrt{1-3\lambda^2}}{15\lambda^2-1-\sqrt{1-3\lambda^2}}\right]. \tag{4.203}$$

Substituting in N_d and N_2 gives

$$\theta_d = \theta_{02} + \tan^{-1}\left[\frac{\left(\frac{2}{3}\left(1-\sqrt{1-3\lambda^2}\right)-\frac{2+\sqrt{1-3\lambda^2}}{3}\right)\lambda}{\lambda^2+\left(1-\frac{2+\sqrt{1-3\lambda^2}}{3}\right)\left(1-\frac{2}{3}\left(1-\sqrt{1-3\lambda^2}\right)\right)}\right]. \tag{4.204}$$

Reducing gives

$$\theta_d = \theta_{02} + \tan^{-1}\left[\frac{-\lambda\sqrt{1-3\lambda^2}}{\lambda^2+1-\frac{2}{3}\left(1-\sqrt{1-3\lambda^2}\right)-\frac{2+\sqrt{1-3\lambda^2}}{3}+\frac{2}{3}\left(1-\sqrt{1-3\lambda^2}\right)\left(\frac{2+\sqrt{1-3\lambda^2}}{3}\right)}\right]. \tag{4.205}$$

This equation can be reduced to

$$\theta_d = \theta_{02} + \tan^{-1}\left[\frac{-\lambda\sqrt{1-3\lambda^2}}{\lambda^2+1-\frac{4}{3}+\frac{1}{3}\sqrt{1-3\lambda^2}+\frac{2}{9}\left(1-\sqrt{1-3\lambda^2}\right)\left(2+\sqrt{1-3\lambda^2}\right)}\right]. \tag{4.206}$$

Simplification leads to

$$\theta_d = \theta_{02} + \tan^{-1}\left[\frac{-\lambda\sqrt{1-3\lambda^2}}{\lambda^2-\frac{1}{3}+\frac{1}{3}\sqrt{1-3\lambda^2}+\frac{2}{9}\left(2-\sqrt{1-3\lambda^2}-1+3\lambda^2\right)}\right]. \tag{4.207}$$

The equation can be rewritten as

$$\theta_d = \theta_{02} + \tan^{-1}\left[\frac{-9\lambda\sqrt{1-3\lambda^2}}{9\lambda^2-3+3\sqrt{1-3\lambda^2}+2\left(1-\sqrt{1-3\lambda^2}+3\lambda^2\right)}\right]. \tag{4.208}$$

Simplifying gives

$$\theta_d = \theta_{02} + \tan^{-1}\left[\frac{-9\lambda\sqrt{1-3\lambda^2}}{15\lambda^2-1+\sqrt{1-3\lambda^2}}\right]. \tag{4.209}$$

Using the properties of inverse tangent as given by Zwillinger (2003),

$$\theta_d = \theta_{02} - \tan^{-1}\left[\frac{9\lambda\sqrt{1-3\lambda^2}}{15\lambda^2-1+\sqrt{1-3\lambda^2}}\right]. \tag{4.210}$$

4.A.2 SMR Equation Derivations (Nonlinear Damping)

4.A.2.1 Derivation of Equations Related to SIM Projection (Nonlinear Damping)

Beginning with equations (3.20),

$$\dot{\varphi_1} + \frac{i\varepsilon}{2(1+\varepsilon)}(\varphi_1 - \varphi_2) - \frac{i\varepsilon\sigma}{2(1+\varepsilon)}(\varphi_1 + \varepsilon\varphi_2) = \frac{\varepsilon A}{2}$$

and (4.211)

$$\dot{\varphi_2} + \frac{3}{8}\lambda(1+\varepsilon)|\varphi_2|^2\varphi_2 + \frac{i}{2(1+\varepsilon)}(\varphi_2 - \varphi_1) - \frac{i\varepsilon\sigma}{2(1+\varepsilon)}(\varphi_1 + \varepsilon\varphi_2) - \frac{i}{2}(1+\varepsilon)|\varphi_2|^2\varphi_2 = \frac{\varepsilon A}{2},$$

the second equation can be rewritten as

$$-\frac{i}{2(1+\varepsilon)}\varphi_1 - \frac{i\varepsilon\sigma}{2(1+\varepsilon)}\varphi_1 = \frac{\varepsilon A}{2} - \dot{\varphi_2} - \frac{3}{8}\lambda(1+\varepsilon)|\varphi_2|^2\varphi_2 + \frac{i}{2}(1+\varepsilon)|\varphi_2|^2\varphi_2 - \frac{i}{2(1+\varepsilon)}\varphi_2 + \frac{i\varepsilon^2\sigma}{2(1+\varepsilon)}\varphi_2. \tag{4.212}$$

The equation can be reduced to

$$-\frac{i(1+\varepsilon\sigma)}{2(1+\varepsilon)}\varphi_1 = \frac{\varepsilon A}{2} - \dot{\varphi_2} - \frac{3}{8}\lambda(1+\varepsilon)|\varphi_2|^2\varphi_2 + \frac{i}{2}(1+\varepsilon)|\varphi_2|^2\varphi_2 + \frac{i(\varepsilon^2\sigma-1)}{2(1+\varepsilon)}\varphi_2. \tag{4.213}$$

Solving for φ_1 gives

$$\varphi_1 = i\frac{\varepsilon A(1+\varepsilon)}{1+\varepsilon\sigma} - i\frac{2(1+\varepsilon)}{1+\varepsilon\sigma}\dot{\varphi_2} - i\frac{3\lambda(1+\varepsilon)^2}{4(1+\varepsilon\sigma)}|\varphi_2|^2\varphi_2 - \frac{(1+\varepsilon)^2}{1+\varepsilon\sigma}|\varphi_2|^2\varphi_2 - \frac{\varepsilon^2\sigma-1}{1+\varepsilon\sigma}\varphi_2. \tag{4.214}$$

Substituting into the first of equations (4.211) gives

$$\frac{d}{dt}\left[i\frac{\varepsilon A(1+\varepsilon)}{1+\varepsilon\sigma}-i\frac{2(1+\varepsilon)}{1+\varepsilon\sigma}\dot{\varphi}_2-i\frac{3\lambda(1+\varepsilon)^2}{4(1+\varepsilon\sigma)}|\varphi_2|^2\varphi_2-\frac{(1+\varepsilon)^2}{1+\varepsilon\sigma}|\varphi_2|^2\varphi_2-\frac{\varepsilon^2\sigma-1}{1+\varepsilon\sigma}\varphi_2\right]+ \tag{4.215}$$

$$\frac{i\varepsilon}{2(1+\varepsilon)}\left[i\frac{\varepsilon A(1+\varepsilon)}{1+\varepsilon\sigma}-i\frac{2(1+\varepsilon)}{1+\varepsilon\sigma}\dot{\varphi}_2-i\frac{3\lambda(1+\varepsilon)^2}{4(1+\varepsilon\sigma)}|\varphi_2|^2\varphi_2-\frac{(1+\varepsilon)^2}{1+\varepsilon\sigma}|\varphi_2|^2\varphi_2-\right.$$

$$\left.\frac{\varepsilon^2\sigma-1}{1+\varepsilon\sigma}\varphi_2-\varphi_2\right]-\frac{i\varepsilon\sigma}{2(1+\varepsilon)}\left[i\frac{\varepsilon A(1+\varepsilon)}{1+\varepsilon\sigma}-i\frac{2(1+\varepsilon)}{1+\varepsilon\sigma}\dot{\varphi}_2-i\frac{3\lambda(1+\varepsilon)^2}{4(1+\varepsilon\sigma)}|\varphi_2|^2\varphi_2-\right.$$

$$\left.\frac{(1+\varepsilon)^2}{1+\varepsilon\sigma}|\varphi_2|^2\varphi_2-\frac{\varepsilon^2\sigma-1}{1+\varepsilon\sigma}\varphi_2+\varepsilon\varphi_2\right]=\frac{\varepsilon A}{2}.$$

This expression can be reduced to

$$-i\frac{2(1+\varepsilon)}{1+\varepsilon\sigma}\frac{d^2\varphi_2}{dt^2}+\frac{d}{dt}\left[i\frac{\varepsilon A(1+\varepsilon)}{1+\varepsilon\sigma}-i\frac{3\lambda(1+\varepsilon)^2}{4(1+\varepsilon\sigma)}|\varphi_2|^2\varphi_2-\frac{(1+\varepsilon)^2}{1+\varepsilon\sigma}|\varphi_2|^2\varphi_2-\right. \tag{4.216}$$

$$\left.\frac{\varepsilon^2\sigma-1}{1+\varepsilon\sigma}\varphi_2\right]+\left[-\frac{\varepsilon^2 A}{2(1+\varepsilon\sigma)}+\frac{\varepsilon}{1+\varepsilon\sigma}\dot{\varphi}_2+\frac{3\lambda\varepsilon(1+\varepsilon)}{8(1+\varepsilon\sigma)}|\varphi_2|^2\varphi_2-i\frac{\varepsilon(1+\varepsilon)}{2(1+\varepsilon\sigma)}|\varphi_2|^2\varphi_2-\right.$$

$$\left.i\frac{\varepsilon^3\sigma-\varepsilon}{2(1+\varepsilon\sigma)(1+\varepsilon)}\varphi_2-i\frac{\varepsilon\varphi_2}{2(1+\varepsilon)}\right]+\left[\frac{\varepsilon^2\sigma A}{2(1+\varepsilon\sigma)}-\frac{\varepsilon\sigma}{1+\varepsilon\sigma}\dot{\varphi}_2-\frac{3\lambda\varepsilon\sigma(1+\varepsilon)}{8(1+\varepsilon\sigma)}|\varphi_2|^2\varphi_2+\right.$$

$$\left.i\frac{\varepsilon\sigma(1+\varepsilon)}{2(1+\varepsilon\sigma)}|\varphi_2|^2\varphi_2+i\frac{\varepsilon\sigma(\varepsilon^2\sigma-1)}{2(1+\varepsilon\sigma)(1+\varepsilon)}\varphi_2-i\frac{\varepsilon^2\sigma}{2(1+\varepsilon)}\varphi_2\right]=\frac{\varepsilon A}{2}.$$

Since $\frac{i\varepsilon A(1+\varepsilon)}{1+\varepsilon\sigma}$ is not dependent on time, the derivative of this term with respect to time is zero. Thus, this term may be removed from the bracketed expression corresponding to the first derivative with respect to time. Taking this simplification into account and further reducing the system gives

$$-i\frac{2(1+\varepsilon)}{1+\varepsilon\sigma}\frac{d^2\varphi_2}{dt^2}+\frac{d}{dt}\left[-\frac{(3\lambda i+4)(1+\varepsilon)^2}{4(1+\varepsilon\sigma)}|\varphi_2|^2\varphi_2-\frac{\varepsilon^2\sigma-1}{1+\varepsilon\sigma}\varphi_2\right]- \tag{4.217}$$

$$\frac{\varepsilon^2 A(1-\sigma)}{2(1+\varepsilon\sigma)}+\frac{\varepsilon(1-\sigma)}{1+\varepsilon\sigma}\dot{\varphi}_2+\frac{3\lambda\varepsilon(1+\varepsilon)(1-\sigma)-i4\varepsilon(1+\varepsilon)}{8(1+\varepsilon\sigma)}|\varphi_2|^2\varphi_2-i\frac{\varepsilon(1+\varepsilon\sigma)}{2(1+\varepsilon)}\varphi_2+$$

$$i\frac{\varepsilon\sigma(1+\varepsilon)}{2(1+\varepsilon\sigma)}|\varphi_2|^2\varphi_2+i\frac{\varepsilon\sigma(\varepsilon^2\sigma-1)-\varepsilon^3\sigma+\varepsilon}{2(1+\varepsilon\sigma)(1+\varepsilon)}\varphi_2=\frac{\varepsilon A}{2}.$$

Taking into account that $\dot{\varphi}_2 = \frac{d\varphi_2}{dt}$ and reducing the system further gives

$$-i\frac{2(1+\varepsilon)}{1+\varepsilon\sigma}\frac{d^2\varphi_2}{dt^2} + \frac{d}{dt}\left[-\frac{(3\lambda i+4)(1+\varepsilon)^2}{4(1+\varepsilon\sigma)}|\varphi_2|^2\varphi_2 - \frac{\varepsilon^2\sigma-1}{1+\varepsilon\sigma}\varphi_2 + \frac{\varepsilon(1-\sigma)}{1+\varepsilon\sigma}\varphi_2\right] \tag{4.218}$$

$$-\frac{\varepsilon^2 A(1-\sigma)}{2(1+\varepsilon\sigma)} + \frac{3\lambda\varepsilon(1+\varepsilon)(1-\sigma)-i4\varepsilon(1+\varepsilon)}{8(1+\varepsilon\sigma)}|\varphi_2|^2\varphi_2 - i\frac{\varepsilon(1+\varepsilon\sigma)}{2(1+\varepsilon)}\varphi_2 + i\frac{\varepsilon\sigma(1+\varepsilon)}{2(1+\varepsilon\sigma)}|\varphi_2|^2\varphi_2 +$$

$$i\frac{\varepsilon\sigma(\varepsilon^2\sigma-1)-\varepsilon^3\sigma+\varepsilon}{2(1+\varepsilon\sigma)(1+\varepsilon)}\varphi_2 = \frac{\varepsilon A}{2}.$$

Again reducing the system yields

$$-i\frac{2(1+\varepsilon)}{1+\varepsilon\sigma}\frac{d^2\varphi_2}{dt^2} + \frac{d}{dt}\left[-\frac{(3\lambda i+4)(1+\varepsilon)^2}{4(1+\varepsilon\sigma)}|\varphi_2|^2\varphi_2 + \frac{\varepsilon(1-\sigma)-\varepsilon^2\sigma+1}{1+\varepsilon\sigma}\varphi_2\right] - \tag{4.219}$$

$$\frac{\varepsilon^2 A(1-\sigma)}{2(1+\varepsilon\sigma)} + \frac{3\lambda\varepsilon(1+\varepsilon)(1-\sigma)-i4\varepsilon(1+\varepsilon)}{8(1+\varepsilon\sigma)}|\varphi_2|^2\varphi_2 - i\frac{\varepsilon(1+\varepsilon\sigma)}{2(1+\varepsilon)}\varphi_2 + i\frac{\varepsilon\sigma(1+\varepsilon)}{2(1+\varepsilon\sigma)}|\varphi_2|^2\varphi_2 +$$

$$i\frac{\varepsilon\sigma(\varepsilon^2\sigma-1)-\varepsilon^3\sigma+\varepsilon}{2(1+\varepsilon\sigma)(1+\varepsilon)}\varphi_2 = \frac{\varepsilon A}{2}.$$

Multiplying by $\frac{i(1+\varepsilon\sigma)}{2(1+\varepsilon)}$ gives

$$\frac{d^2\varphi_2}{dt^2} + \frac{d}{dt}\left[-i\frac{(3\lambda i+4)(1+\varepsilon)}{8}|\varphi_2|^2\varphi_2 + i\frac{\varepsilon(1-\sigma)-\varepsilon^2\sigma+1}{2(1+\varepsilon)}\varphi_2\right] - i\frac{\varepsilon^2 A(1-\sigma)}{4(1+\varepsilon)} + \tag{4.220}$$

$$i\frac{3\lambda\varepsilon(1-\sigma)-i4\varepsilon}{16}|\varphi_2|^2\varphi_2 + \frac{\varepsilon(1+\varepsilon\sigma)^2}{4(1+\varepsilon)^2}\varphi_2 - \frac{\varepsilon\sigma}{4}|\varphi_2|^2\varphi_2 -$$

$$\frac{\varepsilon\sigma(\varepsilon^2\sigma-1)-\varepsilon^3\sigma+\varepsilon}{4(1+\varepsilon)^2}\varphi_2 = i\frac{\varepsilon A(1+\varepsilon\sigma)}{4(1+\varepsilon)}.$$

Since

$$\varphi_2 = \varphi_2(\tau_0, \tau_1, \ldots),\ \tau_k = \varepsilon^k t,\ k = 0,1,\ldots,$$
$$\frac{d}{dt} = \frac{\partial}{\partial\tau_0} + \varepsilon\frac{\partial}{\partial\tau_1} + \cdots = D_0 + \varepsilon D_1 + \cdots, \tag{4.221}$$

and

$$\frac{d^2}{dt^2} = \left(\frac{\partial}{\partial\tau_0} + \varepsilon\frac{\partial}{\partial\tau_1} + \cdots\right)^2 = (D_0 + \varepsilon D_1 + \cdots)^2, \tag{4.222}$$

the equation can be rewritten as

$$(D_0+\varepsilon D_1+\cdots)^2\varphi_2+(D_0+\varepsilon D_1+\cdots)\left[-i\frac{(3\lambda i+4)(1+\varepsilon)}{8}|\varphi_2|^2\varphi_2+\right.$$
$$\left.i\frac{\varepsilon(1-\sigma)-\varepsilon^2\sigma+1}{2(1+\varepsilon)}\varphi_2\right]-i\frac{\varepsilon^2A(1-\sigma)}{4(1+\varepsilon)}+i\frac{3\lambda\varepsilon(1-\sigma)-i4\varepsilon}{16}|\varphi_2|^2\varphi_2+\frac{\varepsilon(1+\varepsilon\sigma)^2}{4(1+\varepsilon)^2}\varphi_2- \tag{4.223}$$
$$\frac{\varepsilon\sigma}{4}|\varphi_2|^2\varphi_2-\frac{\varepsilon\sigma(\varepsilon^2\sigma-1)-\varepsilon^3\sigma+\varepsilon}{4(1+\varepsilon)^2}\varphi_2=i\frac{\varepsilon A(1+\varepsilon\sigma)}{4(1+\varepsilon)}.$$

Multiplying both sides by $(1+\varepsilon)^2$ gives

$$(D_0+\varepsilon D_1+\cdots)^2(1+\varepsilon)^2\varphi_2+(D_0+\varepsilon D_1+\cdots)\left[-i\frac{(3\lambda i+4)(1+\varepsilon)^3}{8}|\varphi_2|^2\varphi_2+\right.$$
$$\left.i\frac{[\varepsilon(1-\sigma)-\varepsilon^2\sigma+1](1+\varepsilon)}{2}\varphi_2\right]-i\frac{\varepsilon^2A(1-\sigma)(1+\varepsilon)}{4}+i\frac{[3\lambda\varepsilon(1-\sigma)-i4\varepsilon](1+\varepsilon)^2}{16}|\varphi_2|^2\varphi_2+ \tag{4.224}$$
$$\frac{\varepsilon(1+\varepsilon\sigma)^2}{4}\varphi_2-\frac{\varepsilon\sigma(1+\varepsilon)^2}{4}|\varphi_2|^2\varphi_2-\frac{\varepsilon\sigma(\varepsilon^2\sigma-1)-\varepsilon^3\sigma+\varepsilon}{4}\varphi_2=i\frac{\varepsilon A(1+\varepsilon\sigma)(1+\varepsilon)}{4}.$$

Expanding the equation and ignoring terms of ε^2 and above gives

$$\left({D_0}^2+2\varepsilon D_0D_1+\cdots\right)(1+2\varepsilon+\cdots)\varphi_2+ \tag{4.225}$$
$$(D_0+\varepsilon D_1+\cdots)\left[-i\frac{(3\lambda i+4)(1+3\varepsilon+\cdots)}{8}|\varphi_2|^2\varphi_2+i\frac{1+\varepsilon(2-\sigma)+\cdots}{2}\varphi_2\right]+$$
$$i\frac{\varepsilon[3\lambda(1-\sigma)-4i]+\cdots}{16}|\varphi_2|^2\varphi_2+\frac{\varepsilon+\cdots}{4}\varphi_2-\frac{\varepsilon\sigma+\cdots}{4}|\varphi_2|^2\varphi_2+\frac{\varepsilon\sigma-\varepsilon+\cdots}{4}\varphi_2=i\frac{\varepsilon A+\cdots}{4}.$$

Separating the equation based on powers of ε gives

$$\varepsilon^0\colon\ {D_0}^2\varphi_2-D_0\left[i\frac{(3\lambda i+4)}{8}|\varphi_2|^2\varphi_2-\frac{i}{2}\varphi_2\right]=0 \tag{4.226}$$

and

$$\varepsilon^1\colon\ 2{D_0}^2\varphi_2+2D_0D_1\varphi_2-D_0\left[i\frac{3(3\lambda i+4)}{8}|\varphi_2|^2\varphi_2-i\frac{2-\sigma}{2}\varphi_2\right]- \tag{4.227}$$
$$D_1\left[i\frac{(3\lambda i+4)}{8}|\varphi_2|^2\varphi_2-\frac{i}{2}\varphi_2\right]+i\frac{3\lambda(1-\sigma)-4i}{16}|\varphi_2|^2\varphi_2+\frac{1}{4}\varphi_2-$$
$$\frac{\sigma}{4}|\varphi_2|^2\varphi_2+\frac{\sigma-1}{4}\varphi_2=i\frac{A}{4}.$$

Integrating the equation of order ε^0 with respect to τ_0 gives

$$\frac{\partial}{\partial \tau_0}\varphi_2 - i\frac{(3\lambda i+4)}{8}|\varphi_2|^2\varphi_2 + \frac{i}{2}\varphi_2 = C(\tau_1, \ldots), \tag{4.228}$$

where $C(\tau_1, \ldots)$ is a result of the integration. The equation for the fixed points was obtained by omitting the derivative term, thus yielding

$$-i\frac{(3\lambda i+4)}{8}|\varphi_2|^2\varphi_2 + \frac{i}{2}\varphi_2 = C(\tau_1). \tag{4.229}$$

Since the analysis is concerned only with studying the equation with respect to τ_0 and τ_1, note that the fixed points φ_2 are functions only of τ_1. Substituting

$$\Phi(\tau_1) = \varphi_2 \tag{4.230}$$

and rearranging, the fixed point equation can be rewritten as

$$\frac{(3\lambda-4i)}{8}|\Phi|^2\Phi + \frac{i}{2}\Phi = C(\tau_1). \tag{4.231}$$

Letting

$$\Phi(\tau_1) = N(\tau_1)e^{i\theta(\tau_1)}, \tag{4.232}$$

and making the substitution, the equation becomes

$$\frac{(3\lambda-4i)}{8}N^3e^{i\theta} + \frac{i}{2}Ne^{i\theta} = C(\tau_1), \tag{4.233}$$

and thus

$$\left[\frac{(3\lambda-4i)}{8}N^3 + \frac{i}{2}N\right]e^{i\theta} = C(\tau_1). \tag{4.234}$$

Using the relation

$$e^{i\theta} = \cos\theta + i\sin\theta \tag{4.235}$$

and substituting, the equation can be rewritten as

$$\left[\frac{(3\lambda-4i)}{8}N^3 + \frac{i}{2}N\right](\cos\theta + i\sin\theta) = C(\tau_1). \tag{4.236}$$

Separating into real and imaginary parts gives

$$\left[\frac{(3\lambda\cos\theta+4\sin\theta)}{8}N^3-\frac{N}{2}\sin\theta\right]+i\left[\frac{(3\lambda\sin\theta-4\cos\theta)}{8}N^3+\frac{N}{2}\cos\theta\right]=C(\tau_1). \quad (4.237)$$

Taking the magnitude of both sides gives

$$\left[\frac{(3\lambda\cos\theta+4\sin\theta)}{8}N^3-\frac{N}{2}\sin\theta\right]^2+\left[\frac{(3\lambda\sin\theta-4\cos\theta)}{8}N^3+\frac{N}{2}\cos\theta\right]^2=|C(\tau_1)|^2. \quad (4.238)$$

Expanding gives

$$\begin{aligned}&\frac{(3\lambda\cos\theta+4\sin\theta)^2}{64}N^6-\frac{(3\lambda\cos\theta+4\sin\theta)\sin\theta}{8}N^4+\frac{N^2}{4}\sin^2\theta+\\&\frac{(3\lambda\sin\theta-4\cos\theta)^2}{64}N^6+\frac{(3\lambda\sin\theta-4\cos\theta)\cos\theta}{8}N^4+\frac{N^2}{4}\cos^2\theta=|C(\tau_1)|^2.\end{aligned} \quad (4.239)$$

Further expansion and reduction gives

$$\begin{aligned}&\frac{9\lambda^2\cos^2\theta+24\lambda\cos\theta\sin\theta+16\sin^2\theta}{64}N^6-\frac{\sin^2\theta}{2}N^4+\frac{N^2}{4}\sin^2\theta+\\&\frac{9\lambda^2\sin^2\theta-24\lambda\cos\theta\sin\theta+16\cos^2\theta}{64}N^6-\frac{\cos^2\theta}{2}N^4+\frac{N^2}{4}\cos^2\theta=|C(\tau_1)|^2.\end{aligned} \quad (4.240)$$

The equation can be reduced to

$$\begin{aligned}&\frac{9\lambda^2(\sin^2\theta+\cos^2\theta)+16(\sin^2\theta+\cos^2\theta)}{64}N^6-\frac{N^4}{2}(\sin^2\theta+\cos^2\theta)+\\&\frac{N^2}{4}(\sin^2\theta+\cos^2\theta)=|C(\tau_1)|^2.\end{aligned} \quad (4.241)$$

Using trig identities, the equation becomes

$$\frac{9\lambda^2+16}{64}N^6-\frac{N^4}{2}+\frac{N^2}{4}=|C(\tau_1)|^2. \quad (4.242)$$

Eliminating the denominators gives

$$(9\lambda^2+16)N^6-32N^4+16N^2=64|C(\tau_1)|^2. \quad (4.243)$$

Making the substitution,

$$Z(\tau_1)=[N(\tau_1)]^2, \quad (4.244)$$

yields

$$(9\lambda^2+16)Z^3-32Z^2+16Z=64|C(\tau_1)|^2. \quad (4.245)$$

Taking the derivative of the left hand side with respect to Z and setting equal to zero gives

$$3(9\lambda^2 + 16)Z^2 - 64Z + 16 = 0. \tag{4.246}$$

The derivative has the following roots:

$$Z_{1,2} = \frac{64 \mp \sqrt{4096 - 192(9\lambda^2 + 16)}}{6(9\lambda^2 + 16)}. \tag{4.247}$$

Reducing begins with

$$Z_{1,2} = \frac{64 \mp \sqrt{4096 - 3072 - 1728\lambda^2}}{6(9\lambda^2 + 16)}. \tag{4.248}$$

Simplifying gives

$$Z_{1,2} = \frac{64 \mp \sqrt{1024 - 1728\lambda^2}}{6(9\lambda^2 + 16)}, \tag{4.249}$$

which then becomes

$$Z_{1,2} = \frac{64 \mp \sqrt{64(16 - 27\lambda^2)}}{6(9\lambda^2 + 16)}. \tag{4.250}$$

The equation gives

$$Z_{1,2} = \frac{64 \mp 8\sqrt{16 - 27\lambda^2}}{6(9\lambda^2 + 16)}, \tag{4.251}$$

which can be further reduced to

$$Z_{1,2} = \frac{32 \mp 4\sqrt{16 - 27\lambda^2}}{3(9\lambda^2 + 16)}. \tag{4.252}$$

Finally, the equation becomes

$$Z_{1,2} = \frac{32 \mp 4\sqrt{16 - 27\lambda^2}}{27\lambda^2 + 48}. \tag{4.253}$$

Since

$$N(\tau_1) = \pm\sqrt{Z(\tau_1)}, \tag{4.254}$$

the positive values of N corresponding to the roots $Z_{1,2}$ can be written as

$$N_{1,2} = \sqrt{\frac{32 \mp 4\sqrt{16-27\lambda^2}}{27\lambda^2+48}}, \tag{4.255}$$

where N_1 and N_2 define the fold lines.

4.A.2.2 Derivation of Equations Related to Phase Portraits (Nonlinear Damping)

Continuing the analysis with the equation of $O(\varepsilon^1)$, taking the limit as τ_0 approaches $+\infty$ gives

$$-\frac{\partial}{\partial\tau_1}\left[i\frac{(3\lambda i+4)}{8}|\varphi_2|^2\varphi_2 - \frac{i}{2}\varphi_2\right] + i\frac{3\lambda(1-\sigma)-4i}{16}|\varphi_2|^2\varphi_2 + \frac{1}{4}\varphi_2 - \frac{\sigma}{4}|\varphi_2|^2\varphi_2 + \frac{\sigma-1}{4}\varphi_2 = i\frac{A}{4}, \tag{4.256}$$

with D_1 being replaced with the original notation. Note that the derivatives with respect to τ_0 drop out since a finite function is constant with respect to an infinite time. Reducing and again substituting

$$\Phi(\tau_1) = \varphi_2, \tag{4.257}$$

the equation becomes

$$\frac{\partial}{\partial\tau_1}\left[\frac{i}{2}\Phi - i\frac{(3\lambda i+4)}{8}|\Phi|^2\Phi\right] + i\frac{3\lambda(1-\sigma)-4i}{16}|\Phi|^2\Phi - \frac{\sigma}{4}|\Phi|^2\Phi + \frac{\sigma}{4}\Phi = i\frac{A}{4}. \tag{4.258}$$

Simplifying yields

$$\frac{\partial}{\partial\tau_1}\left[\frac{i}{2}\Phi + \frac{(3\lambda-4i)}{8}|\Phi|^2\Phi\right] + i\frac{3\lambda(1-\sigma)}{16}|\Phi|^2\Phi + \frac{1}{4}|\Phi|^2\Phi - \frac{\sigma}{4}|\Phi|^2\Phi + \frac{\sigma}{4}\Phi = i\frac{A}{4}. \tag{4.259}$$

Continuing with the simplification yields

$$\frac{\partial}{\partial\tau_1}\left[\frac{i}{2}\Phi + \frac{(3\lambda-4i)}{8}|\Phi|^2\Phi\right] + i\frac{3\lambda(1-\sigma)}{16}|\Phi|^2\Phi + \frac{1-\sigma}{4}|\Phi|^2\Phi + \frac{\sigma}{4}\Phi = i\frac{A}{4}, \tag{4.260}$$

which reduces further to

$$\frac{\partial}{\partial\tau_1}\left[\frac{i}{2}\Phi+\frac{(3\lambda-4i)}{8}|\Phi|^2\Phi\right]+\frac{(3\lambda i+4)(1-\sigma)}{16}|\Phi|^2\Phi+\frac{\sigma}{4}\Phi-i\frac{A}{4}=0. \tag{4.261}$$

Let

$$G=-\frac{(3\lambda i+4)(1-\sigma)}{16}|\Phi|^2\Phi-\frac{\sigma}{4}\Phi+i\frac{A}{4}, \tag{4.262}$$

then

$$\frac{\partial}{\partial\tau_1}\left[\frac{i}{2}\Phi+\frac{(3\lambda-4i)}{8}|\Phi|^2\Phi\right]=G. \tag{4.263}$$

Distributing the derivative gives

$$\left[\frac{i}{2}+\frac{(3\lambda-4i)}{8}|\Phi|^2\right]\frac{\partial\Phi}{\partial\tau_1}+\frac{(3\lambda-4i)}{8}\Phi\frac{\partial|\Phi|^2}{\partial\tau_1}=G. \tag{4.264}$$

But since

$$|\Phi|^2=\Phi\Phi^*, \tag{4.265}$$

the expression can be rewritten as

$$\left[\frac{i}{2}+\frac{(3\lambda-4i)}{8}|\Phi|^2\right]\frac{\partial\Phi}{\partial\tau_1}+\frac{(3\lambda-4i)}{8}\Phi\left[\frac{\partial}{\partial\tau_1}(\Phi\Phi^*)\right]=G. \tag{4.266}$$

Distributing the derivative gives

$$\left[\frac{i}{2}+\frac{(3\lambda-4i)}{8}|\Phi|^2\right]\frac{\partial\Phi}{\partial\tau_1}+\frac{(3\lambda-4i)}{8}\Phi\left[\Phi^*\frac{\partial\Phi}{\partial\tau_1}+\Phi\frac{\partial\Phi^*}{\partial\tau_1}\right]=G. \tag{4.267}$$

The equation can be reduced to

$$\left[\frac{i}{2}+\frac{(3\lambda-4i)}{8}|\Phi|^2\right]\frac{\partial\Phi}{\partial\tau_1}+\frac{(3\lambda-4i)}{8}|\Phi|^2\frac{\partial\Phi}{\partial\tau_1}+\frac{(3\lambda-4i)}{8}\Phi^2\frac{\partial\Phi^*}{\partial\tau_1}=G. \tag{4.268}$$

Finally, the equation simplifies to

$$\left[\frac{i}{2}+\frac{(3\lambda-4i)}{4}|\Phi|^2\right]\frac{\partial\Phi}{\partial\tau_1}+\frac{(3\lambda-4i)}{8}\Phi^2\frac{\partial\Phi^*}{\partial\tau_1}=G. \tag{4.269}$$

Taking the complex conjugate gives

$$\left[-\frac{i}{2}+\frac{(3\lambda+4i)}{4}|\Phi|^2\right]\frac{\partial\Phi^*}{\partial\tau_1}+\frac{(3\lambda+4i)}{8}\Phi^{*2}\frac{\partial\Phi}{\partial\tau_1}=G^*. \tag{4.270}$$

Solving for the complex conjugate derivative gives

$$\frac{\partial \Phi^*}{\partial \tau_1} = \frac{G^* - \frac{(3\lambda+4i)}{8}\Phi^{*2}\frac{\partial \Phi}{\partial \tau_1}}{-\frac{i}{2}+\frac{(3\lambda+4i)}{4}|\Phi|^2}. \tag{4.271}$$

Substituting into the expression above gives

$$\left[\frac{i}{2}+\frac{(3\lambda-4i)}{4}|\Phi|^2\right]\frac{\partial \Phi}{\partial \tau_1}+\frac{(3\lambda-4i)}{8}\Phi^2\left[\frac{G^* - \frac{(3\lambda+4i)}{8}\Phi^{*2}\frac{\partial \Phi}{\partial \tau_1}}{-\frac{i}{2}+\frac{(3\lambda+4i)}{4}|\Phi|^2}\right] = G. \tag{4.272}$$

Multiplying the equation by 8 gives

$$[4i+2(3\lambda-4i)|\Phi|^2]\frac{\partial \Phi}{\partial \tau_1}+(3\lambda-4i)\Phi^2\left[\frac{G^* - \frac{(3\lambda+4i)}{8}\Phi^{*2}\frac{\partial \Phi}{\partial \tau_1}}{-\frac{i}{2}+\frac{(3\lambda+4i)}{4}|\Phi|^2}\right] = 8G. \tag{4.273}$$

Simplifying the fraction in the bracketed term gives

$$[4i+2(3\lambda-4i)|\Phi|^2]\frac{\partial \Phi}{\partial \tau_1}+(3\lambda-4i)\Phi^2\left[\frac{8G^* - (3\lambda+4i)\Phi^{*2}\frac{\partial \Phi}{\partial \tau_1}}{-4i+2(3\lambda+4i)|\Phi|^2}\right] = 8G. \tag{4.274}$$

Further manipulation yields

$$\left\{4i+2(3\lambda-4i)|\Phi|^2-\left[\frac{(3\lambda-4i)(3\lambda+4i)\Phi^2\Phi^{*2}}{-4i+2(3\lambda+4i)|\Phi|^2}\right]\right\}\frac{\partial \Phi}{\partial \tau_1}+\left[\frac{8G^*(3\lambda-4i)\Phi^2}{-4i+2(3\lambda+4i)|\Phi|^2}\right] = 8G. \tag{4.275}$$

Solving for the derivative term gives

$$\frac{\partial \Phi}{\partial \tau_1} = \frac{8G-\left[\frac{8G^*(3\lambda-4i)\Phi^2}{-4i+2(3\lambda+4i)|\Phi|^2}\right]}{4i+2(3\lambda-4i)|\Phi|^2-\left[\frac{(3\lambda-4i)(3\lambda+4i)\Phi^2\Phi^{*2}}{-4i+2(3\lambda+4i)|\Phi|^2}\right]}. \tag{4.276}$$

Simplifying gives

$$\frac{\partial \Phi}{\partial \tau_1} = \frac{8G\left[-4i+2(3\lambda+4i)|\Phi|^2\right]-8G^*(3\lambda-4i)\Phi^2}{[4i+2(3\lambda-4i)|\Phi|^2][-4i+2(3\lambda+4i)|\Phi|^2]-(3\lambda-4i)(3\lambda+4i)\Phi^2\Phi^{*2}}. \tag{4.277}$$

Expanding the terms in the denominator gives

$$\frac{\partial \Phi}{\partial \tau_1} = \frac{8G\left[-4i+2(3\lambda+4i)|\Phi|^2\right]-8G^*(3\lambda-4i)\Phi^2}{16+8i(3\lambda+4i)|\Phi|^2-8i(3\lambda-4i)|\Phi|^2+4(9\lambda^2+16)|\Phi|^4-(9\lambda^2+16)\Phi^2\Phi^{*2}}. \tag{4.278}$$

Substituting in

$$|\Phi|^4 = \Phi^2\Phi^{*2} \tag{4.279}$$

and reducing gives

$$\frac{\partial \Phi}{\partial \tau_1} = \frac{16G\left[-2i+(3\lambda+4i)|\Phi|^2\right]-8G^*(3\lambda-4i)\Phi^2}{16+8i(3\lambda+4i)|\Phi|^2-8i(3\lambda-4i)|\Phi|^2+3(9\lambda^2+16)|\Phi|^4}. \tag{4.280}$$

Simplifying the denominator gives

$$\frac{\partial \Phi}{\partial \tau_1} = \frac{16G\left[-2i+(3\lambda+4i)|\Phi|^2\right]-8G^*(3\lambda-4i)\Phi^2}{16+8i|\Phi|^2(3\lambda+4i-3\lambda+4i)+3(9\lambda^2+16)|\Phi|^4}. \tag{4.281}$$

Finally, the expression can be written as

$$\frac{\partial \Phi}{\partial \tau_1} = \frac{16G\left[-2i+(3\lambda+4i)|\Phi|^2\right]-8G^*(3\lambda-4i)\Phi^2}{16-64|\Phi|^2+3(9\lambda^2+16)|\Phi|^4}. \tag{4.282}$$

Since

$$\Phi(\tau_1) = N(\tau_1)e^{i\theta(\tau_1)}, \tag{4.283}$$

the following can be written:

$$\frac{\partial \Phi}{\partial \tau_1} = \frac{\partial \mathrm{N}}{\partial \tau_1}e^{i\theta} + iNe^{i\theta}\frac{\partial \theta}{\partial \tau_1}. \tag{4.284}$$

Thus,

$$\frac{\partial \Phi}{\partial \tau_1} = \left(\frac{\partial \mathrm{N}}{\partial \tau_1} + iN\frac{\partial \theta}{\partial \tau_1}\right)e^{i\theta}. \tag{4.285}$$

Setting equal to the expression above gives

$$\left(\frac{\partial \mathrm{N}}{\partial \tau_1} + iN\frac{\partial \theta}{\partial \tau_1}\right)e^{i\theta} = \frac{16G\left[-2i+(3\lambda+4i)|\Phi|^2\right]-8G^*(3\lambda-4i)\Phi^2}{16-64|\Phi|^2+3(9\lambda^2+16)|\Phi|^4}. \tag{4.286}$$

Again substituting

$$\Phi(\tau_1) = N(\tau_1)e^{i\theta(\tau_1)} \tag{4.287}$$

gives

$$\left(\frac{\partial \mathrm{N}}{\partial \tau_1} + iN\frac{\partial \theta}{\partial \tau_1}\right)e^{i\theta} = \frac{16G\left[-2i+(3\lambda+4i)N^2\right]-8G^*(3\lambda-4i)N^2e^{2i\theta}}{16-64N^2+3(9\lambda^2+16)N^4}. \tag{4.288}$$

Moving the exponential term to the right hand side gives

$$\frac{\partial N}{\partial \tau_1} + iN\frac{\partial \theta}{\partial \tau_1} = \frac{16Ge^{-i\theta}\left[-2i+(3\lambda+4i)N^2\right]-8G^*e^{i\theta}(3\lambda-4i)N^2}{16-64N^2+3(9\lambda^2+16)N^4}. \tag{4.289}$$

Substituting

$$\Phi(\tau_1) = N(\tau_1)e^{i\theta(\tau_1)} \tag{4.290}$$

into the expression for G gives

$$G = -\frac{(3\lambda i+4)(1-\sigma)}{16}N^3 e^{i\theta} - \frac{\sigma}{4}Ne^{i\theta} + i\frac{A}{4}. \tag{4.291}$$

Thus,

$$G^* = -\frac{(-3\lambda i+4)(1-\sigma)}{16}N^3 e^{-i\theta} - \frac{\sigma}{4}Ne^{-i\theta} - i\frac{A}{4}. \tag{4.292}$$

Therefore,

$$Ge^{-i\theta} = -\frac{(3\lambda i+4)(1-\sigma)}{16}N^3 - \frac{\sigma}{4}N + i\frac{A}{4}e^{-i\theta}, \tag{4.293}$$

and

$$G^*e^{i\theta} = -\frac{(-3\lambda i+4)(1-\sigma)}{16}N^3 - \frac{\sigma}{4}N - i\frac{A}{4}e^{i\theta}. \tag{4.294}$$

Substituting into the differential equation gives

$$\frac{\partial N}{\partial \tau_1} + iN\frac{\partial \theta}{\partial \tau_1} = \tag{4.295}$$

$$\frac{16\left[-\frac{(3\lambda i+4)(1-\sigma)}{16}N^3 - \frac{\sigma}{4}N + i\frac{A}{4}e^{-i\theta}\right]\left[-2i+(3\lambda+4i)N^2\right] - 8\left[-\frac{(-3\lambda i+4)(1-\sigma)}{16}N^3 - \frac{\sigma}{4}N - i\frac{A}{4}e^{i\theta}\right](3\lambda-4i)N^2}{16-64N^2+3(9\lambda^2+16)N^4}.$$

Manipulating the expression gives

$$\frac{\partial N}{\partial \tau_1} + iN\frac{\partial \theta}{\partial \tau_1} = \tag{4.296}$$

$$\frac{16\left[\frac{2i(3\lambda i+4)(1-\sigma)}{16}N^3 + 2i\frac{\sigma}{4}N + \frac{A}{2}e^{-i\theta} - \frac{(3\lambda i+4)(3\lambda+4i)(1-\sigma)}{16}N^5 - \frac{\sigma(3\lambda+4i)}{4}N^3 + i\frac{A(3\lambda+4i)}{4}e^{-i\theta}N^2\right]}{16-64N^2+3(9\lambda^2+16)N^4} -$$

$$\frac{8\left[-\frac{(-3\lambda i+4)(3\lambda-4i)(1-\sigma)}{16}N^3 - \frac{\sigma(3\lambda-4i)}{4}N - i\frac{A(3\lambda-4i)}{4}e^{i\theta}\right]N^2}{16-64N^2+3(9\lambda^2+16)N^4}.$$

Reducing gives

$$\frac{\partial N}{\partial \tau_1} + iN\frac{\partial \theta}{\partial \tau_1} = \tag{4.297}$$

$$\frac{\left[2i(3\lambda i+4)(1-\sigma)N^3+8i\sigma N+8Ae^{-i\theta}-(3\lambda i+4)(3\lambda+4i)(1-\sigma)N^5-4\sigma(3\lambda+4i)N^3+4iA(3\lambda+4i)e^{-i\theta}N^2\right]}{16-64N^2+3(9\lambda^2+16)N^4}+$$

$$\frac{\left[\frac{(-3\lambda i+4)(3\lambda-4i)(1-\sigma)}{2}N^3+2\sigma(3\lambda-4i)N+2iA(3\lambda-4i)e^{i\theta}\right]N^2}{16-64N^2+3(9\lambda^2+16)N^4}.$$

Further simplification gives

$$\frac{\partial N}{\partial \tau_1} + iN\frac{\partial \theta}{\partial \tau_1} = \tag{4.298}$$

$$\frac{\left[2i(3\lambda i+4)(1-\sigma)N^3+8i\sigma N+8Ae^{-i\theta}-(3\lambda i+4)(3\lambda+4i)(1-\sigma)N^5-4\sigma(3\lambda+4i)N^3+4iA(3\lambda+4i)e^{-i\theta}N^2\right]}{16-64N^2+3(9\lambda^2+16)N^4}+$$

$$\frac{\left[(-3\lambda i+4)(3\lambda-4i)(1-\sigma)N^3+4\sigma(3\lambda-4i)N+4iA(3\lambda-4i)e^{i\theta}\right]N^2}{2[16-64N^2+3(9\lambda^2+16)N^4]}.$$

Multiplying out terms gives

$$\frac{\partial N}{\partial \tau_1} + iN\frac{\partial \theta}{\partial \tau_1} = \tag{4.299}$$

$$\frac{\left[(-6\lambda+8i)(1-\sigma)N^3+8i\sigma N+8Ae^{-i\theta}-\left(16i+9\lambda^2 i\right)(1-\sigma)N^5-4\sigma(3\lambda+4i)N^3+4A(3\lambda i-4)e^{-i\theta}N^2\right]}{16-64N^2+3(9\lambda^2+16)N^4}+$$

$$\frac{\left[\left(-9\lambda^2 i-16i\right)(1-\sigma)N^3+4\sigma(3\lambda-4i)N+4A(3\lambda i+4)e^{i\theta}\right]N^2}{2[16-64N^2+3(9\lambda^2+16)N^4]}.$$

Continuing with reducing the equation gives

$$\frac{\partial N}{\partial \tau_1} + iN\frac{\partial \theta}{\partial \tau_1} = \tag{4.300}$$

$$\frac{2\left[(-6\lambda+8i)(1-\sigma)N^3+8i\sigma N+8Ae^{-i\theta}-i\left(16+9\lambda^2\right)(1-\sigma)N^5-4\sigma(3\lambda+4i)N^3+4A(3\lambda i-4)e^{-i\theta}N^2\right]}{2[16-64N^2+3(9\lambda^2+16)N^4]}+$$

$$\frac{\left[-i\left(9\lambda^2+16\right)(1-\sigma)N^5+4\sigma(3\lambda-4i)N^3+4A(3\lambda i+4)e^{i\theta}N^2\right]}{2[16-64N^2+3(9\lambda^2+16)N^4]}.$$

Substituting

$$e^{i\theta} = \cos\theta + i\sin\theta \tag{4.301}$$

and

$$e^{-i\theta} = \cos\theta - i\sin\theta \tag{4.302}$$

gives

$$\frac{\partial N}{\partial \tau_1} + iN\frac{\partial \theta}{\partial \tau_1} = \tag{4.303}$$

$$\frac{2\left[(-6\lambda+8i)(1-\sigma)N^3+8i\sigma N+8A(\cos\theta-i\sin\theta)-i\left(16+9\lambda^2\right)(1-\sigma)N^5-4\sigma(3\lambda+4i)N^3+4A(3\lambda i-4)(\cos\theta-i\sin\theta)N^2\right]}{2[16-64N^2+3(9\lambda^2+16)N^4]} +$$

$$\frac{\left[-i\left(9\lambda^2+16\right)(1-\sigma)N^5+4\sigma(3\lambda-4i)N^3+4A(3\lambda i+4)(\cos\theta+i\sin\theta)N^2\right]}{2[16-64N^2+3(9\lambda^2+16)N^4]}.$$

Reducing again yields

$$\frac{\partial N}{\partial \tau_1} + iN\frac{\partial \theta}{\partial \tau_1} = \tag{4.304}$$

$$\frac{2[(-6\lambda+8i)(1-\sigma)N^3+8i\sigma N+8A(\cos\theta-i\sin\theta)-i(16+9\lambda^2)(1-\sigma)N^5-4\sigma(3\lambda+4i)N^3+4A(3\lambda i\cos\theta+3\lambda\sin\theta-4\cos\theta+4i\sin\theta)N^2]}{2[16-64N^2+3(9\lambda^2+16)N^4]} +$$

$$\frac{[-i(9\lambda^2+16)(1-\sigma)N^5+4\sigma(3\lambda-4i)N^3+4A(3\lambda i\cos\theta-3\lambda\sin\theta+4\cos\theta+4i\sin\theta)N^2]}{2[16-64N^2+3(9\lambda^2+16)N^4]}.$$

Distributing the 2 in the numerator leads to

$$\frac{\partial N}{\partial \tau_1} + iN\frac{\partial \theta}{\partial \tau_1} = \tag{4.305}$$

$$\frac{[(-12\lambda+16i)(1-\sigma)N^3+16i\sigma N+16A(\cos\theta-i\sin\theta)-i(32+18\lambda^2)(1-\sigma)N^5-4\sigma(6\lambda+8i)N^3+4A(6\lambda i\cos\theta+6\lambda\sin\theta-8\cos\theta+8i\sin\theta)N^2]}{2[16-64N^2+3(9\lambda^2+16)N^4]} +$$

$$\frac{[-i(9\lambda^2+16)(1-\sigma)N^5+4\sigma(3\lambda-4i)N^3+4A(3\lambda i\cos\theta-3\lambda\sin\theta+4\cos\theta+4i\sin\theta)N^2]}{2[16-64N^2+3(9\lambda^2+16)N^4]}.$$

Simplifying gives

$$\frac{\partial N}{\partial \tau_1} + iN\frac{\partial \theta}{\partial \tau_1} = \tag{4.306}$$

$$\frac{[(-12\lambda+16i)(1-\sigma)N^3+16i\sigma N+16A(\cos\theta-i\sin\theta)-i(48+27\lambda^2)(1-\sigma)N^5-4\sigma(3\lambda+12i)N^3+4A(9\lambda i\cos\theta+3\lambda\sin\theta-4\cos\theta+12i\sin\theta)N^2]}{2[16-64N^2+3(9\lambda^2+16)N^4]}.$$

Separating into real and imaginary parts gives

$$\frac{\partial N}{\partial \tau_1} + iN\frac{\partial \theta}{\partial \tau_1} = \tag{4.307}$$

$$\frac{\left[-12\lambda(1-\sigma)N^3+16A\cos\theta-12\sigma\lambda N^3+4A(3\lambda\sin\theta-4\cos\theta)N^2\right]}{2[16-64N^2+3(9\lambda^2+16)N^4]} +$$

$$i\frac{\left[16(1-\sigma)N^3+16\sigma N-16A\sin\theta-\left(48+27\lambda^2\right)(1-\sigma)N^5-48\sigma N^3+4A(9\lambda\cos\theta+12\sin\theta)N^2\right]}{2[16-64N^2+3(9\lambda^2+16)N^4]}.$$

Equating real parts of the equation gives

$$\frac{\partial N}{\partial \tau_1} = \frac{-12\lambda(1-\sigma)N^3+16A\cos\theta-12\sigma\lambda N^3+4A(3\lambda\sin\theta-4\cos\theta)N^2}{2[16-64N^2+3(9\lambda^2+16)N^4]}, \tag{4.308}$$

and equating imaginary parts gives

$$N\frac{\partial \theta}{\partial \tau_1} = \frac{16(1-\sigma)N^3+16\sigma N-16A\sin\theta-(48+27\lambda^2)(1-\sigma)N^5-48\sigma N^3+4A(9\lambda\cos\theta+12\sin\theta)N^2}{2N[16-64N^2+3(9\lambda^2+16)N^4]}. \tag{4.309}$$

Final simplifications give

$$\frac{\partial N}{\partial \tau_1} = \frac{-12\lambda N^3+4A(3\lambda\sin\theta-4\cos\theta)N^2+16A\cos\theta}{2[16-64N^2+3(9\lambda^2+16)N^4]}, \tag{4.310}$$

and

$$\frac{\partial \theta}{\partial \tau_1} = \frac{-(48+27\lambda^2)(1-\sigma)N^4+(16-64\sigma)N^2+4A(9\lambda\cos\theta+12\sin\theta)N+16\sigma-(16A\sin\theta)/N}{2[16-64N^2+3(9\lambda^2+16)N^4]}. \tag{4.311}$$

Letting *g(N)* represent the denominator in both differential equations, the fold lines occur when *g(N)* = 0. Thus, the equations can be rescaled by *g(N)* to avoid infinite responses as follows:

$$N' = -12\lambda N^3 + 4A(3\lambda\sin\theta - 4\cos\theta)N^2 + 16A\cos\theta, \tag{4.312}$$

and

$$\begin{aligned}\theta' = &-(48+27\lambda^2)(1-\sigma)N^4 + (16-64\sigma)N^2 + \\ &4A(9\lambda\cos\theta + 12\sin\theta)N + 16\sigma - (16A\sin\theta)/N.\end{aligned} \tag{4.313}$$

Phase portraits were generated using the rescaled equations. The folded singularities may be obtained by setting $\theta' = 0$ and *g(N)* = 0, thus forcing the situation on the fold lines with unchanging θ. Rewriting gives

$$\begin{aligned}\theta' = &\sigma[(48+27\lambda^2)N^4 - 64N^2 + 16] - (48+27\lambda^2)N^4 + \\ &16N^2 + 4A(9\lambda\cos\theta + 12\sin\theta)N - (16A\sin\theta)/N.\end{aligned} \tag{4.314}$$

Since

$$\frac{1}{2}g(N) = (48+27\lambda^2)N^4 - 64N^2 + 16, \tag{4.315}$$

setting $g(N) = 0$ and $\theta' = 0$ results in

$$-(48 + 27\lambda^2)N^4 + 16N^2 + 4A\left(9\lambda \cos\Theta_{1,2} + 12 \sin\Theta_{1,2}\right)N \\ -\left(16A \sin\Theta_{1,2}\right)/N = 0. \tag{4.316}$$

Rewriting gives

$$-(48 + 27\lambda^2)N^4 + 16N^2 + 36A\lambda N \cos\Theta_{1,2} + \\ (48AN - 16A/N) \sin\Theta_{1,2} = 0. \tag{4.317}$$

Since

$$\sin\Theta_{1,2} = \sqrt{1 - \cos^2\Theta_{1,2}}, \tag{4.318}$$

the following may be written

$$-(48 + 27\lambda^2)N^4 + 16N^2 + 36A\lambda N \cos\theta + \\ (48AN - 16A/N)\sqrt{1 - \cos^2\Theta_{1,2}} = 0. \tag{4.319}$$

MATLAB was used to solve for Θ_1 and Θ_2 (see Section 4.A.4.6).

4.A.2.3 Derivation of Equations Related to 1-D Mapping (Nonlinear Damping)

Since $C(\tau_1)$ is constant, equation

$$(9\lambda^2 + 16)Z^3 - 32Z^2 + 16Z = 64|C(\tau_1)|^2 \tag{4.320}$$

may be rewritten as

$$(9\lambda^2 + 16){Z_{1,2}}^3 - 32{Z_{1,2}}^2 + 16Z_{1,2} \\ = (9\lambda^2 + 16){Z_{u,d}}^3 - 32{Z_{u,d}}^2 + 16Z_{u,d}. \tag{4.321}$$

MATLAB was used to determine $Z_{u,d}$ (see Section 4.A.4.7 for the MATLAB code used). Since one solution for $Z_{1,2}$ (the fold lines) was already determined, the MATLAB code

returns the solutions that are not equivalent to $Z_{1,2}$. Interpretation of the MATLAB output gives

$$N_u = \sqrt{\frac{32+8\sqrt{16-27\lambda^2}}{27\lambda^2+48}}, \tag{4.322}$$

and

$$N_d = \sqrt{\frac{32-8\sqrt{16-27\lambda^2}}{27\lambda^2+48}}. \tag{4.323}$$

From

$$\left[\frac{(3\lambda\cos\theta + 4\sin\theta)}{8}N^3 - \frac{N}{2}\sin\theta\right] + \tag{4.324}$$

$$i\left[\frac{(3\lambda\sin\theta - 4\cos\theta)}{8}N^3 + \frac{N}{2}\cos\theta\right] = C(\tau_1),$$

the argument of $C(\tau_1)$ can be expressed as

$$\arg C(\tau_1) = \tan^{-1}\left[\frac{\frac{(3\lambda\sin\theta-4\cos\theta)}{8}N^3+\frac{N}{2}\cos\theta}{\frac{(3\lambda\cos\theta+4\sin\theta)}{8}N^3-\frac{N}{2}\sin\theta}\right], \tag{4.325}$$

which can be rewritten as

$$\arg C(\tau_1) = \tan^{-1}\left[\frac{(3\lambda\sin\theta-4\cos\theta)N^3+4N\cos\theta}{(3\lambda\cos\theta+4\sin\theta)N^3-4N\sin\theta}\right]. \tag{4.326}$$

Since the coefficient N is common to the numerator and denominator in the inverse tangent expression, N may be factored out, leaving

$$\arg C(\tau_1) = \tan^{-1}\left[\frac{(3\lambda\sin\theta-4\cos\theta)N^2+4\cos\theta}{(3\lambda\cos\theta+4\sin\theta)N^2-4\sin\theta}\right]. \tag{4.327}$$

Dividing the numerator and denominator in the inverse tangent expression by $\cos\theta$ gives

$$\arg C(\tau_1) = \tan^{-1}\left[\frac{(3\lambda\tan\theta-4)N^2+4}{(3\lambda+4\tan\theta)N^2-4\tan\theta}\right], \tag{4.328}$$

which can be rewritten as

$$\arg C(\tau_1) = \tan^{-1}\left[\frac{(3\lambda\tan\theta-4)+\frac{4}{N^2}}{(3\lambda+4\tan\theta)-\frac{4}{N^2}\tan\theta}\right]. \tag{4.329}$$

Reducing gives

$$\arg C(\tau_1) = \tan^{-1}\left[\frac{3\lambda\tan\theta+4\left(\frac{1}{N^2}-1\right)}{3\lambda-4\left(\frac{1}{N^2}-1\right)\tan\theta}\right]. \tag{4.330}$$

Dividing the numerator and denominator by 3λ gives

$$\arg C(\tau_1) = \tan^{-1}\left[\frac{\tan\theta+\frac{4}{3\lambda}\left(\frac{1}{N^2}-1\right)}{1-\frac{4}{3\lambda}\left(\frac{1}{N^2}-1\right)\tan\theta}\right]. \tag{4.331}$$

Let

$$q_1 = \tan\theta, \tag{4.332}$$

and

$$q_2 = \frac{4}{3\lambda}\left(\frac{1}{N^2}-1\right). \tag{4.333}$$

Then,

$$\arg C(\tau_1) = \tan^{-1}\left[\frac{q_1+q_2}{1-q_1q_2}\right]. \tag{4.334}$$

From the formula of the "sum and difference of two inverse circular functions" given by Zwillinger (2003), the equation can be rewritten as

$$\arg C(\tau_1) = \tan^{-1} q_1 + \tan^{-1} q_2. \tag{4.335}$$

Thus,

$$\arg C(\tau_1) = \tan^{-1}(\tan\theta) + \tan^{-1}\left[\frac{4}{3\lambda}\left(\frac{1}{N^2}-1\right)\right]. \tag{4.336}$$

The equation can be reduced to

$$\arg C(\tau_1) = \theta(\tau_1) + \tan^{-1}\left[\frac{4}{3\lambda}\left(\frac{1}{N^2}-1\right)\right]. \tag{4.337}$$

Finally, the following can be written:

$$\theta(\tau_1) = \arg C(\tau_1) - \tan^{-1}\left[\frac{4}{3\lambda}\left(\frac{1}{N^2}-1\right)\right]. \tag{4.338}$$

Since

$$Z(\tau_1) = [N(\tau_1)]^2, \tag{4.339}$$

the relation becomes

$$\theta(\tau_1) = \arg C(\tau_1) - \tan^{-1}\left[\frac{4}{3\lambda}\left(\frac{1}{Z(\tau_1)} - 1\right)\right]. \tag{4.340}$$

Since C(τ_1) is constant, the equation may be written as

$$\arg C(\tau_1) = \theta_{01,02} + \tan^{-1}\left[\frac{4}{3\lambda}\left(\frac{1}{Z_{1,2}} - 1\right)\right] = \theta_{u,d} + \tan^{-1}\left[\frac{4}{3\lambda}\left(\frac{1}{Z_{u,d}} - 1\right)\right]. \tag{4.341}$$

Thus,

$$\theta_{u,d} = \theta_{01,02} + \tan^{-1}\left[\frac{4}{3\lambda}\left(\frac{1}{Z_{1,2}} - 1\right)\right] - \tan^{-1}\left[\frac{4}{3\lambda}\left(\frac{1}{Z_{u,d}} - 1\right)\right]. \tag{4.342}$$

Let

$$p_1 = \frac{4}{3\lambda}\left(\frac{1}{Z_{1,2}} - 1\right), \tag{4.343}$$

and

$$p_2 = \frac{4}{3\lambda}\left(\frac{1}{Z_{u,d}} - 1\right). \tag{4.344}$$

Then,

$$\theta_{u,d} = \theta_{01,02} + \tan^{-1} p_1 - \tan^{-1} p_2. \tag{4.345}$$

From the formula of the "sum and difference of two inverse circular functions" given by Zwillinger (2003), the equation can be rewritten as

$$\theta_{u,d} = \theta_{01,02} + \tan^{-1}\left[\frac{p_1 - p_2}{1 + p_1 p_2}\right]. \tag{4.346}$$

Thus,

$$\theta_{u,d} = \theta_{01,02} + \tan^{-1}\left[\frac{\frac{4}{3\lambda}\left(\frac{1}{Z_{1,2}} - 1\right) - \frac{4}{3\lambda}\left(\frac{1}{Z_{u,d}} - 1\right)}{1 + \left[\frac{4}{3\lambda}\left(\frac{1}{Z_{1,2}} - 1\right)\right]\left[\frac{4}{3\lambda}\left(\frac{1}{Z_{u,d}} - 1\right)\right]}\right]. \tag{4.347}$$

Reducing gives

$$\theta_{u,d} = \theta_{01,02} + \tan^{-1}\left[\frac{\frac{4}{3\lambda}\left(\frac{1}{Z_{1,2}}-\frac{1}{Z_{u,d}}\right)}{1+\frac{16}{9\lambda^2}\left(\frac{1}{Z_{1,2}}-1\right)\left(\frac{1}{Z_{u,d}}-1\right)}\right]. \tag{4.348}$$

Multiplying the numerator and denominator by $9\lambda^2$ gives

$$\theta_{u,d} = \theta_{01,02} + \tan^{-1}\left[\frac{12\lambda\left(\frac{1}{Z_{1,2}}-\frac{1}{Z_{u,d}}\right)}{9\lambda^2+16\left(\frac{1}{Z_{1,2}}-1\right)\left(\frac{1}{Z_{u,d}}-1\right)}\right], \tag{4.349}$$

and thus

$$\theta_{u,d} = \theta_{01,02} + \tan^{-1}\left[\frac{12\lambda\left(\frac{1}{{N_{1,2}}^2}-\frac{1}{{N_{u,d}}^2}\right)}{9\lambda^2+16\left(\frac{1}{{N_{1,2}}^2}-1\right)\left(\frac{1}{{N_{u,d}}^2}-1\right)}\right]. \tag{4.350}$$

Substituting in N_u and N_1 gives

$$\theta_u = \theta_{01} + \tan^{-1}\left[\frac{12\lambda\left(\frac{1}{\frac{32-4\sqrt{16-27\lambda^2}}{27\lambda^2+48}}-\frac{1}{\frac{32+8\sqrt{16-27\lambda^2}}{27\lambda^2+48}}\right)}{9\lambda^2+16\left(\frac{1}{\frac{32-4\sqrt{16-27\lambda^2}}{27\lambda^2+48}}-1\right)\left(\frac{1}{\frac{32+8\sqrt{16-27\lambda^2}}{27\lambda^2+48}}-1\right)}\right], \tag{4.351}$$

which becomes

$$\theta_u = \theta_{01} + \tan^{-1}\left[\frac{12\lambda\left(\frac{27\lambda^2+48}{32-4\sqrt{16-27\lambda^2}}-\frac{27\lambda^2+48}{32+8\sqrt{16-27\lambda^2}}\right)}{9\lambda^2+16\left(\frac{27\lambda^2+48}{32-4\sqrt{16-27\lambda^2}}-1\right)\left(\frac{27\lambda^2+48}{32+8\sqrt{16-27\lambda^2}}-1\right)}\right]. \tag{4.352}$$

The equation can be rewritten as

$$\theta_u = \theta_{01} + \tan^{-1}\left[\frac{12\lambda\left(\frac{27\lambda^2+48}{32-4\sqrt{16-27\lambda^2}}-\frac{27\lambda^2+48}{32+8\sqrt{16-27\lambda^2}}\right)}{(9\lambda^2+16)+16\left[\left(\frac{27\lambda^2+48}{32-4\sqrt{16-27\lambda^2}}\right)\left(\frac{27\lambda^2+48}{32+8\sqrt{16-27\lambda^2}}\right)-\frac{27\lambda^2+48}{32-4\sqrt{16-27\lambda^2}}-\frac{27\lambda^2+48}{32+8\sqrt{16-27\lambda^2}}\right]}\right]. \tag{4.353}$$

Multiplying the numerator and denominator by $\left(32-4\sqrt{16-27\lambda^2}\right)\left(32+8\sqrt{16-27\lambda^2}\right)$ gives

$$\theta_u = \theta_{01} + \tan^{-1}\left[\frac{12\lambda\left[(27\lambda^2+48)(32+8\sqrt{16-27\lambda^2})-(27\lambda^2+48)(32-4\sqrt{16-27\lambda^2})\right]}{(9\lambda^2+16)(32-4\sqrt{16-27\lambda^2})(32+8\sqrt{16-27\lambda^2})+16\left[(27\lambda^2+48)^2-(27\lambda^2+48)(32+8\sqrt{16-27\lambda^2})-(27\lambda^2+48)(32-4\sqrt{16-27\lambda^2})\right]}\right]. \tag{4.354}$$

Reducing gives

$$\theta_u = \theta_{01} + \tan^{-1}\left[\frac{12\lambda(27\lambda^2+48)\left[(32+8\sqrt{16-27\lambda^2})-(32-4\sqrt{16-27\lambda^2})\right]}{(9\lambda^2+16)(32-4\sqrt{16-27\lambda^2})(32+8\sqrt{16-27\lambda^2})+16(27\lambda^2+48)^2-16(27\lambda^2+48)\left[(32+8\sqrt{16-27\lambda^2})+(32-4\sqrt{16-27\lambda^2})\right]}\right]. \quad (4.355)$$

Continuing with the reduction yields

$$\theta_u = \theta_{01} + \tan^{-1}\left[\frac{12\lambda(27\lambda^2+48)(12\sqrt{16-27\lambda^2})}{(9\lambda^2+16)(32-4\sqrt{16-27\lambda^2})(32+8\sqrt{16-27\lambda^2})+16(27\lambda^2+48)^2-16(27\lambda^2+48)(64+4\sqrt{16-27\lambda^2})}\right], \quad (4.356)$$

which leads to

$$\theta_u = \theta_{01} + \tan^{-1}\left[\frac{144\lambda(27\lambda^2+48)\sqrt{16-27\lambda^2}}{(9\lambda^2+16)\left[1024+128\sqrt{16-27\lambda^2}-32(16-27\lambda^2)\right]+16(27\lambda^2+48)^2-16(27\lambda^2+48)(64+4\sqrt{16-27\lambda^2})}\right]. \quad (4.357)$$

Simplifying gives

$$\theta_u = \theta_{01} + \tan^{-1}\left[\frac{144\lambda(27\lambda^2+48)\sqrt{16-27\lambda^2}}{(9\lambda^2+16)\left[1024+128\sqrt{16-27\lambda^2}-512+864\lambda^2\right]+16(27\lambda^2+48)^2-16(27\lambda^2+48)(64+4\sqrt{16-27\lambda^2})}\right], \quad (4.358)$$

becoming

$$\theta_u = \theta_{01} + \tan^{-1}\left[\frac{144\lambda(27\lambda^2+48)\sqrt{16-27\lambda^2}}{(9\lambda^2+16)\left[512+128\sqrt{16-27\lambda^2}+864\lambda^2\right]+16(27\lambda^2+48)^2-16(27\lambda^2+48)\left(64+4\sqrt{16-27\lambda^2}\right)}\right]. \quad (4.359)$$

Multiplying out terms gives

$$\theta_u = \theta_{01} + \tan^{-1}\left[\frac{144\lambda(27\lambda^2+48)\sqrt{16-27\lambda^2}}{\left[4608\lambda^2+1152\lambda^2\sqrt{16-27\lambda^2}+7776\lambda^4+8192+2048\sqrt{16-27\lambda^2}+13824\lambda^2\right]+16(27\lambda^2+48)^2-16\left(1728\lambda^2+108\lambda^2\sqrt{16-27\lambda^2}+3072+192\sqrt{16-27\lambda^2}\right)}\right], \quad (4.360)$$

and then

$$\theta_u = \theta_{01} + \tan^{-1}\left[\frac{144\lambda(27\lambda^2+48)\sqrt{16-27\lambda^2}}{4608\lambda^2+1152\lambda^2\sqrt{16-27\lambda^2}+7776\lambda^4+8192+2048\sqrt{16-27\lambda^2}+13824\lambda^2+16(27\lambda^2+48)^2-27648\lambda^2-1728\lambda^2\sqrt{16-27\lambda^2}-49152-3072\sqrt{16-27\lambda^2}}\right], \quad (4.361)$$

Reducing gives

$$\theta_u = \theta_{01} + \tan^{-1}\left[\frac{144\lambda(27\lambda^2+48)\sqrt{16-27\lambda^2}}{-9216\lambda^2-576\lambda^2\sqrt{16-27\lambda^2}+7776\lambda^4-40960-1024\sqrt{16-27\lambda^2}+16(27\lambda^2+48)^2}\right]. \quad (4.362)$$

Again multiplying out terms gives

$$\theta_u = \theta_{01} + \tan^{-1}\left[\frac{144\lambda(27\lambda^2+48)\sqrt{16-27\lambda^2}}{-9216\lambda^2-576\lambda^2\sqrt{16-27\lambda^2}+7776\lambda^4-40960-1024\sqrt{16-27\lambda^2}+16(729\lambda^4+2592\lambda^2+2304)^2}\right], \quad (4.363)$$

which becomes

$$\theta_u = \theta_{01} + \tan^{-1}\left[\frac{144\lambda(27\lambda^2+48)\sqrt{16-27\lambda^2}}{-9216\lambda^2-576\lambda^2\sqrt{16-27\lambda^2}+7776\lambda^4-40960-1024\sqrt{16-27\lambda^2}+11664\lambda^4+41472\lambda^2+36864}\right]. \quad (4.364)$$

Making another reduction gives

$$\theta_u = \theta_{01} + \tan^{-1}\left[\frac{144\lambda(27\lambda^2+48)\sqrt{16-27\lambda^2}}{32256\lambda^2-576\lambda^2\sqrt{16-27\lambda^2}+19440\lambda^4-4096-1024\sqrt{16-27\lambda^2}}\right], \tag{4.365}$$

which then leads to

$$\theta_u = \theta_{01} + \tan^{-1}\left[\frac{144\lambda(27\lambda^2+48)\sqrt{16-27\lambda^2}}{19440\lambda^4+576\lambda^2\left(56-\sqrt{16-27\lambda^2}\right)-1024\sqrt{16-27\lambda^2}-4096}\right]. \tag{4.366}$$

Dividing the numerator and denominator by 16 gives

$$\theta_u = \theta_{01} + \tan^{-1}\left[\frac{9\lambda(27\lambda^2+48)\sqrt{16-27\lambda^2}}{1215\lambda^4+36\lambda^2\left(56-\sqrt{16-27\lambda^2}\right)-64\sqrt{16-27\lambda^2}-256}\right]. \tag{4.367}$$

Final reductions give

$$\theta_u = \theta_{01} + \tan^{-1}\left[\frac{27\lambda(9\lambda^2+16)\sqrt{16-27\lambda^2}}{1215\lambda^4+36\lambda^2\left(56-\sqrt{16-27\lambda^2}\right)-64\sqrt{16-27\lambda^2}-256}\right]. \tag{4.368}$$

Substituting in N_d and N_2 gives

$$\theta_d = \theta_{02} + \tan^{-1}\left[\frac{12\lambda\left(\frac{1}{\frac{32+4\sqrt{16-27\lambda^2}}{27\lambda^2+48}}-\frac{1}{\frac{32-8\sqrt{16-27\lambda^2}}{27\lambda^2+48}}\right)}{9\lambda^2+16\left(\frac{1}{\frac{32+4\sqrt{16-27\lambda^2}}{27\lambda^2+48}}-1\right)\left(\frac{1}{\frac{32-8\sqrt{16-27\lambda^2}}{27\lambda^2+48}}-1\right)}\right], \tag{4.369}$$

which becomes

$$\theta_d = \theta_{02} + \tan^{-1}\left[\frac{12\lambda\left(\frac{27\lambda^2+48}{32+4\sqrt{16-27\lambda^2}}-\frac{27\lambda^2+48}{32-8\sqrt{16-27\lambda^2}}\right)}{9\lambda^2+16\left(\frac{27\lambda^2+48}{32+4\sqrt{16-27\lambda^2}}-1\right)\left(\frac{27\lambda^2+48}{32-8\sqrt{16-27\lambda^2}}-1\right)}\right]. \tag{4.370}$$

The equation can be rewritten as

$$\theta_d = \theta_{02} + \tan^{-1}\left[\frac{12\lambda\left(\frac{27\lambda^2+48}{32+4\sqrt{16-27\lambda^2}}-\frac{27\lambda^2+48}{32-8\sqrt{16-27\lambda^2}}\right)}{(9\lambda^2+16)+16\left[\left(\frac{27\lambda^2+48}{32+4\sqrt{16-27\lambda^2}}\right)\left(\frac{27\lambda^2+48}{32-8\sqrt{16-27\lambda^2}}\right)-\frac{27\lambda^2+48}{32+4\sqrt{16-27\lambda^2}}-\frac{27\lambda^2+48}{32-8\sqrt{16-27\lambda^2}}\right]}\right]. \tag{4.371}$$

Multiplying the numerator and denominator by $\left(32+4\sqrt{16-27\lambda^2}\right)\left(32-8\sqrt{16-27\lambda^2}\right)$ gives

$$\theta_d = \theta_{02} + \tan^{-1}\left[\frac{12\lambda\left[\left(27\lambda^2+48\right)\left(32-8\sqrt{16-27\lambda^2}\right)-\left(27\lambda^2+48\right)\left(32+4\sqrt{16-27\lambda^2}\right)\right]}{(9\lambda^2+16)\left(32+4\sqrt{16-27\lambda^2}\right)\left(32-8\sqrt{16-27\lambda^2}\right)+16\left[(27\lambda^2+48)^2-(27\lambda^2+48)\left(32-8\sqrt{16-27\lambda^2}\right)-(27\lambda^2+48)\left(32+4\sqrt{16-27\lambda^2}\right)\right]}\right]. \tag{4.372}$$

Reducing gives

$$\theta_d = \theta_{02} + \tan^{-1}\left[\frac{12\lambda(27\lambda^2+48)\left[(32-8\sqrt{16-27\lambda^2})-(32+4\sqrt{16-27\lambda^2})\right]}{(9\lambda^2+16)(32+4\sqrt{16-27\lambda^2})(32-8\sqrt{16-27\lambda^2})+16(27\lambda^2+48)^2-16(27\lambda^2+48)\left[(32-8\sqrt{16-27\lambda^2})+(32+4\sqrt{16-27\lambda^2})\right]}\right]. \quad (4.373)$$

Continuing with the reduction yields

$$\theta_d = \theta_{02} + \tan^{-1}\left[\frac{12\lambda(27\lambda^2+48)\left(-12\sqrt{16-27\lambda^2}\right)}{(9\lambda^2+16)\left(32+4\sqrt{16-27\lambda^2}\right)\left(32-8\sqrt{16-27\lambda^2}\right)+16(27\lambda^2+48)^2-16(27\lambda^2+48)\left(64-4\sqrt{16-27\lambda^2}\right)}\right], \quad (4.374)$$

which leads to

$$\theta_d = \theta_{02} + \tan^{-1}\left[\frac{-144\lambda(27\lambda^2+48)\sqrt{16-27\lambda^2}}{(9\lambda^2+16)\left[1024-128\sqrt{16-27\lambda^2}-32(16-27\lambda^2)\right]+16(27\lambda^2+48)^2-16(27\lambda^2+48)\left(64-4\sqrt{16-27\lambda^2}\right)}\right]. \quad (4.375)$$

Simplifying gives

$$\theta_d = \theta_{02} + \tan^{-1}\left[\frac{-144\lambda(27\lambda^2+48)\sqrt{16-27\lambda^2}}{(9\lambda^2+16)\left[1024-128\sqrt{16-27\lambda^2}-512+864\lambda^2\right]+16(27\lambda^2+48)^2-16(27\lambda^2+48)\left(64-4\sqrt{16-27\lambda^2}\right)}\right], \quad (4.376)$$

becoming

$$\theta_d = \theta_{02} + \tan^{-1}\left[\frac{-144\lambda(27\lambda^2+48)\sqrt{16-27\lambda^2}}{(9\lambda^2+16)\left[512-128\sqrt{16-27\lambda^2}+864\lambda^2\right]+16(27\lambda^2+48)^2-16(27\lambda^2+48)\left(64-4\sqrt{16-27\lambda^2}\right)}\right]. \quad (4.377)$$

Multiplying out terms gives

$$\theta_d = \theta_{02} + \tan^{-1}\left[\frac{-144\lambda(27\lambda^2+48)\sqrt{16-27\lambda^2}}{\left[4608\lambda^2-1152\lambda^2\sqrt{16-27\lambda^2}+7776\lambda^4+8192-2048\sqrt{16-27\lambda^2}+13824\lambda^2\right]+16(27\lambda^2+48)^2-16\left(1728\lambda^2-108\lambda^2\sqrt{16-27\lambda^2}+3072-192\sqrt{16-27\lambda^2}\right)}\right], \quad (4.378)$$

and then

$$\theta_d = \theta_{02} + \tan^{-1}\left[\frac{-144\lambda(27\lambda^2+48)\sqrt{16-27\lambda^2}}{4608\lambda^2-1152\lambda^2\sqrt{16-27\lambda^2}+7776\lambda^4+8192-2048\sqrt{16-27\lambda^2}+13824\lambda^2+16(27\lambda^2+48)^2-27648\lambda^2+1728\lambda^2\sqrt{16-27\lambda^2}-49152+3072\sqrt{16-27\lambda^2}}\right], \quad (4.379)$$

Reducing gives

$$\theta_d = \theta_{02} + \tan^{-1}\left[\frac{-144\lambda(27\lambda^2+48)\sqrt{16-27\lambda^2}}{-9216\lambda^2+576\lambda^2\sqrt{16-27\lambda^2}+7776\lambda^4-40960+1024\sqrt{16-27\lambda^2}+16(27\lambda^2+48)^2}\right]. \quad (4.380)$$

Again multiplying out terms gives

$$\theta_d = \theta_{02} + \tan^{-1}\left[\frac{-144\lambda(27\lambda^2+48)\sqrt{16-27\lambda^2}}{-9216\lambda^2+576\lambda^2\sqrt{16-27\lambda^2}+7776\lambda^4-40960+1024\sqrt{16-27\lambda^2}+16(729\lambda^4+2592\lambda^2+2304)^2}\right], \quad (4.381)$$

which becomes

$$\theta_d = \theta_{02} + \tan^{-1}\left[\frac{-144\lambda(27\lambda^2+48)\sqrt{16-27\lambda^2}}{-9216\lambda^2+576\lambda^2\sqrt{16-27\lambda^2}+7776\lambda^4-40960+1024\sqrt{16-27\lambda^2}+11664\lambda^4+41472\lambda^2+36864}\right]. \quad (4.382)$$

Making another reduction gives

$$\theta_d = \theta_{02} + \tan^{-1}\left[\frac{-144\lambda(27\lambda^2+48)\sqrt{16-27\lambda^2}}{32256\lambda^2+576\lambda^2\sqrt{16-27\lambda^2}+19440\lambda^4-4096+1024\sqrt{16-27\lambda^2}}\right], \quad (4.383)$$

which then leads to

$$\theta_d = \theta_{02} + \tan^{-1}\left[\frac{-144\lambda(27\lambda^2+48)\sqrt{16-27\lambda^2}}{19440\lambda^4+576\lambda^2(56+\sqrt{16-27\lambda^2})+1024\sqrt{16-27\lambda^2}-4096}\right]. \quad (4.384)$$

Dividing the numerator and denominator by 16 gives

$$\theta_d = \theta_{02} + \tan^{-1}\left[\frac{-9\lambda(27\lambda^2+48)\sqrt{16-27\lambda^2}}{1215\lambda^4+36\lambda^2(56+\sqrt{16-27\lambda^2})+64\sqrt{16-27\lambda^2}-256}\right]. \quad (4.385)$$

Final reductions give

$$\theta_d = \theta_{02} + \tan^{-1}\left[\frac{-27\lambda(9\lambda^2+16)\sqrt{16-27\lambda^2}}{1215\lambda^4+36\lambda^2(56+\sqrt{16-27\lambda^2})+64\sqrt{16-27\lambda^2}-256}\right]. \quad (4.386)$$

Using the properties of inverse tangent as given by Zwillinger (2003),

$$\theta_d = \theta_{02} - \tan^{-1}\left[\frac{27\lambda(9\lambda^2+16)\sqrt{16-27\lambda^2}}{1215\lambda^4+36\lambda^2(56+\sqrt{16-27\lambda^2})+64\sqrt{16-27\lambda^2}-256}\right]. \quad (4.387)$$

4.A.3 Method of Creating 1-D Maps

The 1-D maps were created using a four-step process. First, the trajectories between Θ_2 and Θ_1 on N_1 were connected to the corresponding trajectories on the upper branch using a line segment for each connection. In this interval, all of the trajectories jump from the lower stable branch at N_1 to the upper stable branch at N_u. The connection was made using equations (4.33) and (4.30) for θ_u and N_u, respectively.

Next, (θ_u, N_u) was set as the initial condition for the equations originally used to generate the phase portrait. Following the same steps used to generate the phase portrait and keeping only the trajectories above N_2 results in a connection from (θ_u, N_u) to corresponding values of θ on N_2.

These values of θ on N_2 were then connected to the corresponding trajectories on the lower branch, (θ_d, N_d), using a line segment for each connection. Refer to equations (4.34) and (4.31) for θ_d and N_d, respectively.

Finally, the trajectory is mapped back to the (Θ_2, Θ_1) interval by setting (θ_d, N_d) as the initial condition for the equations originally used to generate the phase portrait. Following the same steps used to generate the phase portrait and keeping only the trajectories below N_1 completes the mapping. The resultant 1-D map was generated by connecting the initial points on N_1 between between Θ_2 and Θ_1, and connecting to the return points on N_1 between between Θ_2 and Θ_1 using a line segment for each trajectory. An example of this map creation is shown in Figure 4-19 for the case of linear damping and $\sigma = 0$. The process of 1-D map creation is similar for the nonlinearly damped system.

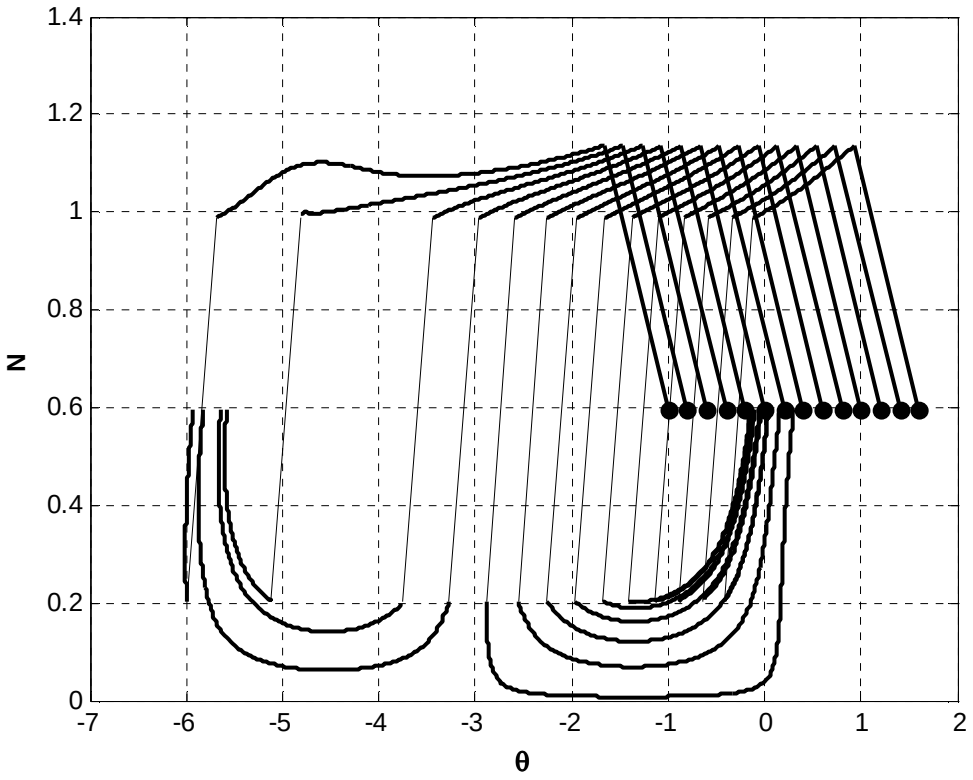


Figure 4-19. Creation of 1-D map for $A = 1$, $\lambda = 0.2$, and $\sigma = 0$ (linear damping)

4.A.4 MATLAB Code

4.A.4.1 Plotting SIM (Linear Damping)

```
%SIM Projection for the Linearly Damped System

close all
clear all

lambda = 0.2; k = 0; ks1 = 0; ks2 = 0; ku = 0;
N_end = 1.6; N_step = 0.0001;
C_sq_4 = zeros(1,N_end/N_step + 1);
N_vec = C_sq_4;

%Fold Lines (N1 and N2):

N1 = sqrt((2 - sqrt(1-3*lambda^2))/3);
N2 = sqrt((2 + sqrt(1-3*lambda^2))/3);
```

```
%Fold Line End Locations (Nu and Nd):

Nu = sqrt(2/3*(1 + sqrt(1-3*lambda^2)));
Nd = sqrt(2/3*(1 - sqrt(1-3*lambda^2)));

for N = 0:N_step:N_end
    k = k + 1;
    C_sq_4 = N^6 - 2*N^4 + (lambda^2+1)*N^2;

    if N < N1
        ks1 = ks1 + 1;
        C_sq_4_stable1(ks1) = C_sq_4;
        N_stable1(ks1) = N;
    elseif N > N2
        ks2 = ks2 + 1;
        C_sq_4_stable2(ks2) = C_sq_4;
        N_stable2(ks2) = N;
    else
        ku = ku + 1;
        C_sq_4_unstable(ku) = C_sq_4;
        N_unstable(ku) = N;
    end
end

C_u = [max(C_sq_4_unstable) max(C_sq_4_unstable)];
C_d = [min(C_sq_4_unstable) min(C_sq_4_unstable)];
fold_u = [N1 Nu]; fold_d = [Nd N2];

figure
hold on; xlabel('\bfN'); ylabel('\bf4|C|^2');
plot(N_stable1,C_sq_4_stable1,'-k','LineWidth',2)
plot(N_stable2,C_sq_4_stable2,'-k','LineWidth',2)
plot(N_unstable,C_sq_4_unstable,'--k','LineWidth',2)
plot(fold_u, C_u, ':k','LineWidth',2)
plot(fold_u(2)-.03, C_u(2), '>k','LineWidth',2,'MarkerFaceColor','k')
plot(fold_d, C_d, ':k','LineWidth',2)
plot(fold_d(1)+.03, C_d(1), '<k','LineWidth',2,'MarkerFaceColor','k')
text(N1, C_u(1)+.01, '\bfN_1')
text(N2, C_d(1)-.01, '\bfN_2')
text(Nu+.03, C_u(1), '\bfN_u')
text(Nd, C_d(1)-.01, '\bfN_d')
axis([0, N, 0, 0.2])
```

4.A.4.2 Determining Θ_1 and Θ_2 (Linear Damping)

```
%Chapter 4 - SMR
%Find THETA_1 and THETA_2 for the system with linear damping:

close all
clear all

lambda = 0.2; A = 1;
```

```
N1 = sqrt((2 - sqrt(1-3*lambda^2))/3); %Lower fold line

k = 0; n = 0;

for theta = -2*pi:.001:2*pi
    k = k + 1;
    q = -3*N1^4 + N1^2 + 3*A*N1*sin(theta) - lambda^2 + ...
        (lambda*A*cos(theta) - A*sin(theta))/N1;
    q_vec(k) = q;
    theta_vec(k) = theta;
    if abs(q) < .01
        n = n + 1;
        near_zero(n) = theta;
    else
    end
end

figure
plot(theta_vec,q_vec,'-k','LineWidth',2)
xlabel('\bf\theta','fontsize',12); ylabel('\bfN'); hold on;

syms c_theta

cos_vec = solve('-3*N1^4 + N1^2 + (3*A*N1 - A/N1)*sqrt(1 - c_theta^2) -
lambda^2 + lambda*A*c_theta/N1 = 0',c_theta);

ct1 = subs(cos_vec(1));
ct2 = subs(cos_vec(2));

theta_1_maybe = acos(ct1);
theta_2_maybe = acos(ct2);

for h = 1:length(near_zero)
    if abs(abs(near_zero(h)) - abs(theta_1_maybe)) < .1
        theta_1 = sign(near_zero(h)) * abs(theta_1_maybe);
    else
    end

    if abs(abs(near_zero(h)) - abs(theta_2_maybe)) < .1
        theta_2 = sign(near_zero(h)) * abs(theta_2_maybe);
    else
    end
end

theta_1_final = max(theta_1,theta_2);
theta_2_final = min(theta_1,theta_2);

theta_1 = theta_1_final
theta_2 = theta_2_final

plot([theta_1 theta_1],[-50 50],':k','LineWidth',2)
plot([theta_2 theta_2],[-50 50],':k','LineWidth',2)
plot([theta_vec(1) theta_vec(k)],[0 0],':k','LineWidth',2)
```

```
text(theta_2 - 0.65,0.1,'\bf\Theta_2','FontSize',12)
text(theta_1 + 0.15,0.1,'\bf\Theta_1','FontSize',12)
axis([theta_vec(1), theta_vec(k), -1.5, 1.5])
```

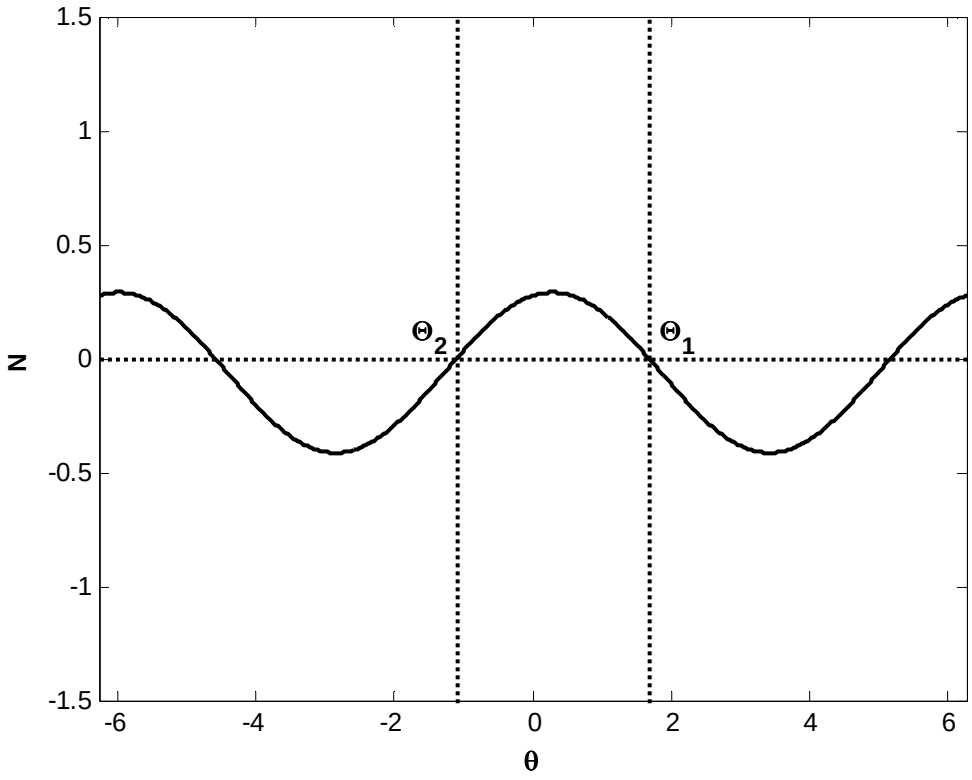
Output:

theta_1 =

1.6942

theta_2 =

-1.0937



4.A.4.3 Determining N_u and N_d (Linear Damping)

```
%SMR - Linearly Damped System
%Finding Zu, Zd
```

```
close all
clear all

syms zu_m zu_p lambda

z_vec1 = simple(solve('zu_m^3 - 2*zu_m^2 + (lambda^2+1)*zu_m = ((2-
sqrt(1-3*lambda^2))/3)^3 - 2*((2-sqrt(1-3*lambda^2))/3)^2 +
(lambda^2+1)*((2-sqrt(1-3*lambda^2))/3)',zu_m));

zu1 = z_vec1(1);
zu2 = z_vec1(2);
zu3 = z_vec1(3);

z_vec2 = simple(solve('zu_p^3 - 2*zu_p^2 + (lambda^2+1)*zu_p =
((2+sqrt(1-3*lambda^2))/3)^3 - 2*((2+sqrt(1-3*lambda^2))/3)^2 +
(lambda^2+1)*((2+sqrt(1-3*lambda^2))/3)',zu_p));

zu4 = z_vec2(1);
zu5 = z_vec2(2);
zu6 = z_vec2(3);

lambda = 0.2;

zu_minus = (2-sqrt(1-3*lambda^2))/3;
zu_plus = (2+sqrt(1-3*lambda^2))/3;

zu1_num = subs(zu1); zu2_num = subs(zu2); zu3_num = subs(zu3);
zu4_num = subs(zu4); zu5_num = subs(zu5); zu6_num = subs(zu6);

if zu_minus ~= zu1_num && zu_plus ~= zu1_num
    if zu1_num > 0
        zu1
    else
    end
else
end

if zu_minus ~= zu2_num && zu_plus ~= zu2_num
    if zu2_num > 0
        zu2
    else
    end
else
end

if zu_minus ~= zu3_num && zu_plus ~= zu3_num
    if zu3_num > 0
        zu3
    else
    end
else
end

if zu_minus ~= zu4_num && zu_plus ~= zu4_num
```

```
    if zu4_num > 0
        zu4
    else
    end
else
end

if zu_minus ~= zu5_num && zu_plus ~= zu5_num
    if zu5_num > 0
        zu5
    else
    end
else
end

if zu_minus ~= zu6_num && zu_plus ~= zu6_num
    if zu6_num > 0
        zu6
    else
    end
else
end
```

Output:

zu3 =

(2*(1 - 3*lambda^2)^(1/2))/3 + 2/3

zu5 =

2/3 - (2*(1 - 3*lambda^2)^(1/2))/3

4.A.4.4 Plotting Phase Portraits and 1-D Maps (Linear Damping)

```
%Chapter 4
%SMR Phase Portraits for the Linearly Damped System

close all
clear all

%SMR vanishes for sigma < -5.07 and sigma > 9.11
%SMR returns for sigma = [9.73,10.07] then vanishes again

for sigma = [-5.07 -2 0 5 9.11 9.73 10.07]

    lambda = 0.2; A = 1;

    %Fold Lines (N1 and N2):
```

```
    N1 = sqrt((2 - sqrt(1-3*lambda^2))/3);
    N2 = sqrt((2 + sqrt(1-3*lambda^2))/3);

    %Fold Line End Locations (Nu and Nd):

    Nu = sqrt(2/3*(1 + sqrt(1-3*lambda^2)));
    Nd = sqrt(2/3*(1 - sqrt(1-3*lambda^2)));

    %Folded singularities:

    theta_1 = 1.6942; %From separate m-file (for A = 1, lambda = 0.2)
    theta_2 = -1.0937; %From separate m-file (for A = 1, lambda = 0.2)

    theta_i = -10; theta_f = 10; theta_span = theta_f - theta_i;
    theta_step = theta_span/100;

    N_test = 0; k_R = 0;

    figure
    hold on; box on;
    axis([0 2*pi 0 Nu]);
    xlabel('\fontsize{12}\bf\theta'); ylabel('\bfN');
    size_of_axes = get(gca,'Position');
    origin_x = size_of_axes(1); %Position of origin relative to figure
    origin_y = size_of_axes(2); %Position of origin relative to figure
    x_conv = size_of_axes(3)/(2*pi);
    y_conv = size_of_axes(4)/Nu;


    for No = [N1-.001 Nu]
        N_test = N_test + 1;

        for theta_o = theta_i : theta_step : theta_f
            theta_vec = zeros(1,theta_span/theta_step + 1);
            N_vec = theta_vec;
            theta = theta_o; N = No; k = 1;
            theta_vec(1) = theta; N_vec(1) = N;

            %*********For 1D
Mapping************************************

            if theta < theta_1 && theta > theta_2 && k == 1 && N_test
== 1
                k_R = k_R + 1;
                R_theta(k_R) = theta_vec(1);
                R_N(k_R) = N_vec(1);
                theta_u(k_R) = R_theta(k_R) + ...
                    atan2(-1 + 15*lambda^2 - sqrt(1 - 3*lambda^2), ...
                    9*lambda*sqrt(1 - 3*lambda^2));
                Nu_vec(k_R) = Nu;
            else
            end
```

```
            %*******Back to phase
portrait*******************************

            k_end = 0; t = 0;

            for t_end = [.0005 .005 .05 5]
                k_end = k_end + 1;

                if k_end == 1
                    dt = 0.00001;
                elseif k_end == 2
                    dt = 0.0001;
                elseif k_end == 3
                    dt = .001;
                else
                    dt = .001;
                end

                for t = dt + t:dt:t_end
                    den = (2*(lambda^2 + 1 - 4*N^2 + 3*N^4));

                    d_theta = (-3*(1 - sigma)*N^4 + (1 - 4*sigma)*N^2 +
...
                        3*A*N*sin(theta) + ...
                        (sigma - lambda^2*(1 - sigma)) + ...
                        (lambda*A*cos(theta) - A*sin(theta))/N);

                    d_N = (-A*N^2*cos(theta) - lambda*N + ...
                        lambda*A*sin(theta) + A*cos(theta));

                    theta = theta + d_theta*dt;
                    N = N + d_N*dt;

                    k = k + 1;

                    theta_vec(k) = theta;
                    N_vec(k) = N;
                end
            end

            kL = 0; kU = 0;

            %Remove unstable response:

            stable_check = 0;

            for k2 = 1:length(theta_vec)

                if N_test == 1
                    if N_vec(k2) < N1 && stable_check == 0
                        kL = kL + 1;
                        theta_lower(kL) = theta_vec(k2);
                        N_lower(kL) = N_vec(k2);
                    else
```

```
                        stable_check = 1;
                    end

                elseif N_test == 2
                    if N_vec(k2) > N2
                        kU = kU + 1;
                        theta_upper(kU) = theta_vec(k2);
                        N_upper(kU) = N_vec(k2);
                    else
                    end
                else
                end
            end

            if kL > 0
                plot(theta_lower,N_lower,'-k','LineWidth',2);

                arrow_L = round(kL/3);

                arrow_min_theta =
[origin_x+theta_lower(arrow_L)*x_conv,...
                    origin_x+theta_lower(arrow_L+1)*x_conv];
                arrow_min_N = [origin_y+N_lower(arrow_L)*y_conv,...
                    origin_y+N_lower(arrow_L+1)*y_conv];

                if theta_lower(arrow_L) < theta_2 + 2*pi && ...
                        theta_lower(arrow_L+1) < theta_2 + 2*pi && ...
                        theta_lower(arrow_L) > theta_1 && ...
                        theta_lower(arrow_L+1) > theta_1
                annotation('arrow',arrow_min_theta, arrow_min_N);
                else
                end
            else
            end

            if kU > 0
                plot(theta_upper,N_upper,'-k','LineWidth',2);

                arrow_U = round(kU/1.2);

                arrow_min_theta =
[origin_x+theta_upper(arrow_U)*x_conv,...
                    origin_x+theta_upper(arrow_U+1)*x_conv];
                arrow_min_N = [origin_y+N_upper(arrow_U)*y_conv,...
                    origin_y+N_upper(arrow_U+1)*y_conv];

                if theta_upper(arrow_U) < theta_2 + 2*pi && ...
                        theta_upper(arrow_U+1) < theta_2 + 2*pi && ...
                        theta_upper(arrow_U) > theta_1 && ...
                        theta_upper(arrow_U+1) > theta_1  && ...
                        N_upper(arrow_U) < Nu - .001
                annotation('arrow',arrow_min_theta, arrow_min_N);
                else
                end
            else
```

```
            end

            clear theta_lower theta_upper N_lower N_upper N_vec
theta_vec
        end
    end



    plot([0 2*pi],[N1 N1],'-k','LineWidth',4)
    plot([0 2*pi],[N2 N2],'-k','LineWidth',4)
    plot([theta_2+2*pi theta_2+2*pi],[-50 50],':k','LineWidth',2)
    plot([theta_1 theta_1],[-50 50],':k','LineWidth',2)
    text(theta_2 + 2*pi - 0.3,N1 + 0.05,'\bf\Theta_2','FontSize',12)
    text(theta_1 + 0.1,N1 + 0.05,'\bf\Theta_1','FontSize',12)
    text(2*pi + 0.15,N1,'\bfN_1')
    text(2*pi + 0.15,N2,'\bfN_2')


%**************************************************************************

%**************************************************************************

%**************************************************************************

    %1D Mapping:

    figure
    hold on; grid on; box on;
    xlabel('\fontsize{12}\bf\theta'); ylabel('\bfN');

    %Plotting from the interval R on N1 to Nu:

    for q = 1:length(R_theta)
        theta_plot = [R_theta(q); theta_u(q)];
        N_plot = [R_N(q); Nu_vec(q)];
        plot(R_theta(q),R_N(q),'.k','MarkerSize',20)
        plot(theta_plot,N_plot,'-k','LineWidth',2)
    end

    %Plotting from Nu to N2:

    k_t = 0;
    dt = .0001; t_end = 10;

    for h = 1:length(theta_u)
        k = 1; k_t = k_t + 1;
        N = Nu; theta = theta_u(h); theta_1D(1) = theta; N_1D(1) = N;

        for t = dt:dt:t_end
            d_theta = (-3*(1 - sigma)*N^4 + (1 - 4*sigma)*N^2 + ...
                3*A*N*sin(theta) + (sigma - lambda^2*(1 - sigma)) + ...
                (lambda*A*cos(theta) - A*sin(theta))/N);
```

```
            d_N = (-A*N^2*cos(theta) - lambda*N + lambda*A*sin(theta) +
...
                A*cos(theta));

            theta = theta + d_theta*dt;
            N = N + d_N*dt;

            if N >= N2 && N <= Nu
                k = k + 1;
                theta_1D(k) = theta;
                N_1D(k) = N;
            else
            end
        end

        theta_1D_min(k_t) = theta_1D(k);
        theta_IC_u(k_t) = theta_u(h);

        plot(theta_1D,N_1D,'-k','LineWidth',2)
        clear theta_1D N_1D
    end

    %Plotting from N2 to Nd:

    k_R2 = 0;

    for g = 1:length(theta_1D_min)
        k_R2 = k_R2 + 1;
        U_theta(k_R2) = theta_1D_min(g);
        U_N(k_R2) = N2;
        theta_d(k_R2) = U_theta(k_R2) - ...
            atan2(-1 + 15*lambda^2 + sqrt(1 - 3*lambda^2), ...
            9*lambda*sqrt(1 - 3*lambda^2));
        Nd_vec(k_R2) = Nd;
    end

    for z = 1:length(U_theta)
        theta_plot2 = [U_theta(z); theta_d(z)];
        N_plot2 = [U_N(z); Nd_vec(z)];
        plot(theta_plot2,N_plot2,'-k')
    end

    %Plotting from Nd to N1:

    k_t = 0;
    dt = .0001; t_end = 1;

    for h = 1:length(theta_d)
        k = 1;
        N = Nd; theta = theta_d(h);
        theta_2D(1) = theta; N_2D(1) = N;

        for t = dt:dt:t_end
            d_theta = (-3*(1 - sigma)*N^4 + (1 - 4*sigma)*N^2 + ...
```

```
                3*A*N*sin(theta) + (sigma - lambda^2*(1 - sigma)) + ...
                (lambda*A*cos(theta) - A*sin(theta))/N);

            d_N = (-A*N^2*cos(theta) - lambda*N + lambda*A*sin(theta) +
...
                A*cos(theta));

            theta = theta + d_theta*dt;
            N = N + d_N*dt;

            if N <= N1
                k = k + 1;
                theta_2D(k) = theta;
                N_2D(k) = N;
            else
            end
        end

        if theta_2D(k) <= theta_1 && theta_2D(k) >= theta_2
            k_t = k_t + 1;
            theta_2D_map(k_t) = theta_2D(k);
            theta_2D_IC(k_t) = theta_d(h);
        else
        end

        plot(theta_2D,N_2D,'-k','LineWidth',2)
        clear theta_2D N_2D
    end

    %Plotting the 1D Map:

    %Backing out the values of theta on N2 corresponding to the values
of
    %theta on Nd (from the values of theta returned to N1):

    k_map2 = 0;

    for k_map1 = 1:length(theta_d)
        for b = 1:length(theta_2D_map)
            if theta_2D_IC(b) == theta_d(k_map1)
                k_map2 = k_map2 + 1;
                theta_N2(k_map2) = U_theta(k_map1);
            else
            end
        end
    end

    %Backing out the values of theta on Nu corresponding to the values
of
    %theta on N2 (from the values of theta returned to N1):

    k_map4 = 0;

    for k_map3 = 1:length(U_theta)
```

```
        for b2 = 1:length(theta_N2)
            if theta_N2(b2) == theta_1D_min(k_map3)
                k_map4 = k_map4 + 1;
                theta_N1(k_map4) = theta_u(k_map3);
            else
            end
        end
    end

    %Backing out the values of theta on N1 corresponding to the values
of
    %theta on Nu (from the values of theta returned to N1):

    k_map6 = 0;

    for k_map5 = 1:length(R_theta)
        for b3 = 1:length(theta_N2)
            if theta_N1(b3) == theta_u(k_map5)
                k_map6 = k_map6 + 1;
                theta_map(k_map6) = R_theta(k_map5);
            else
            end
        end
    end

    figure
    hold on; box on;
    xlabel('\bf\theta','FontSize',12)
    axis([theta_2 - 0.5, theta_1 + 0.5, -0.5, 1.5])
    size_of_axes = get(gca,'Position');
    x_conv = size_of_axes(3)/(theta_1 + 0.5 - (theta_2 - 0.5));
    y_conv = size_of_axes(4)/(1.5-(-0.5));
    origin_x = size_of_axes(1); %Position of origin relative to figure
    origin_y = size_of_axes(2); %Position of origin relative to figure

    for k_map = 1:length(theta_map)
        map_vec = [theta_map(k_map); theta_2D_map(k_map)];
        offset_vec = [0; 1];
        plot(map_vec,offset_vec,'-k','LineWidth',2)
        arrow_min_theta = [origin_x+(map_vec(1)-(theta_2-
0.5))*x_conv,...
            origin_x+(map_vec(1)-(theta_2-0.5))*x_conv + ...
            1/4*(map_vec(2)-map_vec(1))*x_conv];
        arrow_min_N = [origin_y+0.5*y_conv,...
            origin_y+0.5*y_conv+1/4*(1*y_conv)];
        annotation('arrow',arrow_min_theta, arrow_min_N);
    end

    plot([theta_2 theta_1],zeros(1,2),'-k','LineWidth',6)
    plot([theta_2 theta_1],ones(1,2),'-k','LineWidth',6)
    plot([theta_2 theta_2],[-50 50],':k','LineWidth',2)
    plot([theta_1 theta_1],[-50 50],':k','LineWidth',2)
    text(theta_2 - 0.25,0,'\bf\Theta_2','FontSize',12)
    text(theta_1 + 0.1,0,'\bf\Theta_1','FontSize',12)
    set(gca,'ycolor','w','ytick',[])
```

```
    clear all
end
```

4.A.4.5 Plotting SIM (Nonlinear Damping)

```
%SIM Projection for the Nonlinearly Damped System

close all
clear all

lambda = 0.2; k = 0; ks1 = 0; ks2 = 0; ku = 0;
N_end = 1.6; N_step = 0.0001;
C_sq_64 = zeros(1,N_end/N_step + 1);
N_vec = C_sq_64;

%Fold Lines (N1 and N2):

N1 = sqrt((32 - 4*sqrt(16-27*lambda^2))/(27*lambda^2+48));
N2 = sqrt((32 + 4*sqrt(16-27*lambda^2))/(27*lambda^2+48));

%Fold Line End Locations (Nu and Nd):

Nu = sqrt((32 + 8*sqrt(16-27*lambda^2))/(27*lambda^2+48));
Nd = sqrt((32 - 8*sqrt(16-27*lambda^2))/(27*lambda^2+48));

for N = 0:N_step:N_end
    k = k + 1;
    C_sq_64 = (9*lambda^2+16)*N^6 - 32*N^4 + 16*N^2;

    if N < N1
        ks1 = ks1 + 1;
        C_sq_64_stable1(ks1) = C_sq_64;
        N_stable1(ks1) = N;
    elseif N > N2
        ks2 = ks2 + 1;
        C_sq_64_stable2(ks2) = C_sq_64;
        N_stable2(ks2) = N;
    else
        ku = ku + 1;
        C_sq_64_unstable(ku) = C_sq_64;
        N_unstable(ku) = N;
    end
end

C_u = [max(C_sq_64_unstable) max(C_sq_64_unstable)];
C_d = [min(C_sq_64_unstable) min(C_sq_64_unstable)];
fold_u = [N1 Nu]; fold_d = [Nd N2];

figure
hold on; xlabel('\bfN'); ylabel('\bf64|C|^2');
plot(N_stable1,C_sq_64_stable1,'-k','LineWidth',2)
```

```
plot(N_stable2,C_sq_64_stable2,'-k','LineWidth',2)
plot(N_unstable,C_sq_64_unstable,'--k','LineWidth',2)
plot(fold_u, C_u, ':k','LineWidth',2)
plot(fold_u(2)-.03, C_u(2), '>k','LineWidth',2,'MarkerFaceColor','k')
plot(fold_d, C_d, ':k','LineWidth',2)
plot(fold_d(1)+.03, C_d(1), '<k','LineWidth',2,'MarkerFaceColor','k')
text(N1, C_u(1)+.1, '\bfN_1')
text(N2, C_d(1)-.1, '\bfN_2')
text(Nu+.03, C_u(1), '\bfN_u')
text(Nd, C_d(1)-.15, '\bfN_d')
axis([0, N, 0, 2.8])
```

4.A.4.6 Determining Θ_1 and Θ_2 (Nonlinear Damping)

```
%Chapter 4 - SMR
%Find THETA_1 and THETA_2 for the system with nonlinear damping:

close all
clear all

lambda = 0.2; A = 1;

N1 = sqrt((32 - 4*sqrt(16-27*lambda^2))/(27*lambda^2+48)); %Lower fold
line

k = 0; n = 0;

for theta = -2*pi:.001:2*pi
    k = k + 1;
    q = -(48+27*lambda^2)*N1^4 + 16*N1^2 + 36*A*lambda*N1*cos(theta) +
...
        (48*A*N1-16*A/N1)*sin(theta);
    q_vec(k) = q;
    theta_vec(k) = theta;
    if abs(q) < .01
        n = n + 1;
        near_zero(n) = theta;
    else
    end
end

figure
plot(theta_vec,q_vec,'-k','LineWidth',2)
xlabel('\bf\theta','fontsize',12); ylabel('\bfN'); hold on;

syms c_theta

cos_vec = solve('-(48+27*lambda^2)*N1^4 + 16*N1^2 +
36*A*lambda*N1*c_theta + (48*A*N1-16*A/N1)*sqrt(1-c_theta^2) =
0',c_theta);

ct1 = subs(cos_vec(1));
```

```
ct2 = subs(cos_vec(2));

theta_1_maybe = acos(ct1);
theta_2_maybe = acos(ct2);
near_zero;

for h = 1:length(near_zero)
    if abs(abs(near_zero(h)) - abs(theta_1_maybe)) < .1
        theta_1 = sign(near_zero(h)) * abs(theta_1_maybe);
    else
    end

    if abs(abs(near_zero(h)) - abs(theta_2_maybe)) < .1
        theta_2 = sign(near_zero(h)) * abs(theta_2_maybe);
    else
    end
end

theta_1_final = max(theta_1,theta_2);
theta_2_final = min(theta_1,theta_2);

theta_1 = theta_1_final
theta_2 = theta_2_final

plot([theta_1 theta_1],[-50 50],':k','LineWidth',2)
plot([theta_2 theta_2],[-50 50],':k','LineWidth',2)
plot([theta_vec(1) theta_vec(k)],[0 0],':k','LineWidth',2)
text(theta_2 - 0.7,0.4,'\bf\Theta_2','FontSize',12)
text(theta_1 + 0.2,0.4,'\bf\Theta_1','FontSize',12)
axis([theta_vec(1), theta_vec(k), -5, 5])
```

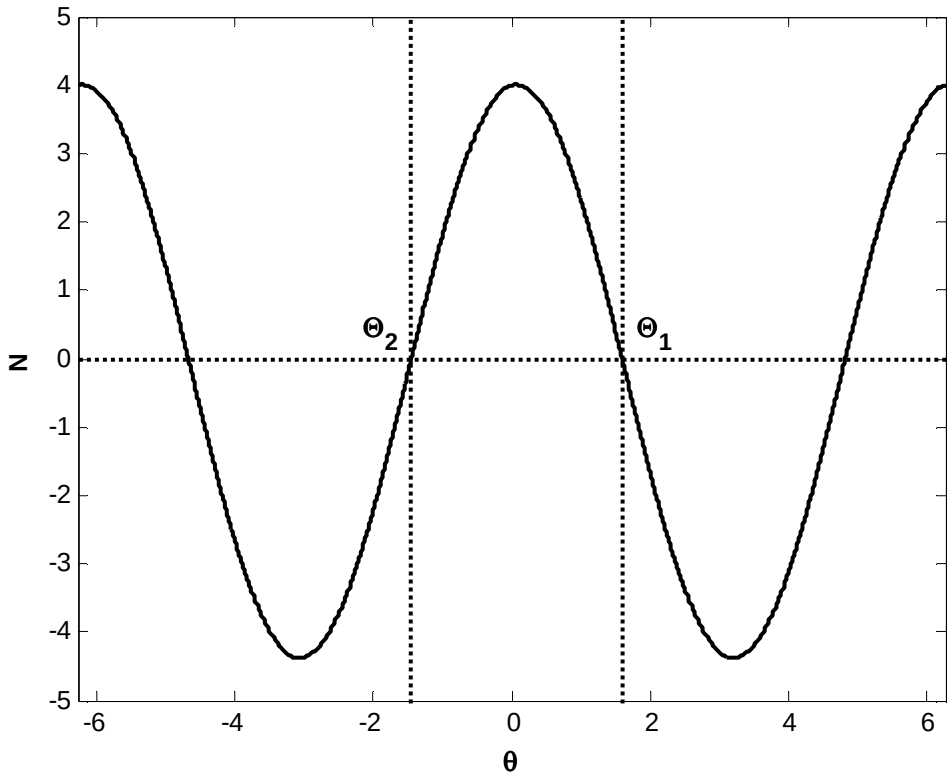
Output:

theta_1 =

1.6028

theta_2 =

-1.4504



4.A.4.7 Determining N_u and N_d (Nonlinear Damping

```
%SMR - Nonlinearly Damped System
%Finding Zu, Zd

close all
clear all

syms zu_m zu_p lambda

z_vec1 = simple(solve('(9*lambda^2+16)*zu_m^3 - 32*zu_m^2 + 16*zu_m =
(9*lambda^2+16)*((32-4*sqrt(16-27*lambda^2))/(27*lambda^2+48))^3 -
32*((32-4*sqrt(16-27*lambda^2))/(27*lambda^2+48))^2 + 16*(32-4*sqrt(16-
27*lambda^2))/(27*lambda^2+48)',zu_m));

zu1 = z_vec1(1);
zu2 = z_vec1(2);
zu3 = z_vec1(3);

z_vec2 = simple(solve('(9*lambda^2+16)*zu_p^3 - 32*zu_p^2 + 16*zu_p =
(9*lambda^2+16)*((32+4*sqrt(16-27*lambda^2))/(27*lambda^2+48))^3 -
32*((32+4*sqrt(16-27*lambda^2))/(27*lambda^2+48))^2 + 16*(32+4*sqrt(16-
27*lambda^2))/(27*lambda^2+48)',zu_p));
```

```
zu4 = z_vec2(1);
zu5 = z_vec2(2);
zu6 = z_vec2(3);

lambda = 0.2;

zu_minus = (32-4*sqrt(16-27*lambda^2))/(27*lambda^2+48);
zu_plus = (32+4*sqrt(16-27*lambda^2))/(27*lambda^2+48);

zu1_num = subs(zu1); zu2_num = subs(zu2); zu3_num = subs(zu3);
zu4_num = subs(zu4); zu5_num = subs(zu5); zu6_num = subs(zu6);

small = 0.0001;

if abs(zu_minus - zu1_num) > small && abs(zu_plus - zu1_num) > small
    if zu1_num > 0
        zu1
    else
    end
else
end

if abs(zu_minus - zu2_num) > small && abs(zu_plus - zu2_num) > small
    if zu2_num > 0
        zu2
    else
    end
else
end

if abs(zu_minus - zu3_num) > small && abs(zu_plus - zu3_num) > small
    if zu3_num > 0
        zu3
    else
    end
else
end

if abs(zu_minus - zu4_num) > small && abs(zu_plus - zu4_num) > small
    if zu4_num > 0
        zu4
    else
    end
else
end

if abs(zu_minus - zu5_num) > small && abs(zu_plus - zu5_num) > small
    if zu5_num > 0
        zu5
    else
    end
else
end
```

```
if abs(zu_minus - zu6_num) > small && abs(zu_plus - zu6_num) > small
    if zu6_num > 0
        zu6
    else
    end
else
end
```

Output:

zu2 =

(8*(16 - 27*lambda^2)^(1/2) + 32)/(27*lambda^2 + 48)

zu6 =

-(8*(16 - 27*lambda^2)^(1/2) - 32)/(27*lambda^2 + 48)

4.A.4.8 Plotting Phase Portraits and 1-D Maps (Nonlinear Damping)

```
%Chapter 4
%SMR Phase Portraits for the Nonlinearly Damped System

close all
clear all

%SMR vanishes for sigma < -7.17 and sigma > 8.47

for sigma = [-7.17 -4 0 4 8.47]

    lambda = 0.2; A = 1;

    %Fold Lines (N1 and N2):

    N1 = sqrt((32 - 4*sqrt(16-27*lambda^2))/(27*lambda^2+48));
    N2 = sqrt((32 + 4*sqrt(16-27*lambda^2))/(27*lambda^2+48));

    %Fold Line End Locations (Nu and Nd):

    Nu = sqrt((32 + 8*sqrt(16-27*lambda^2))/(27*lambda^2+48));
    Nd = sqrt((32 - 8*sqrt(16-27*lambda^2))/(27*lambda^2+48));

    %Folded singularities:
```

```
    theta_1 = 1.6028; %From separate m-file (for A = 1, lambda = 0.2)
    theta_2 = -1.4504; %From separate m-file (for A = 1, lambda = 0.2)

    theta_i = -10; theta_f = 10; theta_span = theta_f - theta_i;
    theta_step = theta_span/100;

    N_test = 0; k_R = 0;

    figure
    hold on; box on;
    axis([0 2*pi 0 Nu]);
    xlabel('\fontsize{12}\bf\theta'); ylabel('\bfN');
    size_of_axes = get(gca,'Position');
    origin_x = size_of_axes(1); %Position of origin relative to figure
    origin_y = size_of_axes(2); %Position of origin relative to figure
    x_conv = size_of_axes(3)/(2*pi);
    y_conv = size_of_axes(4)/Nu;


    for No = [N1-.001 Nu]
        N_test = N_test + 1;

        for theta_o = theta_i : theta_step : theta_f
            theta_vec = zeros(1,theta_span/theta_step + 1);
            N_vec = theta_vec;
            theta = theta_o; N = No; k = 1;
            theta_vec(1) = theta; N_vec(1) = N;

            %*********For 1D
Mapping************************************

            if theta < theta_1 && theta > theta_2 && k == 1 && N_test
== 1
                k_R = k_R + 1;
                R_theta(k_R) = theta_vec(1);
                R_N(k_R) = N_vec(1);
                theta_u(k_R) = R_theta(k_R) + ...
                    atan2(1215*lambda^4 + ...
                    36*lambda^2*(56-sqrt(16-27*lambda^2)) - ...
                    64*sqrt(16-27*lambda^2) - 256, ...
                    27*lambda*(9*lambda^2+16)*sqrt(16-27*lambda^2));
                Nu_vec(k_R) = Nu;
            else
            end

            %*******Back to phase
portrait*********************************

            k_end = 0; t = 0;

            for t_end = [.0005 .005 .05 5]
                k_end = k_end + 1;

                if k_end == 1
```

```
            dt = 0.00001;
        elseif k_end == 2
            dt = 0.0001;
        elseif k_end == 3
            dt = .001;
        else
            dt = .001;
        end

        for t = dt + t:dt:t_end
            den = (2*(lambda^2 + 1 - 4*N^2 + 3*N^4));

            d_theta = -(48+27*lambda^2)*(1-sigma)*N^4 + ...
                (16-64*sigma)*N^2 + ...
                4*A*(9*lambda*cos(theta) + 12*sin(theta))*N +
...
                16*sigma - (16*A*sin(theta))/N;

            d_N = -12*lambda*N^3 + ...
                4*A*(3*lambda*sin(theta) - 4*cos(theta))*N^2 +
...
                16*A*cos(theta);

            theta = theta + d_theta*dt;
            N = N + d_N*dt;

            k = k + 1;

            theta_vec(k) = theta;
            N_vec(k) = N;
        end
    end

    kL = 0; kU = 0;

    %Remove unstable response:

    stable_check = 0;

    for k2 = 1:length(theta_vec)

        if N_test == 1
            if N_vec(k2) < N1 && stable_check == 0
                kL = kL + 1;
                theta_lower(kL) = theta_vec(k2);
                N_lower(kL) = N_vec(k2);
            else
                stable_check = 1;
            end

        elseif N_test == 2
            if N_vec(k2) > N2
                kU = kU + 1;
                theta_upper(kU) = theta_vec(k2);
```

```
                        N_upper(kU) = N_vec(k2);
                    else
                    end
                else
                end
            end

            if kL > 0
                plot(theta_lower,N_lower,'-k','LineWidth',2);

                arrow_L = round(kL/1.5);

                arrow_min_theta =
[origin_x+theta_lower(arrow_L)*x_conv,...
                    origin_x+theta_lower(arrow_L+1)*x_conv];
                arrow_min_N = [origin_y+N_lower(arrow_L)*y_conv,...
                    origin_y+N_lower(arrow_L+1)*y_conv];

                if theta_lower(arrow_L) < theta_2 + 2*pi && ...
                        theta_lower(arrow_L+1) < theta_2 + 2*pi && ...
                        theta_lower(arrow_L) > theta_1 && ...
                        theta_lower(arrow_L+1) > theta_1 && ...
                        N_lower(arrow_L) > 0.001
                annotation('arrow',arrow_min_theta, arrow_min_N);
                else
                end
            else
            end

            if kU > 0
                plot(theta_upper,N_upper,'-k','LineWidth',2);

                arrow_U = round(kU/1.1);

                arrow_min_theta =
[origin_x+theta_upper(arrow_U)*x_conv,...
                    origin_x+theta_upper(arrow_U+1)*x_conv];
                arrow_min_N = [origin_y+N_upper(arrow_U)*y_conv,...
                    origin_y+N_upper(arrow_U+1)*y_conv];

                if theta_upper(arrow_U) < (4*pi + theta_2)/1.8 && ...
                        theta_upper(arrow_U+1) < (4*pi+theta_2)/1.5 &&
...
                        theta_upper(arrow_U) > theta_1 && ...
                        theta_upper(arrow_U+1) > theta_1  && ...
                        N_upper(arrow_U) < Nu - .001
                annotation('arrow',arrow_min_theta, arrow_min_N);
                else
                end
            else
            end

            clear theta_lower theta_upper N_lower N_upper N_vec
theta_vec
        end
```

```
    end



    plot([0 2*pi],[N1 N1],'-k','LineWidth',4)
    plot([0 2*pi],[N2 N2],'-k','LineWidth',4)
    plot([theta_2+2*pi theta_2+2*pi],[-50 50],':k','LineWidth',2)
    plot([theta_1 theta_1],[-50 50],':k','LineWidth',2)
    text(theta_2 + 2*pi - 0.3,N1 + 0.05,'\bf\Theta_2','FontSize',12)
    text(theta_1 + 0.1,N1 + 0.05,'\bf\Theta_1','FontSize',12)
    text(2*pi + 0.15,N1,'\bfN_1')
    text(2*pi + 0.15,N2,'\bfN_2')


%**************************************************************************

%**************************************************************************

%**************************************************************************

    %1D Mapping:

    figure
    hold on; grid on; box on;
    xlabel('\fontsize{12}\bf\theta'); ylabel('\bfN');

    %Plotting from the interval R on N1 to Nu:

    for q = 1:length(R_theta)
        theta_plot = [R_theta(q); theta_u(q)];
        N_plot = [R_N(q); Nu_vec(q)];
        plot(R_theta(q),R_N(q),'.k','MarkerSize',20)
        plot(theta_plot,N_plot,'-k','LineWidth',2)
    end

    %Plotting from Nu to N2:

    k_t = 0;
    dt = .0001; t_end = 10;

    for h = 1:length(theta_u)
        k = 1; k_t = k_t + 1;
        N = Nu; theta = theta_u(h); theta_1D(1) = theta; N_1D(1) = N;

        for t = dt:dt:t_end
            d_theta = -(48+27*lambda^2)*(1-sigma)*N^4 + ...
                (16-64*sigma)*N^2 + ...
                4*A*(9*lambda*cos(theta) + 12*sin(theta))*N + ...
                16*sigma - (16*A*sin(theta))/N;

            d_N = -12*lambda*N^3 + ...
                4*A*(3*lambda*sin(theta) - 4*cos(theta))*N^2 + ...
                16*A*cos(theta);
```

```
        theta = theta + d_theta*dt;
        N = N + d_N*dt;

        if N >= N2 && N <= Nu
            k = k + 1;
            theta_1D(k) = theta;
            N_1D(k) = N;
        else
        end
    end

    theta_1D_min(k_t) = theta_1D(k);
    theta_IC_u(k_t) = theta_u(h);

    plot(theta_1D,N_1D,'-k','LineWidth',2)
    clear theta_1D N_1D
end

%Plotting from N2 to Nd:

k_R2 = 0;

for g = 1:length(theta_1D_min)
    k_R2 = k_R2 + 1;
    U_theta(k_R2) = theta_1D_min(g);
    U_N(k_R2) = N2;
    theta_d(k_R2) = U_theta(k_R2) - ...
        atan2(1215*lambda^4 + ...
        36*lambda^2*(56+sqrt(16-27*lambda^2)) + ...
        64*sqrt(16-27*lambda^2) - 256, ...
        27*lambda*(9*lambda^2+16)*sqrt(16-27*lambda^2));
    Nd_vec(k_R2) = Nd;
end

for z = 1:length(U_theta)
    theta_plot2 = [U_theta(z); theta_d(z)];
    N_plot2 = [U_N(z); Nd_vec(z)];
    plot(theta_plot2,N_plot2,'-k')
end

%Plotting from Nd to N1:

k_t = 0;
dt = .0001; t_end = 1;

for h = 1:length(theta_d)
    k = 1;
    N = Nd; theta = theta_d(h);
    theta_2D(1) = theta; N_2D(1) = N;

    for t = dt:dt:t_end
        d_theta = -(48+27*lambda^2)*(1-sigma)*N^4 + ...
            (16-64*sigma)*N^2 + 4*A*(9*lambda*cos(theta) + ...
            12*sin(theta))*N + 16*sigma - (16*A*sin(theta))/N;
```

```
            d_N = -12*lambda*N^3 + 4*A*(3*lambda*sin(theta) - ...
                4*cos(theta))*N^2 + 16*A*cos(theta);

            theta = theta + d_theta*dt;
            N = N + d_N*dt;

            if N <= N1
                k = k + 1;
                theta_2D(k) = theta;
                N_2D(k) = N;
            else
            end
        end

        if theta_2D(k) <= theta_1 && theta_2D(k) >= theta_2
            k_t = k_t + 1;
            theta_2D_map(k_t) = theta_2D(k);
            theta_2D_IC(k_t) = theta_d(h);
        else
        end

        plot(theta_2D,N_2D,'-k','LineWidth',2)
        clear theta_2D N_2D
    end

    %Plotting the 1D Map:

    %Backing out the values of theta on N2 corresponding to the values
of
    %theta on Nd (from the values of theta returned to N1):

    k_map2 = 0;

    for k_map1 = 1:length(theta_d)
        for b = 1:length(theta_2D_map)
            if theta_2D_IC(b) == theta_d(k_map1)
                k_map2 = k_map2 + 1;
                theta_N2(k_map2) = U_theta(k_map1);
            else
            end
        end
    end

    %Backing out the values of theta on Nu corresponding to the values
of
    %theta on N2 (from the values of theta returned to N1):

    k_map4 = 0;

    for k_map3 = 1:length(U_theta)
        for b2 = 1:length(theta_N2)
            if theta_N2(b2) == theta_1D_min(k_map3)
                k_map4 = k_map4 + 1;
```

```
                theta_N1(k_map4) = theta_u(k_map3);
            else
            end
        end
    end

    %Backing out the values of theta on N1 corresponding to the values
of
    %theta on Nu (from the values of theta returned to N1):

    k_map6 = 0;

    for k_map5 = 1:length(R_theta)
        for b3 = 1:length(theta_N2)
            if theta_N1(b3) == theta_u(k_map5)
                k_map6 = k_map6 + 1;
                theta_map(k_map6) = R_theta(k_map5);
            else
            end
        end
    end

    figure
    hold on; box on;
    xlabel('\bf\theta','FontSize',12)
    axis([theta_2 - 0.5, theta_1 + 0.5, -0.5, 1.5])
    size_of_axes = get(gca,'Position');
    x_conv = size_of_axes(3)/(theta_1 + 0.5 - (theta_2 - 0.5));
    y_conv = size_of_axes(4)/(1.5-(-0.5));
    origin_x = size_of_axes(1); %Position of origin relative to figure
    origin_y = size_of_axes(2); %Position of origin relative to figure

    for k_map = 1:length(theta_map)
        map_vec = [theta_map(k_map); theta_2D_map(k_map)];
        offset_vec = [0; 1];
        plot(map_vec,offset_vec,'-k','LineWidth',2)
        arrow_min_theta = [origin_x+(map_vec(1)-(theta_2-
0.5))*x_conv,...
            origin_x+(map_vec(1)-(theta_2-0.5))*x_conv + ...
            1/4*(map_vec(2)-map_vec(1))*x_conv];
        arrow_min_N = [origin_y+0.5*y_conv,...
            origin_y+0.5*y_conv+1/4*(1*y_conv)];
        annotation('arrow',arrow_min_theta, arrow_min_N);
    end

    plot([theta_2 theta_1],zeros(1,2),'-k','LineWidth',6)
    plot([theta_2 theta_1],ones(1,2),'-k','LineWidth',6)
    plot([theta_2 theta_2],[-50 50],':k','LineWidth',2)
    plot([theta_1 theta_1],[-50 50],':k','LineWidth',2)
    text(theta_2 - 0.25,0,'\bf\Theta_2','FontSize',12)
    text(theta_1 + 0.1,0,'\bf\Theta_1','FontSize',12)
    set(gca,'ycolor','w','ytick',[])

    clear all
end
```

5.0 Time Response Analysis of the Nonlinear Energy Sink

5.1 Introduction

In addition to the applications of NES in harmonically forced systems, NES can be effective in impulsively forced systems as well. This section depicts the temporal dependence of displacement and energy of the system consisting of the linear primary system and nonlinearly damped NES attachment exposed to varying amplitudes of impulsive forcing on the mass of the primary system. For comparison, results using a linearly damped NES also are presented. Following the impulse response discussions, the performance of harmonically forced linear systems with linearly versus nonlinearly damped NES attachments is presented with the use of Poincaré maps.

5.2 System Performance when Subjected to Impulse Loading

The system with linear damping, given by

$$\begin{aligned} \ddot{y}_1 + \varepsilon\lambda(\dot{y}_1 - \dot{y}_2) + (1+\varepsilon\sigma)y_1 + \tfrac{4}{3}\varepsilon(y_1 - y_2)^3 &= \varepsilon A\delta(t) \\ \varepsilon\ddot{y}_2 + \varepsilon\lambda(\dot{y}_2 - \dot{y}_1) + \tfrac{4}{3}\varepsilon(y_2 - y_1)^3 &= 0, \end{aligned} \tag{5.1}$$

was compared with the system with nonlinear damping given by

$$\begin{aligned} \ddot{y}_1 + \varepsilon\lambda(\dot{y}_1 - \dot{y}_2)^3 + (1+\varepsilon\sigma)y_1 + \tfrac{4}{3}\varepsilon(y_1 - y_2)^3 &= \varepsilon A\delta(t) \\ \varepsilon\ddot{y}_2 + \varepsilon\lambda(\dot{y}_2 - \dot{y}_1)^3 + \tfrac{4}{3}\varepsilon(y_2 - y_1)^3 &= 0. \end{aligned} \tag{5.2}$$

As shown by equations (5.1) and (5.2), the systems are initially excited by an impulse of magnitude εA. Additionally, y_1 represents the displacement of the primary system, and y_2 represents the displacement of the NES. In order to help show the effectiveness of the NES, it is of interest to determine the energy of the mass corresponding to each of these

components. Letting E_1 represent the energy of the primary system and E_2 represent the energy of the NES, the energies of the system with linear damping and the system with nonlinear damping are both given by

$$E_1 = \frac{1}{2}\dot{y_1}^2 + \frac{1}{2}(1+\varepsilon\sigma)y_1^2 + \frac{1}{3}\varepsilon(y_1 - y_2)^4$$
$$E_2 = \frac{1}{2}\varepsilon\dot{y_2}^2 + \frac{1}{3}\varepsilon(y_2 - y_1)^4. \tag{5.3}$$

For performance comparisons between the linearly and nonlinearly damped systems, plots of the displacement and energy as functions of time were generated. The Runge-Kutta 4 method was used to perform the numerical integration in MATLAB. Figures 5-1 through 5-5 show the response and energy versus time for varying impulse amplitudes with fixed λ = 0.2, ε = 0.05, and σ = 0.5. The responses corresponding to the lowest impulse amplitude are presented in Figure 5-1, and each subsequent figure represents the responses to increasingly higher forcing amplitude. Custom time intervals were chosen for each plot in order to visually observe the differences between the linearly damped and nonlinearly damped NES results.

For an impulse amplitude of $5x10^2$, Figure 5-1 shows that the linearly damped NES clearly outperforms the nonlinearly damped NES. The displacement of the main system in the chosen time interval of 100 to 200 seconds is lower for the linearly damped NES than for the nonlinearly damped NES. More notably, the plot of system energy unambiguously shows that the energy in the system with the linearly damped attachment is lower than its nonlinearly damped counterpart.

The time responses of the system subjected to an impulse amplitude of $1x10^3$ are presented in Figure 5-2. For the time interval from zero to 80 seconds, greater displacement (in general) of the primary system is observed in the case of nonlinear

damping as opposed to the linear damping case. Additionally, the displacement of the NES is shown to be greater for the nonlinearly damped attachment as well. Initially the same, the energy of the primary system in the linear case is clearly lower than that of the nonlinear case after about 25 seconds in the motion. Energy of the linearly damped NES is lower at almost each instant than that for the nonlinearly damped NES. The effects of the energy reduction are evident due to the decreased displacement of the main system.

As the amplitude of the impulsive force is increased, the differences between the linearly damped and nonlinearly damped responses become less. Eventually, once the amplitude is increased beyond a certain value, the nonlinearly damped system outperforms the linearly damped system. As inferred from inspection of Figures 5-3 through 5-5, this transition occurs near $A = 5\text{x}10^3$. In Figure 5-3, with A near the transition value, the displacements of each system are shown to be very close to one another. The energies of each system are also shown to be very similar, but with the nonlinearly damped NES system having slightly less energy in the main system and NES than the linearly damped NES system.

The plots of displacement and energy versus time shown in Figure 5-4 clearly show that the nonlinearly damped NES outperforms the linearly damped NES for the amplitude $A = 1\text{x}10^4$. The trend toward increased vibration mitigation is further observed in Figure 5-5 for increasing the amplitude to $5\ \text{x}10^4$. The responses shown in Figure 5-5 clearly show the displacements decreasing faster for the nonlinearly damped NES versus the linearly damped NES.

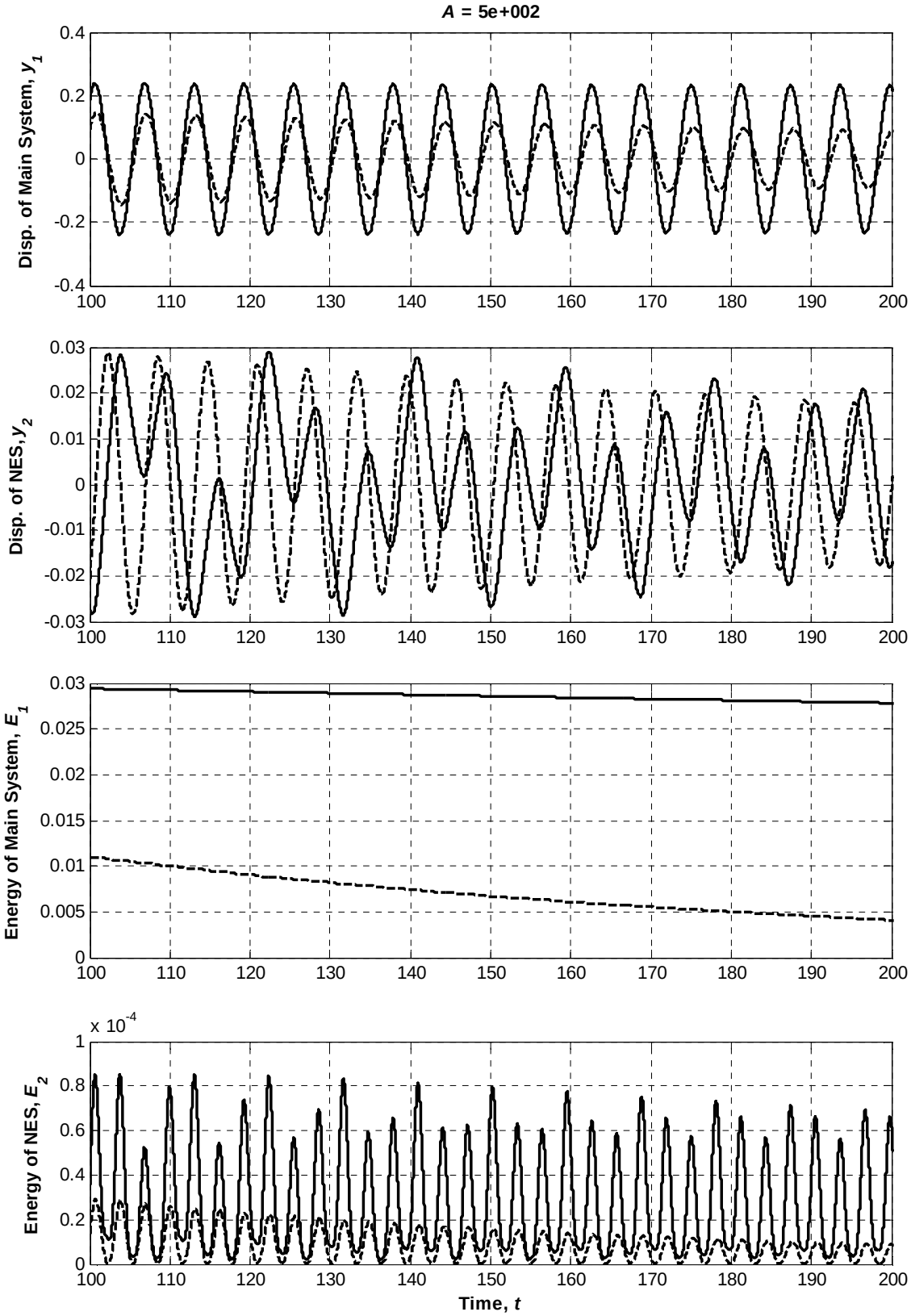


Figure 5-1. Displacement and energy versus time for $A = 5x10^2$
Linear damping: – – – ; Nonlinear Damping: ——

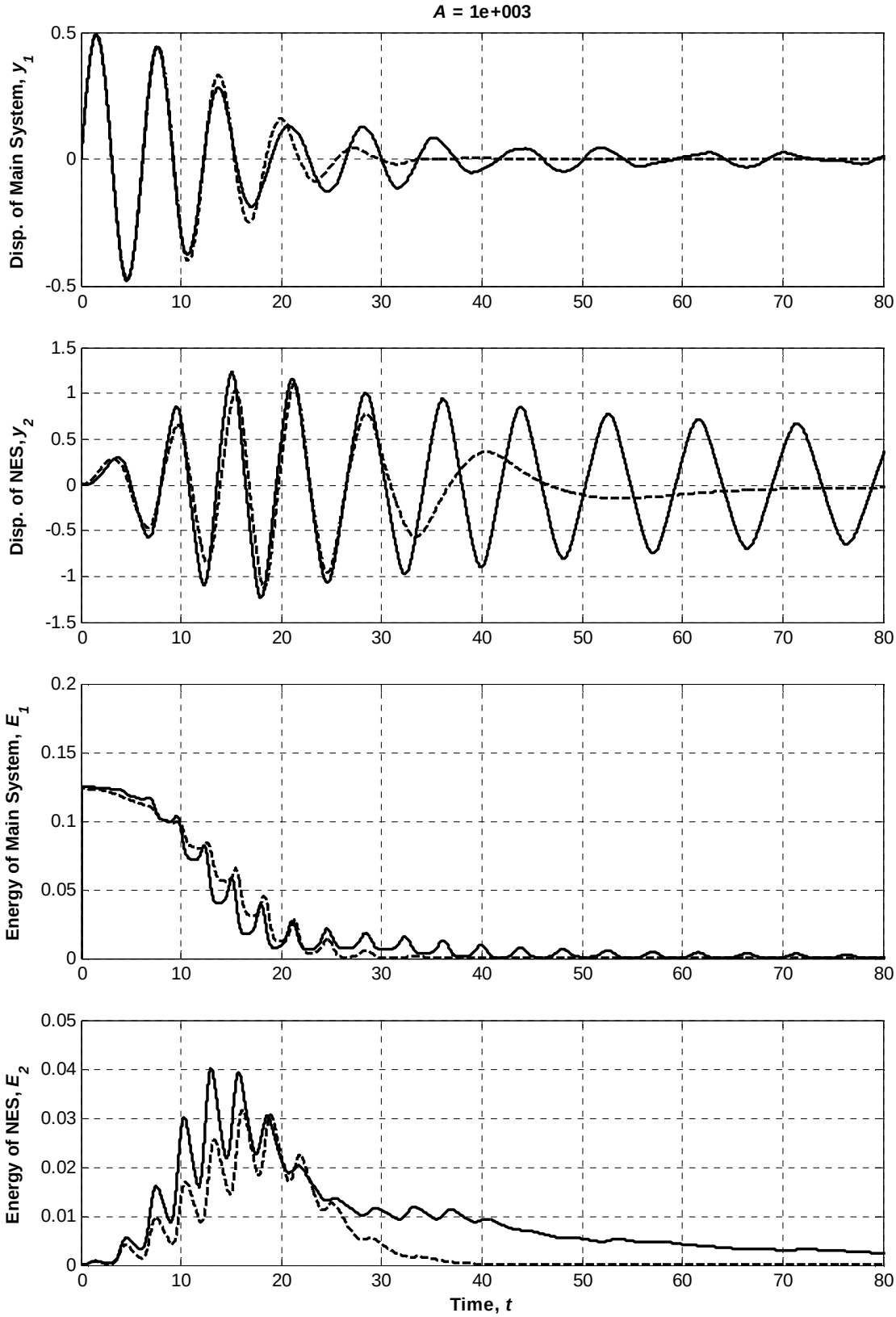


Figure 5-2. Displacement and energy versus time for $A = 1\text{x}10^3$
Linear damping: – – – ; Nonlinear Damping: ——

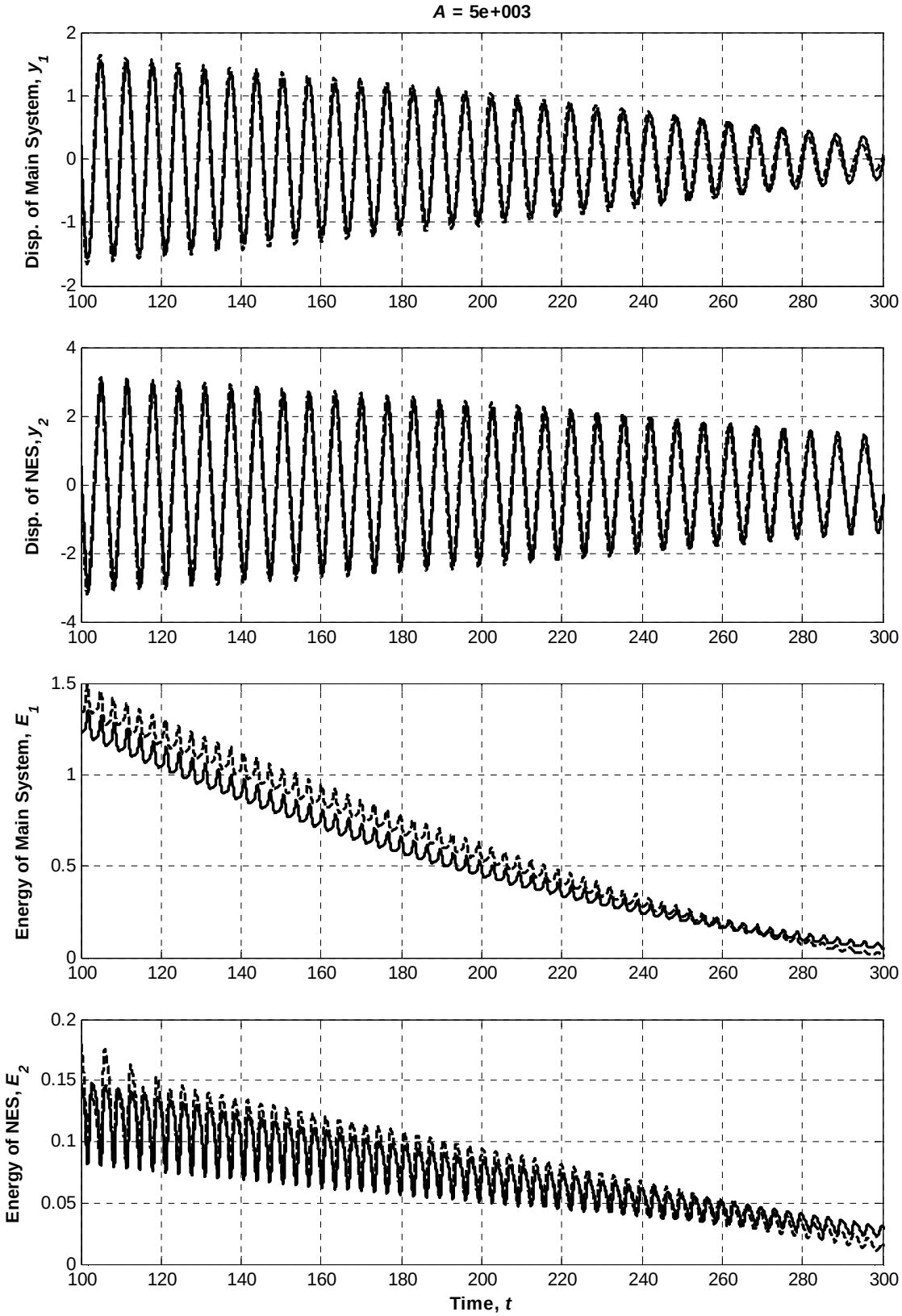


Figure 5-3. Displacement and energy versus time for $A = 5x10^3$
Linear damping: – – – ; Nonlinear Damping: ——

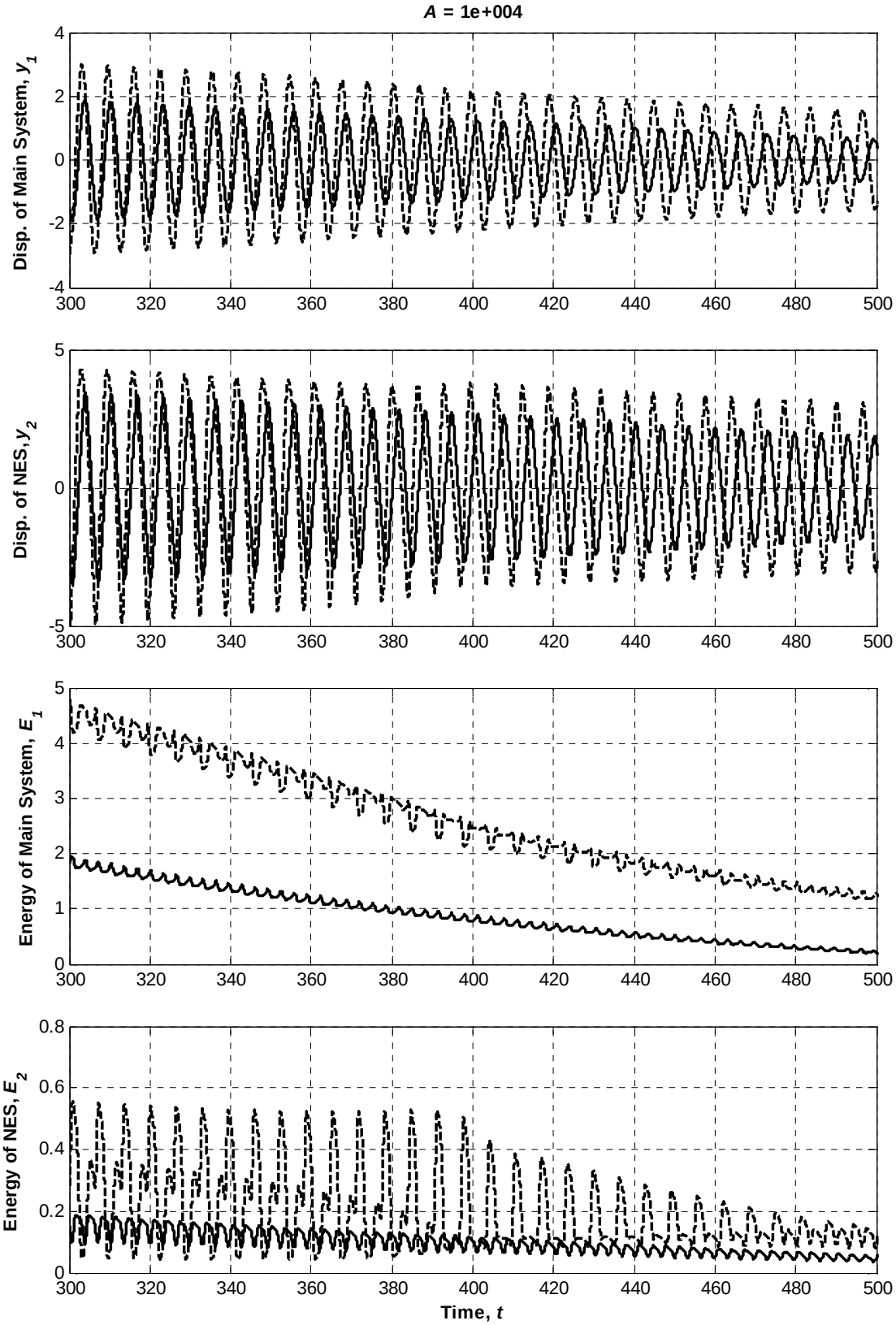


Figure 5-4. Displacement and energy versus time for $A = 1x10^4$
Linear damping: – – – ; Nonlinear Damping: ———

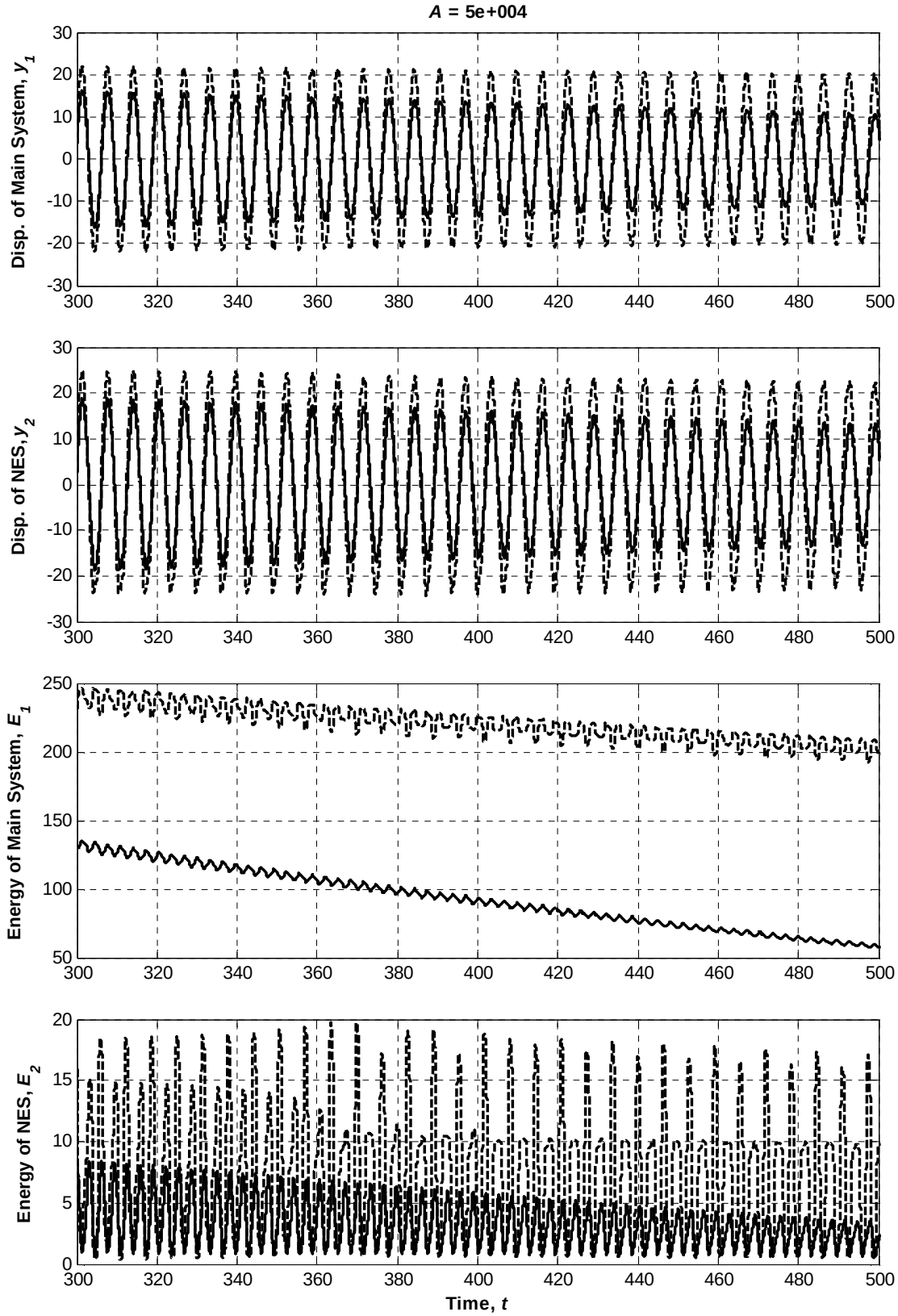


Figure 5-5. Displacement and energy versus time for $A = 5x10^4$
Linear damping: – – – ; Nonlinear Damping: ——

5.3 System Performance Comparisons Using Poincaré Maps

5.3.1 Background on Poincaré Sections and Maps

The purpose of using Poincaré maps is "to transform complicated behavior in the phase space to discrete maps in a lower-dimensional space" (Lynch, 2004). In essence, Poincaré maps provide a simplified representation of the dynamics. Lynch (2004) gives a simple example to introduce the concept of Poincaré maps and sections. As seen in Figure 5-6, the trajectory starting from r_0 on Σ leads to the point r_1 on Σ. The Poincaré section in this example is represented by the line segment Σ. This line segment is considered a Poincaré section because Σ is "crossed transversely (no trajectories are tangential to Σ)" (Lynch, 2004). Since the trajectory from r_0 to r_1 does cross Σ transversely, Σ can be considered as a Poincaré section. If the part of the trajectory not crossing Σ is removed from Figure 5-6, the Poincaré section will show only the two points of intersection. A Poincaré map is simply the function used to relate points on a Poincaré section. In this example by Lynch (2004), the Poincaré map is given by

$$r_{n+1} = \boldsymbol{P}(r_n), \tag{5.4}$$

where $\boldsymbol{P}$ maps Σ into itself. In addition, in the case of

$$r_n = \boldsymbol{P}(r_n), \tag{5.5}$$

r_n is considered a "*fixed point of period one*" (Lynch, 2004) since the point is stationary for each iteration.

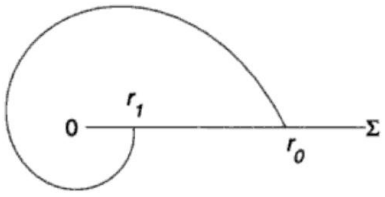


Figure 5-6. Example of a Poincaré section (Lynch, 2004); Courtesy of Google Books

The simplification achieved by using Poincaré maps to represent trajectories in the phase-space is illustrated in an example by Lynch (2004) for the system

$$\ddot{x} + k\dot{x} + (x^3 - x) = \Gamma\cos(\omega t). \tag{5.6}$$

The system is rewritten in state-space as

$$\begin{aligned} \dot{x} &= y \\ \dot{y} &= x - ky - x^3 + \Gamma\cos(\omega t), \end{aligned} \tag{5.7}$$

as shown by Lynch (2004). The Poincaré section in this example is a cross section of a torus at an angle of nπ (n = 0, 2, 4 …). Hence, this Poincaré section corresponds to the period of the harmonic forcing in the system. Lynch (2004) explicitly includes this angle in the system equations as follows:

$$\begin{aligned} \dot{x} &= y \\ \dot{y} &= x - ky - x^3 + \Gamma\cos(\theta), \\ \dot{\theta} &= \omega. \end{aligned} \tag{5.8}$$

Figures 5-7 through 5-12 were generated for varying Γ and keeping k = 0.3 and ω = 1.25 constant. Figures 5-7 and 5-8 show the case for which the period is $2\pi/\omega$, represented as a closed orbit on the phase portrait and a single point on the Poincaré map. This single point corresponds to where the Poincaré section is crossed by the phase trajectory of the system. Since this trajectory has only a single orbit within one period, the corresponding Poincaré map representation is simply one point.

Figures 5-9 and 5-10 show a closed phase trajectory with a period of $4\pi/\omega$. As evident from the phase portrait, two orbits are completed for this period. Thus, two points are plotted in the Poincaré map to represent that the Poincaré section was crossed twice before the trajectory repeated itself.

Figures 5-11 and 5-12 are shown to further illustrate the simplifications achieved by using Poincaré maps versus phase portraits. The seemingly complex trajectory in the phase space is greatly reduced in the Poincaré map, while still maintaining information regarding the periodic behavior of the system. In each of these cases, the Poincaré maps would appear different if a different Poincaré section was chosen.

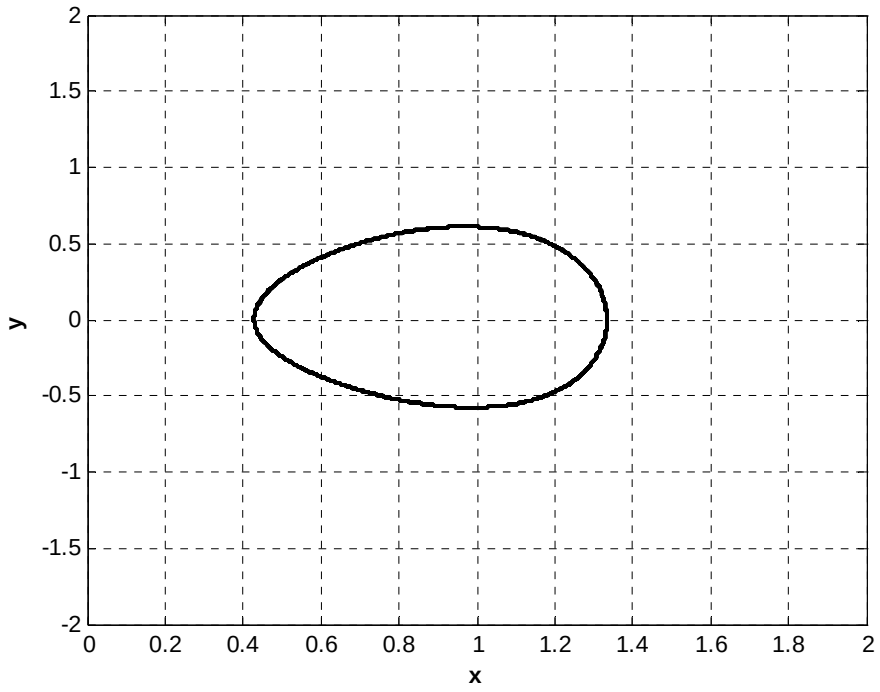


Figure 5-7. Phase portrait for $\Gamma = 0.2$ (Lynch, 2004)

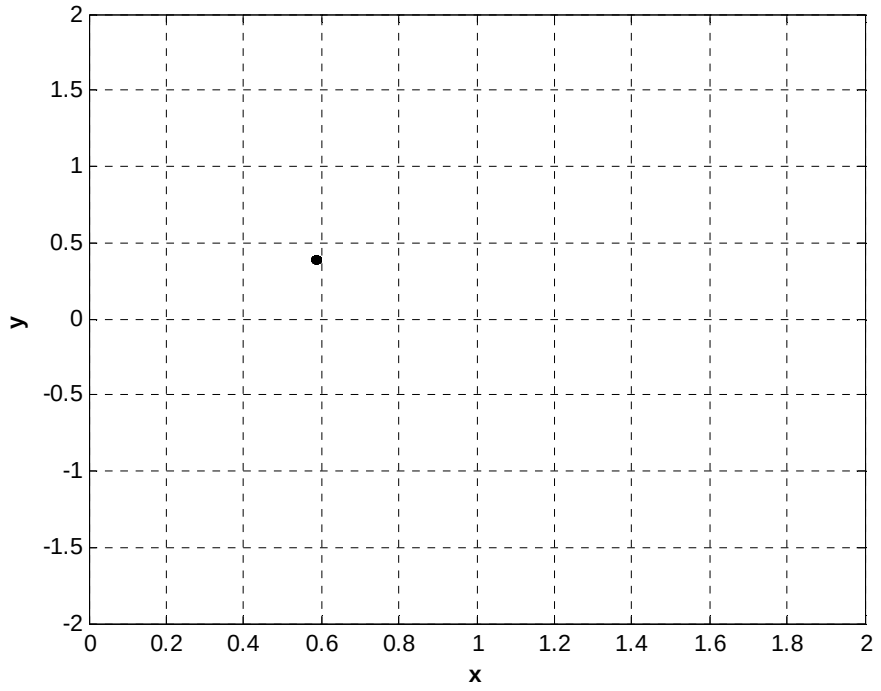


Figure 5-8. Poincaré map for Γ = 0.2 (Lynch, 2004)

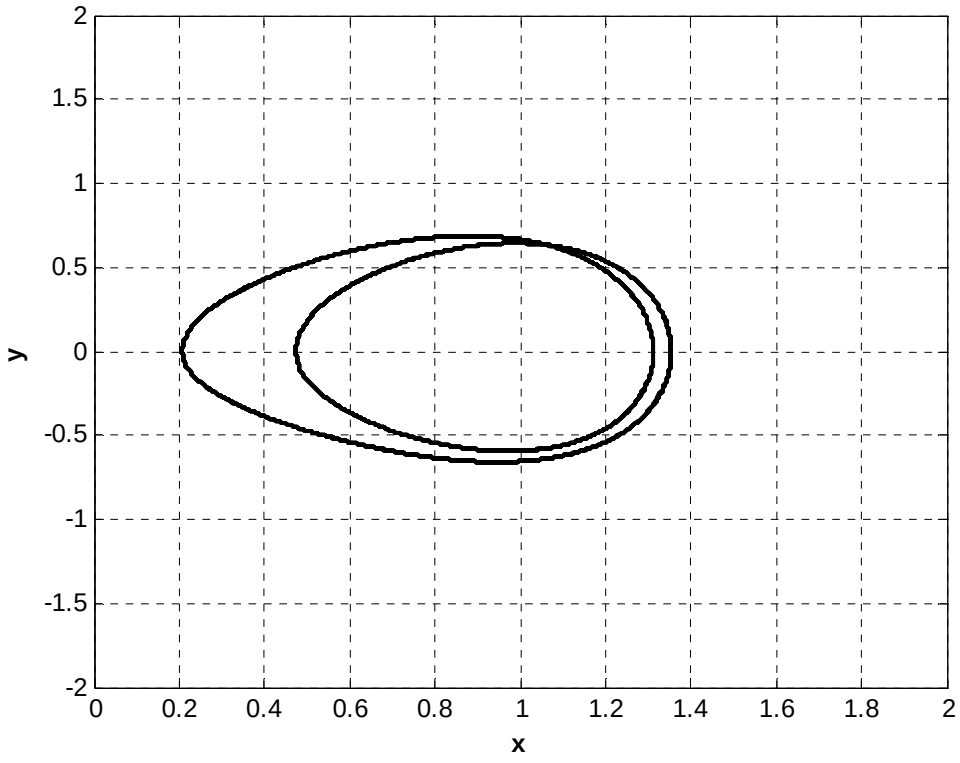


Figure 5-9. Phase portrait for Γ = 0.3 (Lynch, 2004)

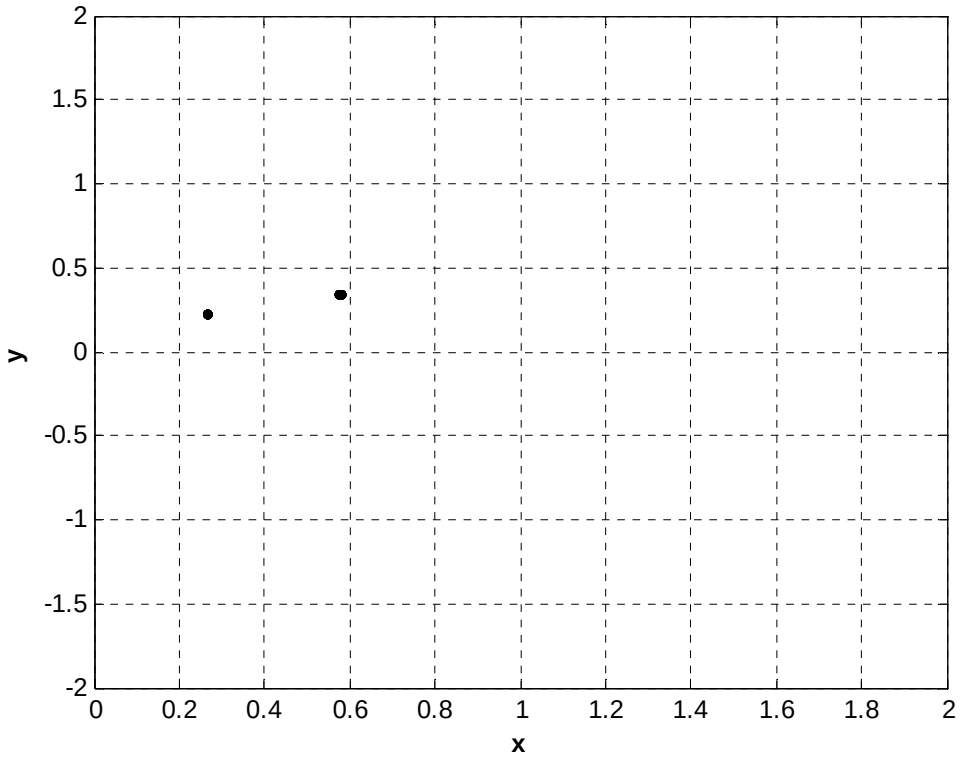


Figure 5-10. Poincaré map for Γ = 0.3 (Lynch, 2004)

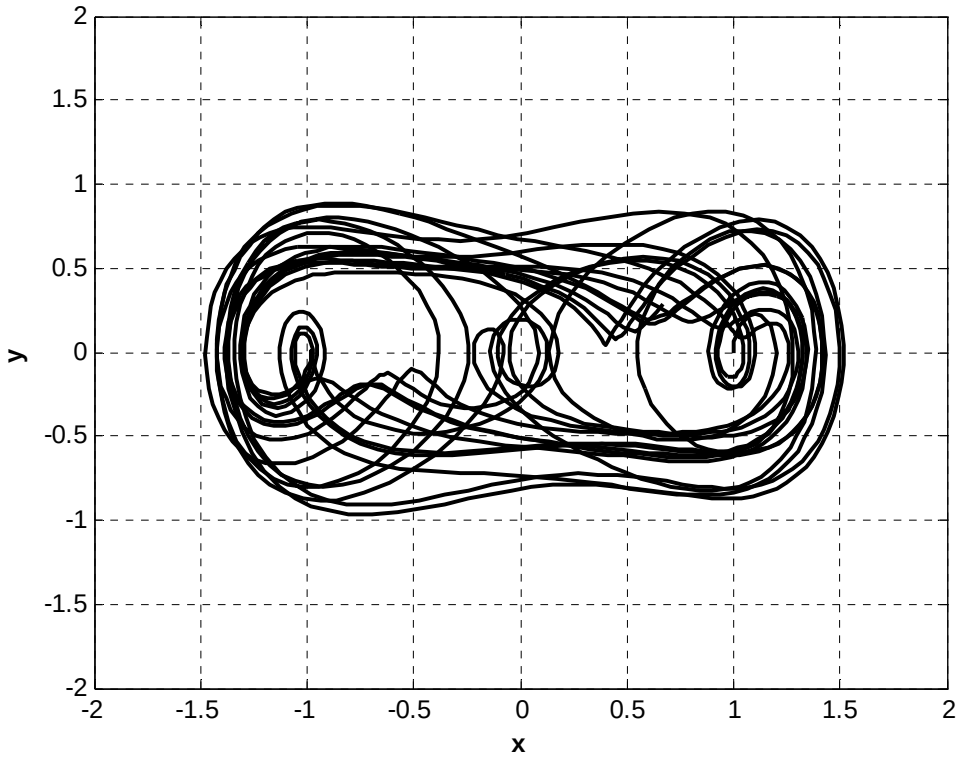


Figure 5-11. Phase portrait for Γ = 0.5 (Lynch, 2004)

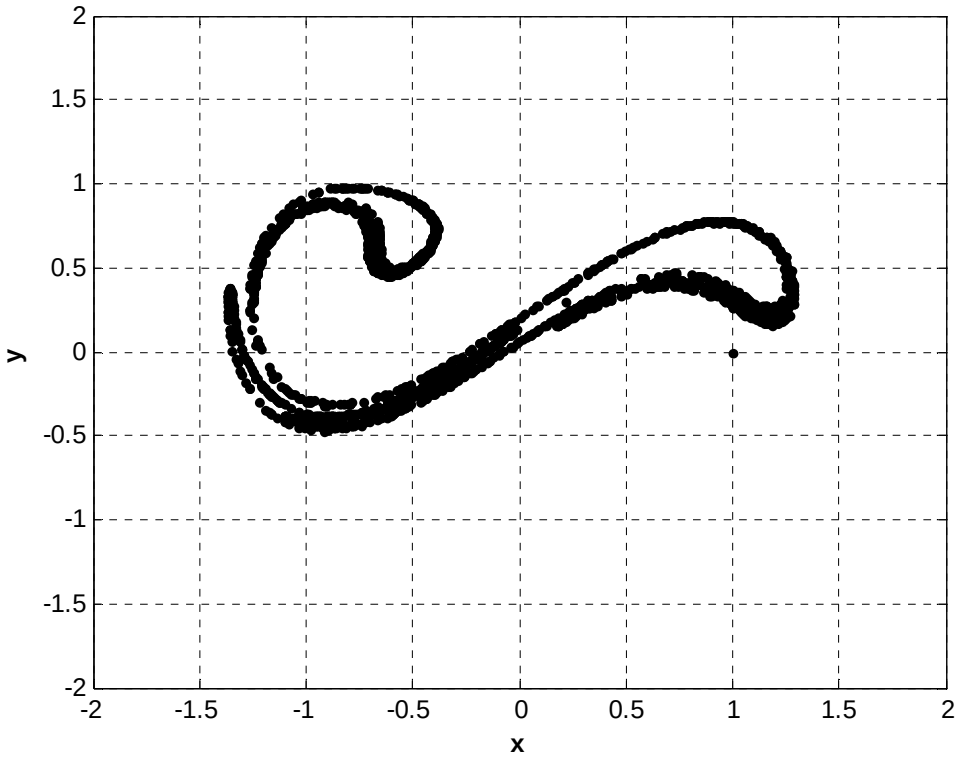


Figure 5-12. Poincaré map for Γ = 0.5 (Lynch, 2004)

5.3.2 Analysis of the Linearly and Nonlinearly Damped Systems using Phase Portraits and Poincaré Maps

The linearly damped system under consideration is

$$\begin{aligned} \ddot{y_1} + \varepsilon\lambda(\dot{y_1} - \dot{y_2}) + (1 + \varepsilon\sigma)y_1 + \tfrac{4}{3}\varepsilon(y_1 - y_2)^3 &= \varepsilon A\cos t \\ \varepsilon\ddot{y_2} + \varepsilon\lambda(\dot{y_2} - \dot{y_1}) + \tfrac{4}{3}\varepsilon(y_2 - y_1)^3 &= 0, \end{aligned} \tag{5.9}$$

and the nonlinearly damped system under consideration is

$$\begin{aligned} \ddot{y_1} + \varepsilon\lambda(\dot{y_1} - \dot{y_2})^3 + (1 + \varepsilon\sigma)y_1 + \tfrac{4}{3}\varepsilon(y_1 - y_2)^3 &= \varepsilon A\cos t \\ \varepsilon\ddot{y_2} + \varepsilon\lambda(\dot{y_2} - \dot{y_1})^3 + \tfrac{4}{3}\varepsilon(y_2 - y_1)^3 &= 0. \end{aligned} \tag{5.10}$$

In order to generate the corresponding Poincaré maps, the system equations had to be converted to state space form. By letting

$$x_1 = y_1,\ x_2 = \dot{y_1},\ x_3 = y_2, \text{ and } x_4 = \dot{y_2}, \tag{5.11}$$

the following state space forms were created for the linearly damped and nonlinearly damped systems, respectively:

$$\begin{aligned} \dot{x_1} &= x_2 \\ \dot{x_2} &= \varepsilon A \cos t - \varepsilon\lambda(x_2 - x_4) - (1 + \varepsilon\sigma)x_1 - \tfrac{4}{3}\varepsilon(x_1 - x_3)^3 \\ \dot{x_3} &= x_4 \\ \dot{x_4} &= -\lambda(x_4 - x_2) - \tfrac{4}{3}(x_3 - x_1)^3 \end{aligned} \tag{5.12}$$

and

$$\begin{aligned} \dot{x_1} &= x_2 \\ \dot{x_2} &= \varepsilon A \cos t - \varepsilon\lambda(x_2 - x_4)^3 - (1 + \varepsilon\sigma)x_1 - \tfrac{4}{3}\varepsilon(x_1 - x_3)^3 \\ \dot{x_3} &= x_4 \\ \dot{x_4} &= -\lambda(x_4 - x_2)^3 - \tfrac{4}{3}(x_3 - x_1)^3. \end{aligned} \tag{5.13}$$

Using MATLAB, the phase portraits and Poincaré maps were generated for the linearly damped and nonlinearly damped systems using fixed values of λ = 0.2, ε = 0.05, σ = 0.5, A = 0.4, and ω = 1. Figures 5-13 and 5-15 (linearly damped case) and Figures 5-17 and 5-19 (nonlinearly damped case) show the phase portraits of the linear oscillator and NES, respectively. Each phase portrait was created for a time interval of [0 500] and with the primary system and NES initially at rest. Additionally, Figures 5-14 and 5-16 (linearly damped case) and Figures 5-18 and 5-20 (nonlinearly damped case) show the Poincaré maps of the linear oscillator and NES, respectively. All Poincaré maps were generated for time intervals of $\left[0, \frac{2000\pi}{\omega}\right]$ and with the primary system and NES initially at rest.

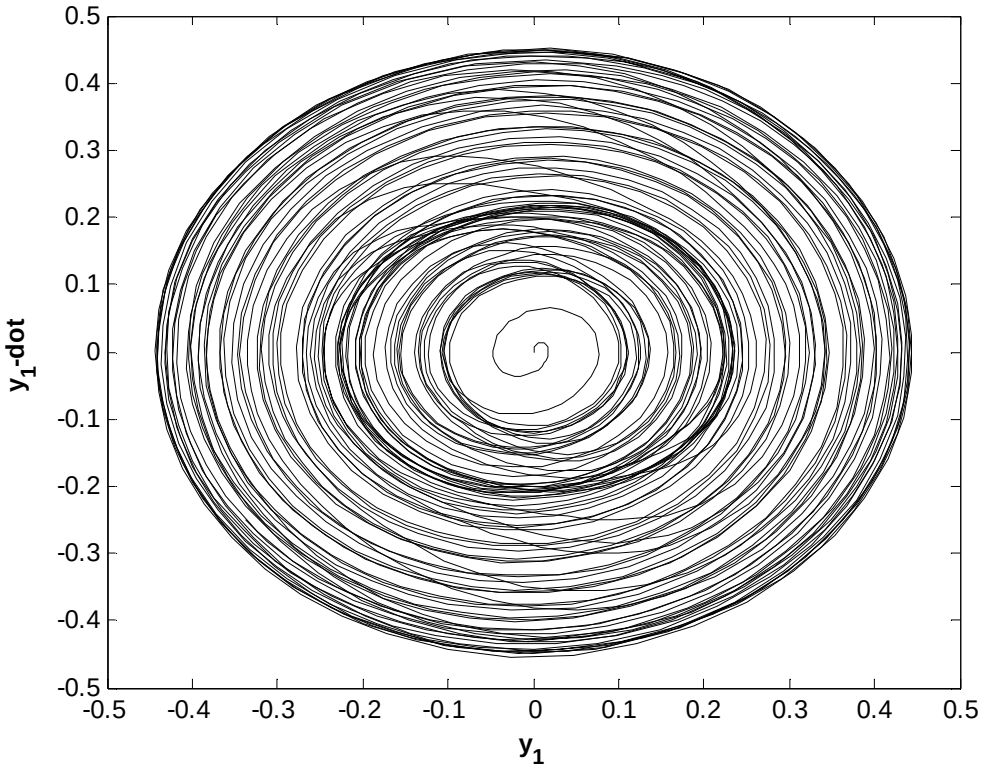


Figure 5-13. Phase portrait for the primary system in the case of linear damping

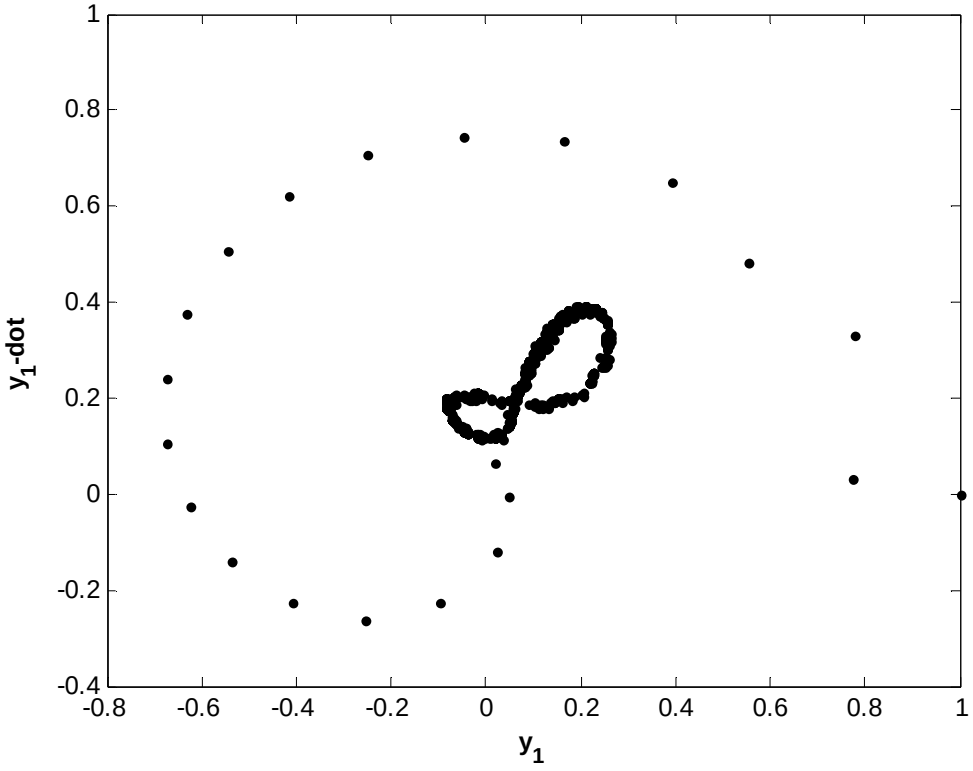


Figure 5-14. Poincaré map for the primary system in the case of linear damping

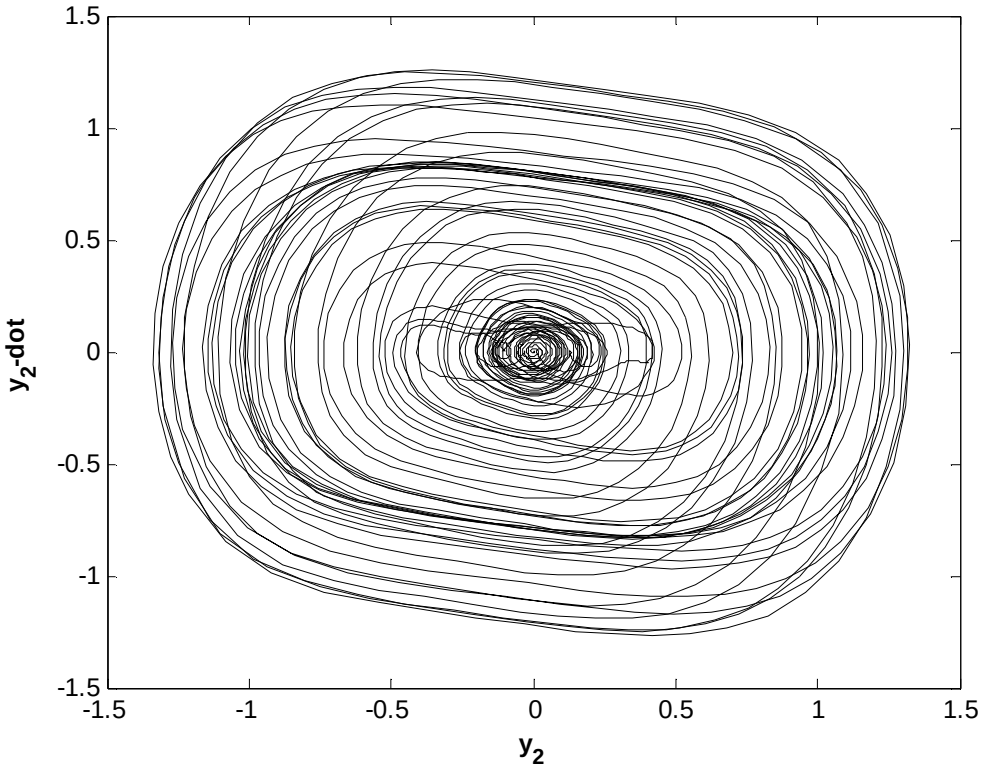


Figure 5-15. Phase portrait for the NES in the case of linear damping

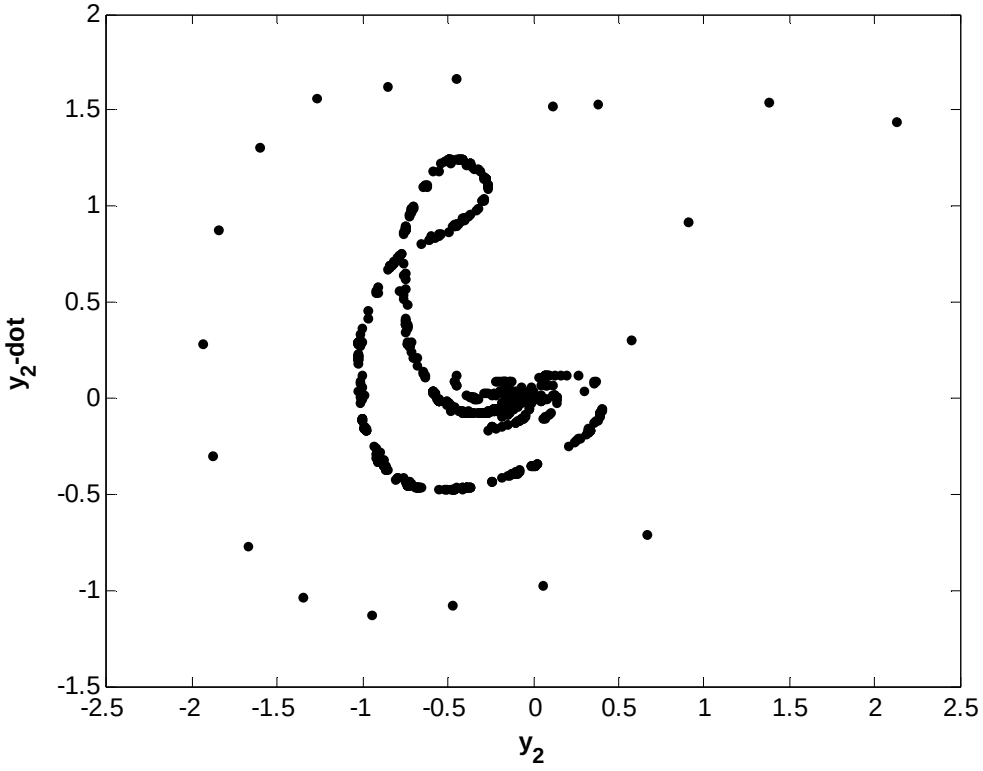


Figure 5-16. Poincaré map for the NES in the case of linear damping

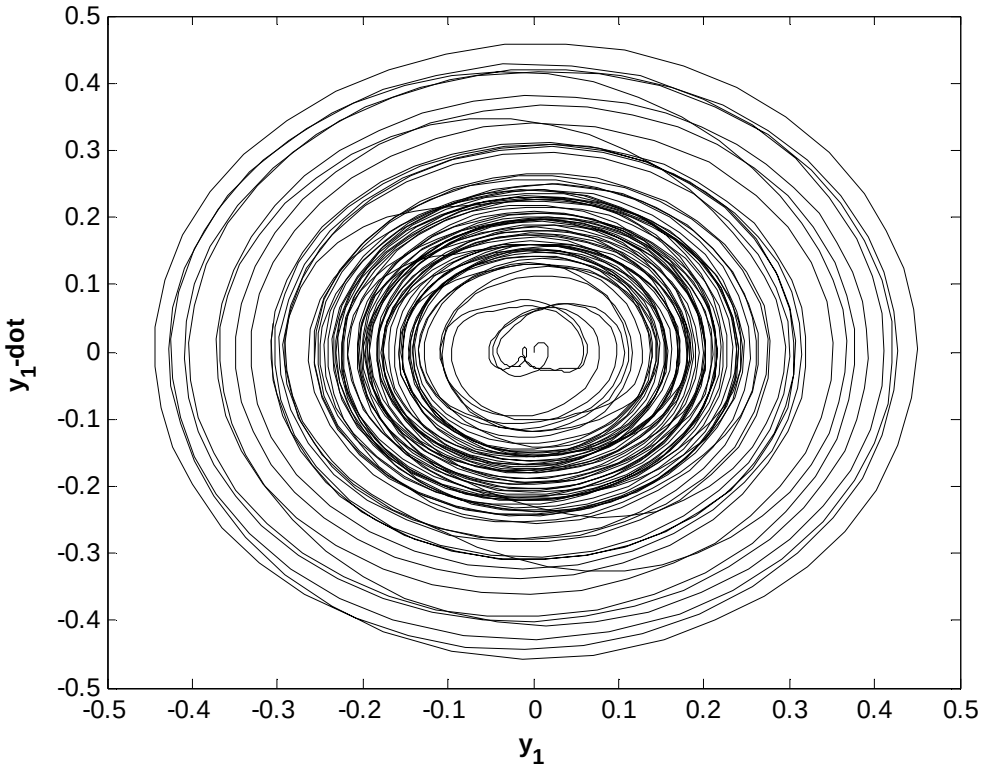


Figure 5-17. Phase portrait for the primary system in the case of nonlinear damping

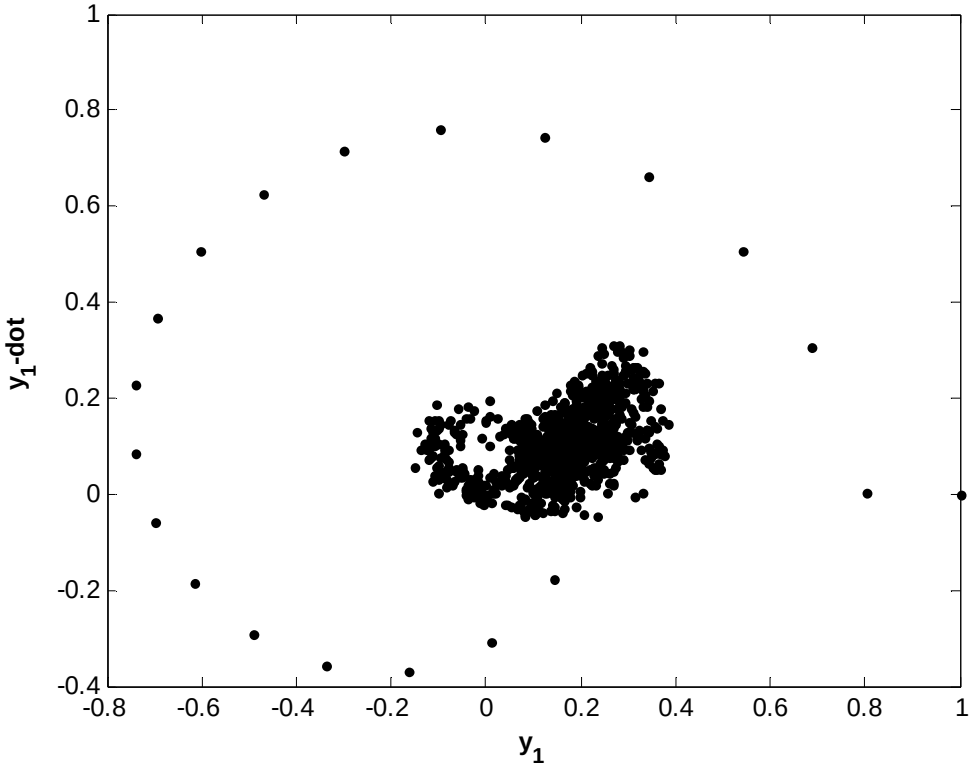


Figure 5-18. Poincaré map for the primary system in the case of nonlinear damping

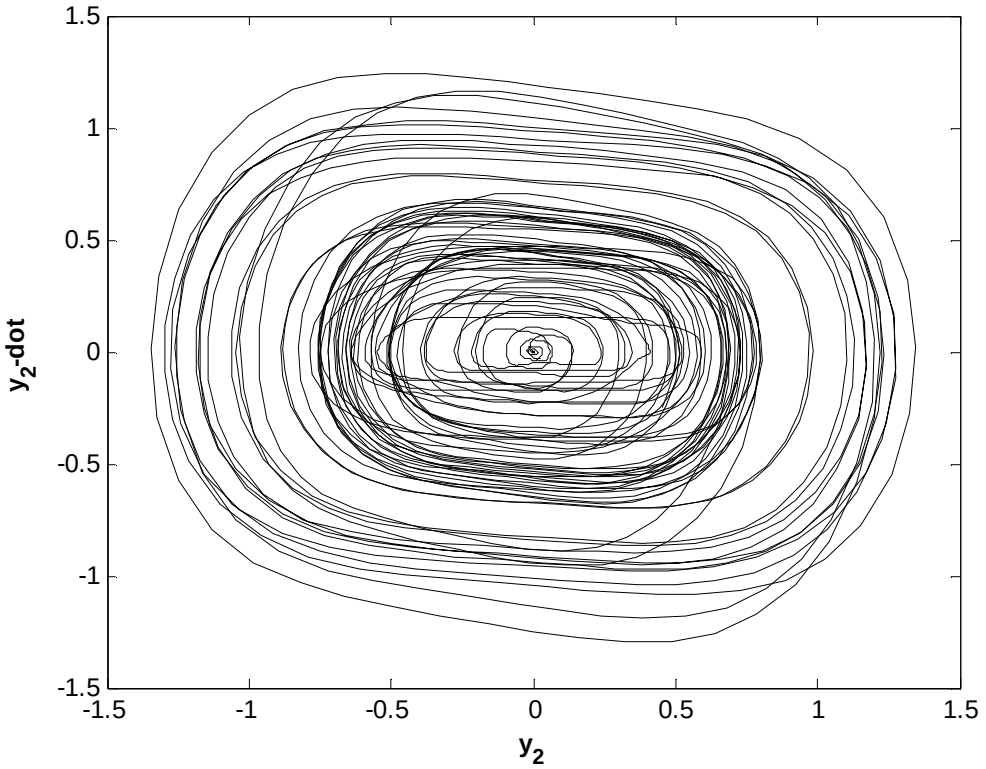


Figure 5-19. Phase portrait for the NES in the case of nonlinear damping

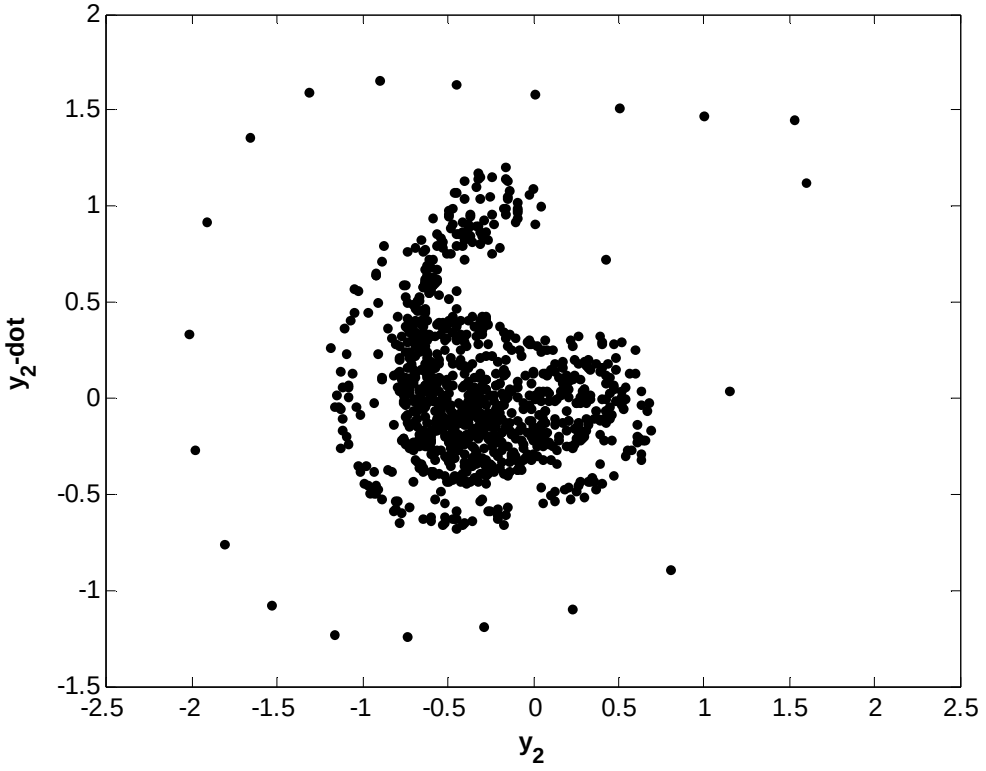


Figure 5-20. Poincaré map for the NES in the case of nonlinear damping

5.3.3 Discussion of Phase Portraits and Poincaré Maps of the Linearly and Nonlinearly Damped Systems

As seen from the Poincaré maps, the information in the phase portraits is greatly simplified. In the linearly damped case, the orbits in the state space have been reduced to a "figure-8" shape on the Poincaré map. Similarly, the orbits from the state space of the linearly damped NES are represented in a simplified shape when sampled in increments of the period of oscillation as shown on the Poincaré map.

In the case of nonlinear damping, the phase portraits of the linear oscillator show more orbits close to the origin than in the linearly damped case. The conclusion to be drawn from these closely spaced orbits is that the displacements and velocities of the nonlinearly damped system are less than those for the linearly damped system for the considered parameters in the case presently studied (harmonic forcing near resonance). Although the phase portrait of the linearly damped NES shows orbits that appear to be closely spaced near the origin, the orbits of the nonlinearly damped NES are also closely spaced, but located farther from the origin, indicating that energy pumping is taking place from the primary system to the nonlinearly damped NES. In addition, the Poincaré maps corresponding to the linear oscillator and NES in the nonlinearly damped case show points more tightly packed than those for the linearly damped system, further reinforcing the previous claim. Note that the structure of the phase portraits and Poincaré maps remains the same in both linearly and nonlinearly damped systems, but the local density of orbits (phase portraits) and points (Poincaré maps) varies between the two cases.

5.4 Conclusions

This chapter has presented the time-dependent performance of the linearly and nonlinearly damped systems. First, the time response of the systems to impulse loading was investigated. From these results, it was evident that the nonlinearly damped NES is more effective than the linearly damped counterpart only if the magnitude of the impulse is sufficiently high. Consequently, if the magnitude of impulse if relatively low, the linearly damped system has a faster reduction in displacement.

The second part of this chapter focused on comparing the phase portraits and Poincaré maps between the two systems. In the case of harmonic forcing near resonance ($\omega = 1 \approx \sqrt{(1 + \varepsilon\sigma)/1}$ for small ε), the chosen parameters result in the nonlinearly damped system having lower displacements and velocities than the linearly damped system. Under these conditions, the nonlinearly damped NES outperforms the linearly damped NES in regards to vibration attenuation.

5.A Appendix

5.A.1 MATLAB Code for Time Response Simulations

```
%Comparison of system with linear damping and system with nonlinear
damping
%when subjecting to varying impulsive loads

close all
clear all

sigma = 0.5; e = 0.05; lambda = 0.2; m1 = 1; m2 = e;

for A = [5e2 1e3 5e3 1e4 5e4]  %Amplitude of impulsive force
    k = 1; %iteration constant
    T = 0.01; %time step
    t_end = 1000; %end of time interval

    y1 = 0; y2 = 0; y1_NL = 0; y2_NL = 0;
    t_vec = zeros(1,t_end/T + 1);
    Y1 = t_vec; Y2 = t_vec; Y1_NL = t_vec; Y2_NL = t_vec;
    E1 = t_vec; E1_NL = t_vec; E2 = t_vec; E2_NL = t_vec;
    Y1(1) = y1; Y2(1) = y2; Y1_NL(1) = y1; Y2_NL(1) = y2;
    z1 = 0; z2 = 0; z3 = 0; z4 = 0;
    z1_NL = 0; z2_NL = 0; z3_NL = 0; z4_NL = 0;

    %Runge Kutta 4
    %From Fundamentals of Vibrations by Leonard Meirovitch, p. 677-679:

    for t = T:T:t_end
        k = k+1; t_vec(k) = t;

        %Condition for impulsive force:

        if k == 2
            F=A;
        else
            F=0;
        end

        %Linear Damping:

        f11 = z2;
        f12 = e*F - e*lambda*(z2-z4) - (1+e*sigma)*z1 - ...
            (4/3)*e*(z1-z3)^3;
        f13 = z4;
        f14 = -lambda*(z4-z2) - (4/3)*(z3-z1)^3;

        g11 = T*f11; g12 = T*f12; g13 = T*f13; g14 = T*f14;
```

```
        f21 = z2 + 1/2*g12;
        f22 = e*F - e*lambda*((z2+1/2*g12)-(z4+1/2*g14)) - ...
            (1+e*sigma)*(z1+1/2*g11) - (4/3)*e*((z1+1/2*g11) - ...
            (z3+1/2*g13))^3;
        f23 = z4 + 1/2*g14;
        f24 = -lambda*((z4+1/2*g14)-(z2+1/2*g12)) - ...
            (4/3)*((z3+1/2*g13)-(z1+1/2*g11))^3;

        g21 = T*f21; g22 = T*f22; g23 = T*f23; g24 = T*f24;


        f31 = z2 + 1/2*g22;
        f32 = e*F - e*lambda*((z2+1/2*g22)-(z4+1/2*g24)) - ...
            (1+e*sigma)*(z1+1/2*g21) - (4/3)*e*((z1+1/2*g21) - ...
            (z3+1/2*g23))^3;
        f33 = z4 + 1/2*g24;
        f34 = -lambda*((z4+1/2*g24)-(z2+1/2*g22)) - ...
            (4/3)*((z3+1/2*g23) - (z1+1/2*g21))^3;

        g31 = T*f31; g32 = T*f32; g33 = T*f33; g34 = T*f34;


        f41 = z2 + g32;
        f42 = e*F - e*lambda*((z2+g32)-(z4+g34)) - ...
            (1+e*sigma)*(z1+g31) - (4/3)*e*((z1+g31)-(z3+g33))^3;
        f43 = z4 + g34;
        f44 = -lambda*((z4+g34)-(z2+g32)) - (4/3)*((z3+g33) - ...
            (z1+g31))^3;

        g41 = T*f41; g42 = T*f42; g43 = T*f43; g44 = T*f44;


        z1 = z1 + 1/6*(g11 + 2*g21 + 2*g31 + g41);
        z2 = z2 + 1/6*(g12 + 2*g22 + 2*g32 + g42);
        z3 = z3 + 1/6*(g13 + 2*g23 + 2*g33 + g43);
        z4 = z4 + 1/6*(g14 + 2*g24 + 2*g34 + g44);

        y1 = z1; y2 = z3;
        Y1(k) = y1; Y2(k) = y2;

        %Energy of mass 1 (primary system):
        E1(k) = 1/2*m1*z2^2 + 1/2*(1 + e*sigma)*y1^2 + 1/3*e*(y1 -
y2)^4;

        %Energy of mass 2 (NES):
        E2(k) = 1/2*m2*z4^2 + 1/3*e*(y2 - y1)^4;



%*****************************************************************

        %Nonlinear Damping:

        f11_NL = z2_NL;
```

```
        f12_NL = e*F - e*lambda*(z2_NL-z4_NL)^3 - (1+e*sigma)*z1_NL -
...
            (4/3)*e*(z1_NL-z3_NL)^3;
        f13_NL = z4_NL;
        f14_NL = -lambda*(z4_NL-z2_NL)^3 - (4/3)*(z3_NL-z1_NL)^3;

        g11_NL = T*f11_NL; g12_NL = T*f12_NL; g13_NL = T*f13_NL;
        g14_NL = T*f14_NL;


        f21_NL = z2_NL + 1/2*g12_NL;
        f22_NL = e*F - e*lambda*((z2_NL+1/2*g12_NL) - ...
            (z4_NL+1/2*g14_NL))^3 - (1+e*sigma)*(z1_NL+1/2*g11_NL) -
...
            (4/3)*e*((z1_NL+1/2*g11_NL) - (z3_NL+1/2*g13_NL))^3;
        f23_NL = z4_NL + 1/2*g14_NL;
        f24_NL = -lambda*((z4_NL+1/2*g14_NL) - (z2_NL+1/2*g12_NL))^3 -
...
            (4/3)*((z3_NL+1/2*g13_NL)-(z1_NL+1/2*g11_NL))^3;

        g21_NL = T*f21_NL; g22_NL = T*f22_NL; g23_NL = T*f23_NL;
        g24_NL = T*f24_NL;


        f31_NL = z2_NL + 1/2*g22_NL;
        f32_NL = e*F - e*lambda*((z2_NL+1/2*g22_NL) - ...
            (z4_NL+1/2*g24_NL))^3 - (1+e*sigma)*(z1_NL+1/2*g21_NL) -
...
            (4/3)*e*((z1_NL+1/2*g21_NL) - (z3_NL+1/2*g23_NL))^3;
        f33_NL = z4_NL + 1/2*g24_NL;
        f34_NL = -lambda*((z4_NL+1/2*g24_NL)-(z2_NL+1/2*g22_NL))^3 -
...
            (4/3)*((z3_NL+1/2*g23_NL) - (z1_NL+1/2*g21_NL))^3;

        g31_NL = T*f31_NL; g32_NL = T*f32_NL; g33_NL = T*f33_NL;
        g34_NL = T*f34_NL;


        f41_NL = z2_NL + g32_NL;
        f42_NL = e*F - e*lambda*((z2_NL+g32_NL)-(z4_NL+g34_NL))^3 - ...
            (1+e*sigma)*(z1_NL+g31_NL) - (4/3)*e*((z1_NL+g31_NL) - ...
            (z3_NL+g33_NL))^3;
        f43_NL = z4_NL + g34_NL;
        f44_NL = -lambda*((z4_NL+g34_NL)-(z2_NL+g32_NL))^3 - ...
            (4/3)*((z3_NL+g33_NL) - (z1_NL+g31_NL))^3;

        g41_NL = T*f41_NL; g42_NL = T*f42_NL; g43_NL = T*f43_NL;
        g44_NL = T*f44_NL;


        z1_NL = z1_NL + 1/6*(g11_NL + 2*g21_NL + 2*g31_NL + g41_NL);
        z2_NL = z2_NL + 1/6*(g12_NL + 2*g22_NL + 2*g32_NL + g42_NL);
        z3_NL = z3_NL + 1/6*(g13_NL + 2*g23_NL + 2*g33_NL + g43_NL);
        z4_NL = z4_NL + 1/6*(g14_NL + 2*g24_NL + 2*g34_NL + g44_NL);
```

```
    y1_NL = z1_NL; y2_NL = z3_NL;
    Y1_NL(k) = y1_NL; Y2_NL(k) = y2_NL;

    %Energy of mass 1 (primary system):
    E1_NL(k) = 1/2*m1*z2_NL^2 + 1/2*(1 + e*sigma)*y1_NL^2 + ...
        1/3*e*(y1_NL - y2_NL)^4;

    %Energy of mass 2 (NES):
    E2_NL(k) = 1/2*m2*z4_NL^2 + 1/3*e*(y2_NL - y1_NL)^4;

end

if A == 1e3
    lim_x_min = 0; lim_x_max = 80; lim_y = 1;
elseif A == 5e3
    lim_x_min = 100; lim_x_max = 300; lim_y = 1;
elseif A == 1e4
    lim_x_min = 300; lim_x_max = 500; lim_y = 1;
elseif A == 5e4
    lim_x_min = 300; lim_x_max = 500; lim_y = 1;
else
    lim_x_min = 100; lim_x_max = 200; lim_y = 1;
end

figure('Units', 'inches', 'OuterPosition', [2 .3 7 10.7])
subplot(4,1,1, 'Units', 'inches', 'OuterPosition', ...
    [-.3, 7.5, 7.7, 2.3]);
plot(t_vec,Y1,'--k','LineWidth', 2)
axis([lim_x_min lim_x_max -lim_y lim_y])
axis 'auto y'
hold on; grid on;
plot(t_vec,Y1_NL,'-k','LineWidth', 2)
title(['\bf{\itA} = ', num2str(A,'%.0e')])
ylabel('\bfDisp. of Main System, \ity_1')

subplot(4,1,2, 'Units', 'inches', 'OuterPosition', ...
    [-.3, 5, 7.7, 2.3]);
plot(t_vec,Y2,'--k','LineWidth', 2)
axis([lim_x_min lim_x_max -lim_y lim_y])
axis 'auto y'
hold on; grid on;
plot(t_vec,Y2_NL,'-k','LineWidth', 2)
ylabel('\bfDisp. of NES, \ity_2')

subplot(4,1,3, 'Units', 'inches', 'OuterPosition', ...
    [-.3, 2.5, 7.7, 2.3]);
plot(t_vec,E1,'--k','LineWidth', 2)
axis([lim_x_min lim_x_max -lim_y lim_y])
axis 'auto y'
hold on; grid on;
plot(t_vec,E1_NL,'-k','LineWidth', 2)
ylabel('\bfEnergy of Main System, \itE_1')

subplot(4,1,4, 'Units', 'inches', 'OuterPosition', ...
```

```
        [-.3, 0, 7.7, 2.3]);
    plot(t_vec,E2,'--k','LineWidth', 2)
    axis([lim_x_min lim_x_max -lim_y lim_y])
    axis 'auto y'
    hold on; grid on;
    plot(t_vec,E2_NL,'-k','LineWidth', 2)
    xlabel('\bfTime, \itt')
    ylabel('\bfEnergy of NES, \itE_2')

end
```

5.A.2 MATLAB Code for Phase Portrait and Poincaré Examples

```
%Examples for Chapter 5
%Based on examples given by Lynch (2004)

close all
clear all

k = 0.3; w = 1.25;

for gamma = [0.2 0.3]
    x = 1; y = 0; theta = 0;
    ind = 0; ind2 = 0; p = 0; n = 2;
    dt = 0.001; t_end = 400; t_steady = t_end - 100; t_period =
t_steady;

    for t = dt:dt:t_end
        ind = ind + 1;
        x_dot = y;
        y_dot = x - k*y - x^3 + gamma*cos(theta);
        theta_dot = w;

        x = x + x_dot*dt;
        y = y + y_dot*dt;
        theta = theta + theta_dot*dt;

        if abs(theta - n*pi) < 0.001
            n = n + 2;
        else
        end

        if t > t_steady
            ind2 = ind2 + 1;
            X(ind2) = x; Y(ind2) = y;

            if abs(theta - n*pi) < 0.01
                p = p + 1;
                X_p(p) = X(ind2);
                Y_p(p) = Y(ind2);
            else
```

```
            end

        else
        end
    end

    %Plot State-Space:
    figure
    plot(X,Y,'-k','LineWidth',2)
    axis([0 2 -2 2]); grid on;
    xlabel('\bfx'); ylabel('\bfy');

    %Plot Poincare Map:
    figure
    plot(X_p,Y_p,'.k')
    axis([0 2 -2 2]); grid on;
    xlabel('\bfx'); ylabel('\bfy');
end

clear all

%The following code is based on those given by Lynch (2004):

state_space = inline('[x(2);x(1)-0.3*x(2)-(x(1))^3 +
0.5*cos(1.25*t)]','t','x');
options = odeset('RelTol',1e-4,'AbsTol',1e-4);
[t,x_ss] = ode45(state_space,[0 200],[1,0], options);
[t,x_p] = ode45(state_space,0:2*pi/1.25:4000*pi/1.25,[1,0]);

%Plot State-Space:
figure
plot(x_ss(:,1),x_ss(:,2),'-k','LineWidth',2)
axis([-2 2 -2 2]); grid on;
xlabel('\bfx'); ylabel('\bfy');

%Plot Poincare Map:
figure
plot(x_p(:,1),x_p(:,2),'.k')
axis([-2 2 -2 2]); grid on;
xlabel('\bfx'); ylabel('\bfy');
```

5.A.3 MATLAB Code for Phase Portraits

```
%Phase portraits for the systems with linear and nonlinear damping
%Based off of code used by Lynch (2004)

close all
clear all

%lambda = 0.2; sigma = 0.5; e = 0.05; A = 0.4;
```

```
%Linear damping case:

w = 1; t_end = 500;

state_space = inline('[x(2); .05*0.4*cos(t) - .05*0.2*(x(2)-x(4)) -
(1+.05*0.5)*x(1) - 4/3*.05*(x(1)-x(3))^3; x(4); -0.2*(x(4)-x(2)) -
4/3*(x(3)-x(1))^3]', 't', 'x');
[t, x_vec] = ode45(state_space,[0 t_end],[0,0,0,0]);

figure
plot(x_vec(:,1),x_vec(:,2),'-k')
xlabel('\bfy_1'); ylabel('\bfy_1-dot')

figure
plot(x_vec(:,3),x_vec(:,4),'-k')
xlabel('\bfy_2'); ylabel('\bfy_2-dot')

clear all

%Nonlinear damping case:

w = 1; t_end = 500;

state_space = inline('[x(2); .05*0.4*cos(t) - .05*0.2*(x(2)-x(4))^3 -
(1+.05*0.5)*x(1) - 4/3*.05*(x(1)-x(3))^3; x(4); -0.2*(x(4)-x(2))^3 -
4/3*(x(3)-x(1))^3]', 't', 'x');
[t, x_vec] = ode45(state_space,[0 t_end],[0,0,0,0]);

figure
plot(x_vec(:,1),x_vec(:,2),'-k')
xlabel('\bfy_1'); ylabel('\bfy_1-dot')

figure
plot(x_vec(:,3),x_vec(:,4),'-k')
xlabel('\bfy_2'); ylabel('\bfy_2-dot')
```

5.A.4 MATLAB Code for Poincaré Maps

```
%Poincare maps for the systems with linear and nonlinear damping
%Based off of code used by Lynch (2004)

close all
clear all

%lambda = 0.2; sigma = 0.5; e = 0.05; A = 0.4;

%Linear damping case:

w = 1;
dt = 2*pi/w; t_end = 2000*pi/w;
```

```
state_space = inline('[x(2); .05*0.4*cos(t) - .05*0.2*(x(2)-x(4)) -
(1+.05*0.5)*x(1) - 4/3*.05*(x(1)-x(3))^3; x(4); -0.2*(x(4)-x(2)) -
4/3*(x(3)-x(1))^3]', 't', 'x');
[t, x_vec] = ode45(state_space,0:dt:t_end,[1,0,0,0]);

figure
plot(x_vec(:,1),x_vec(:,2),'.k')
xlabel('\bfy_1'); ylabel('\bfy_1-dot')
axis([-0.8 1 -0.4 1])

figure
plot(x_vec(:,3),x_vec(:,4),'.k')
xlabel('\bfy_2'); ylabel('\bfy_2-dot')
axis([-2.5 2.5 -1.5 2])

clear all

%Nonlinear damping case:

w = 1;
dt = 2*pi/w; t_end = 2000*pi/w;

state_space = inline('[x(2); .05*0.4*cos(t) - .05*0.2*(x(2)-x(4))^3 -
(1+.05*0.5)*x(1) - 4/3*.05*(x(1)-x(3))^3; x(4); -0.2*(x(4)-x(2))^3 -
4/3*(x(3)-x(1))^3]', 't', 'x');
[t, x_vec] = ode45(state_space,0:dt:t_end,[1,0,0,0]);

figure
plot(x_vec(:,1),x_vec(:,2),'.k')
xlabel('\bfy_1'); ylabel('\bfy_1-dot')
axis([-0.8 1 -0.4 1])

figure
plot(x_vec(:,3),x_vec(:,4),'.k')
xlabel('\bfy_2'); ylabel('\bfy_2-dot')
axis([-2.5 2.5 -1.5 2])
```

6.0 Conclusions and Recommendations

6.1 Summary of Work Presented

This thesis has presented analytical and numerical comparisons between single-degree of freedom linear systems with linearly and nonlinearly damped nonlinear energy sink (NES) attachments. Chapter 1 provided an introduction to NES and a background of previous related work. Concepts such as targeted energy transfer, nonlinear normal modes (NNMs), and resonance capture were discussed to provide the reader prerequisite information for understanding how NES achieve energy pumping. Comparisons between classical linear vibration absorbers and NES were presented in order to instill motivation for studying NES. Finally, experimental methods used in the study of NES were illustrated.

Chapter 2 focused on the differences between various single-degree of freedom systems. A linear system, a NES system, and a nonlinear system with nonlinear damping were compared in terms of time response, system energy as a function of time, and energy decay as a function of time. In each performance comparison, the nonlinear system with nonlinear damping exhibited the most desirable characteristics in regards to vibration attenuation. Clearly, the nonlinear damping allowed the nonlinear system to dissipate energy faster than its linearly damped counterparts, thus providing a more rapid reduction in displacement.

Saddle-node and Hopf bifurcations of the single-degree of freedom linear systems with linearly and nonlinearly damped NES were presented in Chapter 3 in order to study the periodic behavior of the two systems. The saddle-node bifurcation diagrams showed

boundaries between one and three real periodic solutions. Boundaries separating stable and unstable regions were shown by the Hopf bifurcation diagrams. Qualitative differences between the two systems were discussed by comparing the bifurcation diagrams of each system for the same sets of parameters.

Chapter 4 deviates from the analysis of periodic solutions by comparing the Strongly Modulated Response (SMR) between the linearly and nonlinearly damped systems. First, the projection of the slow invariant manifold (SIM) onto a 2-D plane of system parameters was presented to show the locations for the jumps of the system response between stable branches. Phase portraits were presented for the stable branches of the SIM in order to show the manner in which the system response can jump between these branches. For a concise illustration of the trajectories that leave the lower fold line and eventually return, 1-D maps of the phase angles were constructed for varying the system parameter σ. A range of σ that permitted the return trajectories was determined, thus providing the conditions for the existence of the SMR. Due to the correlation between σ and the forcing frequency, frequency ranges were determined that allow for the SMR to occur.

In Chapter 5, time response comparisons were made between the linearly and nonlinearly damped systems. Plots of displacement and energy versus time were presented for the systems with varying amplitudes of impulse loading. From these numerical simulations, it was determined that the displacement and energy of the linear oscillator decay faster in the linearly damped system than the nonlinearly damped system for low impulse amplitudes. However, if the impulse amplitude is increased beyond a certain level, the performance of the nonlinearly damped system becomes more desirable

in terms of vibration attenuation than the linearly damped system. Performance comparisons of the periodically forced system were continued with the use of phase portraits and Poincaré maps. Comparisons between these plots reveal that the nonlinearly damped system in general has lower displacements and velocities than the linearly damped system for the chosen system parameters.

One should keep in mind that this thesis has investigated the performance of the single-degree of freedom systems with linearly and nonlinearly damped NES for only specific cases. Periodic forcing near resonance was studied in Chapters 3 through 5, and the SMR discussed in Chapter 4 was only possible due to the 1:1 resonance with the linear oscillator. Additionally, only specific parameters were chosen to illustrate the system behavior for these cases.

6.2 Suggested Future Work

In order to expand on the work presented in this thesis, the obvious suggestion is to repeat the work using different system parameters. Using the methods presented, this variation should be straightforward to implement. Additionally, the systems could be investigated to determine their behavior when subjected to periodic forcing away from 1:1 resonance, although the SMR will not exist in this case.

Prior to attempting to apply the work shown in this thesis to practical engineering applications, experiments should be run to confirm these conclusions. Even after the analytical and numerical results have been correlated to experiments, one must determine a method of physically implementing the system parameters in the design. Piecewise nonlinear damping can be achieved in practice via the variable orifice discussed by Starosvetsky and Gendelman (2009), but this nonlinear damping differs from the

damping discussed in the work of this thesis. Nonlinear stiffness has been achieved in experiments by using a grounded wire as shown in Section 1.4 of Chapter 1 of this thesis, but the designer of an engineering application involving NES must determine if this method is suitable for a particular situation. This thesis presents only preliminary work on nonlinearly damped NES, and much work still remains before practical applications can be implemented.

6.3 Practical Applications of Nonlinear Energy Sinks

Before deciding to use nonlinear energy sinks in a design, it is important to consider the application. In some cases, a linear vibration absorber may be desired due to its simplicity of design and likely cost savings. For example, if vibration at a single resonant frequency is the only concern, a tuned linear absorber is a logical option. However, if the application demands attenuation at multiple frequencies and/or a more reliable design, a nonlinear energy sink is advantageous. Furthermore, the use of linear versus nonlinear damping in the NES design should be given consideration as well. If the external forces on the system are very high, a NES with nonlinear damping would likely result in more efficient energy reduction in the primary system than a NES with linear damping.

7.0 References

Gendelman, O.V. (2001). Transition of energy to a nonlinear localized mode in a highly asymmetric system of two oscillators, *Nonl. Dyn.* **25**, 237-253.

Gendelman, O.V., and Y. Starosvetsky (2007). Quasiperiodic response regimes of linear oscillator coupled to nonlinear energy sink under periodic forcing, *Journal of Applied Mechanics* **74**, 325-331.

Gendelman, O.V., L.I. Manevitch, A.F. Vakakis, and R. M'Closkey (2001). Energy Pumping in Nonlinear Mechanical Oscillators: Part I – Dynamics of the Underlying Hamiltonian Systems, *Journal of Applied Mechanics* **68**, 34-41.

Gendelman, O.V., Y. Starosvetsky, and M. Feldman (2008). Attractors of harmonically forced linear oscillator with attached nonlinear energy sink I: description of response regimes, *Nonlinear Dynamics* **51**, 31-46.

Lee, Y.S., A.F. Vakakis, L.A. Bergman, D.M. McFarland, G. Kerschen, F. Nucera, S. Tsakirtzis, and P.N. Panagopoulos (2008). Passive non-linear targeted energy transfer and its applications to vibration absorption: a review, *IMechE* **222**, 77-134.

Lynch, S. (2004). *Dynamical Systems with Applications Using MATLAB*, Birkhäuser, Boston.

Meirovitch, L. (2001). *Fundamentals of Vibrations*, McGraw-Hill, New York.

Nayfeh, A. H. and Balachandran, B. (1995). *Applied Nonlinear Dynamics*: *Analytical, Computational, and Experimental Methods*, Wiley, New York.

Nayfeh, A. H. and Mook, D. T. (1995). *Nonlinear Oscillations*, Wiley, New York.

Rand, R. (2005). *Lecture Notes on Nonlinear Vibrations*, Richard H. Rand.

Sanders, J., and F. Verhulst (1985). *Averaging Methods in Nonlinear Dynamical Systems*, Springer-Verlag, New York.

Starosvetsky, Y., and O.V. Gendelman (2008). Response regimes of linear oscillator coupled to nonlinear energy sink with harmonic forcing and frequency detuning, *Journal of Sound and Vibration* **315**, 746-765.

Starosvetsky, Y., and O.V. Gendelman (2008). Strongly modulated response in forced 2DOF oscillatory system with essential mass and potential asymmetry, *Physica D* **237**, 1719-1733.

Starosvetsky, Y., and O.V. Gendelman (2009). Vibration absorption in systems with a nonlinear energy sink: Nonlinear damping, *Journal of Sound and Vibration* **324**, 916-939.

Stewart, J. (1999). *Calculus Early Transcendentals*, Brooks/Cole Publishing Company, Pacific Grove.

Vakakis, A.F. (2001). Inducing Passive Nonlinear Energy Sinks in Vibrating Systems, *Journal of Vibration and Acoustics/Transactions of the ASME* **123**, 324-332.

Vakakis, A.F., L.I. Manevitch, O. Gendelman, and L. Bergman (2003). Dynamics of linear discrete systems connected to local, essentially non-linear attachments, *Journal of Sound and Vibration* **264**, 559-577.

Vakakis, A.F., O.V. Gendelman, L.A. Bergman, D.M. McFarland, G. Kerschen, and Y.S. Lee (2008). *Nonlinear Targeted Energy Transfer in Mechanical and Structural Systems I*, Springer, New York.

Zwillinger, D. (2003). *CRC Standard Mathematical Tables and Formulae*, Chapman & Hall/CRC Press LLC, Boca Raton.